

UNIVERSIDAD POLITÉCNICA DE VALENCIA

Departamento de Organización de Empresas



**PROGRAMACIÓN DE LA PRODUCCIÓN EN UN TALLER
DE FLUJO HÍBRIDO SUJETO A INCERTIDUMBRE:
ARQUITECTURA Y ALGORITMOS. APLICACIÓN A LA
INDUSTRIA CERÁMICA.**

TESIS DOCTORAL

PRESENTADA POR:

D. PEDRO GÓMEZ GASQUET

DIRIGIDA POR:

DR. D. FRANCISCO CRUZ LARIO ESTEBAN

DR. D. CARLOS ANDRÉS ROMANO

VALENCIA, ENERO DE 2010

Esta tesis se ha desarrollado con el apoyo del Centro de Investigación de Gestión e Ingeniería de la Producción (CIGIP) de la Universidad Politécnica de Valencia.



A Cristina, Mar y Claudia.



Quisiera aprovechar estas líneas para recordar a todos los que han hecho posible que este trabajo se haya podido realizar.

En primer lugar quisiera dar las gracias a la familia. A mis Padres por proporcionarme el entorno y la confianza que me han posibilitado embarcarme en esta aventura. A mi mujer, Cristina, que ha conseguido el milagro de sacar tiempo para que yo pudiera trabajar en la tesis, al mismo tiempo que llegaron a nuestras vidas Mar y Claudia y lo revolucionaron todo. Este nuestro tercer "hijo" te lo dedico sobre todo a ti. Tampoco me quiero olvidar de mis suegros sin los cuales todo hubiera sido mucho más difícil.

En el terreno profesional quisiera agradecer a todos mis compañeros del Departamento de Organización de Empresas su colaboración, a los compañeros del Centro de Investigación de Gestión e Ingeniería de la Producción su apoyo en todo lo que he necesitado para el desarrollo de mi trabajo, a los doctores Ángel Ortiz, Raúl Poler, José Pedro García, Juan José Alfaro y Raúl Rodríguez la oportunidad de intercambiar puntos de vista en múltiples ocasiones, y al doctor José Miguel Carot por su apoyo en el análisis estadístico de los datos.

De forma especial quiero dar las gracias a Rubén Darío Franco por dedicar su tiempo, con gran paciencia y mucho acierto, a orientarme cuando más lo necesitaba. Gracias Darío por mostrarme el universo de los Sistemas Multiagente.

Así mismo quisiera expresar mi máximo agradecimiento a las personas que de forma más directa han contribuido a este trabajo, que nos son otros que mis directores. El Dr. Francisco Cruz Lario ha contribuido estructurar y dar consistencia al trabajo con sus comentarios y recomendaciones. El Dr. Carlos Andrés me ha servido de referencia en todo el trabajo relacionado con el planteamiento del problema y la selección de posibles técnicas de resolución. Ambos han aportado con generosidad su experiencia y han contribuido en la mejorar del trabajo final.

Finalmente, quisiera agradecer a los revisores y miembros del tribunal su esfuerzo y dedicación, así como sus comentarios.

Gracias a todos.

Valencia, 2 de noviembre de 2009

RESUMEN

PROGRAMACIÓN DE LA PRODUCCIÓN EN UN TALLER DE FLUJO HÍBRIDO SUJETO A INCERTIDUMBRE: ARQUITECTURA Y ALGORITMOS. APLICACIÓN A LA INDUSTRIA CERÁMICA.

En un marco de competencia global en el cual los tiempos de respuesta son cada vez más relevantes como elemento competitivo y donde, en no pocas ocasiones, las empresas tiende a ofrecer un catálogo de productos amplio y diferenciado de la competencia, existen múltiples retos que las Organizaciones deben afrontar. Dentro de éstas la Dirección de Operaciones tiene el reto de adaptar los procesos de Gestión de los Sistemas Productivos y Logísticos a las actuales necesidades. En este proceso de cambio es habitual partir de Sistemas Productivos poco flexibles y orientados a la producción en masa en los que es fundamental emplear el mejor “saber-hacer” para procurar obtener el rendimiento más adecuado de los recursos disponibles. El despliegue de unas buenas prácticas en el ámbito de la Programación de la Producción puede ayudar en buena medida a mejorar la eficiencia de los recursos.

Tradicionalmente se ha venido considerando a la Programación de la Producción con una visión bastante cuantitativa en la que su misión consistía en asignar, secuenciar y temporizar los diferentes trabajos del periodo en base a los recursos disponibles. No obstante, sin dejar de ser válido este planteamiento, en esta tesis se desea enfatizar como en realidad el fin último de las técnicas y métodos desarrollados durante años en el ámbito de la Programación de la Producción no es otro que el de ser empleados dentro de un Sistemas de Ayuda a la Toma de Decisiones. Y en este sentido, las decisiones operativas que se toman en el área del Programador de la Producción deben estar conectadas en todos los casos, al menos, con su entorno decisional más directo como es el de la Planificación de la Producción.

Una revisión literaria en profundidad al extenso trabajo realizado en más de 50 años de existencia de lo que se ha denominado, empleando la terminología en lengua inglesa, como “Scheduling” pone de manifiesto la existencia una necesidad que debe ser cubierta. Se trata de reducir la distancia entre los problemas planteados por la comunidad científica y los que desean resolver los profesionales. Los problemas académicos se caracterizan por ser simplificaciones de la realidad en la cual los sistemas productivos, las condiciones de trabajo, las interrelaciones organizativas, o la propia aparición de eventos son considerados de forma parcial, o incluso ignorados. Por este motivo en muchos casos los procedimientos y las técnicas aplicadas a la Programación de la Producción en las empresas no han recibido el empuje que precisa.

Uno de los aspectos que es preciso considerar para reducir la brecha existente entre el entorno académico y el profesional es el de la Incertidumbre. Existe relativamente poca bibliografía que considere el problema de la Programación de la Producción en entornos inciertos, y de las propuestas revisadas se ha optado por seguir el modelo que propone dividir el problema en dos fases. La primera fase, que recibe el nombre de Programación Predictiva, consiste en establecer un Programa Productivo en un entorno determinista que sirva de punto de partida. La segunda fase consiste en modificar el Programa Productivo en curso cuando se produzca un evento que lo haga inviable, y se denomina Programación Reactiva. La Programación Predictiva y Reactiva de la Producción sirven para generar Programas a partir de restricciones y objetivos que pueden ser diferentes dado que se emplean en instantes diferentes del periodo Productivo considerado.

Con el objeto de explorar nuevos paradigmas que aportasen un enfoque diferente al tradicional en relación al uso de técnicas y herramientas con los que tratar el problema de Programación de la Producción, en las condiciones actuales, se exploraron las posibilidades de la Inteligencia Artificial, y en concreto el área relacionada con los Agentes Inteligentes y Sistemas Multiagente. El paradigma de los Agentes se ha aplicado a diversas áreas entre las que se encuentra la de Dirección de Operaciones. En el área mencionada su puesta en práctica se ha centrado sobre todo en el ámbito de la Planificación de la Producción y de la Gestión de la Cadena de Suministro. Esto es debido a que la teoría de Agentes ayuda a abordar aspectos como la Integración Empresarial, Organización Distribuida, los Entornos Heterogéneos, la Interoperabilidad, las Estructuras Dinámicas, o la Cooperación entre otros. Dada la necesidad de incluir algunos de estos aspectos en el ámbito de la Programación de la Producción se ha considerado que los Sistemas Multiagentes pueden actuar de facilitadores.

Una vez analizadas las aportaciones científicas relacionadas con el ámbito del "Scheduling" y de los Sistemas Multiagente se identifica que el objetivo de esta tesis debe ser el de plantear un Modelo que permita Automatizar, mediante el empleo de Sistemas Multiagente, la Programación Predictiva y Reactiva de la Producción, basado en una orientación de Procesos de Negocio que incluya Herramientas (Algoritmos) de Ayuda a la Toma de Decisiones bajo Incertidumbre, necesarios para resolver la visión funcional/decisional que contemple la existencia de eventos y su gestión, y que permita dar el primer paso hacia un Modelo Dinámico. El objetivo fijado para dicho modelo es el de mantener una tasa de productividad media lo más alta posible en función de los recursos disponibles, y para ello los procesos de Programación Predictiva y Reactiva dispondrán de funciones objetivas que potencien el mismo.

Como paso previo a la construcción de una propuesta se realiza un importante esfuerzo para delimitar correctamente la validez de la misma. En este aspecto, se analizan aquellas características del problema que se desea tratar destacando sus peculiaridades, pudiéndose caracterizar como un Taller de Flujo Híbrido con Tiempos de Cambio de Partida Dependientes de la Secuencia en condiciones de Incertidumbre. El problema se modela matemáticamente.

Para alcanzar el objetivo propuesto se define una Metodología, unos elementos Arquitectónicos, y unas Herramientas que sirven de guía y de soporte, al mismo tiempo que estructuran, la elaboración de una propuesta. A través de cinco pasos definidos en la metodología, y empleando en cada uno de ellos los elementos constructivos y herramientas necesarios, se sigue un camino que se inicia con la definición elemental del problema y que termina con una prueba de funcionamiento para un caso concreto.

Con los dos primeros pasos de la metodología se determinaron los requerimientos en los que se debería basar la propuesta para conseguir su objetivo. Para el análisis de los requerimientos del proceso de Programación Predictivo-Reactiva y su relación con los procesos de Planificación se emplea la propuesta CIMOSA. Con el tercer paso de la metodología, y en base a los requerimientos identificados, aborda el diseño de una Plataforma Software concebida como un Sistema Multiagente (IPSU-MAS) que facilite la automatización de los procesos de Programación. Para el diseño se emplea la Metodología INGENIAS específica para desarrollos basados en Agentes.

Antes de realizar el cuarto paso, que consiste en la implementación de IPSU-MAS, se diseña, desarrolla, y analiza una serie de algoritmos y procedimientos que necesariamente deben formar parte de la Plataforma Software. Si se considera a la plataforma como una estructura estos procesos serían el motor que realmente posibilita la automatización. Estos se definen a partir de un análisis conceptual y un contraste práctico. Entre los algoritmos más relevantes que se desarrollan hay que destacar aquellos que permiten el cálculo del Programa Predictivo y del Programa Reactivo de la Producción. Aunque en menor medida también abordan aquellos que regulan la coordinación entre los procesos de Programación y Planificación.

Para poder obtener un Programa Productivo Predictivo se propone el diseño e implementación de un nuevo Algoritmo Genético denominado SMAGA. La diferencia fundamental entre la propuesta que realiza en esta tesis y los Algoritmos Genéticos tradicionales, es que en este caso, se realiza una hibridación del algoritmo combinando la estructura básica de esta Metaheurística con las características de los Sistemas Multiagente. El objetivo es que la hibridación se produzca de forma que los Agentes tomen el rol de "seres vivos" y por lo tanto, de individuos de la población. Con la introducción del paradigma de Agentes en este algoritmo la propuesta considera la existencia de un Sistema Multiagente (SMAGA) dentro de otro (IPSU-MAS). Después de una adecuada parametrización de las dos variantes del algoritmo SMAGA, los resultados se comparan con los obtenidos por otras propuestas realizadas para el mismo tipo de problemas resultando ser altamente eficaces. Tanto para la parametrización como para el análisis comparativo se emplea un juego de datos conocido.

Para realizar el cálculo del Programa Reactivo se realiza un análisis comparativo entre un conjunto de Heurísticas y Metaheurísticas seleccionado, entre las que existe una aportación

propia (SSDS). Todas las técnicas son sometidas a un amplio estudio en diversas circunstancias pudiéndose obtener conclusiones relevantes en cuanto a su comportamiento.

Finalmente se realiza como parte del quinto paso de la metodología una aplicación del prototipo IPSU-MAS, generado a partir del trabajo realizado anteriormente, a una empresa concreta de la Industria Cerámica. Se considera que las empresas de este sector son muy adecuadas para realizar una experiencia práctica debido a la configuración de sus talleres y a la situación de competencia de su entorno. En la experiencia práctica se compararon los resultados que obtenidos empleando los métodos actuales de la Empresa Seleccionada y los obtenidos con IPSU-MAS, en diversos escenarios. La conclusión es que en todos ellos los resultados obtenidos con la IPSU-MAS son mejores, y en muchos de ellos se han alcanzado valores con una tasa de productividad entre un 17% y un 29% mejores que los de las técnicas actualmente empleadas.

ABSTRACT

HYBRID FLOWSHOP SCHEDULING UNDER UNCERTAINTY: ARCHITECTURE AND ALGORITHMS. APPLICATION TO THE CERAMIC TILE INDUSTRY.

In a time of global competence in which the lead times are increasingly more prominent as competitive element and where, in not few occasions, companies tend to offer a catalogue of extensive products and differentiated from the competence, multiple challenges exist that the Organizations should confront. Operations Management has the challenge to adapt the processes of Management of the Logistic and Productive Systems to the current needs. In this process of change is habitual to leave from flexible small Productive Systems and to take the mass production approach in which is fundamental to employ the best "know-how" to try to obtain the most adequate performance of the available resources. The unfold of some good practices in the environment of the Production Scheduling can greatly help to improve the efficiency of the resources.

Traditionally, Production Scheduling has been taken under a rather quantitative vision whose mission consisted of assigning, to arrange and to temporize the different works of the period based on the available resources. Nevertheless, although still considering valid this approach, this thesis aims to emphasize that, in reality, the last goal of the techniques and methods developed during years within the environment of the Production Scheduling is not another than being applied within Decision Making Aid Systems. In this sense, the operative decisions taken in the Production Scheduler area should be connected in all the cases, at least, with its more direct decisional environment, as is that of the Production Planning.

An in depth literary review within the extensive work carried out in more than 50 years of existence of what has been called as "Scheduling" shows the existence of a need that should be covered. It is a matter of reducing the distance among the problems presented by the scientific community and the ones that professionals want to solve. The academic problems are characterized for being simplifications of the reality in which the productive systems, the working conditions, the organizing interrelations, or the own apparition of events are considered from a partial point of view, or even ignored. For that reason in many cases the procedures and the techniques applied to the Production Scheduling in the organizations have not received the attention needed.

One of the aspects to be considered to reduce the gap between the academic environment and the professional one is that of the Uncertainty. Relatively few bibliography exists that consider the problem of the Production Scheduling under uncertain environments, and from

the revised proposals, it has been decided to follow the model that proposes to divide the problem into two phases. The first phase, that receives the name of Predictive Scheduling, consists of establishing a Productive Schedule in a deterministic environment that serve of starting point. The second phase consists of modifying the Productive Schedule in process in the occurrence of an event that do it unfeasible, and it is called Reactive Scheduling. The Predictive and Reactive Production Scheduling serve to generate Schedules from restrictions and objectives that can be different given since they are employed in different instants from the Productive period considered.

With the objective of exploring paradigms that contributed a different focus to the traditional one relating to the use of techniques and tools to deal with the problem of Production Scheduling, in the current conditions, the possibilities of the Artificial Intelligence were explored, and in concrete the area related to the Intelligent Agents and Multi-agents Systems. The paradigm of the Agents has been applied to diverse areas like Operations Management. In the area mentioned its put in practice has been mainly focused on the environment of the Production Planning and of the Supply Chain Management. This due to that the aid Agents theory to undertake aspects as the Business Integration, Organization Distributed, the Heterogeneous Environments, the Interoperability, the Dynamic Structures, or the Cooperation among others. From the need of including some of these aspects in the Production Programming ambit, it has been considered that the Multi-agents Systems can act as facilitators.

Once the scientific contributions related to the environment of the "Scheduling" and the Multi-agents Systems have been analyzed, it has been identified that the objective of this thesis should be to introduce a Model that permit to Automate, by means of the employment of Multi-agents Systems, the Predictive and Reactive Production Scheduling, based on a Business Processes orientation that include Tools (Algorithms) for aiding to make decisions under Uncertainty, which are necessary to solve the functional/decisional vision that encloses the existence of events and its management, and that permit to take the first step toward a Dynamic Model. The objective set for the model is that of maintaining a rate of average productivity as high as possible in function of the available resources, having available the processes of Predictive and Reactive Scheduling objective functions that potentiate it.

As a previous step to the construction of a proposal it has been carried out an important effort to correctly delimit its own validity. In this aspect, those characteristics of the problem desired to treat are analyzed, emphasizing its peculiarities and being able to characterize as a Hybrid Flow Shop with Sequence Dependent Setup Times in conditions of Uncertainty. The problem is mathematically modeled.

In order to reach the proposed objective a Methodology, some Architectural elements, and some Tools that serve of guide and support are defined. These elements, at the same time,

structure the elaboration of a proposal. Through five steps defined within the methodology, and employing in each one of them the constructive elements and necessary tools, it is followed a path that is initiated with the elementary definition of the problem and that finishes with a test of operation for a particular case.

With the two first steps of the methodology were determined the requests in which the proposal should be based to obtain its objective. For the analysis of the requirements of the Predictive and Reactive Scheduling process and its relation with the Production Planning is employed the proposed named CIMOSA. With the third step of the methodology, and in base to the identifying requests, it is undertaken the design of a Software Platform conceived as a Multi-agent System (IPSU-MAS) that facilitates the automation of the processes of Scheduling. As design Methodology is employed INGENIAS, which is specific for developments based on Agents.

Before carrying out the fourth step, that consists of the implementation of IPSU-MAS, it is designed, developed, and analyzed a series of algorithms and procedures that necessarily should be included into the Software Platform. If the platform is considered as a structure, these processes would be the motor that really enables the automation. These are defined from a conceptual analysis and a practical contrast. Among the most prominent algorithms develop it is necessary to emphasize those that permit the calculation of the Predictive Scheduling and of the Reactive Production Scheduling. Although in smaller measure they also undertake those algorithms that regulate the coordination between the processes of Scheduling and Planning.

In order to be able to obtain a Predictive Production Schedule it is proposed the design and implementation of a new Genetic Algorithm called SMAGA. The main difference among the proposal carried out in this thesis and the traditional Genetic Algorithms, is that in this case, a hybridization of the algorithm is carried out by combining the basic structure of this Metaheuristic with the characteristics of the Multi-agent Systems. The objective is that the hybridization be produced so that the Agents take the role of "alive beings" and therefore, of individuals of the population. With the introduction of the paradigm of Agents in this algorithm the proposal considers the existence of a Multi-agent System (SMAGA) inside another (IPSU-MAS). After a proper parameter setting of the two variants of the SMAGA algorithms, the results are compared with the obtained by other proposals carried out for the same type of problems, turning out to be highly efficient. For both the parameter setting and for the comparative analysis a known set of data is employed.

In order to carry out the calculation of the Reactive Schedule, it is carried out a comparative analysis between a selected set of Heuristics and Metaheuristics, existing an own contribution (SSDS). All the techniques have undertaken an extensive study under diverse circumstances, being able to obtain prominent conclusions regarding their behavior.

Finally it is carried out, as part of the fifth step of the methodology, an application of the prototype IPSU-MAS, generated from the work carried out previously, to a real world company of the Ceramic Tile Industry. It is considered that the companies of this sector are very adequate to carry out a practical experience due to the configuration of their manufacturing systems and competence level reached within their environment. In the practical experience were compared the results obtained when employing the current methods of the Company Selected and when applying IPSU-MAS, in diverse settings. The conclusion is that in all cases the results obtained with the IPSU-MAS are better, and many of them have reached values of productivity rate of between a 17% and a 29% better than the ones obtained with current techniques.

RESUM

PROGRAMACIÓ DE LA PRODUCCIÓ EN UN TALLER DE FLUX HÍBRID SUBJECTE A INCERTESA: ARQUITECTURA I ALGORISMES. APLICACIÓ A LA INDÚSTRIA CERÀMICA.

En un marc de competència global en què els temps de resposta són cada vegada més rellevants com a element competitiu i on, en no poques ocasions, les empreses tendeixen a oferir un catàleg de productes ampli i diferenciat de la competència, hi ha múltiples reptes que les organitzacions han d'afrontar. Dins d'aquestes, la direcció d'operacions té el repte d'adaptar els processos de gestió dels sistemes productius i logístics a les necessitats actuals. En aquest procés de canvi és habitual partir de sistemes productius poc flexibles i orientats a la producció en massa, en què és fonamental emprar el millor "saber fer" per a procurar obtenir el rendiment més adequat dels recursos disponibles. El desplegament d'unes bones pràctiques en l'àmbit de la programació de la producció pot ajudar en bona mesura a millorar l'eficiència dels recursos.

Tradicionalment, s'ha considerat la programació de la producció amb una visió bastant quantitativa en què la missió que tenia consistia a assignar, seqüenciar i temporitzar els diferents treballs del període sobre la base dels recursos disponibles. No obstant això, sense deixar de ser vàlid aquest plantejament, en aquesta tesi es vol emfatitzar com, en realitat, la finalitat última de les tècniques i els mètodes desenvolupats durant anys en l'àmbit de la programació de la producció no és altra que ser emprats dins d'uns sistemes d'ajuda a la presa de decisions. I en aquest sentit, les decisions operatives que es prenen en l'àrea del programador de la producció han d'estar connectades en tots els casos, almenys, amb el seu entorn de decisió més directe, com és el de la planificació de la producció.

Una revisió literària en profunditat a l'extens treball realitzat en més de 50 anys d'existència del que s'ha denominat, emprant la terminologia en llengua anglesa, *scheduling*, posa de manifest l'existència una necessitat que ha de ser coberta. Es tracta de reduir la distància entre els problemes plantejats per la comunitat científica i els que volen resoldre els professionals. Els problemes acadèmics es caracteritzen perquè són simplificacions de la realitat, en què els sistemes productius, les condicions de treball, les interrelacions organitzatives o la mateixa aparició d'esdeveniments són considerats d'una manera parcial, o fins i tot ignorats. Per aquest motiu, en molts casos els procediments i les tècniques aplicades a la programació de la producció en les empreses no han rebut l'embranchida que caldria.

Un dels aspectes que cal considerar per a reduir la bretxa que hi ha entre l'entorn acadèmic i el professional és el de la incertesa. Hi ha relativament poca bibliografia que considere el problema de la programació de la producció en entorns incerts, i de les propostes revisades s'ha optat per seguir el model que proposa dividir el problema en dues fases. La primera fase, que rep el nom de programació predictiva, consisteix a establir un programa productiu en un entorn determinista que servisca de punt de partida. La segona fase consisteix a modificar el programa productiu en curs quan es produïska un esdeveniment que el faça inviable, i es denomina programació reactiva. La programació predictiva i reactiva de la producció serveixen per a generar programes a partir de restriccions i objectius que poden ser diferents, atès que s'empren en instants diferents del període productiu considerat.

Amb l'objecte d'explorar nous paradigmes que aportaren un enfocament diferent al tradicional en relació a l'ús de tècniques i eines amb què tractar el problema de programació de la producció, en les condicions actuals, s'han explorat les possibilitats de la intel·ligència artificial, i en concret l'àrea relacionada amb els agents intel·ligents i sistemes multiagent. El paradigma dels agents s'ha aplicat a diverses àrees, entre les quals es troba la de direcció d'operacions. En l'àrea esmentada, la posada en pràctica d'aquest paradigma s'ha centrat sobretot en l'àmbit de la planificació de la producció i de la gestió de la cadena de subministrament. Això es deu al fet que la teoria d'agents ajuda a abordar aspectes com la integració empresarial, organització distribuïda, els entorns heterogenis, la interoperabilitat, les estructures dinàmiques, o la cooperació, entre altres. Donada la necessitat d'incloure alguns d'aquests aspectes en l'àmbit de la programació de la producció, s'ha considerat que els sistemes multiagents hi poden actuar de facilitadors.

Una vegada analitzades les aportacions científiques relacionades amb l'àmbit de l'*scheduling* i dels sistemes multiagent, s'identifica que l'objectiu d'aquesta tesi ha de ser el de plantejar un model que permeta automatitzar, mitjançant l'ús de sistemes multiagent, la programació predictiva i reactiva de la producció, basat en una orientació de processos de negoci que incloga eines (algorismes) d'ajuda a la presa de decisions sota incertesa, necessaris per a resoldre la visió funcional/decisional que considere l'existència d'esdeveniments i la seua gestió, i que permeta donar el primer pas cap a un model dinàmic. L'objectiu fixat per a aquest model és el de mantenir una taxa de productivitat mitja el més alta possible en funció dels recursos disponibles, i per a això els processos de Programació Predictiu i Reactiu disposaran de funcions objectius que potencien el mateix.

Com a pas previ a la construcció d'una proposta, es du a terme un esforç important per a delimitar-ne la validesa correctament. En aquest aspecte, s'hi analitzen les característiques del problema que es vol tractar, destacant-ne les peculiaritats, de manera que es pot caracteritzar com un taller de flux híbrid amb temps de canvi de partida dependents de la seqüència en condicions d'incertesa. El problema es modela matemàticament.

Per a aconseguir l'objectiu proposat es defineix una metodologia, uns elements arquitectònics, i unes eines que serveixen de guia i de suport, al mateix temps que estructuren l'elaboració d'una proposta. A través de cinc passos definits en la metodologia, i emprant en cadascun les eines i els elements constructius necessaris, se segueix un camí que s'inicia amb la definició elemental del problema i que acaba amb una prova de funcionament per a un cas concret.

Amb els dos primers passos de la metodologia s'han determinat els requeriments en què s'hauria de basar la proposta per a aconseguir-ne l'objectiu. Per a l'anàlisi dels requeriments del procés de programació predictiu-reactiu i la seua relació amb els processos de planificació, s'empra la proposta CIMOSA. Amb el tercer pas de la metodologia, i sobre la base dels requeriments identificats, s'aborda el disseny d'una plataforma de programari concebuda com un sistema multiagent (IPSU-MAS) que facilite l'automatització dels processos de programació. Per al disseny s'empra la metodologia INGENIAS específica per a desenvolupaments basats en agents.

Abans de dur a terme el quart pas, que consisteix en la implementació d'IPSU-MAS, es dissenya, es desenvolupa i s'analitza una sèrie d'algorismes i procediments que necessàriament han de formar part de la plataforma programari. Si es considera la plataforma com una estructura, aquests processos serien el motor que realment possibilita l'automatització. Aquests es defineixen a partir d'una anàlisi conceptual i un contrast pràctic. Entre els algorismes més rellevants que es desenvolupen cal destacar els que permeten el càlcul del programa predictiu i del programa reactiu de la producció. Encara que en menor mesura també s'hi aborden els que regulen la coordinació entre els processos de programació i planificació.

Per a poder obtenir un programa productiu predictiu, es proposa el disseny i la implementació d'un nou algorisme genètic denominat SMAGA. La diferència fonamental entre la proposta que es fa en aquesta tesi i els algorismes genètics tradicionals, és que en aquest cas es porta a terme una hibridació de l'algorisme combinant l'estructura bàsica d'aquesta metaheurística amb les característiques dels sistemes multiagent. L'objectiu és que la hibridació es produïssa de manera que els agents prenguen el rol d'"éssers vius" i, per tant, d'individus de la població. Amb la introducció del paradigma d'agents en aquest algorisme, la proposta considera l'existència d'un sistema multiagent (SMAGA) dins d'un altre (IPSU-MAS). Després d'una adequada parametrització de les dues variants de l'algorisme SMAGA, els resultats es comparen amb els obtinguts per altres propostes realitzades per al mateix tipus de problemes, i resulten altament eficaços. Tant per a la parametrització com per a l'anàlisi comparativa, s'hi emprava un joc de dades conegut.

Per a fer el càlcul del programa reactiu es du a terme una anàlisi comparativa entre un conjunt d'heurístiques i metaheurístiques seleccionat, entre les quals hi ha una aportació pròpia

(SSDS). Totes les tècniques són sotmeses a un ampli estudi en diverses circumstàncies, i es poden obtenir conclusions rellevants pel que fa al comportament d'aquestes.

Finalment, es porta a terme com a part del cinquè pas de la metodologia una aplicació del prototip IPSU-MAS, generat a partir del treball realitzat anteriorment, a una empresa concreta de la indústria ceràmica. Es considera que les empreses d'aquest sector són molt adequades per a realitzar una experiència pràctica a causa de la configuració dels tallers i per la situació de competència del seu entorn. En l'experiència pràctica s'han comparat els resultats obtinguts emprant els mètodes actuals de l'empresa seleccionada i els obtinguts amb IPSU-MAS, en diversos escenaris. La conclusió és que, en tots aquests, els resultats obtinguts amb la IPSU-MAS són millors, i en molts s'han aconseguit valors amb una taxa productivitat entre un 17% i un 29% millors que els de les tècniques que s'empren actualment.

INDICE GENERAL

Capítulo 1 – Introducción.....	23
Capítulo 2 - La Empresa Cerámica y su Programación de la Producción.....	33
Capítulo 3 - Estado del Arte de los Modelos, Métodos y Algoritmos de Resolución en la Programación de la Producción.....	77
Capítulo 4 - Estado del Arte de los Sistemas Multiagente y su Aplicación a los Sistemas de Fabricación y su Programación de la Producción.....	195
Capítulo 5 - Descripción y Modelado Matemático del Problema.....	279
Capítulo 6 - Enfoque Empleado para la Elaboración de la Propuesta.....	335
Capítulo 7 - Análisis, Diseño e Implementación de una Plataforma Software basada en Multiagentes para la Programación de la Producción Predictivo-Reactiva.....	361
Capítulo 8 - Algoritmos para la Programación Predictiva de la Producción en un Taller de Flujo Híbrido.....	461
Capítulo 9 - Algoritmos para la Programación Reactiva de la Producción en un Taller de Flujo Híbrido.....	587
Capítulo 10 - Aplicación de la Propuesta de Programación de la Producción Predictivo/Reactiva a una Empresa Cerámica.....	633
Capítulo 11 - Conclusiones y Trabajo Futuro.....	697

Anexo I - Definición de los Elementos Arquitectónicos de CIMOSA Implementados para Modelado de Requerimientos del Problema Propuesto.....	713
Anexo II - Límites Conocidos y Nuevos Límites Alcanzados por SMAGA-1 y SMAGA-2 para los Bancos de Datos Empleados en la Experimentación.....	749
Anexo III - Análisis Estadístico de los Resultados Obtenidos en la Experimentación para la Programación Predictiva de la Producción.....	775
Anexo IV - Resultados y Análisis Estadístico de los mismos Obtenidos en la Experimentación para la Programación Reactiva.....	861
Anexo V - Productos Empleados para la Simulación.....	959
Anexo VI - Análisis Estadístico de los Resultados Obtenidos en la Parametrización de SMAGA-1 y SMAGA-2 para la Plataforma IPSU-MAS.....	963

Introducción

Capítulo

1

Tabla de contenido

1.	Capítulo 1	27
1.1	Presentación.....	27
1.2	Motivación	27
1.3	Objetivos	28
1.4	Enfoque del Trabajo y Proceso Recorrido.....	29
1.5	Esquema General de la tesis	30

1.1 Presentación

El trabajo que se inicia en este punto se ha desarrollado en el marco del proceso investigador correspondiente a la última fase del programa doctoral, y con él se pretende poner de manifiesto la capacidad investigadora del doctorando.

El programa de doctorado “Gestión de la Cadena de Suministro e Integración Empresarial” en el que se enmarca este documento da cabida a diversas líneas de investigación entre las cuales se encuentra la relacionada con la Programación de la Producción (“Scheduling” en su denominación en lengua Inglesa), en la cual se centra este trabajo. Sin embargo, gracias a la continua interacción con otras disciplinas relacionadas tanto con la Gestión de la Cadena de Suministro como con la Integración Empresarial, la tesis que se presenta no se centra únicamente aspectos cuantitativos sino que intenta ofrecer una visión más amplia del Problema de Programación de la Producción incorporando visiones del problema procedentes de otras áreas. En concreto se ha tenido muy en cuenta la relación existente entre las tareas del Programador y las del Planificador de la Producción y la necesidad de colaboración entre ambos. Así mismo, como parte del proceso investigador se han manejado conceptos y técnicas próximos al modelado de procesos y al desarrollo de aplicaciones software.

Se podría finalizar esta introducción indicando que aunque esta tesis se basa fundamentalmente en ciertos aspectos cuantitativos relacionados con la Programación de la Producción, ofreciendo en este campo diversas aportaciones en forma fundamentalmente de algoritmos, se realiza un esfuerzo importante considerando las condiciones habituales del entorno en donde dichos avances se debería aplicar, ofreciendo como resultado un conjunto de Modelos, Algoritmos y Herramientas fácilmente aplicables en un entorno real.

1.2 Motivación

La investigación ofrece sin duda un gran estímulo cuando se es conocedor de la aplicabilidad de los resultados obtenidos. En el área de Ingeniería de Organización Industrial, y en concreto dentro de la línea de investigación relacionada con la Programación de la Producción, existe un campo que reclama continuamente la necesidad de incorporar nuevos avances debido a los continuos cambios en los ingenios, en los procesos y en las formas de gestionar las personas que ello implica.

Una de las industrias en las que mejor encaja la filosofía de mejora continua es precisamente la Industria Cerámica española, siempre preocupada por ofrecer productos de calidad, fruto de un interés constante por mejorar todos los aspectos relacionados con la fabricación de sus productos cerámicos. En las dos últimas décadas las empresas del Sector Cerámico han sufrido una transformación muy profunda, inicialmente basada en los avances tecnológicos de sus equipos productivos, y posteriormente fundamentada en los aspectos relacionados con su gestión. Es en el marco de esta apuesta por la mejora, ya en el ámbito de la gestión, y en una industria líder mundial, en el que se ha planteado la posibilidad de revisar inicialmente aquellos aspectos mejorables, y posteriormente desarrollar una propuesta que permita alcanzar algún logro concreto en el ámbito de la Programación de la Producción.

En este aspecto se quiere poner de manifiesto que además de la experiencia que el autor de este trabajo ha tenido durante su etapa como consultor, también durante su etapa como investigador y docente ha podido dedicar parte de su esfuerzo a trabajar en diversos proyectos relacionados con la Industria Cerámica. Algunos de estos proyectos han sido: ESPRIT Project 26736 'Performance Measurement System for Total Quality Management' (TQM-Tile), GV04A/543 Resolución Negociada para el Problema Real de la Programación de Producción Colaborativa de Procesos en Línea en la Industria Cerámica (Agent-Flow), DPI2004-02594- Interoperabilidad de Procesos Extendidos, PSS-370500-2006-1 a PSS-370500-2006-7 Proyecto de Potenciación de la Competitividad del Tejido Empresarial Español a través de la Logística como Factor Estratégico en un Entorno Global (GLOBALOG).

1.3 Objetivos

El objetivo de esta tesis debe ser el de plantear un Modelo que permita Automatizar mediante el empleo de Sistemas Multiagente la Programación de la Producción durante todo el periodo que se desee abarcar, basado en una orientación de Procesos de Negocio que incluya Herramientas de Ayuda a la Toma de Decisiones bajo Incertidumbre, necesarios para resolver la visión funcional/decisional que contemple la existencia de eventos y en su caso la reprogramación de las operaciones, y que permita dar el primer paso hacia un Modelo de más Flexible y orientado al mercado.

Desde el punto de vista investigador esta tesis doctoral tiene como metas es **identificar** una situación que afecta al ámbito operativo de la dirección de operaciones, **analizar** detenidamente el problema, **modelarlo** atendiendo al tipo de solución que se identifique más ajustada al mismo, y **estudiar los resultados** que se obtienen con la propuesta y **validarlos**.

1.4 Enfoque del Trabajo y Proceso Recorrido

En sus orígenes este trabajo se empezó a desarrollar a partir de propuestas preliminares desarrollara fundamentalmente con un enfoque cuantitativo basados en la búsqueda de técnicas que permitiesen obtener Programas Productivos eficientes para problemas deterministas caracterizados como de **Taller de Flujo Híbrido con Tiempos de Cambio de Partida Dependientes de la Secuencia**. En la actualidad la tesis doctoral que se presenta ha sufrido un profundo proceso de transformación fundamentalmente por tres motivos:

- En primer lugar el problema de base se ha modificado considerando no sólo la solución al caso determinista sino también a posibles situaciones de **Reprogramación**.
- En segundo lugar, y como consecuencia del cambio anterior, se ha considerado que una fuente de mejora para un problema de programación-reprogramación, podía proceder de los aspectos relacionados con la **relación planificador-programador** de la producción.
- En tercer lugar, y como consecuencia del cambio anterior, se ha cambiado el enfoque meramente **cuantitativo** por uno mixto en el que los aspectos **cualitativos** tuvieran un peso aunque fuese menor que los primeros.

Al mismo tiempo que se modificó el enfoque del problema original, cubriendo un espectro diferente, necesariamente se empezaron a plantear alternativas para construir la solución. Hasta el momento de iniciar este trabajo los fuentes habituales que el autor empleaba para abordar los problemas de secuenciación eran fundamentalmente los modelos matemáticos y los algoritmos heurísticos y metaheurísticos. Sin embargo, motivado por la complejidad del problema se empezó a explorar la posibilidad de aplicar tecnologías emergentes como es el caso de los **Sistemas Multiagente**.

En el documento actual se analizan las relaciones entre los proceso de Planificación y de Programación de la Producción con el fin de plantear mecanismos que, en el marco de la Industria Cerámica, faciliten la Toma de Decisiones para obtener Programas Productivos que permitan ser eficientes tanto al inicio del periodo como en cualquier instante del mismo si surge la necesidad de modificarlo. Esta Toma de Decisiones se basará tanto en herramientas aplicables a la resolución de modelos cuantitativos como en Políticas Discrecionales al servicio de los interés de las Empresas.

1.5 Esquema General de la tesis

Para abordar los objetivos planteados en esta tesis doctoral se ha realizado un trabajo de aproximación al Sector Cerámico. En el capítulo 2 se presentan las características más relevantes del mismo, su significación económica y social, y los procesos básicos que dentro de éste realizan las empresas de producción de baldosas cerámicas. Este capítulo finaliza analizando las oportunidades de mejora, que dentro del ámbito de la Programación de la Producción, se identifican en este tipo de empresas.

Una vez se ha identificado las líneas generales que pueden constituir el camino a explorar, en los capítulos 2 y 3 se analizan las contribuciones más relevantes que pueden ayudar entender, modelar y solucionar un problema como el que se ha esquematizado en el capítulo 2. En concreto en el capítulo 3 se realiza un profundo análisis de estado del arte en los Modelos, los Métodos y los Algoritmos de resolución empleados en la Programación de la Producción. Y en el capítulo 4 se amplía el estudio del estado del arte a los Sistemas Multiagente aplicados a los sistemas de Fabricación y a su Programación de la Producción.

Una vez se dibuja una idea clara del marco de actuación que se disponen en base a lo desarrollado por la ciencia en años anteriores en referencia al entorno de trabajo seleccionado, en el capítulo 5 se realiza la primera descripción del problema. En este capítulo se comentan los aspectos característicos de la parte que se conoce del problema y se modela matemáticamente. Esta es una descripción que se toma como punto de partida pero que aún no se considera completa.

El capítulo 6 constituye el hito a partir del cual se empieza a afrontar el proceso de construcción de una propuesta de solución. En este capítulo se describe la metodología con la que alcanzar el fin deseado, así como los elementos constructivos y las herramientas que deben formar parte de la solución. Además se realiza una aproximación a la solución propuesta.

En el capítulo 7 se realiza el análisis de requerimientos, diseño e implementación de la solución a nivel macro. En este capítulo se da por completada la descripción del problema que se desea

abordar (requisitos a satisfacer) así como se dibuja el esquema básico de los que constituirá la solución: La Plataformaⁱ Software IPSU-MAS.

En los capítulos 8 y 9 se completa el diseño de la plataforma. En el capítulo 8 se diseña, implementa y analizan los algoritmos para la programación de la producción determinista o inicial (Programación Predictiva). En el capítulo 9 se diseñan, implementan e analizan los algoritmos que forman parte del sistema de reprogramación de la Producción (Programación Reactiva).

En el capítulo 10 se realiza una aplicación de la propuesta desarrollada a una empresa de la Industria Cerámica, y se analizan los resultados obtenidos en relación a los métodos empleados actualmente en la empresa. El análisis que se realiza pretende determinar la factibilidad de la propuesta, así como la eficiencia en algunos aspectos como es el de la productividad.

Finalmente, en el capítulo 11 se realiza una reflexión a modo de conclusión del trabajo realizado, y se plantean líneas de investigación para el futuro.

ⁱ Se debe entender que una Plataforma es, en este contexto, un software sobre el cual se pueden ejecutar diversos protocolos para la Toma de Decisión y Procedimientos de Cálculo, en este caso, orientados a la resolución de un problema de Programación de la Producción.

**La Empresa Cerámica y su
Programación de la Producción**

Capítulo

2

Tabla de contenido

2.	Capítulo 2	39
2.1	Introducción	39
2.2	El Sector de Pavimentos y Revestimientos Cerámicos	40
2.2.1	Perspectiva General	40
2.2.2	Evolución Histórica del Sector	42
2.2.3	Innovaciones en el Sector	43
2.2.4	Factores que Obligan a Flexibilizar el Proceso Productivo	45
2.3	Descripción de la Empresa Cerámica	47
2.3.1	Descripción de los Productos	47
2.3.2	Descripción del Proceso de Fabricación	48
2.3.2.1	Descripción General	48
2.3.2.2	Recepción y Tratamiento de Materias Primas	50
2.3.2.3	Molienda	51
2.3.2.4	Atomización	52
2.3.2.5	Prensado y Secado	52
2.3.2.6	Esmaltado	54
2.3.2.7	Preparación de Esmaltes y Tintas Serigráficas	56
2.3.2.8	Cocción	57
2.3.2.9	Clasificación y Embalado	58
2.3.2.10	Paletizado	59
2.3.2.11	El Transporte	59
2.4	Procedimiento de Programación de la Producción de Algunas Empresas Cerámicas...	60

2.5	Oportunidades para la Mejora del Proceso de Programación de la Producción en la Industria Cerámica Actual.	64
2.5.1	El problema de la Incertidumbre.....	64
2.5.2	El Problema de los Tiempos de Cambio de Partida.....	67
2.5.3	El problema de la Programación de la Producción en el Sector Cerámico	68
2.6	Propuesta de un Nuevo Marco de Trabajo	69
2.6.1	Ventajas que se Obtendrían con una Programación ajustada a las Singularidades Identificadas	72
2.7	Conclusiones	73
2.8	Bibliografía	74

Índice figuras

Figura 2-1. Proceso general de monococción porosa (Dalmau, J. I. & De Miguel, E. 1991). ..	49
Figura 2-2. Ciclo de cocción en el horno (ASCER 2007).....	57
Figura 2-3. Visión general del proceso de fabricación de baldosas cerámicas(ASCER 2007). 60	
Figura 2-4. Sistema de fabricación de baldosas cerámicas modelado como taller de flujo (elaboración propia).....	63
Figura 2-5. Relación de causas y efectos de la Incertidumbre en el sistema productivo de las Empresas Cerámicas (elaboración propia).....	65

2.1 Introducción

En este capítulo se realiza la presentación de un entorno industrial, y de un tipo de empresa característica como es la de Fabricación de Pavimentos y Revestimientos Cerámicos. En los próximos apartados se irán comentado las características, peculiaridades y relevancia de este tipo industria con la intención de dirigir el enfoque hacia los aspectos relacionados con la Dirección de Operaciones y las necesidades que actualmente se demandan en este ámbito para convertirlas en oportunidades para desarrollar un proyecto de investigación.

Como condición paso previo en este capítulo se plantea dar a conocer la forma de trabajar de algunas empresas del sector cerámico en relación a su actividad de Programación de la Producción. Este recorrido, fundamentalmente descriptivo, se realizará mediante una aproximación tomando como punto de partida los aspectos generales, por ello en primer lugar, se describe el Sector Cerámico. Se ha considerado que un análisis de la Industria Cerámica requiere como mínimo poner de manifiesto su **importancia económica y social**, tanto en el ámbito nacional como internacional, comentar la **evolución histórica del sector** que permita comprender el estado actual del mismo, subrayar las **innovaciones tecnológicas** más relevantes que ha supuesto hitos evolutivos destacados, y por supuesto considerar algunas de las **tendencias actuales en el ámbito de la dirección de operaciones**. El siguiente estrato que se debe analizar es el de la empresa, como elemento de transformación, por lo que en este segundo paso los **productos y los procesos** son los objetos de estudio. Esta descripción define las restricciones, que por generalizadas en el sector, deberán de tenerse en cuenta en el transcurso de la investigación. El tercer paso que se ha planteado es la descripción general del proceso de **Programación de Producción** en algunas empresas cerámicas.

A partir de un adecuado conocimiento del entorno empresarial y de las prácticas operativas más habituales del sector se plantea fase, básicamente de análisis crítico, con el fin de encontrar nuevas oportunidades de investigación. En primer lugar se identifica la existencia de **incertidumbre en el proceso**, parte de la cual se relaciona con características del producto (tonos, calibres, etc.) pero también de las operaciones que se realizan, como puede ser la existencia de **Tiempos de Cambio de Partida dependientes de la Secuencia**. A partir de esta peculiaridad se cuestiona la validez de los métodos actuales de Programación de la Producción y la posibilidad de plantear métodos nuevos que permitan obtener programas más realistas y ajustados a las necesidades. Se plantea la existencia de **Programación y Reprogramación de la Producción** como forma de adaptarse a los cambios motivados por la incertidumbre.

El resto del capítulo está estructurado como sigue: En el apartado 2.2 se realiza una aproximación a la relevancia y posicionamiento del Sector Cerámico. En el apartado 2.3 se

describen las operaciones básicas en los que se descompone el proceso de fabricación de baldosas cerámicas. En el apartado 2.4 se analiza el Sistema de Gestión de la Producción de algunas empresas significativas del Sector Cerámico. En el apartado 2.5 se analizan las oportunidades de mejora que ofrecen el actual Sistema de Gestión de la Planificación/Programación. En el apartado 2.6 realiza valoración las ventajas que supondría un nuevo marco de trabajo. Finalmente, en el apartado 2.7 se presentan las conclusiones.

2.2 El Sector de Pavimentos y Revestimientos Cerámicos

2.2.1 Perspectiva General

La industria de la cerámica española engloba una amplia gama de actividades con el denominador común de fabricar productos obtenidos a partir de la arcilla que, una vez moldeada y tratada con otros productos, se somete a cocción. Produce materiales para la construcción, para uso industrial y para uso doméstico, empleando procesos de producción muy distintos, con actividades tradicionales y prácticamente artesanales y otras muy industrializadas y de alto contenido tecnológico.

España es el primer productor europeo, con más del 38% del total de la cuota de la Unión Europea a 27. La producción anual se situó en 2008 en 494,7 millones de m² (585 en 2007). Las ventas totales de productos cerámicos alcanzó en 2008 millones de euros (4.120 en 2007), con un elevado componente exportador, pues alrededor del 60 % de sus ingresos, 2.211 millones de euros, proceden de la exportación a más de 179 países, situándose en **una cuota sobre comercio mundial de alrededor del 19%**. El mayor mercado exportador fue la propia Unión Europea (a 27 países) con una cuota superior al 51%. Los datos económicos que se muestran relativos al sector cerámico se basan en (ASCER 2008).

En los últimos años la producción se distribuye fundamentalmente entre productos de pavimentos esmaltados (46,0 %), azulejos (39,1 %), gres porcelánico (12,0) y extruido (2,9). El porcelánico esmaltado alcanza hasta el 75,2% de total de porcelánico.

Aunque ya en el año 2003 España era el segundo productor mundial, con más del 38,5% del total de la cuota. Se podía observar como, según (ASCER 2003), se estaba asistiendo a un incremento de la competencia a escala mundial, debido al aumento de la capacidad productiva de China con el 32,5% de la producción mundial. Existían otros países como Brasil, Indonesia, la India y Turquía con cuotas del 8,4%, 4,0%, 3,8% y 3,2% respectivamente del total mundial. No obstante, el cambio más significativo en el mercado mundial fue el importante incremento

del coeficiente exportador de China, que pasó de una cuota del 2,1% en el año 2000 al 19,8% registrado en 2004, y que en 2007 y 2008 sigue en aumento.

El número total de empresas españolas era de 200 en 2007, a las que hay que sumar una veintena dedicadas a la atomización de tierras y fabricación de bizcocho. El empleo directo es de unos 26.000 trabajadores y se calcula que el indirecto es de unos 7.000 empleos adicionales. **La mayor parte de las empresas son PYMES.** Una de las principales características del sector azulejero español es la alta concentración de la industria en la provincia de Castellón, en especial en el área delimitada al norte por Alcora y Borriol, al Oeste por Onda, al sur por Nules y al este por Castellón de la Plana. Aproximadamente el 94% de la producción nacional tuvo origen en esta provincia, donde están ubicadas el 81% de las empresas del sector. La actual concentración de empresas ha sido justificado por (De Miguel, E. 1996) mediante las siguientes razones:

- Proximidad de los lugares de extracción de arcillas.
- Abundancia de empleados técnicos especializados.
- Posibles economías de escala derivadas de una red empresarial.

En (Molina, J. 1999) se ha mostrado empíricamente como empresas ubicadas en el área de concentración descrita obtenían resultados superiores a las empresas del mismo sector fuera de dicha área. El sector cerámico es el segundo que más riqueza genera en la Comunidad Valenciana, de un total de 104 sectores analizados, sólo superado por el sector de la distribución de mercancías (grandes superficies y detallistas), según el estudio Ardán 2004 del IMPIVA (IMPIVA 2004). Con datos de 2002, el sector azulejero generó el 6,8% del total de la riqueza creada en la Comunidad. Ya en el año 2009, y en plena crisis económica, los productos cerámicos han supuesto un 10% del total de las exportaciones de la Comunidad Valenciana hasta el mes de junio, manteniéndose como tercera fuerza exportadora sólo superada por los automóviles y los frutos comestibles (Instituto Valenciano de Estadística 2009).

La producción española rompió en 2003 su crecimiento ininterrumpido desde 1982, pero en 2004 volvió a mostrar una tendencia ascendente, que ha continuado hasta 2007. Aunque en 2008 se empezó a ver afectada por inicio de la crisis económica mundial. Se estima que en 2007 las ventas tuvo un ascenso del 5,1%, aunque ya en 2008 se produjo un descenso cercano al 12%. Además de la crisis económica, la aparición de China como superpotencia, no sólo productora, sino exportadora ha tenido un gran impacto en el sector cerámico español. Por este motivo las **Empresas Cerámicas españolas se encuentran en un periodo de adaptación a las nuevas circunstancias del mercado, afrontando las nuevas condiciones competitivas.**

En los próximos años la Industria Cerámica deberá desplegar una estrategia competitiva que no se base exclusivamente en el precio del producto. Aspectos como la calidad entendida de forma global (propiedades del producto, diseño producto, atención al cliente, etc.) están siendo la base de las innovaciones adoptadas en la Industria Cerámica española.

2.2.2 Evolución Histórica del Sector

Tradicionalmente los responsables de las Empresas Cerámicas han considerado que una política de inversión en I+D era suficiente para competir con éxito en los mercados, puesto que el factor diferenciador del producto era la calidad del mismo. Esto propició que se llegase a una situación en la que sistemas productivos tecnológicamente muy desarrollados y automatizados, se siguiesen gestionando de una forma muy tradicional con la aparición de diversos problemas de adaptación a las nuevas situaciones del mercado.

Así, en los años 70, el coste había determinado la ventaja competitiva de las Empresas Cerámicas y, por lo tanto, una gestión centrada en los costes fue suficiente para el buen funcionamiento de las mismas. En los 80 se hizo énfasis en la calidad, intentando llevar los conceptos de calidad más allá del producto, e integrándolo en la filosofía global de la empresa (aún hoy muchos sectores se encuentran en una fase de implantación de sistemas globales de calidad). Además, en el sector azulejero, se incorporan mejoras tecnológicas, como son los hornos de monococción (Dalmau, J. I., De Miguel, E., & Miquel, S. 1993)

Así en la actualidad, el **bajo coste** y la **alta calidad** son premisas básicas para desarrollar una actividad empresarial viable en el sector de fabricación de pavimentos y revestimientos cerámicos. A estos se añade el **servicio adecuado al cliente** sin importar su localización o preferencias. Esto propicia una alta personalización de los productos y grandes esfuerzos de distribución. Sin embargo, este rápido ritmo de crecimiento ha favorecido la aparición de una serie de problemas que, la mayoría de las veces, han supuesto un excesivo énfasis en la mejora tecnológica de los procesos dejando de lado la mejora y modernización de su gestión. Esto se ha agravado porque, desde mediados de los 90, se ha producido un estancamiento en los precios y un progresivo aumento de los inventarios (Ybarra, J. A., Giner, J. M., & Santa, M. J. 1996).

En este contexto se ha observado un interés creciente por integrar a todos los eslabones de la cadena logística que intervienen en la elaboración del producto. Esto implica que, para mantenerse en los mercados, además de fabricar buenos productos a un precio asequible y en unos plazos razonables, **es necesario que los proveedores y distribuidores sincronicen sus actividades con las de la empresa**. En este campo cobra cada vez más importancia los

conceptos de **Gestión de la Cadena/Red de Suministro** y de **integración empresarial** que pretende lograr este objetivo último a partir de unas propuestas y unas herramientas modelado, análisis y resolución global de los diferentes procesos desarrollados en las empresas.

En los próximos años se continuarán estas tendencias, puesto que cada vez se acentúa más la expansión global de los mercados fruto de las políticas comunitarias, y la personalización de los productos para clientes cada vez más exigentes y distintos propician la aparición de una gran variedad de productos con baja demanda, que obligan a la empresa (y en particular al subsistema productivo) a ser **fiable** y **flexible** para adaptarse rápidamente a los cambios en las necesidades de los clientes en un entorno de Cadena/Red de Suministro.

2.2.3 Innovaciones en el Sector

En el sector de fabricación de pavimentos y revestimientos cerámicos se han ido sucediendo una serie de hitos, fundamentalmente tecnológicos, que han llevado al sector a la situación actual, donde una gran automatización del proceso es la base de la competitividad del sector en tecnología, calidad y diseño, estos han sido:

- El primer cambio relevante fue la sustitución a mediados de la década de los sesenta del sistema de fabricación tradicional basado en hornos intermitentes por los hornos túneles que aumentaron considerablemente la productividadⁱ de las plantas.
- Después de esto, no es sino hasta la década de los ochenta cuando se incorpora la tecnología de los hornos de monococción (Molina-Morales, F. X. 2002), traídos desde Italia. Esta modernización coincide con una recuperación importante del mercado a escala mundial y un aumento de la producción basado en el tirón del sector de la construcción que duró hasta el principio de la década actual. La implantación del sistema de monococción conlleva el aumento de la demanda de personal cualificado (Técnicos de grado medio y FP II) (Escardino, A. 2001).
- En los últimos años se han alcanzado importantes avances tecnológicos que ha permitido fabricar productos de gran formato (500 mm x 500 mm, 600 mm x 600

ⁱ En este contexto la productividad se debe entender como m² producidos por unidad de tiempo.

mm) basados en la calidad de las materias primas locales. Como con secuencia de esta ventaja, las firmas han desarrollado tecnología mecánica, de esmaltado y de cocción (Molina-Morales, F. X. 2002).

- Con la entrada del siglo XXI la política medioambiental comunitaria se orienta hacia la protección integral del medio ambiente, de manera que el sector industrial debe adaptarse a las nuevas normativas mediante un enfoque de gestión empresarial que integre la variable medio ambiental. (Gabaldón, S., López, S., & Carda, J. B. 2003) analizan las principales mejoras obtenidas como resultado de aplicar innovaciones medio ambientales en el proceso de producción de baldosas cerámicas, destacando: la optimización del grado de reutilización de los residuos, la inertización de los residuos cerámicos, la disminución del consumo de materias primas y energía, la mayor rentabilidad del proceso productivo y la mejora de las características técnicas del producto acabado. Así mismo, concluyen en su trabajo con una previsión a corto plazo sobre la implantación de mayor número de innovaciones medio ambientales que permitan mejorar el proceso productivo.

Cabe destacar el papel jugado por la iniciativa empresarial y la disponibilidad tecnológica y del personal técnico cualificado necesarios para implantar la innovación industrial que ha permitido el extraordinario crecimiento experimentado en los últimos diez años, basado en un entramado socio-económico y de entidades comprometidas con el proceso de innovación (Escardino, A. 2001).

Estos cambios industriales, han tenido las siguientes consecuencias (Gallego, J. R., Bono, E., & Tomas, J. A. 1997):

- Aumento de la relación m^2 /operario. La necesidad económica de reducir los costes salariales en los países industriales ha conducido a automatizar en mayor medida los procesos industriales.
- Ahorro energético (Molina-Morales, F. X. 2002). El proceso de fabricación cerámico, consume grandes cantidades de energía calorífica. Gracias a la construcción del gasoducto procedente de Argelia, se ha podido obtener el combustible necesario para los procesos de fabricación del gres a precios económicos. Además, las ventajosas condiciones ofrecidas por las compañías eléctricas, han posibilitado que las empresas se conviertan en suministradoras de energía eléctrica a la red nacional, merced a la implantación de los procesos de cogeneración.
- La automatización ha influido en los importadores de cerámica. Los países tradicionalmente consumidores de productos cerámicos, hoy en día están en

condiciones de autoabastecerse, convirtiéndose en países competidores, en vez de clientes (por ejemplo China). Como consecuencia de esto, las facturaciones de la Industria Cerámica en Alemania, Italia, Gran Bretaña y Francia, han sufrido un estancamiento e incluso un ligero retroceso, exigiendo un mayor esfuerzo si cabe en la mejora de la calidad y el abaratamiento de los costes de cara a la exportación.

- (Gabaldón, S. et al. 2003) concluyen en su trabajo la previsión sobre la implantación de mayor número de innovaciones medio ambientales a corto plazo que permitan mejorar el proceso productivo.
- Actualmente, parece ser que la situación se ha estancado en lo que se refiere a las grandes innovaciones tecnológicas. En este momento el sector se empieza a sensibilizar por lograr mejoras en lo que se refiere a mejorar sus sistemas de información para la gestión (Vallada, E., Maroto, C., Ruiz, R., & Segura, B. 2005), cambiar o mejorar sus procesos decisionales y aumentar la flexibilidad del proceso productivo para hacer frente a las variaciones del mercado.

2.2.4 Factores que Obligan a Flexibilizar el Proceso Productivo

Según el estudio realizado por (egre-Vidal, J., Lapiedra-Alcami, R., & Chiva-Gomez, R. 2004) en el cual se analizó la relación entre las operaciones estratégicas de una compañía y su capacidad innovadora, se concluyó que en la Industria Cerámica española las firmas más innovadoras se diferenciaban del resto por dos factores fundamentales: la calidad y la flexibilidad. Este estudio coincide en sus conclusiones con los modelos de (Ferdows, K. & De Meyer, A. 1990) según el cual los objetivos de todas las operaciones son fundamentalmente, la calidad en primer lugar, y luego la entrega y la flexibilidad, y finalmente la eficiencia en el coste. Así mismo, en (Segura, B., Vallada, E., Maroto, C., & Ruiz, R. 2004) se realizó un estudio estadístico basado en encuestas con empresas del sector cerámico en el cual se concluyó las prioridades medias más altas se obtienen para aspectos vinculados con al **servicio, seguido del coste y la flexibilidad**.

Debido a la necesidad de incrementar el grado de innovación, en las Empresas Cerámicas se ha venido observando en los últimos años una necesidad creciente de flexibilizar el sistema productivo, debido a los siguientes **factores**:

- **Variabilidad del volumen demandado.** La demanda de los productos cerámicos procede básicamente del sector de la construcción, tanto en lo que se refiere a nuevas edificaciones como a obras de renovación y rehabilitación. El hecho de que se traten de elementos visibles y que forman parte del mobiliario y la decoración de casa determinan que su renovación se lleve a cabo por razones estéticas y no solamente debido a su deterioro. El diseño y la moda son factores muy importantes

en este tipo de productos, que eleva su valor añadido y posibilita su comercio exterior. En la primera mitad de la década de los 90 el sector tuvo una ligera recesión originada por la caída del consumo y la disminución de la construcción de viviendas, que originó un aumento considerable de los inventarios de producto acabado y puso en evidencia que el mercado tradicional había cambiado y que las empresas debían comenzar a gestionar de una manera menos rígida sus procesos de fabricación.

- **Reducción de los ciclos de vida de los productos.** El azulejo se ha convertido en un producto de moda y productos que antes tenían un ciclo de vida en torno a la década, actualmente se renuevan a los tres o cuatro años. Es necesario, permanentemente renovar los catálogos, con lo que el peligro de obsolescencia (si se fabrica en grandes lotes) es cada vez mayor.
- **Aumento de la variedad y personalización de la oferta.** En el segmento de pavimentos cerámicos se ha acrecentado una variación en los hábitos de compra de los consumidores motivadas por un aumento de la exigencia (en precio y en servicios) y una mayor madurez (y por lo tanto infidelidad a la marca). Un exceso de oferta de productos se ha traducido en un aumento de la competencia y por lo tanto en la necesidad de diferenciarse frente a ella. Frente a esto, las empresas han reaccionado multiplicando los formatos y acabados por lo que la tendencia general es tener extensos catálogos que complican la gestión de los almacenes y la Planificación de la Producción.
- **Disminución de los plazos de entrega.** La poca diferenciación entre los productos propicia que el servicio al cliente se haya convertido en un factor clave en la competitividad de las Empresas Cerámicas. La venta directa al cliente es rara en el sector, siendo lo más común utilizar uno o varios almacenistas intermedios. Esto los convierte en elementos de la cadena logística con gran capacidad de presión sobre las empresas fabricantes. Entre los criterios tenidos en cuenta por estos almacenistas a la hora de elegir un fabricante, uno es precisamente el ajuste a los plazos de entrega, que obliga a las Empresas Cerámicas a reducirlos a base de acumular inventario de producto acabado y mejorar la Planificación de la Producción (ASCER 1998).

Como conclusión a todo lo dicho anteriormente, se ha venido observando en el sector cerámico una aspiración creciente en conjugar las ventajas de fabricar en serie pero con la filosofía de fabricación de pequeños lotes. Sin embargo, tal y como se discutirá el siguiente apartado, existen varios impedimentos que dificultan la consecución de este objetivo de fabricar en pequeños lotes y con rápidos tiempos de respuesta.

2.3 Descripción de la Empresa Cerámica

2.3.1 Descripción de los Productos

En el proceso fabricación tradicional de gres se puede hablar de dos tipos de procesos, denominados de monococción y bicocción. En cada tipo de producción se pueden distinguir varios tipos de formatos (tamaño). A su vez, cada formato posee diferentes modelos que serán función de la forma y de las aplicaciones que reciben (dibujo, esmaltes, etc.).

En el proceso de monococción el esmalte se aplica directamente sobre la pasta prensada y cruda, ambas se cuece simultáneamente para dar el acabado final. En la bicocción, la pasta prensada se pasa por el horno para formar el bizcocho y, posteriormente se aplica el esmalte sobre éste y se cuece nuevamente para dar el acabado final.

Durante muchos años ha existido la controversia acerca de cual de los dos métodos es mejor. En realidad, teniendo la formulación adecuada tanto del cuerpo como del esmalte, y observando un rígido control de todas las etapas de fabricación, es posible producir un buen azulejo con cualquiera de los dos métodos.

Tradicionalmente era más utilizado el proceso de bicocción, con ciclos de cocción de cuarenta y veinte horas para la primera y segunda cocción respectivamente (cocción del soporte y del esmalte). Actualmente es más interesante el proceso de monococción, con ciclos de sólo cuarenta o cincuenta minutos de duración.

Además, junto a la economía del proceso de monococción va unida una gran facilidad para la automatización de los diferentes procesos de fabricación, con el consiguiente resultado en la reducción de costes.

Los pavimentos y revestimientos cerámicos se obtienen preparando una composición de materias primas depuradas formada por silicatos aluminicos, siendo las composiciones diferentes para el caso de los pavimentos y revestimientos cerámicos en pasta roja o en pasta blanca.

En todo caso, en cualquier grupos de productos continuamente se van eliminando modelos que no tienen suficiente demanda e incorporando otros nuevos, por lo que el número de modelos fabricados sufre continuas modificaciones, tendiendo a aumentar el número de un año a otro.

Ambos procesos son totalmente independientes (esto es, cada producto se puede fabricar solo en uno de los procesos y no comparten recursos), por lo que en la tesis, el **análisis se centrará en el primer proceso (monococción)**, puesto que es el que más complejidad representa debido al número de recursos involucrados (Alfaro, J. J. 1997).

Un producto es una pieza cerámica caracterizada por un modelo, un formato, un calibre, un color y tono. El modelo está relacionado con los aspectos estéticos y con el diseño (textura, dibujos, etc.) y se identifica con un nombre propio y de la serie al que pertenece (por ejemplo modelo Países de la serie España). El formato identifica sus dimensiones en el plano (ancho y largo en centímetros). El calibre determina el grosor de la pieza (en milímetros). El color está relacionado el valor numérico grueso que se asociará a la pieza en el pantone de colores, y se identifica con un nombre (rojo, negro, etc.). El tono determina el valor numérico fino que se asociará a la pieza en el pantone de colores. Se puede decir que el producto está formado por características esenciales y auxiliares. Las primeras son aquellas por las cuales son conocidos por los clientes y solicitados por los mismos, es decir modelo, formato y color. Mientras que las segundas, tono y calibre, se determinan definitivamente en el proceso de fabricación y diferencian a los productos, incluso de un mismo lote, en subconjunto diferenciados por no poder ser utilizados en un mismo cliente final.

2.3.2 Descripción del Proceso de Fabricación

2.3.2.1 Descripción General

La fabricación del revestimiento cerámico se realiza a partir de arcillas, que se someten a un tratamiento de molturación vía húmeda y, posteriormente, de atomización. La arcilla atomizada se prensa, formando unas piezas, que en el caso de bicocción se cuecen antes de ser esmaltadas (bizcocho), luego se esmalta y sufre una segunda cocción (fino), como ya se ha comentado en el apartado 2.3.1, en monococción directamente pasan a esmaltarse.

Por último, sea cual sea el proceso, el producto esmaltado pasa al horno. Entre las líneas de esmaltado y los hornos existen almacenes intermedios debido al distinto ritmo de producción que hay en cada sección.

Una vez cocido el producto, el producto se transporta a una zona de almacenamiento a la espera de ser clasificado en diferentes calidades y características físicas (tono y calibre) por medio de máquinas sofisticadas. Al mismo tiempo, un operario analiza los defectos de superficie. La propia máquina de selección confecciona las cajas de cartón en las que se envasa

el producto. Un robot recoge estas cajas y las almacena en los palets que se transportan al almacén de producto acabado. El producto queda ya listo para su expedición.

Tradicionalmente la utilización de un sistema de fabricación u otro ha correspondido sobre todo al tamaño del formato a fabricar, así para la fabricación de formatos de gran tamaño, se ha empleado el proceso de bicocción, y para formatos inferiores sea empleado el proceso de monococción porosa. Esta diferenciación se ha debido a la fragilidad que presentan los formatos grandes cuando se prensa el bizcocho por lo que, para evitar roturas a lo largo del proceso, los azulejos debían ser cocidos antes de pasar por las distintas aplicaciones en la línea de esmaltado. Por ello se les daba una primera pasada por el horno, para que adquieran cierta dureza y así poder ser manipulados en las líneas de esmaltado sin riesgo de rotura (Tormo, G. 1997). Actualmente, el proceso de monococción predomina en la Industria Cerámica, incluso para los formatos de gran tamaño, debido a los avances técnicos.

A continuación, se muestra un diagrama (Figura 2-1) donde se detallan las fases que tienen lugar en el proceso productivo de monococción porosa. Estas fases se explican en los siguientes apartados.

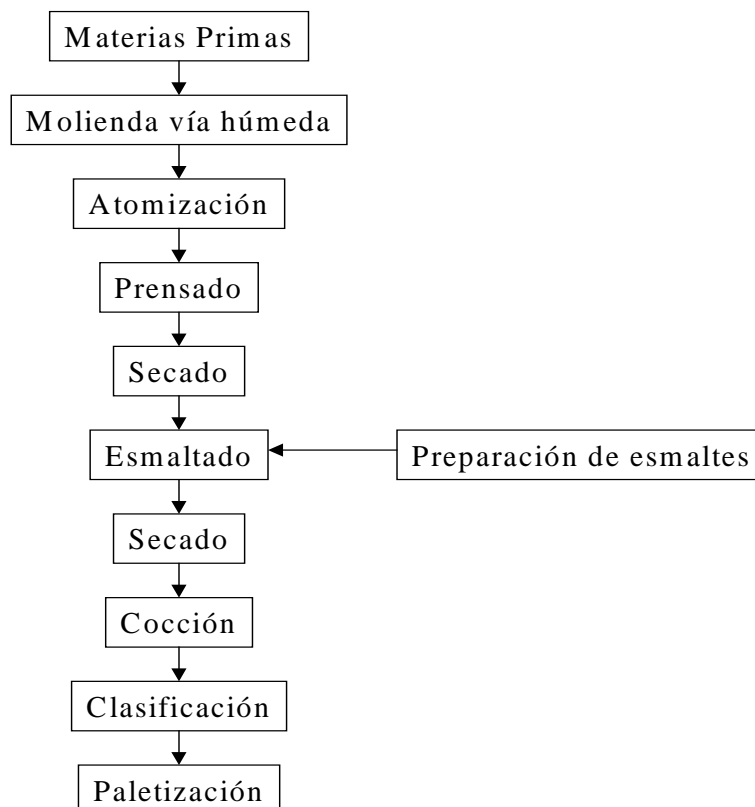


Figura 2-1. Proceso general de monococción porosa (Dalmau, J. I. & De Miguel, E. 1991).

A continuación se describen uno por uno todos los procesos detallados en la Figura 2-1, basados en (Andrés, C. 2001).

2.3.2.2 Recepción y Tratamiento de Materias Primas

El proceso se inicia con la obtención de la materia prima con la que se fabrica la base del azulejo. Las arcillas llegan a la empresa mediante camiones desde minas a cielo abierto distribuidas en las proximidades de la zona.

Las arcillas se deposita en estratos horizontales, en una superficie (era) dedicada al almacenamiento de materias primas, cuyo principal objetivo es conseguir una homogeneización óptima para conseguir una mayor uniformidad en las propiedades de las arcillas.

Cada uno de los tres tipos de arcilla con las que se trabaja, es depositado en montones (eras) independientes y se van formando capas de un espesor aproximado de medio metro entre cada par de estratos, introduciéndose entre ambos una fina capa de polvo que se recupera del ciclo productivo. Esto se hace con objeto de evitar que cuando llueva se filtre el agua de los estratos inferiores y estos se humedezcan por encima de un valor mínimo óptimo para trabajar.

Se puede disponer de dos tipos de eras:

- Eras de consumo. Se extraen las materias primas que van a ser utilizadas en el proceso. La extracción se realiza mediante palas mecánicas que toman la arcilla perpendicularmente a como han sido formados los estratos para conseguir un material lo más homogéneo posible.
- Eras de formación. Se van formando los estratos para abastecer las eras de consumo.

Para obligar a que la lluvia no detenga la producción, se dispone de unos graneros techados en los que las arcillas también se almacenan en montones independientes. Diversos controles a las arcillas aseguran la óptima calidad de las mismas, así se realiza uno a la entrada a eras y otro antes de la molienda.

Para evitar problemas en la etapa de molienda ya que las arcillas tienen una granulometría muy variada, se realiza una reducción de tamaño de las arcillas mediante unos trituradores de martillos que consiguen reducirlas, gracias a varios cilindros dentados, a partículas no superiores a tres o cuatro centímetros. Una pala mecánica se encarga de llevar las arcillas hasta el triturador.

Una vez triturada cada arcilla, pasa mediante una cinta transportadora, a unos silos de acopio, desde los que se realiza la dosificación a los molinos. Estos silos suelen estar conectados a un sistema informatizado de pesada que marca los tipos y la cantidad de arcilla que se requiere. Una vez realizada la pesada de los materiales éstos pasan mediante cintas transportadoras móviles cubiertas (para evitar la acción de agentes meteorológicos durante su trayecto), a una serie de tolvas que se encuentran situadas en la zona de carga de molinos. Para cada molino la carga se realiza mediante una tolva de arcilla y un depósito de agua situados en la parte superior.

2.3.2.3 Molienda

Esta etapa tiene dos objetivos. Por un lado, aumentar la superficie del material por unidad de masa, y por otro, obtener una distribución de tamaños de partícula adecuada a las necesidades del proceso y producto.

La molienda se lleva a cabo mediante molinos de bolas y se realiza por vía húmeda (en presencia de agua), ya que se consigue una mayor homogeneización de las materias primas y permite obtener polvo granulado con mayor fluidez que el obtenido por vía seca, facilitando la obtención de mejoras en el proceso productivo y con ello mayor calidad en el producto final.

Los molinos contienen en su interior bolas de sílex como elementos de molienda, siendo éstos de diferentes tamaños y capacidades. Al final del proceso se obtiene una mezcla de la arcilla, agua y defloculante que se denomina barbotina. Para ello, el molino está en marcha durante unas 10-15 horas.

Una vez finalizado el proceso de molienda, el contenido del molino (barbotina de arcilla) se vierte y almacena en constante agitación en la llamada balsa de sucio. Este líquido espeso es tamizado en tamices vibratorios, eliminando todas las partículas de tamaño excesivo, y guardado en las balsas de limpio quedando listo para el siguiente proceso de atomización.

El agua utilizada en el proceso proviene de unas balsas en las que se almacena los volúmenes provenientes de la limpieza de la fábrica y lluvia, con lo que se evitan vertidos al medio ambiente.

2.3.2.4 Atomización

Mediante la operación de atomización o globulación se logran dos objetivos básicos:

- Secar la barbotina para adecuar la humedad del sólido a la óptima requerida en la etapa de prensado.
- Obtener un polvo cuyos gránulos presentan una morfología esférica que mejora la fluidez del sólido y facilita que durante el llenado del alvéolo de la prensa se obtenga una distribución de masa uniforme.

Desde las balsas de limpio la barbotina se bombea al atomizador mediante bombas de pistón doble refrigeradas por agua, pasando antes por unos tamices. El objetivo de los atomizadores es pulverizar la barbotina en contracorriente con un flujo de aire caliente (que puede provenir de un quemador vertical de gas o una turbina de cogeneración). Al contacto con los gases calientes sufre un secado violento quedando convertida en polvo ligero.

El total de atomizado producido es conducido mediante cintas transportadoras a unos silos de almacenamiento.

Los silos presentan aspiraciones internas para eliminar el polvo producido en la carga así como la humedad, evitando de este modo que se pegue el atomizado a las paredes internas del silo produciendo costras. De los silos, el atomizado parte mediante cintas transportadoras hacia la sección de prensas.

2.3.2.5 Prensado y Secado

Con la mezcla previamente atomizada, se alimenta al carro de la prensa, donde se configura el producto final. Para ello, se suele disponer de prensas oleodinámicas de alta potencia.

Con el prensado se da forma a la pieza y se le dota de una resistencia mecánica que permita que la pieza conformada pueda ser transportada en las siguientes etapas del proceso.

Los moldes son los que dan la forma geométrica exacta a la pieza aprovechando la fuerza ejercida por el pistón de la prensa. En ellos pueden diferenciarse tres partes:

- Punzones superiores (marca comercial).
- Punzones inferiores (características superficie vista).
- Matriz (tamaño y calibre).

La configuración de la prensa, en especial su potencia, determinan el tipo de formato que puede fabricarse con este recurso.

Los trabajos de preparación de las prensas consisten en el ajuste de estos tres elementos hasta obtener la configuración física deseada. Dichos trabajos suelen suponer la manipulación de maquinaria pesada por lo que los recursos necesarios para llevarlos a cabo son importantes evitándose los cambios simultáneos en varias prensas. Esta fase preparatoria suele ser la más laboriosa de todo el proceso de fabricación.

El siguiente proceso al que es sometida la pieza después del prensado es el secado, cuya misión es eliminar agua de la misma para, posteriormente, conseguir que la adherencia de las aplicaciones de esmalte no provoque fallos de tipo superficial.

A través de cintas transportadoras se conducen las piezas al secadero. Durante este transporte se someten a un proceso de cepillado, tanto de la cara superior como de la cara inferior, con el fin de eliminar el polvo que queda adherido a la pieza después del proceso de prensado.

El principio de funcionamiento de los secaderos automáticos verticales se basa en realizar un flujo contracorriente entre el material a secar y aire caliente producido por quemadores de gas natural. Los azulejos se sitúan en cestas con movimiento descendente, mientras que el aire caliente asciende.

El tiempo de permanencia de la pieza en el secadero, así como la temperatura utilizada, depende fundamentalmente del espesor. La temperatura y la humedad se controlan periódicamente.

La operación de secado dura aproximadamente unos 12 minutos y la temperatura de entrada del aire al secadero es de 220º dependiendo del modelo que se esté realizando.

2.3.2.6 Esmaltado

Las instalaciones de una planta de monococción porosa constan de líneas de esmaltado disponiendo cada una de ellas en su cabecera de su correspondiente prensa y secadero vertical. El tipo de producto a fabricar en una línea depende del formato y, principalmente, del número de aplicaciones que va a llevar el modelo a esmaltar.

Las piezas salen del secadero vertical a una temperatura óptima para el proceso de esmaltado. Seguidamente, las piezas sufren un proceso de cepillado por la parte inferior, para evitar que el polvo que puedan llevar adherido contamine las correas en las campanas de engobe y base, contaminando a su vez al esmalte. Un soplador se encarga de eliminar el polvo en la parte superior de las piezas.

Un aerosol de agua homogeneiza la temperatura superficial de la pieza al mismo tiempo que tapa los poros superficiales. En la primera parte de la línea, cada unidad recibe dos aplicaciones por medio de una serie de campanas denominadas engobe y base, respectivamente. La aplicación de la base se debe realizar justo cuando el engobe recién aplicado empieza a secar (esto se consigue regulando la velocidad de la línea). La finalidad de estas capas es ocultar el color rojizo de la pasta, eliminar o corregir los defectos superficiales del soporte y mejorar la adherencia esmalte-soporte.

A continuación, hay un largo tramo de línea que se encuentra techado para evitar que el polvo del ambiente se adhiera a la pieza ocasionando defectos. Durante este tiempo la base se va secando y la pieza va evaporando agua. Más tarde se friega o rasca lateralmente todas las piezas, mediante unas muelas eléctricas al mismo tiempo que un soplador elimina las posibles virutas que hayan podido saltar sobre la pieza.

Seguidamente, se pasa a la decoración de la pieza mediante distintas aplicaciones, que son realizadas por diversos dispositivos específicamente diseñados. Estas instalaciones se pueden montar y desmontar sobre la cinta transportadora, para variar la configuración de la línea en función del tipo de producto a fabricar. Las posibles **aplicaciones** que puede recibir un modelo son:

- **Aerosol de agua:** La función de este aerosol es la de regular la humedad superficial de la pieza, además el agua penetra en los poros de la pieza llenándolos con lo que la aplicación del engobe se hace sobre una superficie de muy poca porosidad con lo que se evitará la formación de pinchazos.

- **Aplicación de esmalte:** Los esmaltes se aplican a campana, con ello se logra un mayor control sobre la aplicación. Este esmalte es el que proporciona el color de fondo del azulejo. Los esmaltes más utilizados son las cristalinas (negra y coloreadas), blanco de circonio y mates.
- **Serigrafiado:** Es un sistema consistente en hacer pasar un esmalte (tinta serigráfica) a través de una tela especial (pantalla). La tinta pasa al presionarla mediante una espátula de goma por los puntos de la tela donde existe un orificio, como la pantalla se encuentra tocando el esmalte de la pieza la tinta quedará adherida a la pieza.

La espátula realiza dos movimientos, en el primero recoge tinta y pasa sobre la pantalla sin apretar, llenando así los orificios de ésta. En el segundo movimiento, en sentido contrario al anterior, la espátula ya aprieta la pantalla sobre la pieza. La alimentación de tinta en la pantalla puede ser manual o automática.

El serigrafiado define el ritmo de producción en el esmaltado. Existen varias de estas máquinas de serigrafiado en cada una de las líneas de esmaltado, para poder así aplicar variedad de dibujos y colores a cada modelo.

Antes del serigrafiado propiamente dicho, se aplica a la pieza una ligera capa de fijador para mejorar la adherencia de la serigrafía.

- **Serigrafiado rotatorio:** se disponen las pantallas alrededor de un cilindro el cual gira a una velocidad determinada, consiguiendo mejorar la productividad de este proceso.
- **Efecto veteadado:** El esmalte se aplica en seco sobre la pieza una vez que ya ha sido esmaltada para crear efectos de veteadado, intentando imitar piedras naturales. Este polvo se aplica con una máquina especial. Esta aplicación se lleva a cabo entre el engobe y el esmaltado. Se requiere que el engobe este suficientemente tierno, para que se lleve a cabo la absorción del polvo y no se den pinchados de esmalte.
- **Cabañas de discos:** permiten la aplicación de efectos lanzando el esmalte a través de unos discos en rápida rotación, lo que da un resultado irregular y de goteo.
- **Fumé:** sistema que consiste en rociar el esmalte con un aerógrafo utilizando una plantilla para el dibujo.
- **Granilladoras:** depositan esmalte en polvo sobre el azulejo, dando resultados rugosos.

- **Compensadores:** distribuidos a lo largo de la línea hay varios compensadores que se encargan de almacenar la producción que llega por la cinta cuando se produce algún pequeño paro aguas abajo.
- **Varios:** elementos para la limpieza de los bordes, volteadores, guías de dirección, sopladores de polvo, cintas transportadoras, etc.

A lo largo de toda la línea, distintos controles aseguran la correcta calidad de las piezas y de los componentes utilizados.

Las piezas una vez esmaltadas se pasan por unos secaderos para reducir el contenido de humedad hasta aproximadamente el 0.5 %. De esta manera, se evitan posibles explosiones y defectos que podrían originarse si se introducen directamente al horno.

Los secaderos son unas estructuras con bandejas de rodillos, donde se introducen los azulejos a los que se les inyecta una mezcla de aire externo y aire caliente procedente de la zona de enfriamiento del horno, mediante una serie de conducciones. El tiempo de residencia de las piezas en el secadero depende del tiempo de almacenamiento de éstas en el parque, de la descarga a hornos y del espesor de las piezas.

Una vez salen del secadero las piezas se traspan a la línea de alimentación a hornos, la cual dispone de otro compensador para evitar la discontinuidad en la entrada al horno.

2.3.2.7 Preparación de Esmaltes y Tintas Serigráficas

Estos procesos auxiliares se realizan previamente al inicio de la producción. Los esmaltes suelen ser de fabricación propia. Para fabricar los esmaltes se utilizan fritas, colores, aditivos y agua. La frita llega en sacos de producto y su forma es la de cristales. Todas estas materias primas se introducen en unos molinos especiales, en unas cantidades adecuadas y se muelen hasta que la granulometría y viscosidad son las prescritas. Se almacenan entonces en cubas constantemente homogeneizadas mediante unos batidores hasta el momento de su uso.

La pasta serigráfica se fabrica a partir de polvo base. Éste se dispersa en un vehículo serigráfico (que generalmente es una mezcla de resinas sintéticas), con ayuda de un agitador a alta velocidad. La preparación de la pasta se hace a través de dos métodos:

- La pasta se hace pasar por una máquina denominada tricilíndrica, la cual consta de tres rodillos que giran en sentido diferente cada uno. Con esto, se consigue una laminación del material, evitando así la formación de pequeños grumos en la pasta y consiguiendo una mayor integración de los componentes de la mezcla.
- La pasta ya dispersada se consigue pasar por un tamiz, ejerciendo presión mediante una espátula para facilitar el paso del material a su través. De este modo se consigue la eliminación de grumos que podrían ocasionar problemas posteriores en las pantallas serigráficas.

2.3.2.8 Cocción

Para esta función se utilizan los hornos. Su función es cocer el bizcocho esmaltado para dar el producto final. La alimentación de los hornos es mediante gas natural. En su paso por el horno el bizcocho atraviesa varias secciones denominadas pre-horno, pre-calentamiento, cocción, enfriamiento natural y enfriamiento forzado, respectivamente, siendo el sistema de movimiento de las piezas mediante rodillos. El control del horno para la regulación de la temperatura se realiza mediante un sistema de control por microprocesadores.

Del horno sale el producto a una temperatura elevada, pero ya con las características finales de resistencia y dureza adecuadas. Los factores que afectan principalmente a la cocción son el rango de cocción y el tiempo de permanencia a la máxima temperatura. Otras variables muy importantes para el óptimo funcionamiento del horno son la uniformidad de la temperatura del horno, el control de la curva de cocción y el control de la atmósfera del horno. La temperatura máxima de cocción del bizcocho es 1120° C aproximadamente, siendo la permanencia de cada pieza en el horno de unos 42 minutos aproximadamente.

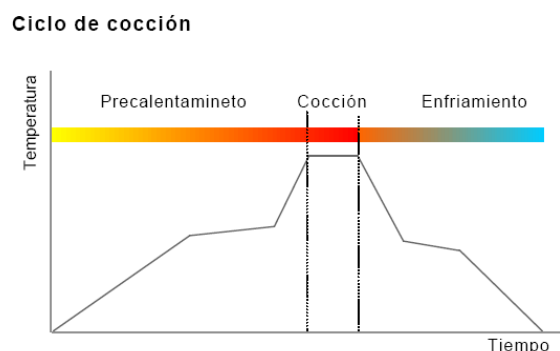


Figura 2-2. Ciclo de cocción en el horno (ASCER 2007).

Las variables fundamentales a considerar en la etapa de cocción son, el ciclo térmico (temperatura-tiempo, Figura 2-2), y la atmósfera del horno, que deben adaptarse a cada composición y tecnología de fabricación, dependiendo del producto cerámico que se desee obtener.

Sin embargo, estas temperaturas pueden cambiar ligeramente en función del tamaño y del formato. A mayor tamaño y espesor mayor tiempo de cocción.

2.3.2.9 Clasificación y Embalado

Las piezas una vez han salido del horno pasan al almacén donde se almacenan para su posterior clasificación. En esta sección se somete a los azulejos, uno a uno, a varias pruebas para clasificarlos por calidades, calibres y tonos (resistencia mecánica, clasificación visual, planaridad, calibres y tono).

En la clasificación de las piezas se distinguen tres calidades:

- Primera: el azulejo no presenta ningún defecto visible.
- Segunda: el azulejo incluye despuntados, grietas, hoyos, picaditas de tamaño pequeño y en poca cantidad, etc.
- Tercera: el azulejo contiene los mismos defectos que los de segunda pero en mayor cantidad y tamaño, además de defectos de serigrafía y otras aplicaciones.

Los azulejos con defectos más graves como despuntados o desconchados se clasifican como tiesto, las piezas así clasificadas son desviadas y caen a un depósito. Posteriormente serán molidas para su reincorporación a las nuevas arcillas de las eras.

Al mismo tiempo que se realiza la clasificación por calidades los azulejos se clasifican por tonos y calibres dentro del mismo producto que ya venía definido por un formato, un diseño y un color. Dicho de otra forma el tono y el calibre son características variables del producto, que si bien se mantienen dentro de un rango, no se conocen con precisión hasta el final del proceso.

La tecnología actual permite que el proceso de clasificación este altamente automatizado. Este proceso consiste en el paso, mediante una cinta transportadora, por un sistema formado por elementos de visión artificial y otros sensores que informan a un sistema informático que

asigna al azulejo a un grupo. El azulejo continúa por la línea hasta que llega a la posición asignada a los azulejos de su grupo donde es desviado a un nivel inferior donde permanecerá hasta completar un número preestablecido dependiente del formato. Cuando ese número se alcanza se traslada mediante otra cinta transportadora a la entrada de la máquina empaquetadora que identifica en su envoltorio no sólo el modelo, formato y color sino su tono y calibre.

Finalmente las cajas son transportadas por una cinta hasta la siguiente sección de paletizado.

2.3.2.10 Paletizado

De la sección de la clasificación sale el producto embalado en cajas codificadas según calibre, calidad y tono. La paletizadora se encarga de llenar los palets de forma automática teniendo en cuenta estos datos. Una vez está el palet ya conformado se enfunda manualmente con un plástico biodegradable mediante la aplicación de calor. Los palets se llevan a otra nave donde se ubica el almacén mediante la utilización de vehículos mecánicos.

La situación de los palets en el almacén atiende a un programa de control de almacenes diseñado para un óptimo aprovechamiento del mismo, así como para una eficaz gestión y control de cada palet.

2.3.2.11 El Transporte

Entre líneas de esmaltado y hornos, y hornos y líneas de clasificación las unidades se depositan mediante sendos manipuladores en unas estructuras compuestas por diferentes repisas denominadas vagonetas. Para el transporte de estas vagonetas se utilizan vehículos filoguiados motorizados eléctricamente (AGV), que se desplazan sin necesidad de raíles mediante la acción de un sistema de radio control. El recorrido de los vehículos viene prefijado por un trazado de cables bajo el suelo que se encarga de dirigir la trayectoria. Todo el mecanismo está gestionado por un autómatas, que manda la información a los vehículos mediante ondas de radio. Este sistema proporciona una gran flexibilidad a la hora de programar la entrada de los trabajos a la siguiente sección.

En la Figura 2-3 se puede apreciar de forma resumida el proceso completo de fabricación de baldosas cerámicas.

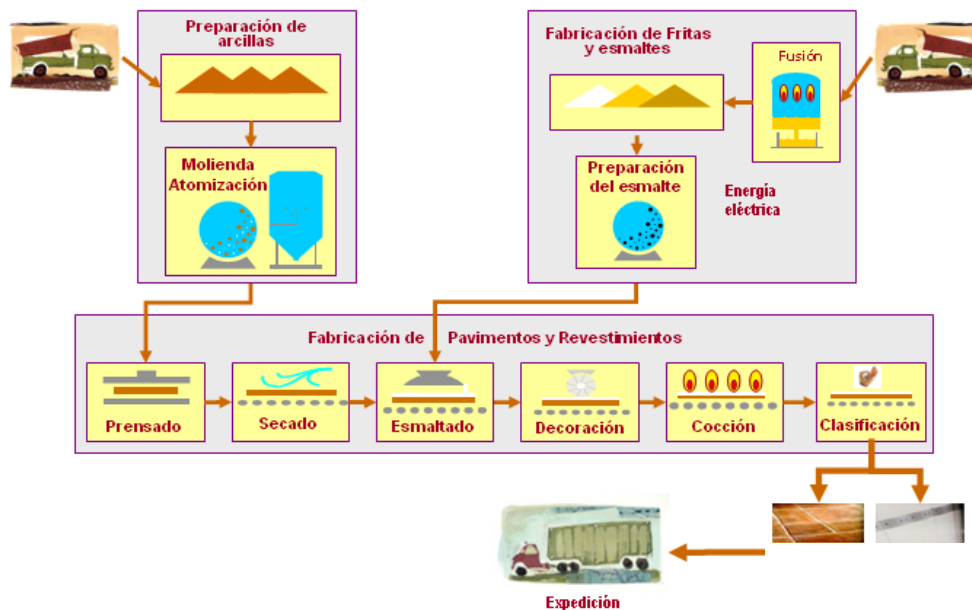


Figura 2-3. Visión general del proceso de fabricación de baldosas cerámicas(ASCER 2007).

2.4 Procedimiento de Programación de la Producción de Algunas Empresas Cerámicas.

Actualmente, en las empresas del Sector Cerámico existe una situación predominante de lo que se conoce tradicionalmente como fabricación contra almacén. La tendencia a largo plazo es llegar a una situación cercana a la fabricación bajo pedido, debido a las exigencias cada vez mayores de un producto personalizado y cambiante, que provoca una rápida obsolescencia. Sin embargo, debido a los elevados lotes de fabricación, tiempos de ajuste y la gran variedad de modelos que se suele dar en los Programas de Producción, los **plazos de fabricación**ⁱⁱ (Lead Time) **son elevados** (se estiman en una semana y media en promedio). Esto provoca la necesidad de una Planificación basada en la previsión y mantenimiento de cierto nivel de inventario de algunos productos, por lo que la mayoría de las veces, se fabrica contra almacén para asegurar una capacidad de respuesta rápida.

ⁱⁱ Tiempo promedio transcurrido entre que se origina la orden de fabricación y se finaliza su producción

Para comprender el proceso de Programación es necesario introducir una serie de conceptos relacionados dicho ámbito táctico-operativo de la Gestión de los Sistemas Productivos que aunque habituales en el ámbito de la Dirección de Operaciones, serán ampliamente comentados en el capítulo 3, correspondiente al análisis del Estado del Arte de la Programación de la Producción.

En la fabricación contra inventario el proceso de definición del Plan Maestro de Producciónⁱⁱⁱ es fundamental. Aunque dicho proceso queda fuera de los objetivos de esta tesis se reconoce el interés de analizar la problemática de definición del mismo, puesto que se ha observado en muchas Empresas Cerámicas que uno de los problemas que aparece en el ámbito de **Programación de Producción** es una deficiente definición del Plan Maestro, que provoca frecuentes cambios y distorsiones en el desarrollo del mismo. Será necesario determinar los requerimientos en la relación entre el Plan Maestro de Producción y el Programa de Producción.

Debido a la vinculación entre Planificación y Programación que se vislumbra es interesante realizar una breve introducción, que será ampliada en capítulos posteriores, respecto al procedimiento que se sigue en algunas Empresas Cerámicas para alimentar su sistema productivo. En las algunas Empresas Cerámicas^{iv} se suele definir un Plan Maestro que sufre sucesivas modificaciones en un proceso de refinamiento continuo. En general, el proceso se suele iniciar con un Plan Maestro a Capacidad Infinita y con un horizonte de doce meses basado en demanda prevista, datos históricos, estrategias corporativas, lotes mínimos, capacidad almacenamiento y aprovisionamiento, etc. Este plan marca las grandes líneas de actuación a un año vista. A partir del plan anterior se generan otros planes más detallados, normalmente un mínimo de dos planes simultáneos, con horizontes de tres meses y de seis semanas respectivamente. Estos planes se basan en la agrupación por modelos a partir de los **Pedidos**^v de los distintos clientes y **Previsiones** de stock deseadas. En el plan con menor horizonte se generan la **Lista de Órdenes**^{vi} **de Fabricación** donde se especifican los distintos

ⁱⁱⁱ Entendido como “un Programa anticipado de producción” que representa lo que la compañía planifica que va a producir expresado en configuraciones, cantidades y fechas específicas” (Apics 1994)

^{iv} Información extrapolada a partir de conversaciones mantenidas con diferentes responsables de producción de empresas fabricantes de baldosas cerámicas.

^v Estos pedidos se componen de diferentes modelos y formatos y son realizado por el mismo cliente.

^{vi} Lista de Órdenes: Listado de Trabajos a realizar agrupados en Órdenes de Fabricación. Cada Orden supone una mera agrupación temporal de trabajos.

Trabajos^{vii} (donde se especifica modelos y tamaño de lote a fabricar). Esto se hace en función del ratio de agotamiento de cada modelo en el almacén (puesto que un objetivo es el mantenimiento de un cierto nivel de Inventario de Seguridad de cada producto). A veces, cuando se detecta que un producto está próximo a agotarse, se origina un Trabajo, que se añade a la Lista de Órdenes de Fabricación, cuyo tamaño de lote suele oscilar entre un valor mínimo que ronda los 1.500 m² y máximo situado alrededor de los 10.000 m² definidos en función de la experiencia de los responsables de fabricación y en previsiones proporcionadas por el **Departamento de Planificación**. Así se puede observar que en el mismo Plan Maestro coexisten distintos tipos de Trabajos: por un lado están aquellos que han sido originados por pedidos en firme de clientes (descontando las posibles existencias), mientras que por otro se pueden dar Trabajos Previsionales, destinados a mantener un cierto nivel de inventario para satisfacer demandas imprevistas.

En este Plan Maestro, a cada orden de fabricación se le asigna una fecha de entrega marcada por el compromiso alcanzado con los clientes, basándose en criterios fundamentalmente comerciales adoptados por la dirección (importancia de los clientes cuyos pedidos han originado las órdenes de fabricación). Esto supone que normalmente el conjunto de trabajos asociado a dicha orden tienen la misma fecha de entrega^{viii}.

Por el sistema que se sigue, normalmente se verifica la capacidad casi de forma simultánea a la generación del Plan Maestro, por lo que la lista de Órdenes de Fabricación suele convertirse en un verdadero Plan de Fabricación^{ix}. La generación del Programa de Producción se realiza mediante una reunión de los expertos en base al Plan de Fabricación y a los informes de incidencias de los responsables de las diferentes secciones. Revisándose los programas definidos en reuniones previas y añadiendo nuevos trabajos al Plan de Fabricación si fuera necesario. Así se define un calendario de lanzamientos, basándose en criterios de disponibilidad de material, maximización de la utilización de los recursos y minimización del inventario en curso, **con el requisito de lograr el cumplimiento de las prioridades prefijadas y el objetivo de la utilización eficiente de las instalaciones**. El Programa suele tener un

^{vii} El trabajo considera solamente un único modelo o formato de diferentes clientes agrupados en función de las fechas de entrega y donde se han tenido en cuenta el inventario existente del producto.

^{viii} Se considera como fecha de entrega aquella en la que el producto final debe haber realizado todas sus operaciones, estar disponible, para ser entregado al cliente.

^{ix} Plan de Fabricación: Listado de Fabricación sobre el que se ha realizado el análisis de Capacidad, y por lo tanto se considera factible la realización de los trabajos incluidos en las órdenes en las fechas previstas.

horizonte de dos a cuatro semanas, con un periodo máximo de una semana. La generación de los informes de incidencias es lenta (varios días) y dificulta la toma de decisiones por parte de los programadores.

Una gran parte de los Sistemas Productivos de la Industria Cerámica podrían ser modelados como un **Taller de Flujo Híbrido**, tal y como se justificará y profundizará en los próximos capítulos. En el modelo de Taller de Flujo se pueden identificar de forma general tres etapas comunes a una gran mayoría de Empresas Cerámicas, donde se realizan sendas operaciones: **Etapas de Prensado-Secado-Esmaltado, Etapa de Cocción, y Etapa de Clasificación y Embalado**. En cada etapa suelen existir recursos alternativos, con diferente eficiencia, para realizar la misma operación.

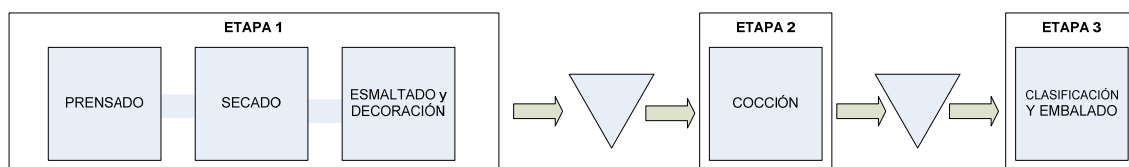


Figura 2-4. Sistema de fabricación de baldosas cerámicas modelado como taller de flujo (elaboración propia).

En este taller, cada responsable de etapa se debe preocupar de ajustarse a lo fijado en el Programa pudiendo hacer ligeros cambios en el orden de procesamiento de cada trabajo. Sin embargo, como la definición del **calendario de lanzamientos se consensua** basándose en la experiencia de los asistentes a la reunión, cuando se produce una situación no prevista y no se puede esperar a la próxima reunión para resolverla, se toman decisiones en el ámbito del taller o de una sección que pueden perjudicar a la ejecución del Programa de Producción. Esto provoca **distorsiones** en el Plan Maestro de Producción que a veces pueden llegar a ser graves.

Por otro lado, no son pocas las veces en las que se lanzan trabajos que no se han tenido en cuenta en la reunión, debido a que aparecen necesidades de última hora, no contempladas en el Plan Maestro de Producción desplegado, y que a su vez, provocan perturbaciones en lo inicialmente programado. Se observa por lo tanto, una relación entre Planificación y Programación poco fluida, casi limitado a una reunión semanal, pero que **no dispone de mecanismos de Toma de Decisiones ni Eficientes, ni Colaborativos** en caso de eventos no esperados.

2.5 Oportunidades para la Mejora del Proceso de Programación de la Producción en la Industria Cerámica Actual.

Actualmente se tiende a ofrecer un catálogo de productos muy amplio donde además se pretende que estos productos estén diferenciados de la competencia. **Los clientes son cada vez más exigentes en términos de Precio, Calidad y Servicio.** Además existe un conjunto de países en vías de desarrollo (China, Turquía, Brasil, etc.) que son una fuente de fuerte competencia en el ámbito de los precios. Todos estos factores, unidos al hecho de que, por norma general, las empresas disponen de Sistemas de Producción poco flexibles y orientados hacia la Producción en Masa, hacen que sea necesaria una eficiente utilización de los recursos para poder mantener el nivel competitivo.

A continuación se analizan con mayor detalle aquellos aspectos de la Industria Cerámica que inciden de forma negativa, dificultando la gestión del proceso de Programación de la Producción. En concreto se han seleccionado 2 problemas diferentes:

- La Incertidumbre. Se analizará de que manera afecta al proceso, y se profundiza sobre aquella que procede de los Tiempos de Cambio de Partida por ser característica de este Sector.
- La distancia existente entre las soluciones necesarias para los problemas reales y las existentes en el entorno científico.

A partir de esta revisión se deberían identificar con mayor facilidad las oportunidades de mejora que permitan alcanzar una mayor competitividad.

2.5.1 El problema de la Incertidumbre

La **Gestión de la Producción en la Industria Cerámica es un proceso claramente afectado por la Incertidumbre** en los términos establecidos por (Galbraith, J. 1973). Esta Incertidumbre procede **tanto del entorno como del propio sistema**, es decir procede de los dos grupos establecidos por (Ho, C. 1989).

La Incertidumbre del entorno que afecta al proceso productivo en la Industria Cerámica es fundamentalmente la debida a la Demanda de los Clientes, que está relacionada con la Incertidumbre en los Pedidos de los Clientes y/o Previsiones.

La Incertidumbre del sistema está relacionada directamente con el proceso productivo, y entre las que afectan de forma singular a la producción de baldosas cerámicas destacan: **Los procedentes del proceso de transformación** que provocan variaciones en la calidad del producto (ver apartado 2.3.2.9), **la variabilidad del Tiempo de Proceso y de Cambio de Partida** (ver apartado 2.5.2) que implica variaciones en las fechas de finalización. Existen otras Incertidumbres comunes a otros procesos industriales como: Averías del equipamiento, variación de los recursos, etc. En la Figura 2-5 se identifican las causas de Incertidumbre y se relacionan con los impactos que tienen sobre la gestión del sistema, que será comentada a continuación.

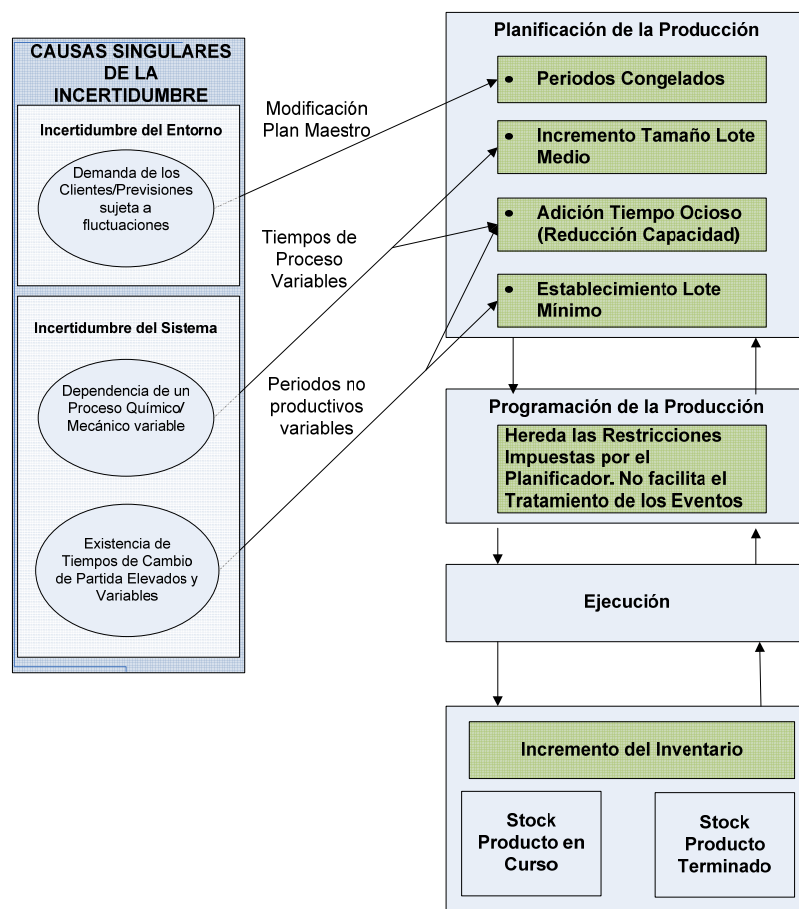


Figura 2-5. Relación de causas y efectos de la Incertidumbre en el sistema productivo de las Empresas Cerámicas (elaboración propia).

Como consecuencia de los impactos causados por la Incertidumbre del entorno y/o del sistema, el coste total de la fabricación se incrementa. En (Murthy, D. N. P. & Ma, L. 1991) se enumeraron algunas de las causas de dicho incremento de coste, entre las cuales se destacan por su relación con el caso de las Empresas Cerámicas las siguientes: Pedidos no servidos, inventarios excesivos o insuficientes.

Las Empresas Cerámicas han luchado de forma constante para **reducir el impacto que sobre la Productividad y el nivel de Servicio al Cliente tiene la Incertidumbre**. Se han hecho grandes esfuerzo para minimizar la Incertidumbre, especialmente las debidas a la singularidad del entorno y del sistema productivo objeto de estudio, es decir, las producidas por la variación de la demanda, de las calidades del producto, o de los Tiempos de Proceso y de Cambio de Partida (ver Figura 2-5). La organización y la gestión de la producción en la Industria Cerámica se han visto condicionadas por estas peculiaridades.

Desde el punto de vista de la Programación de la Producción se ha observado como los Programas, al verse afectados por la Incertidumbre del sistema, han optado en general por:

- **Incorporar periodos de tiempo ocioso extra** para absorber pequeños retrasos en los tiempos estándares de producción o cambio de partida.
- **Fabricar unos lotes mínimos** que permitan que el producto llegue al final del proceso para ser clasificado, y que entonces se realicen los ajustes necesarios en las variables controlables del proceso para corregir las anomalías que puedan aparecer. Esto está motivado, porque pese a todos los esfuerzos realizados en mejorar el proceso, es imposible asegurar una calidad 100%. Por lo tanto, se debe realizar al final del proceso una clasificación por calidades.
- **Aumentar el tamaño de los lotes**, para evitar las pérdidas debidas a defectos de calidad, debido a la dificultad de asegurar un producto final uniforme de primera calidad. Estos problemas propician que se tiendan a sobredimensionar el tamaño de los lotes a fabricar con lo que los inventarios de producto acabado aumentan.

Así mismo, los programadores han evitado que la Incertidumbre del entorno les afecte trabajando, por un lado, con periodos congelados donde los órdenes de producción, y por lo tanto la ejecución del Plan Maestro, no pueden ser modificadas, y por otro lado, suelen disponer de ***Inventarios Sobredimensionados***.

Queda patente que alguno de los planteamientos actuales podrían ser mejorados si las empresas pudieran ser capaces de absorber parte de la Incertidumbre mediante la **Flexibilización y el aumento de la robustez del Sistema de Gestión de la Programación de la Producción**.

La correcta comprensión de las causas de Incertidumbre y de sus efectos, directos e indirectos, es importante para entender la motivación de esta tesis. Algunos impactos quedan fuera del objeto de la tesis, como es el caso de la variación de la demanda, otras se han explicado en

apartados anteriores (ver apartado 2.3.2.9) donde se ha presentado el problema de los tonos y calibres así como el de la clasificación en categorías (primera, segunda, tercera y tiesto) de los productos cerámicos.

A continuación se profundiza en el problema de los Tiempos de Cambio de Partida característicos del sector cerámico.

2.5.2 El Problema de los Tiempos de Cambio de Partida

Al inicio de cada nueva orden de fabricación aparecen unos **Tiempos de Ajuste**, necesarios para calibrar las máquinas de manera que se obtenga el producto deseado. A estos tiempos de ajuste se les deben de añadir los tiempos invertidos en modificar las máquinas y su posición en la línea, denominados **Tiempos de Preparación**, con lo que la suma de ambos origina lo que se denomina **Tiempo de Cambio de Partida**.

Inicialmente, cuando el catálogo de las Empresas Cerámicas no era muy extenso, el sistema productivo era muy eficiente puesto que en la misma línea se fabricaban uno o dos modelos. Sin embargo, hoy en día en una misma línea de esmaltado (y debido a su gran automatización y coste), se deben fabricar gran diversidad de productos. Para ello, se tienen que cambiar y ajustar las máquinas situadas en la misma. **Estos Tiempos de Cambio de Partida pueden llegar a ser muy elevados** y, por lo tanto, se hace necesario una gestión adecuada de los mismos que minimice su impacto sobre la capacidad del sistema productivo. En (Vallada, E. et al. 2005) se realiza un estudio sectorial en el que se establece la relación entre el número de formatos diferentes y el número de líneas dedicadas a cada formato, así como la frecuencia media de cambio de partida en la línea de prensas-esmaltado.

Los tiempos de cambio de partida afectan a todas las etapas del proceso productivo. No obstante, el grado de impacto no es el mismo en todas, pudiendo invertirse horas o minutos dependiendo de la tarea a realizar.

Los cambios de partida condicionan los **costes** en los que incurre la empresa, la **productividad** de la misma, la **flexibilidad** de operaciones, el **tamaño de lote** aplicado a cada producto, la **capacidad de los almacenes** tanto de producto acabado como de producto en curso, etc. El mayor o menor tiempo empleado en la realización de los cambios de partida condiciona la Programación de la Producción, puesto que en función de lo que se esté fabricando y de lo que queda por fabricar, se incurrirán en unos tiempos de ajuste más o menos elevados en función de la decisión que se tome y, por lo tanto la fecha de finalización de los trabajos se verá afectada.

Se pueden identificar dos tipos de trabajo en relación con los tiempos de cambio de partida. Las labores asociadas a la modificación del formato y las vinculadas a la variación del modelo entre dos productos sucesivos. Las primeras tienen que ver con la modificación del ancho y largo de la baldosa (el cambio de matrices, ancho de carriles, etc.), los segundos están relacionados con el cambio de las aplicaciones situadas en la línea de esmaltado y de las materias primas. A grosso modo, el tiempo de cambio relacionado con el formato suele estar en torno a las ocho horas, y afecta sobre todo a la configuración de la prensa. De igual forma el tiempo relacionado con el cambio de un modelo suele situarse alrededor de una hora, y afecta a casi todas las etapas productivas.

2.5.3 El problema de la Programación de la Producción en el Sector Cerámico

Una excelente estrategia empresarial y una buena Planificación de la Producción necesitan también de una buena Programación de la Producción para conseguir los objetivos de productividad y servicio.

Un problema que afronta el Programador en Taller de Flujo consiste básicamente en el establecimiento de la secuencia de los trabajos, y la determinación de unas fechas de inicio y fin de cada operación. En el Taller de Flujo clásico, el flujo u orden de procesamiento de cada trabajo en las máquinas es el mismo para todos ellos. Sin embargo, si el Taller de Flujo es Híbrido dispondrá de múltiples recursos disponibles para realizar algunas de sus operaciones. Por lo tanto, en el Taller de Flujo Híbrido el Programador debe resolver el problema de Asignación, además del de Secuenciación y Temporización.

Las siguientes hipótesis aparecen frecuentemente en la literatura de la teoría de secuenciación y son debidas a (Conway, R. W., Maxwell, W. L., & Miller, L. W. 1967): (1) Las máquinas están siempre disponibles y nunca dejan de funcionar; (2) Cada máquina puede procesar a lo sumo un trabajo a la vez; (3) Cualquier trabajo puede ser procesado únicamente en una máquina a la vez; (4) Los tiempos de preparación de todos los trabajos son cero, por ejemplo, todos los trabajos están disponibles al comienzo del proceso; (5) No se permiten interrupciones, es decir, cuando una operación ha comenzado debe terminarse antes de empezar otra en la misma máquina; (6) Los tiempos de cambio son independientes de los programas/secuencias y están incluidos en los tiempos de procesado, y (7) Los tiempos de procesado y las restricciones tecnológicas son deterministas y se conocen de antemano, y similarmente ocurre con las fechas de entrega.

Sin embargo, en muchos problemas industriales reales estas hipótesis no son válidas y es por eso que en los últimos años han aparecido modelos y procedimientos de resolución que relajan una o varias de las hipótesis anteriores. En (MacCarthy, B. L. & Liu, J. 1993) se comenta que **existe una gran distancia en los desarrollos de la teoría de secuenciación y sus aplicaciones prácticas**, que los autores achacan a dos factores: el primer factor es un desconocimiento de la teoría de secuenciación y de los procedimientos para resolver este tipo de problemas por parte de los usuarios finales (léase jefes de producción, mandos intermedios y personal encargado de tomar este tipo de decisiones); mientras que el segundo factor se basa en que la mayoría de resultados existentes resuelven problemas demasiado alejados de la realidad. El autor de este trabajo está de acuerdo con la apreciación de (MacCarthy, B. L. et al. 1993), sin embargo se considera que en muchos casos no es posible reducir la distancia entre el planteamiento teórico y el práctico en base a modelos y/o algoritmos más próximos a la realidad. Efectivamente, por la complejidad de muchos sistemas no es eficaz, ni eficiente, disponer modelos finales tan complejos, y en algunas ocasiones los avances técnicos no posibilitan la obtención resultados de la forma deseada. Por este motivo, en este trabajo se desea avanzar para reducir esta distancia entre teoría y práctica focalizando los esfuerzos tanto en el enfoque de Programación de la Producción como un **Proceso de Negocio**, así como en la mejora de la **Colaboración entre Planificación y Programación de la Producción**. La revisión y adaptación de procesos que se propondrá también significará una **revisión y adaptación de algunos Algoritmos/Técnicas** de Programación de la Producción que permitan alcanzar los objetivos de los Procesos Establecidos.

En este trabajo se propone aplicar los métodos desarrollados a entornos reales de producción, concretamente en el Sector Cerámico. La creciente diversificación y diferenciación de productos y la fuerte competencia han hecho que las empresas de este sector tengan que dotar de un valor añadido a los productos/servicios para vender todo lo que producen, y cada vez este valor añadido tiene un peso específico mayor. El valor añadido está directamente vinculado con la mejora del servicio al cliente, que junto con la necesidad de ofrecer un precio muy competitivo, implica realizar una mejora integral de todos los procesos. **La mejora de los procesos debe de incluir entres sus objetivos realizar una eficaz Programación de la Producción**. En este contexto un objetivo que se plantea es el diseño de un prototipo de software para la Programación de la Producción que incluya todos los algoritmos desarrollados en la presente Tesis Doctoral. La finalidad del software es que sea capaz de

2.6 Propuesta de un Nuevo Marco de Trabajo

Según lo analizado en el apartado anterior la Gestión de los Programas Productivos de las Empresas Cerámicas actuales presenta al menos dos líneas de trabajo que permitirían una mejora sustancial de los resultados en dicho ámbito. Dichas líneas son:

- La utilización de **Aplicaciones con una Arquitectura Orientada a la Programación de la Producción** embebiendo algoritmos que eficientes sean capaces de proporcionar un marco que dote de flexibilidad y robustez al proceso.
- La aplicación de Algoritmos Eficientes en función de su adecuación al tipo de taller y condiciones productivas reales que garanticen un rendimiento medio elevado.

Parte de la flexibilidad y robustez que se busca debe proceder de una **adecuada respuesta ante los eventos** que permita considerar la evolución temporal de los Programas. Este es un aspecto que constituyen una pieza clave en la propuesta que se realizada en esta tesis. Un sistema razonablemente sensible a los eventos podría suponer una mejora, ya que proporcionaría mayor agilidad al sistema productivo pudiendo incrementar la capacidad de reacción ante Incertidumbres lo que podría dar lugar a perfeccionar alguno de los siguientes aspectos:

- Mayor fiabilidad del Plan Maestro.
- Mayor automatización del sistema de Gestión de la Producción.
- Reducción del inventario.

En la actualidad (ver apartado 2.4) los Programas de Producción abarcan un espacio temporal reducido. La reducción del Programa está causada por la inexistencia de un sistema automatizado de gestión de la producción que ayude a manejar los eventos. La fórmula tradicional es reducir el periodo programado para reducir el número de eventos que inciden sobre el mismo e intentar absorberlos, normalmente mediante: tiempos muertos explícitamente preparados para circunstancias inesperadas o desplazando el evento al siguiente periodo. Excepcionalmente, los eventos implican una reprogramación. En todo, caso la aparición de un evento implica la dedicación de un conjunto importante de recursos humanos y tiempo para su análisis y la presentación de soluciones alternativas.

La existencia de un Modelo Automatizado de Programación de la Producción, basado en una orientación de Procesos de Negocio que incluyen Herramientas (Algoritmos) de Ayuda a la Toma de Decisiones bajo Incertidumbre necesarias para resolver la visión funcional/decisional, que contemple la existencia de eventos y su gestión, permite dar el primer paso hacia un **modelo dinámico**. Se entenderá por un sistema de Programación de la Producción dinámico aquel que dispone de un horizonte de programación infinito (ver capítulo 3). Un sistema de programación dinámico permitiría:

- Mayor integración entre los Planes y los Programas de Producción. El Programa de Producción puede tener en cuenta trabajos previstos más allá del periodo actual, lo que puede ser usado para evaluar la viabilidad y prestaciones de los planes de Producción.
- Disponer de un periodo de programación extenso (disponer simultáneamente de varias tandas de trabajos) implica que el número de trabajos a programar en cada periodo es mayor. El aumento del número de trabajos en un solo Programa aumenta las posibilidades combinatorias del problema, frente a la alternativa anterior, y las posibilidades de mejorar funciones de utilidad.

En esta tesis se realiza una propuesta para la Programación de la Producción basada en un **enfoque semi-dinámico y estocástico** del sistema productivo. Esta propuesta se diferencia por completo del modelo tradicional. Este cambio de enfoque permite realizar una propuesta que contempla el problema de la Programación de la Producción con una perspectiva lo más realista posible, donde aspectos como la No Disponibilidad de Recursos, aparición de Trabajos Urgentes no contempladas, modificación de Tiempos de Ejecución, Cambio de Partida, etc. puedan ser tenidos en cuenta.

Para poder llevar a cabo la automatización se ha considerado determinante el análisis de nuevos ámbitos tecnológicos que complementen a los aspectos Metodológico y Arquitectónico, y actúen de facilitadores. Desde el punto de vista tecnológico hay que subrayar que en los últimos años ha surgido una disciplina procedente de la inteligencia artificial, que aborda el paradigma de los **Sistemas Multiagentes**, Agentes Inteligentes o simplemente Agentes.

Los Sistemas Multiagente tienen como una de sus características fundamentales la de afrontar problemas complejos. Según (Lesser, V. R. 1995) los Agentes solucionan problemas interdependientes, ya sea por el uso de recursos o por objetivos compartidos. Para los que Agentes alcancen soluciones compatibles (no conflictivas) y óptimas/subóptimas en entornos de interdependencia con el mínimo uso de recursos requieren tener una visión actualizada, consistente, aunque no necesariamente completa del conjunto de problema (visión global).

Se puede concluir indicando que el objetivo será el planteamiento de un Modelo Programación de la Producción, basado en una orientación de Procesos de Negocio que incluyen Herramientas (Algoritmos) de Ayuda a la Toma de Decisiones bajo Incertidumbre necesarias para resolver la visión funcional/decisional soportado en un Plataforma Integradora desarrollada bajo el paradigma de los Sistemas Multiagente.

2.6.1 Ventajas que se Obtendrían con una Programación ajustada a las Singularidades Identificadas

La utilización de una aplicación software capaz de gestionar el conjunto de Incertidumbres asociadas al Proceso de Programación de la Producción expuesto con anterioridad y de generar Programas Productivos con un alto grado de eficiencia, generarían un conjunto de mejoras en el Sistema Productivo. Se pueden destacar las siguientes:

- **Mejor uso de los recursos** al aplicar algoritmos que mejoren las medidas de eficiencia mediante una mejor adaptación a las singularidades del proceso.
- **Mejora en la robustez** al poder adaptarse a las modificaciones que vayan ocurriendo en el sistema productivo, proponiendo soluciones alternativas.
- **Mejora del Servicio al Cliente** si el sistema es capaz de solventar posibles eventualidades con soluciones alternativas podrá evitar posibles retrasos o cancelaciones de pedidos.
- **Aumento de la Productividad** si es capaz de reducir el tiempo de espera actual para la gestión de un evento. Dicho tiempo podrá ser utilizado en avanzar producción de otros periodos o atender pedidos urgentes.
- **Automatización de la Gestión** al ser capaces de utilizar pequeñas rutinas informatizadas para solucionar eventos habituales.

Si además el sistema que se proponga es capaz de afrontar el problema de los Cambios de Partida con éxito se obtendrían algunas economías en las Empresas Cerámicas que podrían ser:

- Disminución del **número de cambios** en los distintos recursos del sistema (moldes en prensas, ajuste de guías y diversos elementos en las líneas de esmaltado, ajustes en hornos y ajustes en las máquinas de clasificación y embalaje).
- Disminución de los **costes de personal**. Al reducirse el número de cambios, no serían necesarios los mismos operarios de mantenimiento que atienden estas variaciones.
- Ahorro de **tiempo de ejecución** en la consecución del **Plan Maestro** de Producción, ya que se evitarían innecesarios cambios de partida. Con ello, se puede aumentar la producción y el coste unitario por metro de producto acabado sería menor.

El mayor o menor tiempo empleado en la realización de los Cambios de Partida condiciona la Programación de la Producción, puesto que en función de lo que se esté fabricando y de lo que queda por fabricar, se incurrirán en unos tiempos de ajuste más o menos elevados en función de la decisión que se tome y, por lo tanto la fecha de finalización de los trabajos se verá afectada

La **capacidad de adaptación** de los Programas de Producción a las singularidades del sistema durante el proceso de generación del Programa, y a las eventualidades que se pueden dar en la fase de ejecución puede convertirse en una ventaja competitiva de primer orden.

2.7 Conclusiones

En resumen, el sector azulejero español es **altamente competitivo** en los mercados internacionales. Este comportamiento se ha alcanzado debido a una política efectiva y constante a lo largo del tiempo basada en inversiones en gran medida aplicadas a innovaciones tecnológicas en los procesos. Sin embargo, sus éxitos esconden algunas posibilidades de mejora que en los últimos años se están intentando mejorar.

En lo que respecta al Sistema de Planificación y Control de Operaciones se han observado un conjunto de causas que generan **Incertidumbre**, que a su vez, condicionan el proceso productivo, originando un **incremento del nivel de inventarios** y una especialización en las tareas que son agrupadas en diversas etapas. Para mejorar la competitividad, al mismo tiempo que se desarrollan y aplican tecnologías que permiten mejorar operaciones concretas, se ha detectado como las empresas muestran un creciente interés por desarrollar **procedimientos y técnicas que ayuden a programar adecuadamente la producción** como forma de conseguir un entorno de trabajo que facilite la **gestión de eventos de forma dinámica**, y atendiendo a la singularidad de cada fase productiva. Con este enfoque se intuye como las empresas inician una orientación sus sistemas productivos para favorecer el **incremento de flexibilidad**, convirtiendo el sistema de programación en una pieza fundamental, para la mejora de la fiabilidad de los planes productivos.

Como conclusión a todo lo expresado en el presente capítulo, se identifica la existencia de un campo para la investigación en relación con la **mejora de la Gestión del Programa de Producción** en las Empresas del Sector Cerámico. En concreto toman relevancia los enfoques relacionados con los sistemas **semi-dinámicos y con Incertidumbre** que se podrían abordar con técnicas **Predictivo-Reactivas**, es decir un sistema basado en un Programa preestablecido que sea capaz de reaccionar ante ciertos eventos Tomando Decisiones que puedan suponer una reparación del Programa o una Reprogramación completa. Este ámbito de investigación debe de plantear diversas medidas de eficiencia para diversos Procesos, que por lo identificado

de forma preliminar, debería de contar con medidas de eficiencia para el proceso Global de Gestión del Programa, y otras medidas de eficiencia para las actividades de Programación, o de Reprogramación.

El ámbito de investigación que se plantea debe ser confirmado, después del cual se definirá en todo caso el problema concreto que se desea resolver. No obstante, como avance se puede proponer, que dadas las características del entorno de trabajo, podría ser interesante analizar las posibilidades de una solución fundamentada en un modelo basado en un **Sistema Multiagente** embebido en una Plataforma Software. El próximo paso que se propone realizar dentro de este trabajo es el análisis del estado del arte de los Modelos, Métodos y Algoritmos de Programación de la Producción, así como en el de los Sistemas Multiagente y su aplicación a los Sistemas de Fabricación y a su Programación de la Producción.

2.8 Bibliografía

Alfaro, J. J. 1997. Análisis mediante simulación de reglas heurísticas. Aplicación a empresas cerámicas. Proyecto Final de Carrera. Universidad Politécnica de Valencia.

Andrés, C. 2001. Programación de la Producción en Talleres de Flujo Híbridos con Tiempos de Cambio de Partida dependientes de la secuencia. Modelos, Métodos y Algoritmos de Resolución. Aplicación a Empresas del Sector Cerámico. Universidad Politécnica de Valencia.

Apics. Diccionario. 1994. American Production and Inventory Control Society.

ASCER. El azulejo. Estudio sectorial. 1998. Castellón, ASCER. Informe anual ASCER.

ASCER. Los sectores español y mundial de fabricantes de baldosas cerámicas 2003. 2003. 2003. Castellón, ASCER.

ASCER. Proceso de Fabricación de Baldosas Cerámicas. 2007.

ASCER. Evolución de las exportaciones del sector cerámico Español. 2008. Castellón, ASCER.

Conway, R. W., Maxwell, W. L., & Miller, L. W. 1967. Theory of Scheduling. Addison-Wesley Publishing Company.

Dalmau, J. I. and De Miguel, E. El azulejo. Estudio Sectorial. 1991. Valencia, Universidad Politécnica de Valencia.

Dalmau, J. I., De Miguel, E., and Miquel, S. Análisis estratégico de los sectores industriales y del turismo en la Comunidad Valenciana. 1993. Universidad Politécnica de Valencia. Valencia.

De Miguel, E. *Estructura de las PYMES en la Comunidad Valenciana*. 1996. Valencia, Universidad Politécnica de Valencia.

egre-Vidal, J., Lapiedra-Alcami, R., & Chiva-Gomez, R. 2004. *Linking operations strategy and product innovation: an empirical study of Spanish ceramic tile producers*. *Research Policy*, 33(5): 829-839.

Escardino, A. 2001. *Technological innovation in the Castellon ceramic industry*. *Boletin De La Sociedad Espanola De Ceramica y Vidrio*, 40(1): 43-51.

Ferdows, K. & De Meyer, A. 1990. *Lasting improvements in manufacturing performance: in search of new theory*. *Journal of Operations Management*, 9(2): 168-184.

Gabaldón, S., López, S., & Carda, J. B. 2003. *Legistación y gestión medioambiental en la producción de baldosas cerámicas*. *Boletin De La Sociedad Espanola De Ceramica y Vidrio*, 42(3): 169-179.

Galbraith, J. 1973. *Designing complex organizations*. Massachusetts: Addison-Wesley Pub. Co.

Gallego, J. R., Bono, E., and Tomas, J. A. *Cambio tecnológico y transformación de sistemas industriales localizados: el caso de la industria española de pavimentos y revestimientos cerámicos*. 502. 1997. Departament d'Economia Aplicada.

Ho, C. 1989. *Evaluating the impact of operating environments on MRP systems nervousness*. *International Journal of Production Research*, 27: 1115-1135.

IMPIVA. *Informe Económico y financiero de la competitividad. C. Valenciana*. Ardán. 35. 2004. Valencia, IMPIVA.

Instituto Valenciano de Estadística. *Comercio Exterior e Intracomunitario de la Comunidad Valenciana*. 2009.

Lesser, V. R. 1995. *Multiagent Systems: An Emerging Subdiscipline of AI*. *ACM Computing Surveys*, 27(3): 340-342.

MacCarthy, B. L. & Liu, J. 1993. *Addressing the gap in scheduling research : a review of optimization and heuristic methods in production scheduling*. *International Journal of Production Research*, 31: 59-79.

Molina, J. *Los Sistemas Productivos Descentralizados: Factor territorial y estrategias empresariales*. 1999. Valencia, Fundación Bancaja. Comité Económico y Social de la Comunidad Valenciana, Serie Economía y Empresa, n.º 5.

Molina-Morales, F. X. 2002. *Industrial districts and innovation: the case of the Spanish ceramic tiles industry*. *Entrepreneurship and Regional Development*, 14: 317-335.

Murthy, D. N. P. & Ma, L. 1991. *MRP with uncertainty: a review and some extensions*. *International Journal of Production Economics*, 25(1-3): 51-64.

Segura, B., Vallada, E., Maroto, C., & Ruiz, R. 2004. Operations strategy in Spanish tile industry firms. Boletín De La Sociedad Española De Cerámica y Vidrio, 43(6): 929-932.

Tormo, G. 1997. Desarrollo de una metodología para la agrupación de productos en familias de producción. Aplicación a una empresa de fabricación de pavimentos y revestimientos cerámicos. Proyecto Final de Carrera. Universidad Politécnica de Valencia.

Vallada, E., Maroto, C., Ruiz, R., & Segura, B. 2005. Analysis of production scheduling in Spanish tile industry. Boletín De La Sociedad Española De Cerámica y Vidrio, 44(1): 39-44.

Ybarra, J. A., Giner, J. M., & Santa, M. J. 1996. Una política industrial para la PYME. La experiencia de la cerámica española. Economía Industrial, 308: 175-185.

**Estado del Arte de los Modelos,
Métodos y Algoritmos de Resolución
en la Programación de la Producción**

Capítulo

3

Tabla de contenido

3.	Capítulo 3	83
3.1	Introducción	83
3.2	Introducción a la Programación de la Producción	84
3.3	Teoría de la Programación de Producción	91
3.3.1	Introducción	91
3.3.2	Programación en contexto Determinista o Incierto.....	92
3.3.3	Programación en contextos Estáticos, Dinámicos, en Línea, o en Tiempo Real.	95
3.4	Descripción del Problema Dinámico y con Incertidumbre de Programación de la Producción. Enfoques Generales de Resolución Distribuidos.	98
3.4.1	Definiciones e Hipótesis	98
3.4.2	Tipos de Configuraciones Productivas y Nomenclatura.....	100
3.4.3	Estrategias de Programación de la Producción en un entorno Dinámico y con Incertidumbre.	103
3.4.4	Medidas de Eficacia sobre el Rendimiento, la Robustez y la Estabilidad de un Programa en entornos Dinámicos.....	110
3.4.4.1	Medidas de Eficacia sobre el Rendimiento	110
3.4.4.2	Medidas de Eficacia sobre la Robustez	113
3.4.4.3	Medidas de Eficacia sobre la Estabilidad	115
3.4.5	Políticas de Reprogramación en Estrategias Predictivo-Reactivas: Cuando Actuar ante los Eventos	116
3.4.6	Procedimientos de Resolución para Programaciones Predictivo-Reactivas	120
3.4.7	Descentralización en la Programación de la Producción.	126
3.5	El Taller de Flujo, Flujo Híbrido y los Tiempos de Cambio de Partida Dependientes de la Secuencia.....	131

3.5.1	Introducción	131
3.5.2	Definiciones de Taller de Flujo y Flujo Híbrido.....	131
3.5.3	El Problema de los Tiempos de Cambio de Partida.....	133
3.5.3.1	El Tiempo de Preparación.	134
3.5.4	Definición y Problemática del Taller de Flujo Híbrido con SDST	137
3.5.5	Enfoque Distribuido de la Programación del Taller de Flujo.....	138
3.6	Procedimientos para abordar un Problema de Optimización	139
3.6.1	La Teoría de la Complejidad	139
3.6.2	Tipos de Procedimiento de Resolución del Problema de Programación de la Producción	142
3.7	Métodos de Resolución de Programación de la Producción para el Taller de Flujo Híbrido: Los Modelos y Algoritmos	146
3.7.1	El Caso del Taller de Flujo con Tiempos de Cambio de Partida Dependientes de la Secuencia.....	148
3.7.1	El Caso del Taller de Flujo Híbrido.....	157
3.7.2	El Caso del Taller de Flujo Híbrido con Tiempos de Cambio de Partida Dependientes de la Secuencia.	167
3.8	Conclusiones	172
3.9	Bibliografía	173

Índice de figuras

Figura 3-1. Representación del Sistema de Planificación y Control de Operaciones (Vicens, E., Poler, R., Ortiz, A., Albarracín, J. M., García, J. P., Andrés, C. et al. 1999).....	85
Figura 3-2. Representación del Sistema de Planificación y Control de Operaciones (adaptado de la propuesta elaborada en (Vicens, E. et al. 1999)).....	86
Figura 3-3. Una relación entre las configuraciones de los sistemas productivos (a partir de (Maccarthy, B. L. et al. 1993)).....	101
Figura 3-4. Representación de un Taller de Flujo con m máquinas o Etapas (elaboración propia).	132
Figura 3-5. Representación de un Taller de Flujo Híbrido con r Etapas (elaboración propia). .	132
Figura 3-6. Caso de tiempos de montaje y desmontaje fijos. Adaptado de (Artigues, C. 1997).	135
Figura 3-7. Representación gráfica de algunos tipos de Tiempos de Cambio (elaboración propia).	137
Figura 3-8. Representación de los tipos de problema en base a su dificultad (elaboración propia).	140

Índice de tablas

Tabla 3-1. Resumen de las principales estrategias de Programación de la Producción en un entorno dinámico (elaboración propia).	106
Tabla 3-2. Clasificación de los procedimientos reparadores en un Taller General (Raheja, A. S. et al. 2002).....	122
Tabla 3-3. Clasificación de los tipos de tiempos de cambio(Andrés, C. 2001).	136
Tabla 3-4. Clasificación de los métodos de resolución de los Problemas de Programación de la Producción (elaboración Propia).	142
Tabla 3-5. Clasificación de los métodos de resolución de los Problemas de Programación de la Producción en un Taller de Flujo Híbrido (Quadt, D. et al. 2007).	146
Tabla 3-6. Resumen contribuciones al problema del Taller de Flujo con SDST con soluciones óptimas (elaboración propia).	150
Tabla 3-7. Resumen contribuciones al problema del Taller de Flujo con SDST con Soluciones Aproximadas (elaboración propia).	151
Tabla 3-8. Resumen contribuciones al problema del Taller de Flujo con SDST con Soluciones Aproximadas (continuación de Tabla 3-7)(elaboración propia).	152
Tabla 3-9. Resumen contribuciones al problema del Taller de Flujo Híbrido con Soluciones Exactas (elaboración propia).	158
Tabla 3-10. Resumen contribuciones al problema del Taller de Flujo Híbrido con Soluciones Aproximadas (elaboración propia).	159
Tabla 3-11. Resumen contribuciones al problema del Taller de Flujo Híbrido con Soluciones Aproximadas (continuación de Tabla 3-10).....	161
Tabla 3-12. Resumen contribuciones al problema del Taller de Flujo Híbrido con SDST con Soluciones Aproximadas (elaboración propia).	168
Tabla 3-13. Resumen contribuciones al problema del Taller de Flujo Híbrido con SDST con Soluciones Aproximadas (continuación de Tabla 3-12 (elaboración propia)).....	169

3.1 Introducción

En este capítulo se realiza una revisión de las publicaciones más relevantes relacionadas con los Modelos, Métodos y Algoritmos para la Programación de la Producción a las que se ha tenido acceso. El fin que se persigue en este capítulo es doble. Por un lado, se desea confirmar que el caso que se identificó en el capítulo anterior es interesante, desde el punto de vista investigador interesante, es decir que existen enfoques de éste que aún no han sido resueltos, y que por lo tanto, hay campo para realizar aportaciones. Por otro lado, se desea tomar referencias que permitan realizar una buena aproximación al futuro modelado del problema y a los métodos más adecuados de resolución.

Aunque este capítulo está dedicado por completo a un aspecto muy concreto de la Dirección de Operaciones, como es el de Programación de la Producción, este no deja de ser un campo de conocimiento bastante amplio que obliga a realizar una aproximación hasta poder analizar los casos más similares al objetivo. En primer lugar se ha considerado fundamental realizar una revisión sobre el **Sistema de Planificación y Control de Operaciones** como forma de ubicar el proceso de **Programación de la Producción** en su contexto, y especialmente en su relación con el procedimiento de **Planificación de la Producción**. Posteriormente, y ya centrados en el ámbito de la secuenciación, se realiza un resumen general del estado actual de la misma, y se analiza la influencia del contexto (**determinista/estocástico**) en el que se **Toman las Decisiones** en el momento de abordar un problema. En este paso se precisa sobre el significado de conceptos como **Estático, Dinámico, En Línea** (“on line”), o en **Tiempo Real**. Estos conceptos se utilizan para definir de forma más adecuada el problema identificado, y aunque se empieza por describir aspectos relativos al problema de Programación de la Producción generales, como los tipos de taller existentes, las hipótesis de trabajo habituales, también se tratan aspectos específicos de un proceso dinámico y estocástico, como las **Estrategias** (Predictivas-Reactivas), **Políticas y Procedimientos** de Programación o las **Medidas de Eficiencia** en entornos con incertidumbre. No obstante, dado que según el análisis preliminar del capítulo 2 el tipo de sistema productivo que se ha detectado se puede clasificar como Taller de Flujo Híbrido con Tiempos de Cambio de Partida dependientes de la Secuencia se ha considerado necesario definir formalmente este tipo de problemas. El último paso que se realiza en este capítulo es el de analizar los **tipos de métodos** con los que se puede abordar el problema esbozado, identificar el grado de dificultad asociado a este tipo de problemas, es decir su **complejidad**, y estudiar con detenimiento el conjunto de **Modelos** y de **Algoritmos** que se han empleado en casos similares y que pueden ayudar a aproximar la solución de este trabajo.

El resto del capítulo ha quedado estructurado de la siguiente forma: En el apartado 2 se introduce la función de Programación de la Producción como aproximación al contexto de trabajo; En el apartado 3 se amplía el apartado anterior abordando aspectos formales de la Teoría de la Programación de la Producción, sobre todo relacionados con la incertidumbre y los contexto de Toma de Decisiones; En el apartado 4 se analiza el ámbito de Programación de la Producción Dinámico y con Incertidumbre; En el apartado 5 se define el problema de Programación para el Taller de Flujo Híbrido con sus diferentes variantes, prestando especial atención al caso de Tiempos de cambio de Partida dependientes de la Secuencia; En el apartado 6 se tratan los aspectos relacionados con las alternativas para resolver el problema y la complejidad para obtener soluciones; En el apartado 7 se estudian un conjunto de referencias afines a la estructura de problema preliminar con el fin de aprender respecto del método propuesto para su resolución; Y finalmente en el apartado 8 se revisa el trabajo realizado como conclusión del mismo.

3.2 Introducción a la Programación de la Producción

Dentro del **Sistema de Planificación y Control de Operaciones**, de manera abreviada **SPCO**, se inscribe el proceso denominado de Programación de la Producción en cuyo ámbito se desarrollan los temas tratados en esta tesis. No obstante, tal y como se puede concluir del análisis de la Figura 3-1, *el proceso de Programación de la Producción no se puede entender de forma aislada* ya que interactúa en gran medida con otros procesos del **SPCO** como son el proceso de Planificación de la Producción y Capacidad y el proceso de Ejecución también conocido por proceso de Lanzamiento. También se puede decir que *existe una importante relación entre los procesos mencionados y el periodo de vigencia (medio, corto y muy corto plazo) de las Decisiones que en ellos se Toman*. Y finalmente, se debe tener en cuenta que el **SPCO** no es más que una parte del subconjunto de sistemas de la Empresa, y en ese sentido el proceso de Programación de la Producción se encuentra sujeta a múltiples factores, que en ocasiones como consecuencia de una visión sesgada, tienden a perderse.

Como ya ha dicho la función de Programación de la Producción debe de interactuar con otras funciones de la Organización. Por ejemplo, viene afectada por el proceso de Planificación de Producción, el cual desarrolla los planes de operaciones a medio y corto plazo para la organización en su totalidad. Este proceso debe considerar los requerimientos de capacidad, los niveles de inventario y las previsiones, para mejorar en la medida de lo posible el Plan de Producción y la asignación de recursos a largo plazo.

Dentro del Sistema de Planificación y Control de Operaciones se diferencian claramente las funciones de Planificador y Programador, aunque atendiendo a la Figura 3-1 debe existir una

relación bidireccional entre ambos. Sin embargo, *se ha identificado que en muchas ocasiones la relación Planificador-Programador es escasa, mal definida, y que el único vínculo formal y robusto es la comunicación unidireccional que tiene como origen al Planificador y destino al Programador, pero que no cuenta con retroalimentación, o ésta es débil o está poco formalizada.* A pesar de ello, parece lógico que debe existir una comunicación formal en ambos sentidos.

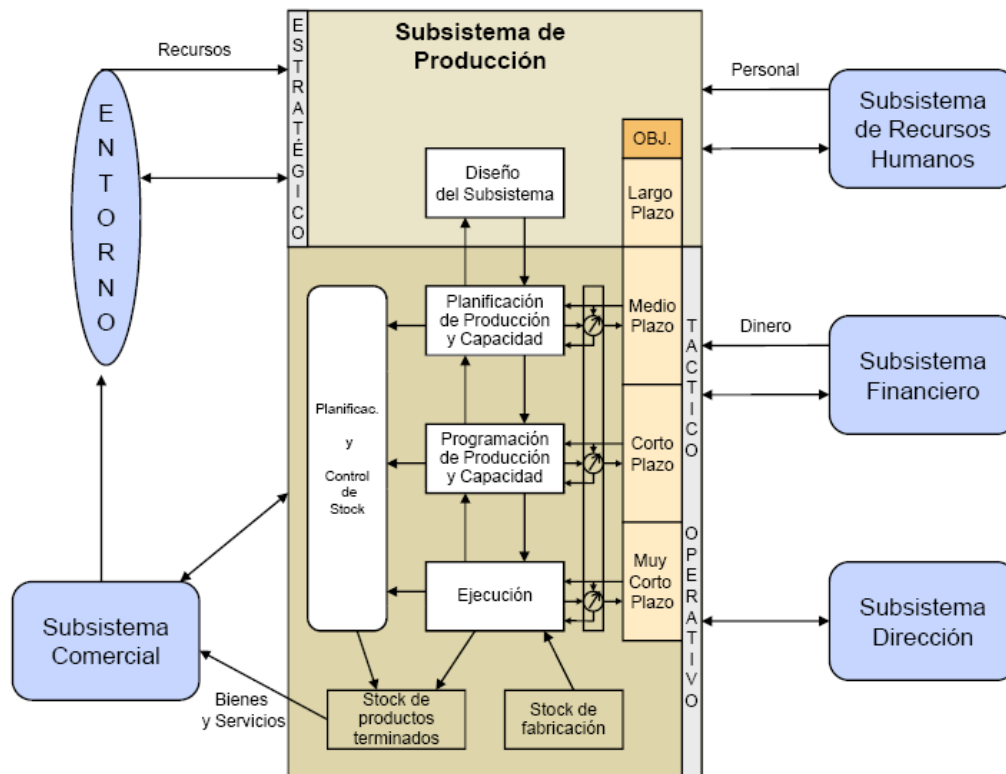


Figura 3-1. Representación del Sistema de Planificación y Control de Operaciones (Vicens, E., Poler, R., Ortiz, A., Albarracín, J. M., García, J. P., Andrés, C. et al. 1999).

Se han identificado algunas de las actividades más relevantes en relación con el proceso de Planificación-Programación de la Producción con el objeto de revisar aquellas que afectan a dicha relación y constituir el glosario de conceptos que serán de uso habitual a lo largo de esta tesis.

En el ámbito del largo plazo de la Figura 3-2 se encuentran las actividades directamente relacionadas con la **Toma de Decisiones de carácter Estratégico**. Como resultado de dichas actividades se elabora un **Plan Agregado de Producción** del cual se puede obtener un **Plan Maestro Previsional**.

Por un lado, se debemos entender que el **Plan Agregado de Producción** especifica los requisitos de producción por grupos (tipos o familias) de productos principales por periodos (mensuales por ejemplo), ya sea en horas o en unidades de producción. Las entradas principales del mismo son la **Previsión Agregada de la Demanda**, y el **Plan de Recursos y Capacidades** a largo plazo. El objetivo de éste suele ser encontrar la distribución de niveles de inventario y mano de obra que minimice los costes durante el horizonte de Planificación, aunque pueden existir otros enfoques.

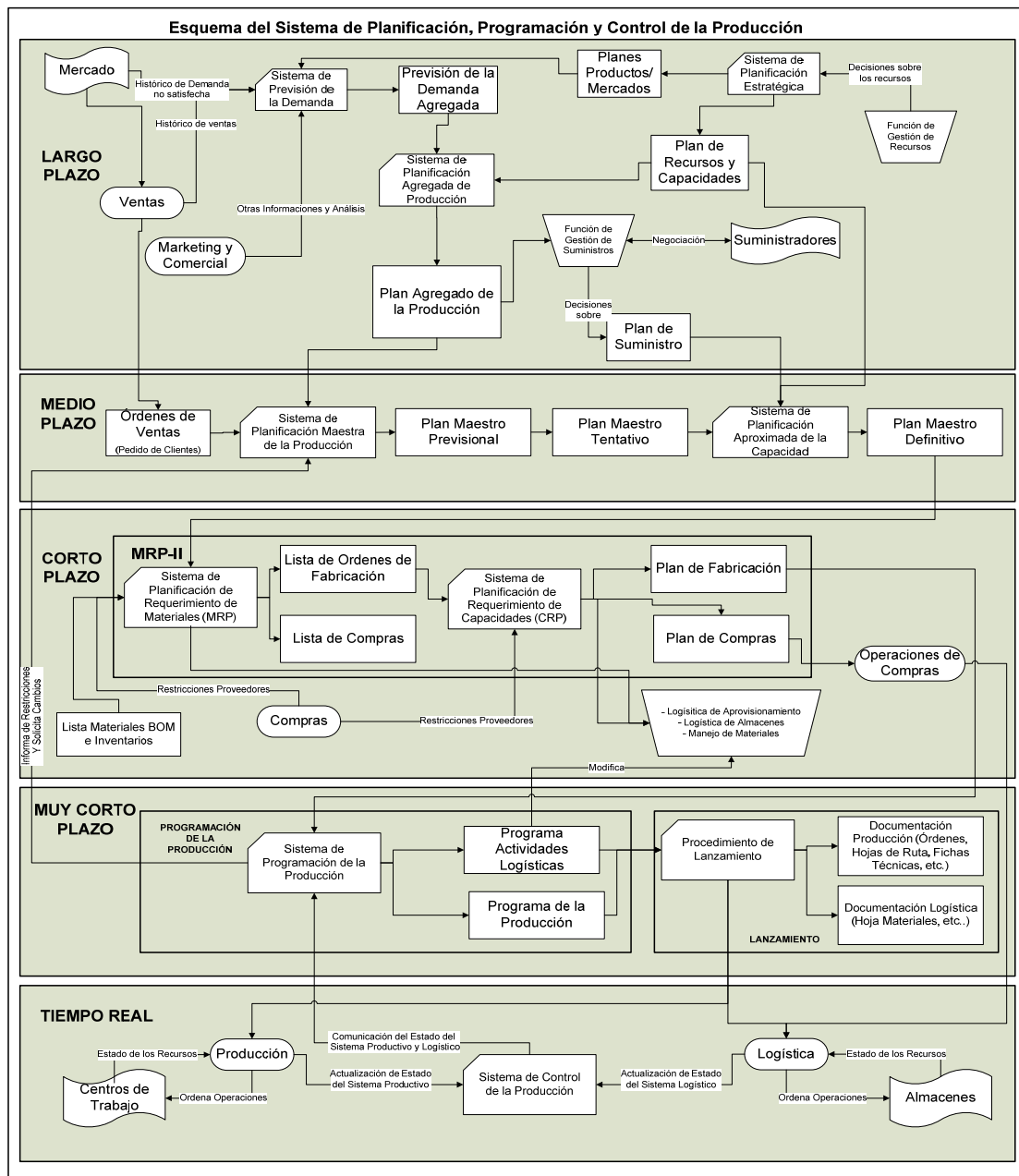


Figura 3-2. Representación del Sistema de Planificación y Control de Operaciones (adaptado de la propuesta elaborada en (Vicens, E. et al. 1999))

Por otro lado, se debe entender que el **Plan Maestro Previsional** proyecta las cantidades y fechas de producción de los artículos específicos de forma que sirve de aproximación al resultado final. Las unidades de productos a fabricar se obtienen mediante desagregación. La entrada principal de este Plan es el **Plan Agregado de Producción**, y por lo tanto, la información que se maneja es poco detallada.

Si se sigue la Figura 3-2 se puede observar como *el ámbito del medio y corto plazo están asociados con el proceso de Planificación de la Producción*. Esta visión modifica la que ofrece la Figura 3-1, ampliando el horizonte de actuación del Planificador del Medio al Medio y Corto Plazo.

Las actividades relacionadas con el medio plazo toman como elemento de partida básico el **Plan Maestro Previsional** a partir del cual se elabora en primer lugar el **Plan Maestro Tentativo** y posteriormente el **Plan Maestro “Definitivo”** (coincide con el concepto de Master Plan Schedule, **MPS**). El **MPS** es consecuencia de la consideración Aproximada de las Capacidad y Recursos. Ambos Planes se pueden considerar como sucesivas actualizaciones del **Plan Maestro Previsional** en donde, con un nivel de detalle más elevado y actualizado, se tienen en cuenta fundamentalmente aspectos relacionados con las ventas y las capacidades y recursos. El **MPS proyecta las cantidades y fechas de producción de los artículos específicos**. En los primeros periodos (corto plazo) el **MPS** se considera fijo.

En el corto plazo se desarrollan las actividades de Planificación de la Producción más próximas al ámbito operativo. Se trata de aquellas actividades que tienen como objetivo generar las peticiones sobre cantidades a producir (piezas, montajes intermedios, productos finales, etc.) y cantidades de materiales a adquirir (materias primas, semi-elaborados, etc.). Los resultados de las actividades de este ámbito repercuten por lo tanto en la Programación de Operaciones Productivas y Logísticas.

Una vez elaborado el **Plan Maestro de Producción**, y teniendo en cuenta adicionalmente las restricciones relacionadas con las compras, se calculan los **Requerimientos de Materiales o MRP** (Material Requirement Planning), del que se obtienen dos resultados. Por un lado, el conjunto de trabajos a fabricar, y por otro las necesidades de materiales que es imprescindible comprar. El primer resultado se ha identificado con el nombre de **Lista de Órdenes de Fabricación** y el segundo con el de **Lista de Compras**. Cada actividad (compras, chequeos de

ⁱ Como se verá a lo largo de esta tesis la consideración de Plan Definitivo es simplemente intencional, ya que se contemplaran posibles modificaciones del mismo incluso durante su ejecución.

materias primas, producción de piezas, etc.) que se deriva del **MRP** tiene asignada una fecha máxima de disponibilidad que garantiza que el producto que ha originado dichas actividades podrá estar disponible en la fecha que se marcó en el **Plan Maestro de Producción**. Dado que el **MRP** no tiene en cuenta la capacidad se pueden dar situaciones de no factibilidad de la propuesta resultante por falta de recursos, por eso se ha denominado "*Lista*" y no "*Plan*" a las propuestas. Por este motivo, sobre la Lista de Órdenes se suele aplicar un análisis o **Planificación de los Requerimientos de Capacidad o CRP** (Capacity Requirement Planning) que puede implicar un alisamiento de las cargas, y en general una redistribución de las actividades, para garantizar la disponibilidad de recursos suficientes para realizar las operaciones. Como resultado del **CRP** se obtiene un **Plan de Fabricación** y un **Plan de Compras**. Ambos elementos son resultado de refinar las **Listas de Órdenes de Fabricación** y de **Compras** para ajustarlas a los recursos existentes. El *sistema formado por la Planificación de Requerimientos y el Análisis de Capacidades (MRP + CRP) es conocido como MRP de Bucle Cerrado (o MRP-II)*.

En el ámbito del muy corto plazo se encuentran los procedimientos de **Programación de la Producción y de Lanzamiento o Implementación del Programa de Producción**. En cuanto a la Programación de la Producción (Apics 1994) define la Programación de Operaciones como la "asignación efectiva de fechas de inicio y/o término a operaciones o grupos de operaciones para calcular el momento en que estas operaciones deben ser realizadas para que la orden de fabricación se complete a tiempo". Estas fechas se usan en la función de asignación, selección y secuenciación de los trabajos. Y como se puede ver en la Figura 3-2 como consecuencia de dicha actividad se deben obtener los programas relativos a las actividades productivas como logísticas. Para poder realizar correctamente el cálculo **del Programa Productivo** es necesario disponer como entradas del proceso el **Plan de Fabricación** y el *Estado Actual del Sistema Productivo y Logístico*.

Una definición de la Programación de la Producción integrada en el **SPCO** es la que ofrecen (Companys, R. & Corominas, A. 1996), para las cuales la Programación de la Producción se define como "el proceso de asignación de órdenes de fabricación y/o las operaciones en que se descomponen, a recursos específicos dentro de intervalos de tiempo concretos". La fase de Programación se engloba dentro del **SPCO**, como una etapa posterior a la Planificación y el cálculo de necesidades. Es el paso previo a la ejecución física de las operaciones de fabricación y, a pesar de su importancia, en muchos entornos productivos, la carga de máquinas y Programación de la Producción es llevada a cabo por personal de nivel operativo (capataces, encargados de línea o los propios operarios).

Estos autores diferencian entre tres subfunciones que tradicionalmente se han distinguido dentro de la función de Programación de la Producción. Éstas pueden ejecutarse simultáneamente o no:

- **Subfunción carga (loading):** Asignación de las operaciones a centros de trabajo, decisión que se adoptará por comparación entre la capacidad disponible del centro y la carga requerida por las operaciones ya asignadas al mismo.
- **Subfunción secuenciación (sequencing):** secuenciación de las operaciones asignadas a un centro de trabajo para establecer su orden de ejecución.
- **Subfunción temporización (scheduling):** Determinación de los instantes de inicio y fin (programados) de cada operación.

Sin embargo, y volviendo sobre la Figura 3-2, se puede apreciar que una vez el Programa Productivo está disponible este debe ser ejecutado para lo cual se necesita realizar un procedimiento de **Lanzamiento**. Las actividades de **Lanzamiento** están encaminadas a proporcionar el conjunto de elementos de soporte que faciliten *la transmisión de los Programas Productivos y Logísticos al conjunto de componentes activos* (básicamente recursos humanos) y que permitan realizar de forma efectiva el trabajo previsto.

Algunos de los objetivos perseguidos por la Programación de la Producción (dentro de las consideraciones generales establecidas por la Planificación) suelen ser:

- Terminar dentro de plazo un alto porcentaje de órdenes.
- Obtener una alta utilización del equipo o del personal.
- Reducir al mínimo las horas extra.
- Reducir al mínimo la obra en curso, etc.

Estos objetivos son claramente antagónicos entre sí, dada una situación concreta del sistema productivo, por lo que se suele establecer una jerarquización entre ellos. Además, el sistema de Programación de Producción tiene que interaccionar con otras funciones necesarias dentro del **SPCO**. Estas interacciones son dependientes del sistema y pueden diferir sustancialmente de una situación a otra (Pinedo, M. 1995). En concreto, dentro del **SPCO**, los pedidos deben ser transformados en órdenes de producción o trabajos con sus fechas de entrega asociadas. Los trabajos deben ser procesados por las máquinas en los centros de trabajo en un determinado orden o secuencia y deben esperar a ser procesados en las máquinas que estén ocupadas, pudiendo ocurrir interrupciones cuando llegan órdenes de una prioridad superior a la que se está ejecutando actualmente. La Programación de la Producción detallada de las tareas a realizar en un sistema de producción es necesaria para mantener la eficiencia y el control de las operaciones.

Finalmente, en el ámbito del tiempo real se encuentran las actividades de **Control de la Producción**. Estas tienen como objetivo monitorizar las actividades que forman parte de las operaciones en curso y contrastar si la información percibida coincide con la prevista. Los controladores deben de establecer rutinas que permitan detectar anomalías.

Un aspecto relevante del proceso de **Planificación-Programación de la Producción** es la influencia que la información procedente desde el ámbito del **Control de la Producción** tiene sobre el ámbito de la **Programación de la Producción** y de cómo la información procedente de éste último puede influir en las decisiones que se tomen en el ámbito de **Planificación**.

En la Figura 3-2 se puede observar como el Sistema de Control de la Producción debe comunicar el estado del sistema Productivo y Logístico al Sistema de Programación de la Producción. A su vez en la misma figura se puede observar como el Sistema de Programación de la Producción informa de restricciones y solicita cambios en el Plan Maestro de Producción mediante comunicación directa con el Sistema de Planificación de la Producción. Un análisis del flujo de información de parte del ámbito del corto plazo y termina en el medio plazo, y que conforma lo que debería ser la comunicación Programador- Planificador, puede dar como resultado que en muchas ocasiones éste no se recoge en algún tipo de procedimiento de trabajo, ya que se suelen plantear casos estáticos y deterministas en el que no son necesarias estas comunicaciones.

Así mismo, como las decisiones tomadas en el ámbito de Planificación, influyen en la Programación, la información generada en el ámbito de control de planta, también repercute en ésta. Una forma de coordinar las diferentes decisiones tomadas consiste en plantear sistemas de Planificación Jerárquica de la Producción (Hax, A. C. & Meal, H. C. 1975; Lario F.C. 1986; Vicens, E. 1988). Los conceptos de agregación de piezas en familias que se usan en Planificación Jerárquica se pueden abordar mediante la aplicación de la tecnología de grupos (TG).

Desde el punto de vista terminológico es importante identificar algunos vocablos comunes en este ámbito pero con significados poco unificados en la literatura. Por un lado, se puede establecer que la unidad básica en la que se puede descomponer el *Plan Maestro de Producción* es el **Lote**, y que este queda determinado por el artículo comercial y la cantidad a fabricar del mismo, así como por la fecha de disponibilidad prevista. Por otro lado, la unidad básica del *Programa de Producción* es el **Trabajo**, que queda determinado por la referencia comercializable o no (pieza, compuesto, sub-montaje o referencia), la cantidad a fabricar, la operación que hay que realizar, el recurso sobre la cual se prevee realizar la operación, y la fecha de inicio y fin de la operación. En la terminología de esta tesis se entiende por **Orden de**

Fabricación al conjunto de Trabajos agrupados con un fin común (normalmente ejecución en un mismo periodo).

3.3 Teoría de la Programación de Producción

3.3.1 Introducción

En este apartado se cree conveniente hacer un breve resumen de los planteamientos tradicionales de abstracción, modelado y resolución matemática del problema. Si bien el *número de referencias que existen en este campo es inmenso*, se ofrece a continuación una visión general de este enfoque teórico junto con aquellas referencias que se han considerado fundamentales para profundizar más en el tema.

A partir de la abstracción del problema real de Programación de Producción, se han desarrollado una serie de técnicas que han originado un gran número de resultados teóricos desde hace más de tres décadas. A todo esto se le denomina frecuentemente Teoría de Programación de Producción-Secuenciación (Theory of Scheduling), inspirándose en el trabajo pionero de (Conway, R. W., Maxwell, W. L., & Miller, L. W. 1967).

Sin embargo, hay que destacar que en el entorno industrial real, en la mayoría de los casos, no se suelen aplicar los resultados obtenidos a partir de las investigaciones académicas en teoría de Programación de Producción, desarrolladas desde hace más de tres décadas, y que podrían ayudar a mejorar los programas de fabricación en las empresas. Esto puede ser debido a varios factores:

- La teoría de Programación de la Producción y los métodos de resolución utilizados, son desconocidos o son difíciles de entender por las personas encargadas de desarrollar la función de Programación de Producción. La solución a este problema, pasa por una mejora en la capacitación del personal encargado de Programar la Producción y el desarrollo de procedimientos, que hagan más amigable la resolución de los problemas por personal no excesivamente cualificado en teoría de secuenciación.
- **Las situaciones analizadas en la teoría no son suficientemente cercanas a las encontradas en la práctica.** Es común analizar problemas simplificados, que después son difíciles de extrapolar al entorno industrial real. Sin embargo, cada vez más, están apareciendo desarrollos que permiten un mejor acercamiento a la realidad. La aplicación de las técnicas heurísticas, mucho más flexibles en este aspecto, permiten evitar este problema.

A continuación se describirán diversos enfoques que se han usado para abordar los problemas en teoría de Programación de la Producción.

3.3.2 Programación en contexto Determinista o Incierto

La mayor parte de las herramientas tradicionales usadas para modelar problemas de toma de decisiones son **Deterministas**. Por **Determinista** se entiende que el modelo se conoce con precisión y certidumbre (Maccarthy, B. L. & Liu, J. Y. 1993; Mula, J. 2004). La precisión asume que los parámetros de un modelo representan exactamente la percepción del fenómeno o las características del sistema modelado. La certidumbre indica que se asume que las estructuras y parámetros se conocen definitivamente y no existen dudas sobre sus valores u ocurrencia.

En los problemas de Programación de la Producción las situaciones reales son frecuentemente **Imprecisas o Inciertas**. Debido a la falta de información el estado futuro del sistema podría no ser completamente conocido. Según (Galbraith, J. 1973) la diferencia entre la cantidad de información requerida para ejecutar una tarea y la cantidad de información que realmente posee una organización se entiende como **Incertidumbre**. Este tipo de **Incertidumbre** de carácter estocástico se ha gestionado tradicionalmente mediante teoría de probabilidad y la estadística (Kolmogoroff, A. 1950; Koopman, B. O. 1940). Para abordar cuantitativamente la imprecisión, se pueden utilizar los conceptos y técnicas de la teoría de la probabilidad, complementados por las herramientas proporcionadas por la Teoría de la Decisión, la Teoría del Control, la Teoría de la Información (Mula, J. 2004).

Muchos de los procesos de **Toma de Decisiones en el mundo real ocurren en entornos en los que las metas, las restricciones y las consecuencias de las posibles acciones no se conocen de forma precisa**. Esta imprecisión puede pertenecer a dos categorías: **Incertidumbre estocástica** (o aleatoria) o **Borrosidad**. En (Zimmermann, H. J. 1996) se diferencia entre la aleatoriedad producida por las imprecisiones del mundo real, que se denomina **Incertidumbre estocástica** en contraste a la imprecisión presente en la descripción del significado semántico de los eventos, fenómenos o sentencias, que denomina **Borrosidad**. (Bellman, R. & Zadeh, L. A. 1970) *diferencia entre aleatoriedad y borrosidad* (fuzziness). La aleatoriedad se relaciona con la Incertidumbre concernida a la pertenencia o no pertenencia de un objeto a un conjunto no difuso. Por otro lado, la **Borrosidad considera clases en las que existen grados de pertenencia entre la completa pertenencia y la no pertenencia**. En esta tesis se consideraran aspectos relacionados con la Borrosidad en los datos del problema a resolver.

Una vez se ha reflexionado sobre el concepto de imprecisión en relación con el proceso de modelado de la realidad para la resolución de problemas es conveniente realizar una revisión

sobre las causas que la producen y su alcance. En (Ho, C. 1989) se clasifica la Incertidumbre en dos grandes grupos, *la correspondiente al entorno y la correspondiente al sistema productivo*. (McKay, K. N., Buzacott, J. A., & Safayeni, F. R. 1989) distingue tres tipos de Incertidumbre en los sistemas productivos: *completamente desconocidas, sospechosas y conocidas*. Las primeras se refiere a los eventos impredecibles, y ante las cuales solo cabe reaccionar. Las segundas son intuitivas en base a la experiencia del programador, pero son difíciles de incorporar en un Programa Productivo. Las terceras son aquellas que pueden ser previstas y que se pueden incorporar en los algoritmos y programas. En (Vieira, G. E., Herrmann, J. W., & Lin, E. 2003) se identifican un conjunto de eventos que producen Incertidumbre en un sistema productivo y que tienen un impacto considerable sobre los Programas de Producción en curso.

No obstante, no se debe considerar la causa como único o principal criterio de clasificación de la Incertidumbre. En (Aytug, H., Lawley, M. A., McKay, K., Mohan, S., & Uzsoy, R. 2005; Henning, G. P. & Cerda, J. 1995; Li, R. K., Shyu, Y. T., & Adiga, S. 1993) se estableció una taxonomía para el análisis de los problemas con Incertidumbre relacionada con el sistema productivo basada en cuatro dimensiones clave. Estas cuatro dimensiones para el análisis son: **causa, contexto, impacto e inclusión**. *Un procedimiento de Programación de la Producción en un contexto de Incertidumbre debe definir el alcance adecuado en cada una de las cuatro dimensiones que le permita cubrir los objetivos propuestos.*

Los problemas de Programación de la Producción, que inicialmente se solían plantear en base a un Modelo Determinista, con el tiempo se han ido desarrollando otros enfoques que pretenden modelar más fielmente la Incertidumbre que caracteriza a este tipo de sistemas (Pinedo, M. 1995). En estos Modelos Estocásticos o Borrosos, se asume que los datos de cada trabajo (tiempos de proceso, fechas de lanzamiento o fechas de entrega) pueden no conocerse con exactitud, sino que se conoce una distribución probabilística de los mismos o una función de pertenencia. Sólo después de ejecutarse el evento se conoce su valor exacto.

A continuación se muestra un conjunto de referencias, relacionadas con la Programación de Producción con Incertidumbre, clasificadas en función del evento que consideran de forma incierta (Vieira, G. E. et al. 2003):

- **Fallos en las máquinas:** (Abumaizar, R. J. & Svestka, J. A. 1997; Akturk, M. S. & Gorgulu, E. 1999; Alvarez, E. & Diaz, F. 2004; Black, G. W., McKay, K. N., & Messimer, S. L. 2004; Bollapragada, R. & Sadeh, N. M. 2004; Chen, J. & Chen, F. F. 2003; Church, L. K. & Uzsoy, R. 1992; Fang, J. & Xi, Y. G. 1997; Guo, B. & Nonaka, Y. 1999; Jain, A. K. & Elmaraghy, H. A. 1997; Jensen, M. T. 2003; Kim, M. H. & Kim, Y. D. 1994; Kumar, R., Tiwari, M. K., & Allada, V. 2004; Li, R. K. et al. 1993; Mason, S. J., Jin, S., & Wessels, C. M. 2004; Mendez, C. A. & Cerda, J. 2004; Rossi, A. & Dini, G. 2000; Sabuncuoglu, I. & Bayiz,

M. 2000;Sabuncuoglu, I. & Comlekci, A. 2002;Sabuncuoglu, I. & Karabuk, S. 1999;Sabuncuoglu, I. & Kizilisik, O. B. 2003;Shafaei, R. & Brunn, P. 2000;Shafaei, R. & Brunn, P. 1999b;Sun, J. & Xue, D. 2001;Wu, S. D., Byeon, E. S., & Storer, R. H. 1999;Yamamoto, M. & Nof, S. Y. 1985).

- **Trabajos urgentes:** (Abumaizar, R. J. et al. 1997;Alvarez, E. et al. 2004;Hall, N. G. & Potts, C. N. 2004;Jain, A. K. et al. 1997;Kim, M. H. et al. 1994;Li, R. K. et al. 1993;Mason, S. J. et al. 2004;Subramaniam, V., Raheja, A. S., & Reddy, K. R. B. 2005b;Sun, J. et al. 2001), o llegadas de nuevos trabajos como, (Hall, N. G. et al. 2004;Jang, W. S. & Klein, C. M. 2002;Lodree, E., Jang, W. S., & Klein, C. M. 2004), o demanda de trabajos impredecible como, (Jensen, M. T. 2003;Kumar, R. et al. 2004;Sabuncuoglu, I. et al. 2002).
- **Cancelación de trabajos:** (Abumaizar, R. J. et al. 1997;Jain, A. K. et al. 1997;Li, R. K. et al. 1993;Sun, J. et al. 2001).
- **Cambio en las fechas de entrega:** (Fang, J. et al. 1997;Henning, G. P. et al. 1995;Lejmi, T. & Sabuncuoglu, I. 2002;Li, R. K. et al. 1993;Sun, J. et al. 2001).
- **Retraso o falta de materia prima:** (Alvarez, E. et al. 2004;Fang, J. et al. 1997;Henning, G. P. et al. 1995;Li, R. K. et al. 1993;Rossi, A. et al. 2000;Sun, J. et al. 2001).
- **Cambio en la prioridad de los trabajos:** (Alvarez, E. et al. 2004;Henning, G. P. et al. 1995;Jain, A. K. et al. 1997).
- **Problemas de calidad:** (Church, L. K. et al. 1992).
- **Desviación en los Tiempos de Proceso previstos:** (Li, R. K. et al. 1993). Adicionalmente, podemos reseñar aquellos que presentan Incertidumbre en el tiempo de proceso como, (Anglani, A., Grieco, A., Guerriero, E., & Musmanno, R. 2005;Daniels, R. L. & Kouvelis, P. 1995;Hauptman, B. & Jovan, V. 2004;Jang, W. S. et al. 2002;Kutanoglu, E. & Wu, S. D. 2004;Lejmi, T. et al. 2002;Mendez, C. A. et al. 2004;Montoya-Torres, J. R. 2003;Sabuncuoglu, I. et al. 2002;Sabuncuoglu, I. et al. 2003;Subramaniam, V. et al. 2005b). (He, Y., Smith, M., & Dudek, R. 1994) realizaron un estudio sobre el efecto de la variación del tiempo de proceso sobre las reglas de despacho en un entorno dinámica, llegando a la conclusión de que una moderada variación no afecta al rendimiento de las reglas. (Lawrence, S. R. & Sewell, E. C. 1997) investigaron el efecto de los tiempos de proceso estocásticos sobre métodos de Programación de la Producción y encontraron que cuando la variabilidad aumenta los métodos óptimos son superados por las heurísticas y las reglas de despacho.
- **Variación en la plantilla de trabajadores:** (Alvarez, E. et al. 2004;Church, L. K. et al. 1992;Jensen, M. T. 2003;Subramaniam, V. et al. 2005b;Sun, J. et al. 2001).

- **Cambios en el taller**ⁱⁱ: (Black, G. W. et al. 2004).
- Pueden existir otro tipo de situaciones que conduzcan a generar Incertidumbre, sin embargo las mencionadas anteriormente suelen ser las más habituales en un entorno fabril.

3.3.3 Programación en contextos Estáticos, Dinámicos, en Línea, o en Tiempo Real.

Existen dos tendencias claras en la literatura en lo que se refiere a *como afrontar un problema de Programación de la Producción con respecto a la evolución temporal* de los eventos, las prioridades, los recursos, las necesidades, el entorno y los requerimientos considerados. Esas tendencias conocidas de forma simplificada como **Estática y Dinámica**, tienen en la segunda un abanico de denominaciones alternativas. En este apartado diferenciaremos entre el enfoque **Estático** y **Dinámico** de la Programación de la Producción, y sus alternativas.

Existen al menos tres aspectos que se deben considerar a la hora de abordar un problema de Programación de Producción en lo que se refiere al aspecto temporal, que son: El **modelo de la realidad** que se desea reflejar, el **momento que se Toman las Decisiones** y el **entorno sobre el cual se actúa**.

Por lo que se refiere al modelo que se desea considerar, los problemas pueden ser divididos en **Estáticos** o **Dinámicos**. Un problema **Estático** es aquél en el que los eventos, las prioridades, los recursos, las necesidades, el entorno y los requerimientos son finitos y son conocidos en el instante inicial. Por lo tanto, en este tipo de problemas, el horizonte de Programación no se extiende más allá de un periodo determinado. Mientras que un programa se entiende como **Dinámico** si se realiza en un entorno volátil que necesita frecuentes cambios para adaptarse a los requerimientos (Stankovic, J. A., Spuri, M., Marco, D. N., & Buttazzo, G. C. 1995; Suresh, V. & Chaudhuri, D. 1993).

Si se atiende al momento en el que se toman las decisiones o estrategia los programas pueden estar **en Línea** o **Fuera de Línea**. Se dice que un programa se ejecuta **en Línea** (on-line) si la

ⁱⁱ En este trabajo se entiende por “Taller” al Sistema Productivo o Fabril que es objeto de análisis. Éste puede estar compuesto por uno o varios Centros de Trabajo, que a su vez pueden estar compuestos por múltiples recursos.

Programación de la Producción se está realizando al mismo tiempo que la producción está en marcha (Narasimhan, S. L., McLeavey, D. W., & Billington, P. J. 1995). Por otro lado, se entiende que se ejecuta **Fuera de Línea** (off-line) si la Programación de la Producción se realiza una vez acabado el programa anterior y en un momento de parada productiva (Riggs, J. L. 1987). Pueden existir procesos mixtos, en donde existen un conjunto de Decisiones Tomadas **Fuera de Línea**, y otras que se toman **en Línea** cuando se detectan eventos que afectan al Programa Inicial.

Por lo que se refiere al sistema que gestiona el Programa de la Producción se entiende que el Programa se realiza **en Tiempo Real** si la actividad se ejecuta simultáneamente a la realidad (Garvey, A. J. & Lesser, V. R. 1993; Sha, L. & Goodenough, J. B. 1990).

Por lo tanto, un problema **Dinámico**, considera el horizonte de funcionamiento del taller ilimitado hacia el futuro, aunque el conocimiento concreto en un instante determinado se limite a la situación actual, y a una cartera de piezas finita a realizar. En el taller **Dinámico** no se puede hallar un Programa Productivo único, sino un conjunto de Programas sucesivos desarrollados a lo largo de ciclos de Reprogramación. Por tanto, *no podrá juzgarse la calidad de un Programa Productivo sino, en todo caso, la de un procedimiento de Reprogramación*, y la naturaleza del criterio que permita juzgar la calidad del procedimiento. Este procedimiento estará entroncado con el comportamiento medio de los Programas Productivos obtenidos utilizándolo frente a las entradas (características de llegada y estructura de las piezas) a que es sometido.

En (Vieira, G. E. et al. 2003) se justifica la no existencia de clasificaciones adecuadas en un entorno de **Dinámico** frente a la existencia clasificaciones en un entorno estático como la de (Herrmann, J. W., Chung-Yee, L., & Snowdon, J. 1993). En (Vieira, G. E. et al. 2003) se realiza una diferenciación del problema **Dinámico-Estático** de la siguiente forma:

- **Estático**: Entendiendo por tal, el caso en el que existen un conjunto de trabajos finitos durante un periodo finito.
- **Determinista**: No se contemplan incertidumbres en el modelo.
- **Estocástico**: Existen cierta variables aleatorias en el modelo. Algunas referencias interesantes se encuentran en (Daniels, R. L. et al. 1995; Herrmann, J. W. 1999; Pinedo, M. & Chao, X. 1999; Wu, S. D. et al. 1999)
- **Dinámico**: Entendiendo por tal, el caso en el que existen un conjunto de trabajos infinitos.

- **Sin variación en las llegadas de trabajos:** Se conoce el momento de entrada en el taller de cada trabajo. Los trabajos siguen un comportamiento cíclico hasta el infinito. Algunas referencias interesantes se encuentran en (Hall, N. G. & Sriskandarajah, C. 1996; Kamoun, H. & Sriskandarajah, C. 1993; Lee, T. E. & Posner, M. E. 1997; Matsuo, H. 1990; Pinedo, M. et al. 1999; Roundy, R. 1992).
- **Con variación en la llegada de los trabajos:** Existen incertidumbre sobre el momento de entrada en el taller de los trabajos, pero las operaciones requeridas son conocidas. Algunas referencias interesantes se encuentran en (Markowitz, D. M. & Wein, L. M. 2001; Mehta, S. V. & Uzsoy, R. M. 1998)
- **Con variación en el flujo de trabajo:** Los trabajos pueden sufrir cambios en la ruta prevista cuando llegan al taller. Algunas referencias interesantes se encuentran en (Church, L. K. et al. 1992).

En (Abumaizar, R. J. et al. 1997; Vieira, G. E., Herrmann, J. W., & Lin, E. 2000b) ya se proponía la misma diferenciación entre Dinámica y Estática. **La Visión Dinámica** basada en la existencia de un conjunto de trabajos que van llegando al taller también está avalada por (Fang, J. et al. 1997).

Tanto los Enfoques Estáticos como Dinámicos se ven afectados por la Incertidumbre. Los modelos estáticos pueden incorporar la imprecisión como una función de probabilidad, pero los modelos dinámicos pueden optar adicionalmente a esperar a que se produzca el evento y ofrecer la respuesta adecuada en función de las circunstancias correspondientes al estado en dicho instante.

El **Enfoque Dinámico** del modelo correspondiente al problema de Programación de la Producción implica la **definición de Estrategias y Métodos de Programación de la Producción en un Entorno Dinámico**. Las estrategias están relacionadas con la Toma de Decisiones (on-line o off-line) y las políticas concretas sobre Reprogramación. Los métodos están relacionados con los algoritmos concretos que son más adecuados utilizar. Ambos aspectos se analizarán en los próximos apartados.

3.4 Descripción del Problema Dinámico y con Incertidumbre de Programación de la Producción. Enfoques Generales de Resolución Distribuidos.

3.4.1 Definiciones e Hipótesis

Suponiendo que existen n trabajos $\{J_1, J_2, \dots, J_n\}$ y m máquinas disponibles $\{M_1, M_2, \dots, M_m\}$. Según (Brucker, P. 2006) un Programa consiste en la ubicación de cada trabajo en uno o más intervalos de tiempo en una o más máquinas. El Problema de Programación consiste en encontrar dicha ubicación bajo ciertas restricciones.

Esta definición puede ser generalizada, reemplazando las máquinas por etapas de un proceso que pueden, a su vez, contener varias máquinas. Se debe entender que una Etapa está asociada a una funcionalidad dentro del Taller y que cada trabajo realiza sus operaciones en diferentes pasos o Etapas. En este caso, aparece un problema de asignación de los trabajos a las máquinas y secuenciación de los trabajos en cada máquina.

Las siguientes hipótesis aparecen frecuentemente en la literatura de la teoría de secuenciación y son debidas a (Conway, R. W. et al. 1967):

- Las máquinas están siempre disponibles y nunca dejan de funcionar.
- Cada máquina puede procesar a lo sumo un trabajo a la vez.
- Cualquier trabajo puede ser procesado únicamente en una máquina a la vez.
- Los tiempos de preparación de todos los trabajos son cero, por ejemplo, todos los trabajos están disponibles al comienzo del proceso.
- No se permiten interrupciones, es decir, cuando una operación ha comenzado debe terminarse antes de empezar otra en la misma máquina.
- Los Tiempos de Cambio son independientes de los programas/secuencias y están incluidos en los Tiempos de Procesado.
- Los tiempos de procesado y las restricciones tecnológicas son deterministas y se conocen de antemano, y similarmente ocurre con las fechas de entrega.

Sin embargo, hay que tener presente que los entornos productivos son por naturaleza Dinámicos y sujetos a diversos tipos de Perturbaciones. Estos eventos pueden cambiar el

estado del sistema y afectar al rendimiento del mismo. La literatura relativa a la Programación Dinámica de la Producción considera un número significativo de eventos producidos sobre sistemas controlados en tiempo real, así como sus efectos en varios sistemas productivos (maquina única, máquinas paralelas, talleres de flujo, talleres de trabajo, etc.). Los autores de (Stoop, P. P. M. & Weirs, V. C. S. 1996) en su trabajo sobre la complejidad de la Programación de la Producción describieron dos categorías de eventos, *los relacionados con los recursos y los relacionados con los trabajos*. Esta misma categorización es apoyada por (Cowling, P. & Johansson, M. 2002; Mehta, S. V. & Uzsoy, R. 1999). En el trabajo de (Suresh, V. et al. 1993) sobre Programación Dinámica de la Producción pusieron de manifiesto el efecto de la presencia de eventos tratados en Línea sobre sistemas en Tiempo Real, y propusieron tres categorías: *los relacionados con los trabajos, los relacionados con los recursos, y los relacionados con las operaciones*.

Los **eventos** relacionados con los recursos hacen referencia a los fallos de las máquinas, bajas de los operarios, no disponibilidad de las herramientas, no disponibilidad de materiales, etc.

Los **eventos** relacionados con los trabajos hacen referencia a la llegada de las órdenes, la cancelación de trabajos, cambios en las fechas de entrega, cambios en la prioridad de los trabajos, cambios en los tiempos de proceso, etc.

Si se aplican las hipótesis de (Conway, R. W. et al. 1967) a las Empresas Cerámicas, contempladas con un enfoque dinámico, es necesario realizar los siguientes comentarios:

- La hipótesis 1 no se cumple. **No se puede asegurar que no se produzca un fallo en una máquina que retrase el proceso productivo**, a pesar del alto grado de automatización. También se puede determinar la necesidad de parar una máquina en el transcurso de un Programa de Producción de acuerdo con una estrategia.
- Las hipótesis 2 y 3, se pueden asumir puesto que las características de la maquinaria hacen que no sea posible procesar trabajos distintos simultáneamente.
- Debido a la aplicación de sistemas M.R.P., y al alto grado de coordinación con los proveedores todos los materiales suelen estar disponibles al inicio de un trabajo. No obstante en los trabajos no previstos no ocurre lo mismo, por lo que la hipótesis 4 no se puede asumir.
- En cuanto a la hipótesis 5, ésta se cumple, puesto que las interrupciones implicarían costes de arranque de la maquinaria.

- Sin embargo, la hipótesis 6 no se cumple, porque en el caso cerámico los Tiempos de Cambio, tal y como se ha discutido, son dependientes de la secuencia y no asimilables a los Tiempos de Proceso.
- Por último, la hipótesis 7 se cumple parcialmente. Por lo que se refiere a los tiempos pueden aparecer, pequeñas variaciones en los Tiempos de Proceso y Cambio, su magnitud suele ser pequeña pero no se pueden descartar variaciones significativas por problemas de ajuste con respecto al patrón de calidad establecido. Por lo que se refiere a las fechas de entrega se puede decir que son deterministas.

3.4.2 Tipos de Configuraciones Productivas y Nomenclatura

Al considerar un problema de Programación de la Producción deberían tenerse en cuenta las restricciones tecnológicas de los trabajos, determinadas principalmente por el tipo de flujo de los mismos a lo largo de las máquinas del taller (hojas de ruta). Así se pueden definir las siguientes configuraciones productivas:

- **Taller monomáquina** (single machine shop).
- **Taller Abierto** (Open shop). No existe predefinido un flujo para cada trabajo.
- **Taller de Flujo** (Flow shop). Todos los trabajos tiene la misma ruta.
- **Taller de Flujo de Permutación** (Permutation flow shop). Es un caso especial del taller de flujo en el que todos los trabajos pasan en la misma secuencia por todas las máquinas del taller.
- **Taller de trabajo** (Job shop). También denominado por algunos autores, taller mecánico, puesto que es en este tipo de industrias donde primero se definió. En este caso, cada trabajo tiene su ruta específica, por lo que el problema de Programación de la Producción se complica enormemente.

En el caso en el que consideran que existe más de una máquina capaz de ejecutar la misma operación aparece el concepto de Etapa. En efecto, si $M(\lambda)$ es el número de máquinas en la Etapa λ ($\lambda=1,\dots,k$) y k el número de etapas, se pueden considerar los siguientes casos:

- **Máquinas paralelas**. M máquinas idénticas sólo en una etapa de trabajo. Cada trabajo necesita solamente una de las máquinas.

- **Taller de Flujo Híbrido** o con máquinas o procesadores duplicados (Flow shop with multiple machines/processor). También llamado taller de flujo híbrido (Hybrid flow shop), o taller de flujo flexible (Flexible flow shop). Cada trabajo sufre k operaciones, cada operación puede ser ejecutada por cualquiera de las $M(\lambda)$ máquinas existentes en cada etapa.
- **Taller de Trabajo con máquinas duplicadas** (Job shop with multiple machines). Es un taller de trabajo en donde existen $M(\lambda)$ máquinas capaces de ejecutar cada operación y cada trabajo que requiera esa operación sólo se procesa en una de esas máquinas.

La Figura 3-3 basada en (Maccarthy, B. L. et al. 1993) expresa gráficamente la clasificación realizada:

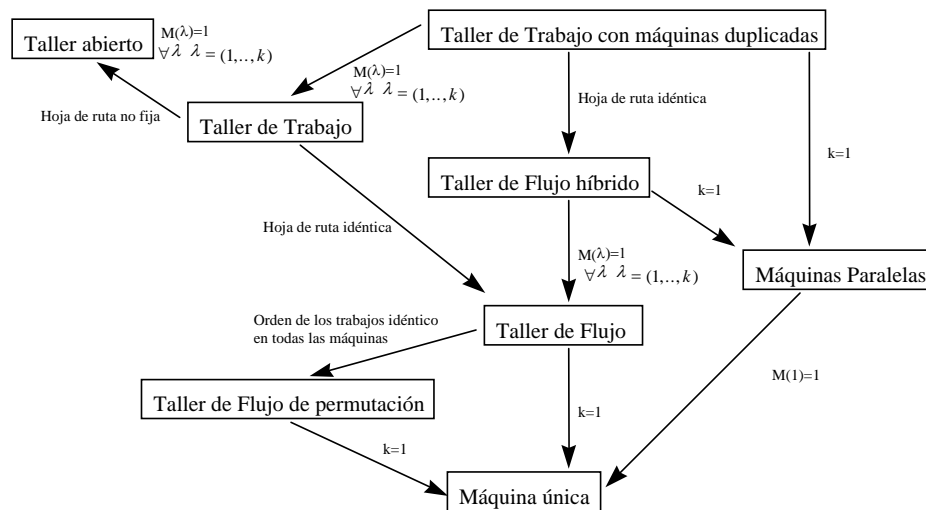


Figura 3-3. Una relación entre las configuraciones de los sistemas productivos (a partir de (Maccarthy, B. L. et al. 1993)).

Se puede observar en la figura que un taller de flujo híbrido puede ser un caso general de un taller de flujo (donde por cada operación hay más de una máquina) o un taller de máquinas paralelas (donde sólo existe una operación o etapa).

Dada la infinita variedad de los problemas de Programación, se han desarrollado diversos sistemas de clasificación que permiten rápidamente describir e identificar cada problema. Uno de los primeros sistemas desarrollados fue el de (Conway, R. W. et al. 1967). En este se describe cada problema mediante cuatro campos A/B/C/D:

- Campo A. Define los trabajos (u órdenes de fabricación) a realizar. Para problemas estáticos, especificará el número de trabajos cuya llegada se conoce exactamente $(1, \dots, n)$. Para problemas dinámicos, A identificará la distribución de probabilidad entre llegadas.
- Campo B. Determina el número y tipo de máquinas que componen el taller $(1, \dots, m)$. Cuando se consideran máquinas paralelas, el valor de este parámetro es el número de etapas de procesado y el número de máquinas de cada etapa es incluido en el descriptor C.
- Campo C. Describe el tipo de configuración del taller. Sus valores posibles son:
 - 1 : máquina única.
 - J : taller de trabajo (Job shop).
 - F : taller de flujo.
 - O : taller abierto.
 - F,perm: taller de flujo de permutación.
 - k-paralel: k-máquinas en paralelo.
 - J,k-paralel: taller de trabajo con k máquinas paralelas en cada etapa.

Con esta nomenclatura no se pueden identificar todos los tipos de problemas posibles del taller mecánico, ni tan siquiera todos los que aparecen en la literatura, muy extensa, existente sobre el tema. Algunos autores han propuesto nomenclaturas más voluminosas que generalmente son una extensión de la indicada. Para ello se añade a este campo C, una serie de abreviaturas que representan restricciones adicionales y que pueden tener lugar en entornos de secuenciación más complejos. Algunas de las abreviaturas utilizadas son:

- r_j : trabajos con diferentes instantes de entrada en el taller (ready times).
 - prec: existencia de precedencias entre las operaciones.
 - prmt: se permiten interrumpir una operación durante su realización.
 - unit: tiempos de procesado unitarios.
 - depend: trabajos dependientes.
 - setup: tiempos de cambio dependientes de la secuencia.
- Campo D. Describe el criterio de eficacia elegido para evaluar los programas, como pueden ser los criterios de F_{max} , C_{max} , F_{med} , T_{max} , etc.

Existen otras nomenclaturas pero esta suele ser la más utilizada.

3.4.3 Estrategias de Programación de la Producción en un entorno Dinámico y con Incertidumbre.

En este apartado se presentan las *diferentes estrategias que un programador puede adoptar frente a un problema de Programación de la Producción en un entorno Dinámico y con Incertidumbre*. Se entenderá por **Estrategia** en este apartado a la determinación por parte del programador de si sus decisiones se tomarán **en Línea, Fuera de Línea o con una Combinación de ambas**. En el marco de esta tesis este aspecto es relevante. Tal y como se detallará en el Capítulo 5, donde se describe con precisión el problema que se plantea abordar, en este trabajo se tiene en cuenta que existen ciertos eventos que pueden suponer una reconsideración de las condiciones del problema. Las condiciones que se consideran susceptibles de cambio son entre otras el tipo y número de trabajos para los cuales se debe realizar la Programación de la Producción. Por lo tanto, se debe analizar el problema de Programación de la Producción en un entorno cambiante que según algunos autores puede ser considerado Dinámico.

No todos los autores que son referenciados en este apartado utilizan la misma nomenclatura presentada en el apartado 3.3.3, es más, algunos de estos autores usan las mismas palabras con diferente significado o incluso intercambiando el significado de estas. Así la Estrategia denominada en el apartado 3.3.3 como “en Línea” podrá ser denominada en este apartado con un conjunto de nombres como Programación: Dinámica, o Reactiva (esta denominación puede ser usada también como estrategia mixta en cuyo caso se indicará). Programación de la Producción Predictivo-Reactiva, Reprogramación, Programación Robusta, basada en el Conocimiento o Reactiva son conceptos utilizados por algunos autores que se incluyen dentro de la esta estrategia denominada “mixta” en el apartado 3.3.3.

Con el objeto de facilitar la lectura a continuación se definirán algunos términos como los de Programación de la Producción Reactiva, Predictivo-Reactiva o el de Programación Robusta basándose en la aportación de las diversas referencias usadas en este apartado. Dichos conceptos serán matizados por los diferentes autores en caso de discrepancia.

La **Programación de la Producción Reactiva**, conocida por algunos autores como dinámica, es aquella que no proporciona ningún programa inicial de forma que las decisiones se toman en el momento que son requeridas y sin avances previos.

La **Programación de la Producción Predictivo-Reactiva** proporciona un programa inicial que no intenta prever ninguna distorsión, y cuando esta se produce el programa dispone de mecanismos para reaccionar, modificando el programa en curso y generando uno nuevo parcial o totalmente.

La **Programación Robusta de la Producción** es aquella que proporciona un programa de forma anticipada (Programación Predictiva) que minimiza el efecto de los eventos sobre el valor de la medida del rendimiento del programa finalmente realizado. Se suele combinar con una acción Reactiva ante eventos o efectos de los mismo no previstos.

En **(Vieira, G. E. et al. 2000b)** se realiza una primera diferenciación entre dos estrategias dinámicas, por un lado las representadas por las reglas de despacho, y por otro lado lo que el autor denomina Programación Reactiva de la Producción (que atendiendo a la categorización del apartado 3.3.3 se debe incluir dentro de la estrategia mixta). Ya en **(Vieira, G. E. et al. 2003)** se establecen dos estrategias bien diferenciadas para hacer frente al Control de la Producción en un entorno de Reprogramación Dinámica de la Producción que presenta Incertidumbre en la entrega de trabajos al taller. Las estrategias son la Programación Dinámica de la Producción (que atendiendo a la categorización del apartado 3.3.3 se debe denominar Reactiva) y la Programación de la Producción Predictivo-Reactiva (o Reprogramación). En la última se diferencian a su vez tres políticas de Reprogramación: **Periódica, Dirigida por Eventos e Híbrida**.

Según **(Mehta, S. V. et al. 1999)** existen cuatro aproximaciones a la Programación de la Producción dinámica en un entorno de Incertidumbre: Completamente Reactiva, Predictivo-Reactiva, Robusta y basada en el conocimiento. La orientación completamente Reactiva está dominada por las técnicas basadas en reglas de despacho. En esta categorización se entiende por métodos basados en el conocimiento aquellos que proporcionan un mecanismo de selección de la política de Reprogramación adecuada frente a un conjunto de alternativas. Dentro de la Programación Predictiva de la Producción, los autores distinguen tres tipos de política de Reprogramación; continua, periódica, y dirigida por eventos, así como cuatro tipos de técnicas; right-shift, programación completa, matchup, y multiobjetivo.

Para **(O'Donovan, R., Uzsoy, R., & McKay, K. N. 1999)** existe una diferencia entre la Programación Predictiva y la Predictivo-Reactiva. En la Programación Reactiva se presenta un programa que debe ser capaz de absorber distorsiones de forma anticipada al mismo tiempo que se mantiene el rendimiento del mismo. Los autores clasifican finalmente las estrategias de Programación de la Producción ante Incertidumbres en cuatro grupos: Completamente Reactivas, Predictivas-Reactivas, Robustas (o Predictivas) y basadas en conocimiento. Los

autores destacan el predominio de las reglas de despacho en la aproximación Completamente Reactiva.

Para un entorno dinámico (Shafaei, R. & Brunn, P. 1999a; Shafaei, R. et al. 1999b) diferencia entre estrategias basadas en Teoría de Colas y Reprogramación (el autor lo denomina “rolling time horizon”). La teoría de colas afronta el problema considerando un sistema de colas donde cada máquina actúa como un servidor. Normalmente, la teoría de colas se ha aplicado a problemas con una máquina haciéndose aconsejable una simulación previa para determinar el comportamiento previsible. La Teoría de Colas y de las reglas de despacho han estado muy ligadas. En el caso de la estrategia de Reprogramación los autores no realizan un estudio exhaustivo realizando un breve recorrido por las reglas de despacho usadas con este propósito. (Shafaei, R. et al. 1999b) introduce el concepto de Programación Robusta.

En **(Raheja, A. S. & Subramaniam, V. 2002)** se realiza una clasificación de las diversas aproximaciones a la Programación de la Producción Estática y Dinámica. En la aproximación dinámica distinguen 3 tipos de Programación: Totalmente Reactivo, Reactivo y Predictivo-Robusto. Los autores ponen de manifiesto la supremacía de las reglas de despacho en las soluciones Totalmente Reactivas. El enfoque Reactivo propuesto en esta referencia es equivalente al Predictivo-Reactivo propuesto por otros autores y definido al inicio del apartado. Mientras que el enfoque Predictivo-Robusto se relaciona con soluciones con toma de decisiones fuera de línea en las que el programa establecido antes del inicio de las operaciones incorpora una serie de parámetros que le permiten absorber las distorsiones que se produzcan. Finalmente los autores proponen una serie de técnicas para ser aplicadas en la Programación Reactiva de la Producción como: Heurísticas (right-shift, affected operation rescheduling (AOR)), Sistemas Multiagente, Programación basada en Lógica Difusa, Programación basada en Restricciones, Programación basada en Razonamiento de los casos, Sistemas basados en el Conocimiento y Técnicas de Inteligencia Artificial (Redes Neuronales o Algoritmos Genéticos).

En **(Ouelhadj, D. 2003)** se categorizan las estrategias de Programación de la Producción en un entorno dinámico en: Programación en Línea, Programación Predictivo-Reactiva y Programación Robusta. La autora destaca que en la Programación en Línea es frecuente el uso de reglas de despacho, ya que proporcionan una solución aceptable en un corto espacio de tiempo. Pero destaca la miopía de dichas reglas al no disponer de información global, ya que se aplican de forma local, lo no garantiza el rendimiento del sistema. La Programación Predictivo-Reactiva, a la que la autora reconoce otros nombre como Programación Reactiva o Reprogramación, es interpretada por la misma como aquella que genera un programa inicial con el objeto de optimizar un parámetro y sin tener en cuenta ningún evento, y cuando se produce el evento la Programación Reactiva proporciona un nuevo programa con el objeto de mantener la factibilidad o mejorar el rendimiento. Para Ouelhadj la Programación Robusta

parte de la Programación de la Producción Predictivo-Reactiva pero se caracteriza por buscar la minimización de los efectos de las distorsiones sobre el rendimiento del Programa Productivo realizado.

En el trabajo de **(Aytug, H. et al. 2005)** se clasifican las estrategias para afrontar los problemas de Programación de la Producción es tres grandes grupos: Completamente Reactivos, Robustos y Predictivos-Reactivos. Las aproximaciones Completamente Reactivas se basan en tomas de decisiones en línea de forma local. Los autores destacan como técnica más representativa de la estrategia Completamente Reactiva la aplicación de reglas de despacho, ya sea de forma directa o aplicando una variante que permita seleccionar la regla en cada instante en función de las condiciones del taller. Otra técnica Completamente Reactiva destacada por los autores es el uso de Agentes Inteligentes. Los Programas Productivos Robustos intentan minimizar el impacto de las diferentes eventualidades sobre el programa implementado. En el caso de la Programación Predictivo-Reactiva los autores hacen hincapié en las políticas de Reprogramación, es decir ¿Cuándo reprogramar? Los autores proponen tres opciones de Reprogramación: continua, periódica y dirigida por eventos.

<i>Autores</i>	<i>Estrategia En Línea</i>	<i>Estrategia Mixta</i>
(Vieira, G. E. et al. 2000b)	- En línea	- Reactiva
(Vieira, G. E. et al. 2003)	- Dinámica (Reactiva)	- Predictivo- Reactiva
(Mehta, S. V. et al. 1999) (O'Donovan, R. et al. 1999)	- Completamente Reactiva	- Predictivo-Reactiva - Robusta - Basada en el Conocimiento
(Shafaei, R. et al. 1999a; Shafaei, R. et al. 1999b)	- Teoría de colas	- Reprogramación - Predictivo-Robusto
(Raheja, A. S. et al. 2002)	- Completamente Reactiva	- Reactiva
(Ouelhadj, D. 2003)	- En línea	- Predictivo-Reactiva - Robusta
(Aytug, H. et al. 2005)	- Completamente Reactiva	- Predictivo-Reactiva

Tabla 3-1. Resumen de las principales estrategias de Programación de la Producción en un entorno dinámico (elaboración propia).

En la Tabla 3-1 se muestra un resumen de las diferentes denominaciones utilizadas por los autores consultados, clasificadas en dos grandes grupos. **Según la revisión literaria realizada se puede decir que existen dos estrategias principales para enfocar la Programación de la Producción en un entorno dinámico: La Programación de la Producción en Línea** (en discordancia con el apartado 3.3.3, algunos autores la denominan dinámica, pero en esta tesis usaremos preferentemente el término en línea) dominada por las reglas de despacho, **y la Programación de la Producción Mixta** basada en un plan previo que se va modificando según se hacen patentes las distorsiones. En la Programación de la Producción Mixta existe una base común denominada habitualmente estrategia Predictivo-Reactiva, a partir de la cual se distinguen dos mejoras importantes: la Programación de la Producción Robusta y la Programación de la Producción basada en el Conocimiento.

La Programación de la Producción en Línea se puede entender como aquella que no crea Programas de Producción de forma anticipada, de manera que las decisiones se suelen tomar de forma local y en el momento que es necesario. Normalmente se utilizan reglas de despacho, que son aplicadas cuando una máquina queda libre, para seleccionar el siguiente trabajo. Con estas reglas de despacho o con heurísticas se priorizan los trabajos en cola de espera para establecer el orden de entrada. La ventaja de las reglas de despacho es su fácil aplicación y su rápido cálculo, pero normalmente adolecen de una visión global del problema y es difícil de establecer cual será su rendimiento en un entorno cambiante. **La Programación de la Producción en Línea no es adecuada para la Empresa Cerámica** dado que la preparación de las materias primas (arcillas, esmaltes, colores, etc.) por parte de los proveedores implica la necesaria disponibilidad de un Programa Previo.

Una vez se han diferenciado las estrategias de Programación de la Producción en entornos dinámicos, que al mismo tiempo están sujetos normalmente a Incertidumbre, y se han destacado las más extendidas, se presentan un conjunto de referencias complementarias. Los autores correspondientes no ofrecen diversas alternativas estratégicas, pero aportan aspectos interesantes que ayudan a entender la situación actual de cada una de ellas.

Dentro de las estrategias en Línea (Completamente Reactiva) destacamos: (Bhaskaran, K. & Pinedo, M. 1991;Chrétienne, P., Coffman, J. E., Lensrta, J. K., & Liu, Z. 1995;Dempster, M. A., Lensrta, J. K., & Rinnooy, K. 1982) presentan una discusión sobre reglas de despacho aplicadas de forma local. (Wu, D. S. & Wysk, R. A. 1989) estudiaron el problema de las reglas de despacho en un sistema de fabricación flexible. Los autores dividieron el horizonte temporal en intervalos y al inicio de cada uno de ellos se simulaba el comportamiento de un conjunto de reglas y seleccionaba la mejor para aplicarla en dicho periodo. (Jeong, K. C. & Kim, Y. D. 1998;Kim, M. H. et al. 1994) realizaron aplicaciones parecidas a las de (Wu, D. S. et al. 1989) introduciendo sensibles mejoras. (Aytug, H., Bhattacharyya, S., Koehler, G. J., & Snowdon, J. L. 1994) realiza una profunda revisión literaria sobre la aplicación de la técnica de inteligencia

artificial denominada máquina de aprendizaje (machina learning) para la selección de la regla de despacho adecuada en un problema de Programación de la Producción.(Piramuthu, S., Park, S. C., Raman, N., & Shaw, M. J. 1991) se basó en la técnica de la máquina de aprendizaje para seleccionar la regla de despacho más adecuada según las condiciones del taller en base a diferentes eventos establecidos en un modelo de simulación. (Chen, C. & Yih, Y. 1996) usaron una red neuronal para predecir la regla de despacho en función del estado del sistema. (Ovacik, I. M. & Uzsoy, R. 1994) propusieron diversos horizontes de Reprogramación para un taller de una sola máquina con Tiempo de Cambio de Partida Dependientes de la Secuencia. (Ovacik, I. M. & Uzsoy, R. 1995) desarrollaron un algoritmo similar para un problema de máquinas paralelas.

Dentro de las estrategias Mixtas se encuentran aportaciones interesantes en el enfoque Predictivo-Reactivo puro o enfoques Robustos y basados en el Conocimiento.

El enfoque Predictivo-Reactivo presenta aportaciones como: (Raman, N., Rachamadugu, R. V., & Talbot, F. B. 1989) que usan una Reprogramación continua para la resolución del problema presentado. (Adam, N. R. & Surkis, J. 1980;Church, L. K. et al. 1992) investigan la frecuencia de Reprogramación en un taller de trabajo dinámico con fallos en máquinas y Tiempos de Proceso variables. (Muhlemann, A. P., Lockett, A. G., & Farn, C. K. 1982) realiza un estudio de simulación para investigar el rendimiento de un conjunto de reglas heurísticas basadas en un algoritmo de generación de Programas Productivos sin retraso. Aunque no se pudo concluir con la supremacía de ninguna reglas, si se detecto que la frecuencia de la Reprogramación influía en el rendimiento del Programa. (Shafaei, R. et al. 1999a) realizan un estudio de simulación para investigar sobre el rendimiento de una serie de reglas de despacho en un entorno de Reprogramación en un taller de trabajo dinámico. Los resultados muestran como la reglas SPT-C/R es la más apropiada para minimizar el coste total y como se puede mostrar una relación entre el rendimiento y el intervalo de Reprogramación. (Yamamoto, M. et al. 1985) realizan un estudio mediante Reprogramación completa en un taller con fallos aleatorios en las máquinas. Los resultados mostrados por los autores concluyen con resultados mejores de dicho método frente a las reglas de despacho o modificaciones parciales del programa. (Church, L. K. et al. 1992) estudiaron el problema de la Reprogramación con una única máquina con llegada dinámica de trabajos. Según su trabajo la política de Reprogramación a intervalos fijos es preferible frente a la dirigida por evento excepto cuando se producen llegadas de trabajos urgentes al inicio del programa.

El enfoque de Programación Robusta presenta aportaciones como:(Dooley, K. J. & Mahmoodi, F. 1992;Graves, C. S. 1981;Yellig, E. J. & Mackulak, G. T. 1997) definieron la robustez como un objetivo para evaluar el rendimiento real de un método de Programación de la Producción en un entorno estocástico. (Leon, V. J., Wu, S. D., & Storer, R. H. 1990) considera la Programación Robusta de un taller de trabajo con fallos en las máquinas, variación de los Tiempos de Proceso

y minimización de C_{max} y la desviación del C_{max} como medidas de rendimiento. (Wu, S. D. et al. 1999) propusieron una descomposición para alcanzar la robustez. Ellos consideraron el problema con variación en los Tiempos de Proceso y con la minimización del retraso ponderado total (weighted tardiness) como función objetivo. (Daniels, R. L. & Carrillo, J. E. 1997; Daniels, R. L. et al. 1995; Kouvelis, P. & Yu, G. 1997) aportan una discusión interesante sobre la Programación Robusta. Definen la Incertidumbre en términos de escenarios y redefinen el problema de Programación como la búsqueda de una secuencia que minimice la máxima desviación entre el rendimiento de la secuencia y las secuencias óptimas asociadas al conjunto de escenarios. (Mehta, S. V. et al. 1998) desarrollaron un algoritmo que minimizaba el retraso (lateness) y la diferencia entre el C_{max} previsto y realizado. Este estudio muestra como los Programas Productivos que son Robustos a las distorsiones estocásticas pueden ser generados sin sacrificar demasiado rendimiento del mismo. (Mehta, S. V. et al. 1999) propone una Programación Predictiva de la Producción, intentando minimizar L_{max} sobre una única máquina con llegada dinámica de trabajos y fallos la máquina. Para ello los autores insertan tiempos muertos mientras se recupera el fallo. La propuesta Predictiva supone mejoras importantes de la robustez a costa de una pequeña disminución de L_{max} . (Mehta, S. V. et al. 1998) aplicaron una estrategia Predictiva sobre un taller de trabajo con máquinas no fiables y obtuvieron resultados similares al anterior caso. (Shafaei, R. et al. 1999b) investiga sobre la robustez de un conjunto de reglas de despacho en un entorno dinámico y estocástico usando Reprogramación cíclica. Las reglas evalúan sobre una función de coste. Se concluye estableciendo una relación entre la influencia de la carga y el balance del taller y la Robustez de un Programa. (Bollapragada, R. & Sadeh, N. M. 1996) desarrollan un Programa Predictivo de la Producción para minimizar el retraso-adelanto total de un taller de trabajo teniendo en cuenta los posibles fallos de máquinas y la variación de los Tiempos de Proceso. (McKay, K., Morton, T. E., Ramnath, P., & Wang, J. 2000) introdujeron la Reprogramación dinámica que sub-optimiza en un determinado periodo de tiempo para permitir al sistema estabilizarse y progresivamente optimizarse. (Singer, M. 2000) aplica la Reprogramación dinámica en un taller de trabajo con Incertidumbre en los Tiempos de Proceso para minimizar el retraso total.

El enfoque de Programación basada en el Conocimiento presenta aportaciones como: (Fox, M. S. & Smith, S. F. 1984) realizaron una de las primeras y más exitosas aportaciones de la IA aplicada al problema de la Programación de la Producción. (Sadeh, N. M., Otsuka, S., & Schnellbach, R. 1993) muestra la propuesta de la Universidad de Carnegie Mellon sobre propagación de restricciones aplicada a la Reprogramación y usada en la propuesta de Micro-Boos como sistema de control y secuenciación. (Smith, S. F. 1992) discute sobre las posibilidades de Reprogramación en OPIS (Opportunistic Intelligent Scheduler), un sistema basado en el Conocimiento desarrollado por la Universidad Carnegie Mellon. En (Smith, S. F. 1987; Smith, S. F., Keng, N., & Kempf, K. 1990) se puede encontrar sobre OPIS. (McKay, K. N. et al. 1989) presenta el sistema de Programación de carga finita en un taller de trabajo, WATPASS. En este caso la Reprogramación se realiza por actuación directa del programador. (Szelke, E. & Kerr, R. M. 1994) proporciona una extensa revisión de la Programación Reactiva

basada en el conocimiento. (Dutta, A. 1990) propone un mecanismo de control monitorizando el entorno y tomando decisiones correctivas en función del evento. (Baptiste, P. & Favrel, J. 1993) compara esquemas para representar Programas Productivos alternativos mientras se desarrolla el Programa Predictivo.

Los apartados siguientes prestarán mayor atención a los diversos aspectos relacionados con las estrategias Mixtas. Al considerar que **el enfoque Predictivo-Reactivo resulta una estrategia bastante acertada (a priori) para abordar el problema que se propone resolver en esta tesis** en los siguientes apartados se presentarán las diferentes políticas de Reprogramación en relación con dicha estrategia Mixta.

3.4.4 Medidas de Eficacia sobre el Rendimiento, la Robustez y la Estabilidad de un Programa en entornos Dinámicos

3.4.4.1 Medidas de Eficacia sobre el Rendimiento

Para evaluar la calidad de un Programa de Producción se deberá utilizar una medida o parámetro de eficacia. Estas medidas pueden estar basadas en términos temporales, económicos u otros.

Existen una alta variedad de medidas de eficacia que son definidas en función de los objetivos a optimizar. Por lo que se refiere a las medidas basadas en el tiempo, la teoría clásica de secuenciación, se ha venido hablando de parámetros o medidas de eficacia regulares y no regulares.

Una medida de eficacia es regular, si se puede poner en función de los instantes de finalización de los trabajos en el taller y aumenta sólo si por lo menos uno de los instantes de finalización de las piezas en el taller aumenta. Para cada trabajo "i" procesado en el taller se pueden calcular los siguientes parámetros:

- Fecha de finalización, C_i .
- **Tiempo de Flujo**, $F_i = C_i - r_i$, tiempo de permanencia de la pieza en el taller.
- **Tiempo de Espera**, $W_i = C_i - r_i - \sum t_{ij}$
- **Holgura**, $L_i = C_i - d_i$

- Tardanza o Retraso, $T_i = \text{Max}$
- Adelanto, $E_i = \text{Max} \{0, -L_i\}$
- Tiempo de desocupación de la máquina j , $I_j = C_{\text{max}} - \sum t_{ij}$
- $i=1, \dots, n$. $j=1, \dots, m$.

A partir de las anteriores definiciones se pueden plantear algunas medidas de eficacia regulares:

- **Máxima fecha de finalización (o makespan), C_{max} .** La mejora de esta medida de eficacia implica una utilización eficiente de los recursos (Blazewicz, J., Ecker, K. H., Pesch, E., Schmidt, G., & Weglarz, J. 1996; Mehta, S. V. et al. 1998; Sabuncuoglu, I. et al. 1999; Wu, S. D., Storer, R. H., & Chang, P. C. 1993; Yamamoto, M. et al. 1985).
- Maximo tiempo de flujo, varianza del tiempo de flujo (Jayamohan, M. S. & Rajendran, C. 2000).
- Fecha de finalización media C_{med} , tiempo de flujo medio, o tiempo de espera medio W_{med} . Implica una respuesta rápida a la demanda y la minimización del tiempo en proceso de los trabajos.
- Cumplimiento de la fecha tope de entrega: Tardanza media T_{med} (Henning, G. P. et al. 1995; Jain, A. K. et al. 1997; Kim, M. H. et al. 1994), tardanza máxima T_{max} (Church, L. K. et al. 1992) , raíz cuadrada del retraso medio (Gargeya, V. B. & Deane, R. H. 1999), varianza del número retraso (Jayamohan, M. S. et al. 2000) o el número de trabajos tardíos NT.

Algunos de los criterios que se utilizan son equivalentes. En general la mayoría de las medidas son susceptibles de ser utilizadas en función de su valor absoluto, su media, varianza, valor máximo, valor mínimo, porcentaje, etc. según el objetivo perseguido.

Algunas medidas no regulares son:

- Utilización media de las máquinas (Tabe, T. & Salvendy, G. 1988).
- Minimizar la desocupación de todas las máquinas.

- Porcentaje de los cambios de herramientas realizados como medio de reflejar las distorsiones operativas asociadas a los cambios frecuentes de recursos (Gargeya, V. B. et al. 1999).

En (Pinedo, M. et al. 1999) se establecieron los diferentes objetivos como una composición de seis objetivos básicos que se combinan, que son los relacionados con: makespan, fechas de entrega. Como se puede observar existen dos objetivos fundamentales basados en aspectos temporales y cuatro en aspecto económicos. Por lo que se refiere a los primeros, los autores resaltan, que:

- El makespan es especialmente interesante maximizarlo en talleres con máquinas paralelas.
- Los objetivos relacionados con las Fechas de Entrega, como la medida del: retraso de los trabajos, retraso máximo, número de trabajos retrasados, retrasos totales, media de retrasos, media de retrasos ponderada. Destacando que no sólo el retraso debería ser medido en un sistema productivo, sino también el adelanto.

Por lo que se refiere a las medidas de evaluación basadas en términos económicos (Abdel-Wahab, H. M. & Kameda, T. 1978; Aggarwal, S. C. & McCart, B. A. 1974; Hoffmann, T. R. & Scudder, G. D. 1983; Jones, C. H. 1973; Wilson, H. G. & Mardis, B. J. 1983) intentan cubrir un ámbito al que no pueden llegar las medidas basadas en aspectos temporales. Por ejemplo, el beneficio de un trabajo no puede ser adecuadamente medido con parámetros temporales.

- Valor actual de la red, (Lawrence, S. R. 1991; Rohlder, T. R. & Scudder, G. D. 1993; Scudder, G. D. & Smith, D. 1989) toma como base de los cálculos el tiempo para tener en cuenta los ingresos, coste de los materiales, coste de espera, coste de las entregas fuera de plazo, etc.
- En (Pinedo, M. et al. 1999) establecieron a los costes de cambio de partida, los costes del trabajo en curso, los costes de inventario de producto final, y los costes de personal como sus cuatro objetivos básicos basados en la economía
- Otras medidas basadas en costes son: Beneficio medio de los trabajos, minimización del coste total, coste de entregas fuera de plazo, coste de reprogramar.

Por lo que se refiere a medidas basadas en otros costes se pueden destacar:

- Las medidas basadas en el coste computacional (Church, L. K. et al. 1992; Sabuncuoglu, I. et al. 1999).
- Medidas basadas en el coste de la replanificación de herramientas o reubicación de material (Olumolade, M. O. & Norrie, D. H. 1996).
- Medidas basadas en el transporte del material por el taller (Olumolade, M. O. et al. 1996).

Algunos autores como (Min, H. S. & Yih, Y. 2003) establecen la necesidad de disponer no sólo medidas de rendimiento del sistema, sino **medidas complementarias del estado del mismo para evaluar la calidad de un programador en tiempo real y multiobjetivo**. Algunos de los parámetros propuestos para identificar el estado del sistema son: Trabajo en curso total, carga total de las máquinas críticas, número medio de operaciones pendientes en cada lote, media del tiempo de proceso restante, media del tiempo sobrante.

En la Programación en línea los valores relativos al periodo de Reprogramación y a la media del Tiempo Total de Proceso requeridos para un trabajo afectarán a la medida usada en el Programa Predictivo-Reactivo de la Producción. Cuando el periodo de Reprogramación es relativamente largo, los trabajos pueden ser iniciados y finalizados entre las Reprogramaciones. Los objetivos de la Programación de la Producción se centran normalmente en acabar los trabajos disponibles dentro del periodo. Cuando el periodo de Reprogramación es relativamente pequeño, el sistema tendrá en cada punto de Reprogramación, algunos trabajos que están disponibles y esperando para empezar y otros que se encuentren en proceso y cuyo inicio se habrá realizado en periodos anteriores al actual.

3.4.4.2 Medidas de Eficacia sobre la Robustez

Muchas de las estrategias de la Programación Predictivo-Reactiva de la Producción están basadas en simples ajustes del Programa Productivo, que pueden llevar al taller a un rendimiento bajo. Por lo tanto, es deseable generar Programas Predictivos-Reactivos de la Producción robustos (ver apartado 3.4.3). Ejemplos significativos sobre la investigación en según este enfoque se pueden encontrar en (Wu, D. S., Storer, R. H., & Chang, P. C. 1991; Wu, S. D. et al. 1993), (Leon, V. J., Wu, S. D., & Storer, R. H. 1994), (Daniels, R. L. et al. 1995), (Yellig, E. J. et al. 1997), (Abumaizar, R. J. et al. 1997), (Shafaei, R. et al. 2000; Shafaei, R. et al. 1999b), (O'Donovan, R. et al. 1999), (Mehta, S. V. et al. 1999) y recientemente (Jensen, M. T. 2001), (Cowling, P. et al. 2002; Pinedo, M. 2002).

El trabajo de **(Leon, V. J. et al. 1994)** consistió en el desarrollo de medidas de robustez y Programas Productivos robustos para tratar con casos de fallos en las máquinas y Tiempos de Proceso variables. La robustez se definió como la minimización del rendimiento en términos de valor esperado de makespan y retraso. El retraso esperado es la desviación entre el makespan antes del evento y el makespan real después de aplicar la técnica right-shift. El resultado experimental mostró como los Programas Productivos robustos superan significativamente a los Programas basados en makespan exclusivamente.

En **(Shafaei, R. et al. 2000;Shafaei, R. et al. 1999b)** se experimentó con la robustez de algunas reglas de despacho en un entorno dinámico y estocástico. Los resultados indicaron que junto al efecto del método de Programación de la Producción y los intervalos de Reprogramación, la carga del taller y el balance de la misma tienen un impacto significativo sobre la robustez del Programa Productivo.

Los autores de **(Wu, D. S. et al. 1991;Wu, S. D. et al. 1993)** definieron un doble criterio de robustez para una máquina en un problema de Reprogramación con fallos en máquinas. La propuesta incluía un criterio de eficacia (makespan) y un criterio de estabilidad (impacto del cambio de programa). Para medir el impacto del cambio los autores investigaron dos medidas: La desviación de la fecha de inicio de los trabajos y la desviación de la secuencia respecto de la original. Los resultados mostraron como la efectividad de la medida de la robustez puede significar un incremento significativo de la estabilidad manteniendo o perdiendo ligeramente el valor de makespan.

De forma similar en **(Abumaizar, R. J. et al. 1997)** usaron dos medidas, una de eficacia (makespan) y otra de estabilidad (desviación de fechas de inicio de los trabajos y desviación de la secuencia) para definir un Programa Productivo Robusto. El objetivo de la propuesta fue maximizar la medida de eficacia y al mismo tiempo minimizar el impacto causado en el sistema por los cambios.

En **(Jensen, M. T. 2001)** se experimentó con diferentes medidas de robustez para mejorar la robustez de talleres de trabajo con fallos en máquinas y con medidas de eficacia sobre el retraso y el tiempo de flujo total.

Los autores de **(Daniels, R. L. et al. 1995)** definieron medidas de robustez para entornos de máquinas únicas para tratar con la Incertidumbre en el Tiempo de Proceso donde el objetivo del programador es minimizar el tiempo de flujo de los trabajos. La robustez del Programa Productivo se definió como la minimización del tiempo de flujo y de la desviación absoluta del tiempo de flujo real frente al óptimo. La medida de robustez se utilizó para seleccionar la mejor heurística.

En el trabajo de **(Cowling, P. et al. 2002)** se desarrolló un marco de trabajo general para definir Programas Productivos robustos. Los autores definieron medidas generales de utilidad y estabilidad para facilitar las decisiones sobre las estrategias usadas para reaccionar ante eventos en tiempo real. Las medidas de utilidad mejoran el objetivo original mediante la revisión del Programa Productivo. La medida de estabilidad evalúa la desviación del Programa original causado por la revisión. Los autores han experimentado sobre un número de medidas de utilidad y de estabilidad para modelos de máquinas únicas con el objetivo de minimizar el C_{max} .

En la propuesta de **(Pinedo, M. 2002)** se definió una medida de robustez para el caso de C_{max} con una máquina única. Esta medida es $f = \frac{Z'(\delta) - Z}{\delta}$ donde $Z'(\delta)$ es el valor de la nueva función objetivo, Z es el valor de la función objetivo inicial, y δ es el tiempo de retraso sufrido por un trabajo debido a un evento que ha interferido en el Programa Productivo.

En **(Mehta, S. V. et al. 1999)** propusieron un Programa Predictivo de la Producción para una máquina única en la que se producían fallos y se conocía el modelo que seguían las averías así como el modelo de la llegada de los trabajos. El modelo considerado minimizaba la holgura. La medida de robustez se basaba en la diferencia entre el C_{max} previsto en el Programa Predictivo y el C_{max} obtenido en la realidad.

En **(O'Donovan, R. et al. 1999)** utilizaron técnicas predictivas como **(Mehta, S. V. et al. 1999)** para crear un Programa robusto capaz de soportar los diversos eventos manteniendo un valor de medida de rendimiento dentro de lo previsto. En concreto los autores extendieron los trabajos de **(Mehta, S. V. et al. 1999)** para medidas de retraso.

3.4.4.3 Medidas de Eficacia sobre la Estabilidad

La *estabilidad de un Programa de Producción no es una medida utilizada en propuestas estáticas*, es decir en Programas Productivos que no necesitan ser actualizados. Como se ha visto en el apartado 3.4.4.2 las medidas de robustez incorporan, en la mayoría de los casos, una componente que evalúa la estabilidad.

Los cambios constantes en un *Programa Productivo son la característica opuesta a la estabilidad, denominada Nerviosismo*. Así un Programa Productivo que realiza pocas modificaciones incrementa la estabilidad y decrementa el nerviosismo. En una Programación de la Producción en Línea es importante considerar esta medida, ya que influye no solo en la calidad de la medida de rendimiento (apartado 3.4.4.2), sino en otros aspectos como

aprovisionamiento, número de reconfiguraciones de la planta, actitud de los trabajadores (estrés), etc.

Los autores de (Wu, S. D. et al. 1993) consideraron que el impacto de un cambio en un Programa Productivo es una medida no regular que se define en dos sentidos:

- La desviación de los instantes de inicio entre el programa original y el nuevo.
- La medida de la diferencia entre secuencias entre los dos programas.

En (Abumaizar, R. J. et al. 1997) propusieron ideas similares estableciendo que las medidas de estabilidad tratan con la desviación con el Programa inicial.

Otros autores como (Church, L. K. et al. 1992; Cowling, P. et al. 2002; Wu, S. D. et al. 1993) proporcionan una aproximación general a la medición de la estabilidad de un Programa de Producción y presentan una medida de estabilidad específica que calcula el cambio medio absoluto del instante de inicio y final de dos Programas Productivos.

3.4.5 Políticas de Reprogramación en Estrategias Predictivo-Reactivas: Cuando Actuar ante los Eventos

Este apartado trata el problema relacionado con la forma de abordar la revisión de las decisiones tomadas por el programador y plasmadas en un Programa Predictivo de la Producción. En definitiva *se trata de saber cuando es conveniente revisar las decisiones anteriores, y en caso de modificarlas, cual es el mejor procedimiento en función de las circunstancias.*

Según (Vieira, G. E. et al. 2003) la Reprogramación de la Producción es el proceso de actualizar un Programa Productivo existente en respuesta a un evento.

En (Church, L. K. et al. 1992) se proporciona una taxonomía de las diferentes aproximaciones que existían al problema de la Reprogramación de la Producción. Los autores consideran la **Programación continua, la periódica y la dirigida por eventos**. (Sabuncuoglu, I. et al. 2000; Vieira, G. E. et al. 2003) proponen tres políticas de Reprogramación de la Producción alternativas: **Periódica, Dirigida por Eventos o Híbrida**.

La **política de Reprogramación de la Producción Periódica** consiste en la revisión del Programa en curso a intervalos regulares, de forma que antes de revisar el Programa se suele realizar un proceso de actualización de la información que permita disponer el estado real del taller y de las circunstancias del mismo. Los Programas Productivos con Reprogramaciones periódicas pueden ser vistos como un conjunto sucesivo de Programas estáticos encadenados. Una de las ventajas de esta política es que las técnicas clásicas de Programación de la Producción en entornos estáticos pueden ser aplicadas en cada Reprogramación. Así mismo los Programas Productivos alcanzan cierta estabilidad y su grado de nerviosismo es bajo. El inconveniente más importante de esta política es que los Programas Productivos pueden volverse inestables frente a eventos de gran impacto ya que su modificación no se realizaría hasta el inicio del nuevo periodo.

En la **política de Reprogramación de la Producción Dirigida por Eventos** el programador actúa cuando el rendimiento del Programa en ejecución se ve disminuido por causa de un evento inesperado. En (Church, L. K. et al. 1992) se denomina Programación Continua de la Producción a aquella en la que se actualiza el Programa cuando se produce cualquier evento sin determinar si este afecta al rendimiento del sistema.

En la **política de Reprogramación de la Producción Híbrida** los Programas Productivos se revisan en intervalos regulares, y cuando algún evento inesperado reduce el rendimiento del Programa en curso. En las políticas híbridas es fundamental realizar una correcta categorización de los tipos de eventos para evitar que el sistema de revisión se active muy frecuentemente. Se debe establecer un equilibrio entre el posible beneficio de corregir el impacto del evento y el coste de Reprogramación, teniendo en cuenta que existe una revisión cíclica.

Las Reprogramaciones de la Producción periódicas e híbridas han recibido especial atención bajo el concepto "Rolling Time Horizon" o periodificación del espacio temporal. Algunas referencias significativas como las de (Adam, N. R. et al. 1980;Chen, J. et al. 2003;Muhlemann, A. P. et al. 1982;Ovacik, I. M. et al. 1994;Qi, J. G., Burns, G. R., & Harrison, D. K. 2000;Shafaei, R. et al. 1999b;Shafaei, R. et al. 1999a;Singer, M. 2000;Sun, D. & Lin, L. 1994) aplican el Rolling Time Horizon.

En (Adam, N. R. et al. 1980) se realiza un estudio basado en un taller dinámico en el cual se aplican las reglas de despacho que necesitan el cálculo del estado actual de las prioridades de los trabajos remanentes en el taller. Los autores proponen un nuevo procedimiento mediante el cual la actualización de la información y la Programación de la Producción se realiza a intervalos.

Una de las primeras aplicaciones de este concepto a la Programación de la Producción en entornos dinámicos fue realizada por **(Muhlemann, A. P. et al. 1982)**. Los autores realizan un estudio de simulación para investigar el rendimiento de un conjunto de reglas heurísticas basadas en un algoritmo de generación de Programas Productivos sin retraso. Aunque no se pudo concluir con la supremacía de ninguna reglas, si se detecto que la frecuencia de la Reprogramación influía en el rendimiento del Programa. Este trabajo profundiza en el análisis de la frecuencia de Reprogramación en un taller de trabajo dinámico con fallos en máquinas y tiempos de proceso variables.

En **(Ovacik, I. M. et al. 1994)** se propone diversos horizontes de Reprogramación para un sistema de máquina única con Tiempos de Cambio de Partida Dependientes de la Secuencia.

Los autores de **(Sun, D. et al. 1994)** plantean el problema de Programación de la Producción en un taller general. Los autores plantean un algoritmo estático de vuelta atrás. El algoritmo debe hacer frente a la disminución del rendimiento provocada por los fallos en máquinas. Para ello se utiliza una política de Reprogramación dirigida por eventos. Se plantean varias medidas de rendimiento: tiempo de flujo medio, makespan, penalización por retraso de los trabajos. El Programa Productivo se modifica completamente en cada revisión si es necesario.

En **(Shafaei, R. et al. 1999a)** se realiza un estudio de simulación para investigar sobre el rendimiento de una serie de reglas de despacho en un entorno de Reprogramación en un taller de trabajo dinámico. Los resultados muestran como la reglas SPT-C/R es la más apropiada para minimizar el coste total y como se puede mostrar una relación entre el rendimiento y el intervalo de Reprogramación. **(Shafaei, R. et al. 1999b)** investiga sobre la robustez de un conjunto de reglas de despacho en un entorno dinámico y estocástico usando Reprogramación cíclica Las reglas evalúan sobre una función de coste. Se concluye estableciendo una relación entre la influencia de la carga y el balance del taller y la robustez de un Programa Productivo.

La propuesta de **(Singer, M. 2000)** consistió en usar un procedimiento heurístico de descomposición basado en un procedimiento centrado en el cuello de botella desarrollado por **(Pinedo, M. & Singer, M. 1999)**.

En el trabajo de **(Qi, J. G. et al. 2000)** se propone un procedimiento heurístico de descomposición basado en un algoritmo genético multi-población.

En **(Chen, J. et al. 2003)** se realizan una propuesta de algoritmo dividido en dos etapas: asignación de un trabajo a una máquina y asignación de una herramienta a una operación, en un taller flexible. El algoritmo debe hacer frente a los diversos eventos taller como los fallos en máquinas. Para ello se utiliza una política de Reprogramación de la Producción dirigida por

eventos. Se plantean tres medidas de rendimiento: tiempo de flujo medio, retrasos de los trabajos y porcentaje de trabajos retrasados. El Programa Productivo se modifica completamente en cada revisión si es necesario.

Por lo que se refiere a la Reprogramación de la Producción por eventos la mayoría de los autores estudiados están de acuerdo en que ésta suele ser más eficiente que la Reprogramación periódica. Algunas referencias interesantes son (Church, L. K. et al. 1992; Cowling, P. et al. 2002; Rossi, A. et al. 2000; Vieira, G. E., Herrmann, J. W., & Lin, E. 2000a; Vieira, G. E. et al. 2000b; Yamamoto, M. et al. 1985).

En **(Yamamoto, M. et al. 1985)** se realiza un estudio mediante Reprogramación completa en un taller con fallos aleatorios en las máquinas. Los resultados mostrados por los autores concluyen con resultados mejores de dicho método frente a las reglas de despacho o modificaciones parciales del Programa Productivo.

Los autores de **(Church, L. K. et al. 1992)** estudiaron el problema de la Reprogramación con una única máquina con llegada dinámica de trabajos. Según su trabajo la política de Reprogramación a intervalos fijos es preferible frente a la dirigida por evento excepto cuando se producen llegadas de trabajos urgentes al inicio del Programa Productivo.

En algunas referencias como las de **(Cowling, P. et al. 2002; Rossi, A. et al. 2000; Vieira, G. E. et al. 2000b)** se presenta un estudio para la búsqueda de un modelo que pueda predecir el rendimiento de un sistema de una máquina con Reprogramación dirigida por eventos y periódica en un entorno donde diferentes tipos de trabajos llegan dinámicamente existiendo un tiempo de cambio de partida entre trabajos.

En el trabajo de **(Vieira, G. E. et al. 2000a)** se extiende el estudio anterior a un sistema de máquinas paralelas. Este trabajo demuestra que la frecuencia de Reprogramación puede afectar de forma determinante al rendimiento del sistema, dado que influye sobre el número de Cambios de Partida del sistema.

Como conclusión a este apartado se debe remarcar toda política de Reprogramación está condicionada por los eventos que se producen en el sistema, ya que son éstos los que determinan el grado de degradación en la ejecución del Programa Original a través de diferentes impactos. La política elegida debe, en todo caso, intentar evitar la degradación del rendimiento del sistema. En la medida que la política seleccionada alcance este objetivo será o no adecuada para el problema en el cual se aplica.

3.4.6 Procedimientos de Resolución para Programaciones Predictivo-Reactivas

Hasta el momento se ha comentado que un Programador que desarrolle sus funciones en un entorno de Incertidumbre debería establecer en primer lugar su estrategia, y según se ha definido esta podía ser básicamente Reactiva o Predictivo-Reactiva. Si este opta por una estrategia Predictivo-Reactiva debería establecer, en segundo lugar, su política, es decir bajo que criterio reaccionara y considerará la modificación del programa en curso. Y en cualquier caso, el Programador debe de aplicar una técnica para generar o modificar el Programa Productivo. **En el caso de una estrategia Predictivo-Reactiva existen claramente dos fases, y por lo tanto dos técnicas, diferenciadas. Por un lado se necesita un procedimiento para generar Programas nuevos, y por otro lado se necesita un procedimiento para modificar los Programas en curso si es necesario.** En este apartado se van a tratar estos últimos. Existen diversas técnicas para afrontar la revisión de un Programa Productivo en curso, es decir su modificación.

Los autores de **(Raheja, A. S. et al. 2002)** realizan un análisis para un Taller General o Job Shop del conjunto de métodos que se han utilizado hasta la fecha para afrontar la revisión y reparación de un programa en curso afectado por algún tipo de incidencia que ha impedido ejecutar el Programa tal y como estaba previsto. Los autores diferencia claramente entre dos tipos de métodos: **procedimientos Reparadores y procedimientos de Reprogramación Completa.**

Los **procedimientos Reparadores** se basan en la realización de alguna modificación local del Programa Productivo, de forma que el programa inicial se mantiene sin cambios en su mayoría. Este tipo de solución proporciona en la mayoría de los casos estabilidad al sistema, evitando lo que se denomina nerviosismo del Programa Productivo. Así mismo, las Reparaciones permiten reacciones inmediatas ya que desde el punto de vista computacional no consumen demasiados recursos.

Los **procedimientos de Reprogramación Completa** generan un nuevo Programa Productivo en cada ocasión. La Reprogramación puede llevar a buenas prestaciones en los parámetros de rendimiento, pero producen por el contrario grandes perturbaciones en el sistema.

En el trabajo de **(Raheja, A. S. et al. 2002)** se proporciona una interesante tabla en la cual se clasifican los procedimientos Reparadores en base al método utilizado y sobre cada método comentan sus ventajas, inconvenientes, medidas de eficacia relacionadas o favorecidas por su uso, y aplicaciones habituales. Esta aportación puede ser aprovechada en diversos talleres. A continuación se presenta un resumen (ver Tabla 3-2).

<i>Método</i>	<i>Ventajas</i>	<i>Inconvenientes</i>	<i>Medidas de Eficacia</i>	<i>Aplicación</i>
Heurísticas basadas en RSR y AOR.	Simplicidad en la implementación.	El tipo de eventos que se pueden tratar son reducidos. La calidad inicial del Programa no se recupera.	Desviación sobre el Makespan inicial	Talleres estables con Interrupciones esporádicas y cortas.
Sistemas Multiagente	Completa automatización del sistema de reparación. Trabajo paralelo.	Difícil coordinación entre agentes. La integración humanos-agentes es difícil de mejorar.	Calidad del Programa obtenido. Tiempo de reacción.	Talleres dinámicos con Incertidumbres aleatorias.
Basados en Inteligencia artificial (Basados en conocimiento o razonamiento de casos)	Bien ajustados a dominios concretos. Aprendizaje continuo. Amplio campo de eventos modelados.	Consumo excesivo de tiempo en búsquedas en bases de datos. Esencial contar con experiencia previa.	Desviación del Programa inicial.	Talleres con amplia experiencia registrada.
Programación basada en restricciones	Facilita la intervención humana. Posibilita la incorporación de franja temporales	Se necesita un ajuste muy fino para trabajar en tiempo real. Necesita contar con un Sistema	Tiempo de ejecución del algoritmo. Calidad del Programa.	Talleres dinámicos con múltiples interrupciones.

	de reacción	Multiagente.		
Lógica Difusa	Revisión completa de las variables violadas. Respuesta rápida.	Necesita incorporar conocimiento en el algoritmo. No puede aprender y mejorar.	Tiempo de ejecución del algoritmo. Calidad del Programa.	Talleres con variabilidad en los tiempos de procesos y gran número de restricciones asociadas al mismo.
Redes Neuronales	Respuesta muy rápida. Fiabilidad en las predicciones.	Necesita casos de aprendizaje previos muy buenos. Necesita una base de conocimiento.	Tiempo de ejecución del algoritmo. Calidad del Programa.	Talleres con alimentación continua y repetición del mismo tipo de interrupciones.

Tabla 3-2. Clasificación de los procedimientos reparadores en un Taller General (Raheja, A. S. et al. 2002)

En la clasificación anterior se han incluido algunas **Procedimientos heurísticos, RSR y AOR**, a continuación se comentan brevemente, incluyendo otra de las heurísticas más utilizadas en este ámbito:

- **La técnica Right-Shift Repair (RSR)** (Brandimarte, P., Rigodanza, M., & Roero, L. 1997;Efstathiou, J. 1996;Leon, V. J. et al. 1994;Sadeh, N. M. et al. 1993) es una heurística que consiste en mover toda la secuencia hacia el futuro, hasta que el evento es superado en el tiempo. La secuencia se respeta y queda tal cual fue establecida inicialmente, permitiendo introducir tiempos muertos.
- **Affected Operation Rescheduling (AOR)** (Abumaizar, R. J. et al. 1997;Hasle, G. & Smith, S. F. 1994;Leon, V. J. et al. 1994) es una técnica heurística en la cual sólo las operaciones que se ven afectadas por los efectos de un evento son modificadas. El concepto básico de esta técnica es el desplazamiento hacia delante de las fechas de inicio de algunas operaciones manteniendo las secuencias de inicio de cada máquina.

- **La técnica matchup** (Bean, J. C., Birge, J. R., Mittenthal, J., & Noon, C. E. 1991) consiste en hacer coincidir en el tiempo la nueva secuencia con la anterior en algún punto de la misma.

El resto de métodos incluidos en la clasificación tuvieron su origen en el campo de la **Inteligencia Artificial**:

- Las **propuestas basadas en Sistemas Multiagente** se basan en un conjunto de agentes independientes coordinados para resolver un problema en el que cada individuo proporciona su conocimiento, y sus habilidades para conseguir el objetivo común. Estas técnicas serán ampliamente revisadas en el capítulo siguiente. Algunas aplicaciones basadas en sistemas multiagente son (Shen, W. & Norrie, D. H. 1999; Sycara, K. P., Roth, S. F., Sadeh, N., & Fox, M. S. 1991).
- La Programación de la Producción basada en el **Razonamiento de los Casos**. Esta consiste en establecer el conjunto de características fundamentales de un determinado problema de forma que el sistema mantiene una base de datos de los problemas caracterizados y los resultados obtenidos. De esta forma se puede conocer el resultado anterior con respecto a la aplicación de una u otra solución en un determinado problema. Este tipo de técnicas se suelen combinar con las de lógica difusa o algoritmos de Búsqueda Tabú. Algunos casos prácticos de muestran en (Ovacik, I. M. et al. 1994; Sycara, K. P. & Miyashita, K. 1994; Szelke, E. & Markus, G. 1997).
- Las **técnicas basadas Restricciones**. Se basa en la teoría de satisfacción de restricciones ((Barber, F. 2003; Barták, R. 2001)) que permite tratar problemas en los que la solución pueda especificarse mediante el cumplimiento de unas determinadas restricciones. Algunas aplicaciones son analizadas en (Miyashita, K. 1995; Sycara, K. P. et al. 1994).
- Los **métodos basados en Lógica Difusa o Borrosa**. En la Programación de la Producción basada en lógica difusa los tiempos de proceso son reemplazados por distribuciones de tiempos difusos y las violaciones de restricciones temporales son expresadas como porcentaje. El grado de solapamiento entre la fecha de entrega y la fecha de difusa de finalización establece la calidad de la solución. Algunos ejemplos se muestran en (Dorn, J., Kerr, R., & Thalhammer, G. 1994; Slany, W. 1996).
- Las **Redes Neuronales**. Las redes neuronales son estructuras de procesamiento distribuido, paralelo y adaptativo que son capaces de aprender a partir de ejemplos y se inspiran en la estructura cerebral humana (Martin, B. & Sanz, A. 1997). Algunos ejemplos se pueden ver en (Garner, B. J. & Ridley, G. J. 1994; Rovithakis, G. A., Perrakis, S. E., & Christodoulou, M. A. 2001).

Las técnicas que se acaban de mostrar hay que entenderlas desde una óptica exclusiva de la Reprogramación. Son técnicas que han sido introducidas por algunos autores con el objeto de dotar de un procedimiento que permita combatir la pérdida de prestaciones de los talleres producidos por un continuo cambio en la ejecución programa inicial. Dichos autores han observado, que en un sistema dinámico, no es suficiente disponer de un algoritmo, basado en una metaheurística, heurística o incluso en un modelo matemático, y obtener un Programa Productivo que permita alcanzar una eficacia cercana a la óptima o incluso la óptima. En un sistema cambiante hay que mantener un equilibrio entre la modificación continua del Programa Productivo para mantener la medida de eficacia adecuada, y la estabilidad del mismo.

A continuación se presentan algunas referencias donde se contribuye de forma significativa en el ámbito de la Reprogramación de la Producción. En las referencias de **(Abumaizar, R. J. et al. 1997; Daniels, R. L. et al. 1995; Jensen, M. T. 2001; Wu, D. S. et al. 1991; Wu, S. D. et al. 1993)** se realizaron diversos estudios en los que utilizaron diversas medidas de robustez, medidas de eficacia y de estabilidad simultáneamente, para decidir sobre la mejor técnica de Reprogramación. En **(Abumaizar, R. J. et al. 1997)** concluyeron en su estudio que la reparación de un Programa Productivo era la más extendida en la práctica y que la Reprogramación completa se utilizaba en contadas ocasiones. En **(Sabuncuoglu, I. et al. 2000)** se demostró la potencia de la reparación de un Programa Productivo en términos de estabilidad del sistema. Por el contrario, **(Yamamoto, M. et al. 1985)** realizan un estudio mediante Reprogramación completa en un taller con fallos aleatorios en las máquinas. Los resultados mostrados por los autores concluyen con que el rendimiento del sistema es mejor con dicho método frente a las reglas de despacho o modificaciones parciales del programa.

En **(Subramaniam, V., Raheja, A. S., & Reddy, K. R. B. 2005a)** se analiza el problema de reconstrucción de un Programa Productivo en curso en un Taller General afectado por Incertidumbre. Los autores proponen la modificación de la heurística (Affected Operation Rescheduling AOR) para contemplar más de una causa de interrupción. Los autores indican que la eficacia de la propuesta es superior a la de la heurística Right-Shift.

El trabajo de **(Petrovic, D. & Duenas, A. 2006)** permite aplicar un modelo de lógica difusa al caso de un taller de Máquina en Paralelo con Incertidumbre en su rendimiento que puede implicar no disponibilidad de las mismas. Para realizar en modelo se consideran como parámetros inciertos la duración la cantidad de interrupciones. Se plantea una aproximación Predictiva-Reactiva de forma que el Programa Productivo inicial sea reparado sólo si las interrupciones son importantes. . El objetivo del Programa Predictivo (PP) es minimizar el C_{max} y el del Programa Reactivo (PR) minimizar la diferencia de las de fechas de finalización de los programas Predictivo y Reactivo ($\sum |C_j(PP) - C_j(PR)|$). Se proponen dos conjuntos de reglas

para la Reprogramación. El primero define cuando hay que reprogramar y el segundo como el método utilizado. La evaluación experimental confirma la eficacia del método.

En la propuesta de **(Duenas, A. & Petrovic, D. 2008)** se plantea un método de Programación de la Producción Predictivo-Reactivo para un Taller de Máquinas en Paralelo con interrupciones inesperadas. El objetivo del Programa Predictivo (PP) es minimizar el C_{max} y el del Programa Reactivo (PR) minimizar la diferencia de las de fechas de finalización de los programas Predictivo y Reactivo ($\sum |C_j(PP) - C_j(PR)|$). El Programa Predictivo se construye de forma que pueda absorber los eventos de menor impacto. Los autores proponen dos tipos de procedimientos Reactivos. El primero es la aplicación de la técnica Right-Shift ó desplazamiento de la secuencia hasta el instante de recuperación del evento. El segundo es la Reprogramación total de los trabajos pendientes.

Los autores de **(Tang, L. X. & Wang, X. P. 2008)** proponen la utilización de un sistema Predictivo-Reactivo en un caso aplicado a la industria del acero. Tratan de minimizar el impacto de los frecuentes eventos que suponen la no disponibilidad de alguna máquina. El objetivo del Programa Predictivo es la minimización de tres objetivos; la penalización en el cambio de partida, el retraso en la entrega de productos, y el inventario generado. El Programa Reactivo (PR) tiene que minimizar los tiempos no productivos y mantener la calidad de los productos. El PR se diseña para trabajar en tres fases. En la primera se busca trabajos que puedan sustituir a los no realizables utilizando el método Match-UP. En la segunda se verifica la factibilidad de los resultados y se reparan con una heurística. En la tercera fase se mejora la secuencia utilizando una Búsqueda Tabú. Los autores realizan un análisis experimental.

Las técnicas mostradas en este apartado alcanzan, en algunos casos, un alto grado de abstracción (Sistemas Multiagente o técnicas basadas en el Conocimiento y la Inteligencia Artificial) constituyéndose en verdaderos métodos generales de resolución orientados o aplicados a sistemas dinámicos. Estos métodos pueden ser combinados en su mayoría con otros métodos o instancias de los mismos (algoritmos genéticos, reglas de despacho, búsquedas tabú, etc.) de forma que los primeros velan por el equilibrio entre eficacia y estabilidad y los segundos buscan la obtención del mejor valor posible de algún parámetro concreto.

Finalmente, se debe comentar que las técnicas presentadas no han obtenido un reconocimiento masivo y su uso no se ha extendido demasiado. En la actualidad, no es frecuente encontrar casos prácticos en los que dichas técnicas se apliquen de forma explícita.

3.4.7 Descentralización en la Programación de la Producción.

Las nuevas formas de abordar los problemas, considerando la diversidad de intereses y desarrollos cada vez más complejos, no pueden estar sustentadas por el paradigma clásico de la toma de decisiones centralizadas. De hecho, más que una aproximación, la **Toma de Decisiones Distribuidas** ha llegado a ser la metodología predominante de sistemas complejos para la **Toma de Decisiones** (Schneeweiss, C. 2003) . Así pues la Toma de Decisiones Distribuidas (en inglés distributed decision making, **DDM**) se ha desarrollado rápidamente en el campo de la teoría decisional. Las **DDM** comprenden áreas tan diversas como la optimización multi-nivel, la Programación de la Producción Estocástica Multi-Etapa, la Planificación de Producción Jerárquica o la Gestión de la Cadena de Suministro entre otras.

Desde el punto de vista de la teoría de decisiones las áreas anteriormente mencionadas en **DDM** abarcan desde configuraciones de una sola persona hasta situaciones antagonistas multi-personas, como en la Teoría principal de Agenciaⁱⁱⁱ(Agentes y Sistemas Multiagentes). Así pues, se requiere cubrir un área que va desde los problemas estructurales de matemáticas hasta interferencias de la vida real. Para la situación de una sola persona, la **DDM** puede ser útil para entender mejor o manipular una situación de decisiones complejas. Esto es particularmente necesario para el caso de los sistemas dinámicos donde las Decisiones deben ser Distribuidas en el Tiempo y cuando se adquiere nueva información. Para una situación multi-persona, la **DDM** es mucho más obvia. A parte de la información diversa que los decisores deben poseer, se deben considerar los derechos de decisión y las competencias de decisión, así como la comunicación entre las unidades decisoras.

Con frecuencia estos problemas de decisión separados no tienen la misma importancia, lo que implica en muchos casos algún tipo de dependencia jerárquica. De hecho, las decisiones que no presentan algún tipo de característica jerárquica son excepcionales. Las decisiones tomadas en diferentes situaciones temporales son ejemplos característicos de dependencias jerárquicas asimétricas.

La **DDM** tiene dos ámbitos diferentes, separar un sistema en varios subsistemas y definir las tareas de coordinación. Se pueden distinguir al menos cuatro grados diferentes de coordinación (Schneeweiss, C. 2003):

ⁱⁱⁱ En el marco de trabajo de esta tesis esta teoría debe entenderse desde el enfoque que se realiza en el contexto de los Sistemas de Información e Inteligencia artificial.

- **Integración de Datos:** describe el nivel más bajo. Es el típico nivel de integración que se encuentra en la informática. Se centra principalmente en la consistencia de los datos y sus estructuras garantizando un buen intercambio de datos entre sistemas.
- **Integración de Sistemas a través de Negociación:** describe una situación auto-organizativa a través de las reglas de mercado. Los sistemas multiagente son con frecuencia de este tipo. En muchas situaciones existe un agente coordinador que posee las reglas de negociación y facilita los procesos de comunicación.
- **Integración a través de Planificación de Actividades:** Intenta coordinar los procesos de decisión de los sistemas separados a través de instrucciones y anticipaciones. Las instrucciones influyen en la estructura o en el campo de las decisiones de los sistemas. La influencia se limita, sin embargo a los cambios transaccionales. El concepto de anticipación ocupa un importante rol en contraste con las negociaciones miopes.
- **Integración a través de Actividades de Liderazgo:** no solo se deben realizar transacciones sino que se deben intentar compartir actividades que aporten valor, como podría ser el transporte. En este campo, los procesos de comunicación entre socios ocupan un papel fundamental, como por ejemplo convencer a éstos para llegar a un acuerdo de compartir actividades.

La teoría conceptual a desarrollarse en **DDM** estará fundamentalmente focalizada en aspectos de Programación de la Producción. Además de las actividades de Programación de la Producción, se considerarán actividades de negociación. La negociación será considerada como una parte de tareas de Programación de la Producción.

A continuación se comentan algunas referencias significativas que han proporcionado teoría y sobre todo han afrontado prácticas, en el ámbito de la Programación de la Producción Distribuida.

En (**Carvalho, D., Protti, F., De Gregorio, M., & Franca, F. M. G. 2005**) se presenta un algoritmo distribuido para Programación de Recursos Compartidos en un entorno de reconfiguración dinámica de la carga de trabajo. El algoritmo SERH - Scheduling by Edge Reversal with Hibernation es capaz de trabajar en condiciones de gran carga y desde el punto de vista de la comunicación puede ser visto como un grafo dinámico donde el intercambio de mensajes entre el nodo de proceso y sus vecinos se muestra como un arco reversible. Se ilustra el funcionamiento propuesto con un caso de un sistema de control de semáforos.

Para un Problema de Programación Distribuida en un contexto de multi-empresa y multi-producto (**Chan, F. T. S., Chung, S. H., & Chan, P. L. Y. 2005**) proponen un algoritmo genético

adaptativo. Plantean los problemas de asignación de trabajos a centros productivos y la correcta Programación Temporal de la Producción de los mismos para alcanzar el rendimiento esperado del sistema. En este trabajo se introduce un nuevo mecanismo de cruce denominado cruce de gen dominado que permite mejorar el rendimiento de la búsqueda y elimina el problema de la tasa óptima de cruce. Se realiza un ejemplo con un sistema de cinco fábricas.

Los autores de **(Jeong, I. J. & Leon, V. J. 2005)** consideran el problema de una sola máquina donde los decisores y la información están distribuidos en varios subsistemas. Los subsistemas comparten la máquina y deben cooperar entre sí para minimizar una función lineal basada en la fecha de finalización de los trabajos. Ni los subsistemas ni las máquinas disponen de una información completa. La solución propuesta se basa en una relajación Lagrangiana modificada.

El trabajo de **(Cicirello, V. A. & Smith, S. F. 2004)** presenta un nuevo enfoque al problema de coordinación, esbozando varios aspectos de un modelo computacional de coordinación de actividades individuales y de las tareas para alcanzar las necesidades colectivas. Se centran especialmente en el problema de configuración de máquinas paralelas multipropósito de una fábrica para mejorar la satisfacción de la demanda a lo largo del tiempo. El sistema se basa en Agentes con arquitecturas jerárquicas. La propuesta se ha experimentado en el problema de asignación en un sistema de simulación de pintado de vehículos.

En **(Siwamogsatham, T. & Saygin, C. 2004)** se muestra un algoritmo de subasta para la Programación de la Producción en tiempo real de un sistema de fabricación flexible con rutas alternativas. La efectividad del modelo propuesto se demuestra por comparación mediante simulación con varias reglas de despacho. La función de coste incorpora el aspecto temporal como criterio prioritario. Las medidas de rendimiento analizadas son la media del retraso, media de la tardanza, media de la desviación de la fecha de entrega, media del tiempo de espera, balance en la utilización y coste total. El análisis muestra un mejor rendimiento del sistema de subasta.

En **(Chun, H. W. & Wong, R. Y. M. 2003)** se presenta un esquema generalizado basado en Agentes que usa negociación para programar dinámica y óptimamente en función de eventos. Los eventos pueden ser creados dinámicamente por cualquier Agente del entorno. Cada evento puede requerir recursos o la colaboración potencial con uno o más Agentes. La asignación de recursos es negociada iterativamente hasta alcanzar un compromiso. Se usa un modelo de evaluación y un protocolo de negociación llamado Nstar (N^*). La propuesta se evaluó en un entorno de gestión de citas basado en agentes llamado MAFOA.

El trabajo de **(Gargouri, E. & Hammadi, S. 2003)** trata el problema de las industrias de alimentación en las cuales usan productos primarios caracterizados por fecha de caducidad. Se

propone un sistema de Toma de Decisión Distribuido para afrontar el proceso de Programación de la Producción en tiempo real. El sistema propuesto pretende proporcionar opciones robustas al decisor y está distribuido a lo largo de la cadena de fabricación. En cada decisión se tiene en cuenta a los vecinos, con el objetivo de tener en cuenta sus restricciones para satisfacer simultáneamente los diferentes criterios. El sistema está preparado para tratar diferentes grados de importancia en los criterios mediante un modelo de evaluación.

En **(Jian, H. Z., Nee, A. Y. C., Fuh, J. Y. H., & Zhang, Y. F. 2003)** se realiza una reflexión sobre la aplicación de los algoritmos genéticos (AG) aplicados a los problemas de Programación de la Producción y la evolución de los problemas de un entorno mono taller a un entorno distribuido. Los autores proponen un AG modificado el cual es capaz de resolver los problemas de Programación de la Producción tradicionales así como los distribuidos. Con la propuesta se alcanzan varios objetivos como el de minimizar el makespan, el coste y criterios múltiples ponderados.

Los autores de **(Jeong, I. J. & Leon, V. J. 2002)** presentan una metodología de Programación de la Producción Distribuida para el problema de un taller de flujo con dos máquinas. Se asume que los decisores y la información se encuentran distribuidos en varios subsistemas productivos que comparten dos máquinas para satisfacer la demanda. Los problemas de Programación de la Producción son resueltos mediante una técnica de relajación Lagrangiana modificada para trabajar en entornos con información compartida muy restringida. El principal objetivo del algoritmo es encontrar un estado de compromiso donde las restricciones locales y globales sean satisfechas, y se minimice la suma ponderada total de tiempos de finalización.

El estudio de los sistemas distribuidos de Programación de la Producción donde los nodos pueden sufrir sobrecarga y esta puede ser transferida a otros nodos menos cargados es analizado por **(Barroso, A. M., Leite, J. C. B., & Loques, O. G. 2002)**. Especialmente, analizan sistemas que utilizan sistemas Bayesianos para inferir el estado de carga del sistema y seleccionar el nodo adecuado para transferir para de esta. Los autores detectan que el método basado en Bayesianos se basa en funciones objetivos difíciles de definir, y en ocasiones, subjetivas. En este trabajo se propone una mejora para añadir objetivos a los algoritmos Bayesianos y mejorar el número de trabajos ejecutados en un sistema distribuido de tiempo real.

En **(Dewan, P. & Joshi, S. 2002)** se presenta una nueva formulación del taller de trabajo que Programa los trabajos usando subasta para distribuir el control. Se presentan las bases teóricas para la descomposición del problema, la construcción de ofertas, y la evaluación de la subasta usando herramientas de Programación matemática. Los resultados muestran que la propuesta supera los despachos distribuidos y que puede ser usada para crear Programas Productivos para un amplio conjunto de objetivos de Programación de la Producción.

El autor de **(Ghosh, S. 2001)** examina críticamente la naturaleza de los sistemas reales y de gran escala, y observa que muchas de las complejidades están compuestas de entidades. Dichas entidades son unidades de decisión independientes, concurrentes y auto-contenidas, que interactúan unas con otras asincrónamente. Los autores comentan algunos ejemplos de sistemas de decisión asíncronos distribuidos.

En la propuesta de **(Nakamura, M., Nakada, N., Kinjo, H., & Onaga, K. 2001)** se propone un esquema distribuido autónomo para el problema de máquinas paralelas. A través de simulación se observa que el esquema propuesto puede reducir rápidamente la tardanza superando otras propuestas y que puede adaptarse flexiblemente a situaciones no usuales como la llegada de más trabajos.

En el trabajo de **(Bemelman, R., Tharumarajah, A., Welgama, P., & Wells, A. 1999)** se considera el problema de la Programación de la Producción de líneas de montaje, y su evolución desde enfoques centralizados hasta los más recientes distribuidos o heterárquicos. Así los autores extienden su trabajo previo en el área de control distribuido basado en el comportamiento y examinan el rendimiento del enfoque basado en un control centralizado basado en el comportamiento para un problema de Programación de la Producción de líneas de ensamblado. Se analiza el rendimiento con problemas de tres y cuatro estaciones de ensamblado comparando los resultados del modelo propuesto con los resultados de los modelos basados en Programación matemática entera.

En **(Miyashita, K. 1998)** se propone una nueva arquitectura distribuida para la Planificación y Programación de la Producción. Dicha arquitectura alcanza el cumplimiento de las restricciones mediante la *descomposición y coordinación del problema*. El objetivo que se propone es desarrollar un método eficiente para tratar problemas Planificación/Programación de la Producción de gran complejidad de forma distribuida y sin perder calidad en la solución. Se propone un prototipo (CAMPS) implementado con Agentes Inteligentes que coordinan sus acciones manejando restricciones intra-agente e inter-agente. Se describe la metodología para la Planificación y Programación Distribuida de la Producción junto con los mecanismos basados en restricciones de la creación de coaliciones dinámicas entre agentes.

La propuesta de **(Tharumarajah, A. & Bemelman, R. 1997)** presentan un entorno de creciente autonomía y distribución en los talleres de trabajo, en el que las características operacionales de dichos talleres derivan en acciones independientes pero cooperativas. Los autores revisan, con una visión eminentemente práctica, los métodos de negociación y basados en el comportamiento para la Programación de la Producción y coordinación de entidades distribuidas en estructuras jerárquicas y heterárquicas.

Los autores de (Kim, K. H., Bae, J. W., Song, J. Y., & Lee, H. Y. 1996) sugieren Programar y Controlar un taller de trabajo mediante un sistema basado en la filosofía de mercado y basado en Agentes. Cada recurso se negocia en función de ofertas evaluadas según su precio y un mecanismo de revisión del mismo. Se consideran rutas alternativas para cada orden de trabajo. Finalmente los autores proponen un sistema de simulación para estimar el inicio y fin de cada tarea.

Otros ejemplos destacables de aplicación de sistemas distribuidos a resolución de problemas de Programación de la Producción anteriores a 1996 son: (Duffie, N. A. & Prabhu, V. V. 1994; Lu, S. H. & Kumar, P. R. 1991; Tang, Z. B. & Shi, L. Y. 1992). Así mismo por lo que hace referencia a arquitecturas y marcos conceptuales propuestos con anterioridad a dicha fecha se pueden destacar: (Deen, S. M. 1993; Hadavi, K., Hsu, W. L., Chen, T., & Lee, C. N. 1992).

3.5 El Taller de Flujo, Flujo Híbrido y los Tiempos de Cambio de Partida Dependientes de la Secuencia

3.5.1 Introducción

En el presente apartado, y una vez centrado el marco teórico de aproximación al problema de Programación de Producción en un entorno Dinámico y con Incertidumbre, se analiza un tipo de configuración productiva, que en los últimos años, comienza a ser estudiada con detalle, denominada Taller de Flujo Híbrido con Tiempos de Cambio de Partida Dependientes de la Secuencia (en Inglés flowshop with Sequence Dependent Setup (**SDST**)). El interés de abordar este tipo de configuraciones es que, como se verá en el capítulo 5, uno de los sistemas productivos objeto de estudio en esta tesis es asimilable a dicho Taller.

Para comprender adecuadamente la forma de operar en un taller de flujo **SDST** es necesario partir del concepto de **Taller de Flujo**, posteriormente introducir el concepto de **Taller de Flujo Híbrido** y finalmente aplicar sobre este la problemática de los **Tiempos de Cambio de Partida Dependientes de la Secuencia**.

3.5.2 Definiciones de Taller de Flujo y Flujo Híbrido

En un taller de flujo estándar o "*flowshop*" tenemos un conjunto n de trabajos que deben ser procesados en un conjunto m de máquinas. En principio cada trabajo debe de procesarse en todas y cada una de las máquinas, con la particularidad de que el orden de proceso es el

mismo para todos los trabajos. Es decir, existe un “flujo” común de los trabajos en las máquinas. El flujo de trabajo en las máquinas implica una relación de precedencia en las operaciones, es decir, para cualquier trabajo i , el orden de procesamiento de sus m operaciones es el siguiente: $O_{1i}, O_{2i}, \dots, O_{mi}$. Se puede contemplar la circunstancia en la cual el Tiempo de Proceso en una máquina sea cero, es decir no se realiza ninguna operación, en este caso se habla de Taller de Flujo Generalizado (Baker, K. R. 1974).

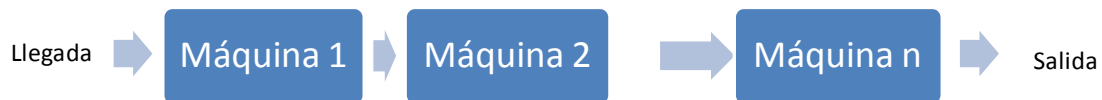


Figura 3-4. Representación de un Taller de Flujo con m máquinas o Etapas (elaboración propia).

En (Riane, F., Artiba, A., & Elmaghraby, S. E. 1998) se define el Taller de Flujo Híbrido como una configuración de máquinas organizadas en ' r ' Secciones o Etapas donde se procesan una serie de ' n ' Piezas (Trabajos, en la terminología utilizada en esta tesis). Cada pieza recibe como máximo ' r ' Operaciones (una en cada etapa o sección). Una sección contiene un conjunto de ' m_r ' Recursos (máquinas) susceptibles de ejecutar una misma operación, estos son equivalentes en cuanto a su funcionamiento, aunque pueden no serlo en su eficacia, debido a que la duración de una operación puede depender del recurso elegido dentro de una sección (en el caso de máquinas se podrá hablar de máquinas paralelas idénticas, proporcionales o no relacionadas). Las máquinas solamente pueden procesar un único trabajo a la vez y cada producto recibe una única operación por etapa. El flujo de los trabajos es unidireccional desde la etapa 1 a la ' r '.

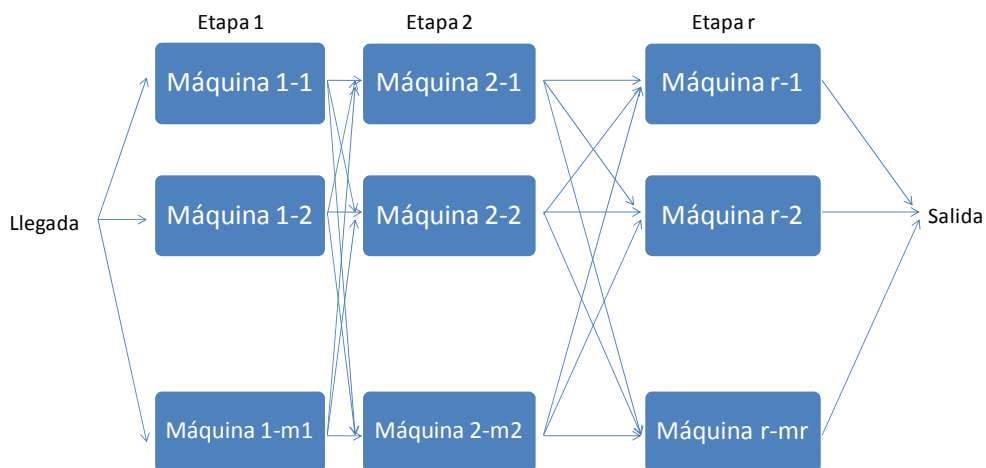


Figura 3-5. Representación de un Taller de Flujo Híbrido con r Etapas (elaboración propia).

Estas definiciones pueden ser matizadas en las diferentes aproximaciones a problemas reales o simplificados que realizan los diversos autores. Aspectos como la inclusión de Tiempos de Cambio de Partida Dependientes o no de la Secuencia, aparición o no de Almacenes Intermedios, Aparición o no de Máquinas Idénticas, Uniformes o no Relacionadas, Restricciones en la ruta de algunos trabajos, Disponibilidad de recursos, etc. pueden condicionar la forma de afrontar la resolución del problema. En los próximos capítulos se tratarán algunos de los casos más relevantes.

3.5.3 El Problema de los Tiempos de Cambio de Partida

Se puede definir a los Tiempos (o costes) de Cambio de Partida como una cantidad de tiempo (o un coste monetario) invertida en ajustar los recursos para pasar de realizar un trabajo a otro diferente (Andrés, C. 2001).

Según (Artigues, C. 1997) la preparación de los recursos se puede **modelar** mediante dos visiones diferentes:

- Los **Tiempos de Preparación** (setup o changeover time).
- Los **Costes de Preparación** (setup o changeover cost).

Usualmente los problemas de Programación de Producción relacionan en una escala temporal, la fecha de inicio de las diferentes operaciones de los trabajos entre sí, por lo que se suele utilizar el concepto de Tiempo de Cambio o ajuste, y es por ello que es más frecuente en la literatura.

Sin embargo, algunas veces se utilizan restricciones basadas en conceptos expresados en unidades monetarias (inventarios, porcentajes de defectos, etc,...) con lo que en estos casos se traducen estos Tiempos de Cambio de Partida en unos costes dados por el coste de oportunidad debido al mantenimiento de la maquinaria inactiva, costes debido a la aparición de piezas iniciales defectuosas, etc., y que permiten manejar los diferentes tipos de restricciones bajo una unidad de medida común.

Se observa que se usa principalmente la visión temporal a la hora de representar las situaciones donde aparecen Cambios de Partida, y a continuación se destacan las diferentes aportaciones encontradas en la literatura, al respecto.

3.5.3.1 El Tiempo de Preparación.

El tiempo de preparación es una cantidad de tiempo durante la que el recurso está inactivo con posterioridad a la finalización de un trabajo y previamente a la ejecución de otro trabajo diferente. Se suele expresar al inicio de la operación (tiempo de ajuste), al final (tiempo de desmontaje) o ambos.

Normalmente, se definen los Tiempos de Cambio en función de los trabajos a realizar o la máquina donde se van a realizar. Así se pueden distinguir:

- **Tiempos de Cambio fijos.** Su valor es idéntico sea cual sea el trabajo a secuenciar o el recurso donde se secuencian. En estos casos, el Tiempo de Cambio se puede incluir o no dentro del Tiempo de Procesamiento del trabajo. Como caso particular se puede considerar el valor nulo.
- **Tiempos de Cambio variables.** Su valor es diferente en función de los trabajos o máquinas donde se secuencian. A su vez se pueden clasificar en dos subtipos:
 - Tiempos de montaje/desmontaje independientes de la secuencia pero dependientes del recurso.
 - Tiempos de preparación Dependientes de la Secuencia y Dependientes del Recurso

Los **tiempos de montaje/desmontaje independientes de la secuencia pero dependientes del recurso** (ver Figura 3-6), denominados S_{nsd} / R_{nsd} (non sequence dependent Setup/Removal Times). El Tiempo de montaje/desmontaje que inmoviliza el recurso antes o después de una operación i y que no depende de i , se denomina S_i / R_i . Estos Tiempos de Cambio representan la duración de las operaciones de ajuste, fijación o preparación de máquina antes de empezar un nuevo producto (o al finalizar un producto en el caso de tiempos de desmontaje). En algunos casos, un **Tiempo de Cambio Dependiente de la Secuencia se puede descomponer** fácilmente en el par S_{nsd} / R_{nsd} , de tal forma que aunque por separado no sean dependientes de la secuencia su suma sí lo sea.

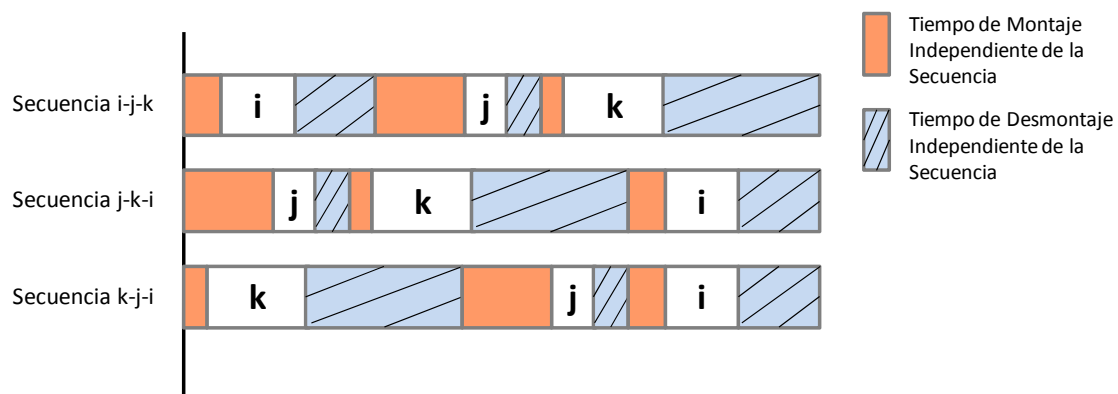


Figura 3-6. Caso de tiempos de montaje y desmontaje fijos. Adaptado de (Artigues, C. 1997).

Los **Tiempos de Preparación Dependientes de la Secuencia y Dependientes del Recurso** son denominados S_{sd} (sequence dependent Setup Times). En (Baker, K. R. 1974) se asegura que los Tiempos de Preparación Dependientes de la Secuencia son una característica definitoria de la secuenciación en taller mecánico general. El tiempo de preparación necesario entre dos operaciones i y j que depende a la vez de la operación precedente i y de la operación sucesora j , se denomina S_{ijk} . En este caso se considera como un todo, el conjunto de operaciones de montaje y desmontaje puesto que no admiten fácilmente una separación. Para un problema concreto se define el conjunto $\{S_{ij}\}^k$ definido por una matriz de tiempos de preparación del recurso k . También pueden existir tiempos de cambios cuyo valor no sólo depende de la secuencia y de las máquinas, sino además de otros recursos auxiliares como pueden ser mano de obra involucrada (tipo y cantidad), utillaje o sistemas de transporte (tipo, velocidad,...). En función de las características de la matriz de tiempos de cambio podemos tener:

- **La matriz de cambios admite una buena agrupación por familias de piezas.** En estos casos se aplican los planteamientos basados en conceptos de tecnología de grupos (TG) (Burbidge, J. L. 1992; Burbidge, J. L. 1963; Hashiba, S. & Chang, T. C. 1991; Potts, C. N. & Vanwassenhove, L. N. 1992). Así, se pueden plantear dos partes del tiempo de ajuste. Un tiempo de ajuste grande (mejor setup) que significa la duración entre familias, y un tiempo de ajuste pequeño o inferior al mayor setup (minor setup) que considera las pequeñas pérdidas de tiempo por ajustes entre trabajos de la misma familia.
- **La matriz de tiempos de cambio no admite una buena agrupación por familias.** Este es el caso más general y complejo, aunque raramente se encuentra en la práctica.

Dentro de los problemas multioperación o dentro de los problemas mono-operación que incluyen las fechas de llegada, es necesario indicar si la preparación puede ser realizada con

antelación a la llegada de las piezas a la máquina. Si la preparación no puede ser anticipada se definen como Tiempos de Cambio unidos al Tiempo de Proceso y se incorporan al mismo a efectos de cálculo (Johnson, S. M. 1954). Mientras que si la preparación puede ser anticipada, se habla entonces de Tiempos de Preparación separados del Tiempo de Proceso y no se deben tratar junto a los tiempos de ejecución según (Yoshida, T. & Hitomi, K. 1979), existiendo la posibilidad de insertar tiempos muertos entre el tiempo de preparación y el de operación. Esto posibilita que la máquina ya esté preparada cuando la pieza llega de la operación anterior. Muchas veces esto no sólo depende de las características de los trabajos o máquinas sino más bien de la forma de gestión del taller. Si hasta el último momento no se toma la decisión de secuenciación, no es posible preparar con antelación la máquina. Esto sucede en sistemas de control de talleres muy descentralizados donde las decisiones de secuenciación no dependen demasiado de un sistema de Programación central, sino que son tomadas por los operarios de los diferentes puestos con un periodo de anticipación muy bajo.

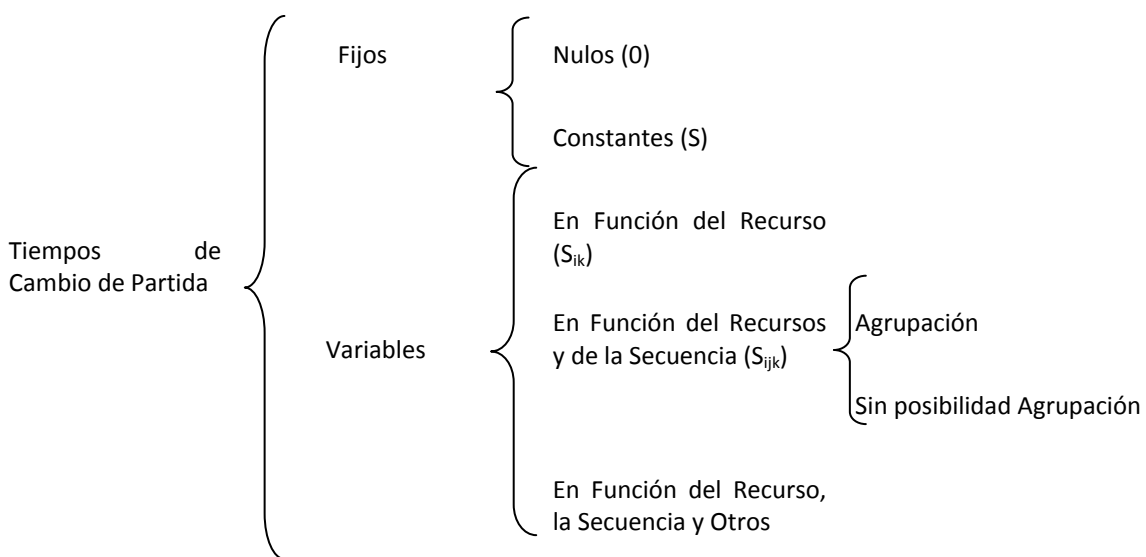


Tabla 3-3. Clasificación de los tipos de tiempos de cambio(Andrés, C. 2001).

La Figura 3-7 resume los diferentes modelos planteados dando para cada uno de ellos su representación según un diagrama de Gantt.

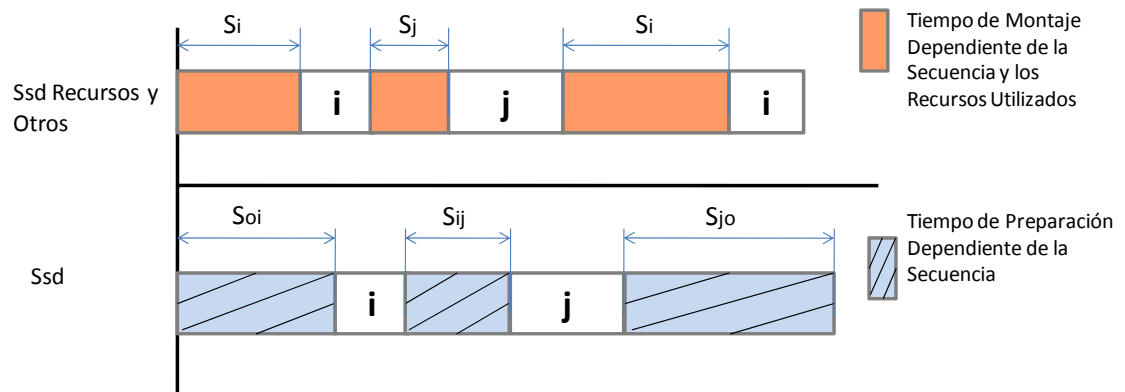


Figura 3-7. Representación gráfica de algunos tipos de Tiempos de Cambio (elaboración propia).

El criterio a minimizar más frecuentemente usado ligado a los tiempos de preparación es el tiempo total de preparación o suma de todos los tiempos de preparación $\sum S_{ij}$

Este criterio se puede combinar con otros criterios clásicos mediante una suma ponderada o también puede no estar presente en la función objetivo. En este caso, el objetivo a minimizar puede ser exclusivamente algún criterio clásico (C_{max} , L_{max} ,...) que tenga en cuenta las restricciones de preparación.

3.5.4 Definición y Problemática del Taller de Flujo Híbrido con SDST

Se puede decir que el Taller de Flujo Híbrido con Tiempo de Cambio de Secuencia Dependiente de la Secuencia (Taller de Flujo **HSDST**) es una extensión del Taller de Flujo definido por (Riane, F. et al. 1998). En éste existe un Tiempo de Preparación necesario entre dos operaciones i y j , que depende a la vez de la operación precedente i y de la operación sucesora j (S_{ijk}) por lo que se necesita un conjunto $\{S_{ij}\}^k$ definido por una matriz de tiempos de preparación para cada recurso k . Se contempla por lo tanto sólo el caso de Tiempos de Cambio de Partida variables en función del recurso y de la secuencia.

Dentro de un problema de Taller de Flujo “estándar” o Flow Shop sólo es necesario Tomar un tipo de Decisión que es la relativa a la secuenciación de tareas. El problema de asignación no existe, puesto que sólo hay un recurso por etapa. Este no es el caso del Taller de Flujo Híbrido donde podemos encontrarnos máquinas en paralelo. Así pues, deben Tomarse dos Decisiones a la hora de realizar la Programación de un Trabajo. Por un lado existe el **problema de Asignación de los Trabajos a las máquinas de cada Etapa**, mientras que por otro se debe

resolver un **problema de Secuenciación de los Trabajos en las diferentes máquinas** (Vignier A., Billaut, J. C., & Proust, C. 1995). Ambos problemas están íntimamente relacionados.

Cuando se trata con Taller de Flujo con HSDST además del problema de Secuenciación y Asignación hay que **considerar de forma simultánea la aparición de tiempos adicionales (Tiempos de Cambio de Partida) variables en función de las decisiones de Asignación y Secuencia** realizadas. Es decir hay que considerar $n_i!$ (permutaciones del número de trabajos considerados en el recursos i) combinaciones adicionales.

3.5.5 Enfoque Distribuido de la Programación del Taller de Flujo.

Un posible enfoque para afrontar el problema de Programación de la Producción en los Talleres de Flujo o Flujo Híbrido es la de *considerarlos como una composición de etapas con diferente grado de vinculación*. Según el problema que se afronte, y en función de las características del mismo, el programador puede encontrarse ante un caso de un alto grado de dependencia inter-etapas o de gran desacople.

En base a las aportaciones de los diferentes autores consultados y presentados en el apartado 3.4.7, es posible considerar adecuadas las técnicas de resolución de problemas de Programación de la Producción Distribuidas para el caso que se aborda en esta tesis. **La Programación de la Producción en la Industria Cerámica, y en especial en la empresa que se analiza, presentan la característica de asemejarse a un Taller de Flujo que en ocasiones puede presentar un alto grado de desacople.**

Aunque en el apartado 3.4.7 se han referenciado algunas de las comunicaciones más interesantes sobre Programación de la Producción Distribuida en el ámbito industrial, resulta complicado encontrar trabajos que aborden dicho problema aplicado a los Talleres de Flujo o Flujo Híbrido. No obstante, (Jeong, I. J. et al. 2002; Nishi, T., Konishi, M., & Hasebe, S. 2004) proporcionan dos trabajos sobre Programación de la Producción Distribuida en el ámbito del Taller de Flujo, que respaldan este enfoque.

En (Nishi, T. et al. 2004) se propone un método de Programación de la Producción descentralizado para un Taller de Flujo con restricción de recursos usando descomposición Lagrangiana y un sistema de coordinación. En este trabajo la descomposición en subproblemas se realiza mediante un algoritmo de recocido simulado combinado con Programación dinámica. Mediante la descomposición del problema en máquinas el coste del cambio de partida es incorporado a la función de coste de forma sencilla. El método es contrastado con

los resultados obtenidos mediante recocido simulado aplicado al taller completo mostrando ser más competitivo que el método tradicional.

Los autores de (Jeong, I. J. et al. 2002) presentan una metodología de Programación de la Producción distribuida para el problema de un Taller de Flujo con dos máquinas. Se asume que los decisores y la información se encuentran distribuidos en varios subsistemas productivos que comparten dos máquinas para satisfacer la demanda. Los problemas de Programación de la Producción son resueltos mediante una técnica de relajación Lagrangiana modificada para trabajar en entornos con información compartida muy restringida. El principal objetivo del algoritmo es encontrar un estado de compromiso donde las restricciones locales y globales sean satisfechas, y se minimice la suma ponderada total de tiempos de finalización.

Ambos autores, aunque desde perspectivas diferentes, **asumen un grado de descentralización de la Información y de las Decisiones (Johnson, S. M. 1954) en un Taller de Flujo**. Esta práctica puede extenderse al Taller de Flujo Híbrido característico en la Industria Cerámica. En la producción de baldosas cerámicas la Programación Productiva de cada etapa, constituida por un conjunto de máquinas con diversas características, puede alcanzar un alto grado de complejidad de forma que, en un gran número de ocasiones, la información y las decisiones son tomadas en dicho nivel.

En esta tesis se valoró la posibilidad de plantear un enfoque de Taller de Flujo Híbrido con Toma de Decisiones Distribuidas. No obstante, después de un periodo de reflexión se descartó ya que presentaba dificultades importantes cuando se trataba con Talleres de Flujo Híbrido. La mayor dificultad observada fue la “coordinación de la decisiones tomadas” por las diversas etapas. En todo caso, se cree que dicho enfoque puede ser implementable para algunos casos particulares, y que combinados con el uso de Sistemas Multiagente pueden dar resultados interesantes.

3.6 Procedimientos para abordar un Problema de Optimización

3.6.1 La Teoría de la Complejidad

La teoría de la complejidad se basa en el estudio de la dificultad intrínseca de resolución de los problemas de optimización y los problemas de decisión (Garey, M. R. & Johnson, D. S. 1979; Lenstra, J. K., Rinnooykan, A. H., & Brucker, P. 1977).

En la teoría de la complejidad la palabra “problema” se asocia a una descripción genérica de un problema, mientras que el término “instancia”, “ejemplo” o “ejemplar” se aplica a un problema donde se han especificado todos los datos exactamente.

Un **problema de optimización es aquel que la solución a una instancia del mismo implica un valor óptimo de una función objetivo**. La resolución de un problema de decisión sólo puede tomar dos valores (sí o no). Cualquier problema de optimización se reduce a un problema de decisión.

Según esta teoría **la eficiencia de un algoritmo se mide por el máximo (caso más desfavorable) número de pasos de ordenador necesarios para obtener una solución óptima** en función del tamaño de la instancia considerada.

Así pues, se puede clasificar cualquier problema en dos clases:

- **La clase P** contiene todos los problemas para los que existe un algoritmo de resolución, tal que el número de pasos de cálculo que requiere se puede expresar mediante una función polinomial en relación al tamaño del problema.
- **La clase NP** contiene todos los problemas para los que una solución determinada puede ser verificada si es factible o no (problema de decisión), mediante un algoritmo en el que el número de pasos de cálculo se puede expresar mediante una función polinomial, por lo que la clase NP incluye a la clase P ($P \subseteq NP$).

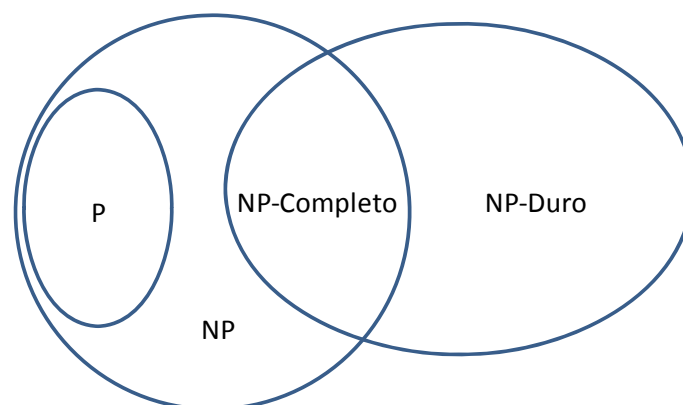


Figura 3-8. Representación de los tipos de problema en base a su dificultad (elaboración propia).

Un problema se denomina NP-duro, si cualquier problema de la clase NP se puede transformar en dicho problema. Un problema es NP-Completo si es de clase NP-duro y también pertenece al conjunto NP. Como se puede ver en la Figura 3-8 no todos los problemas NP son HP-duros.

Debido a que los problemas NP-Complejos incluyen problemas clásicos difíciles (Circuito Hamiltoniano, Problema del Viajante, programación Entera,...) para los que, a pesar de muchos intentos, nadie ha sido capaz de encontrar algoritmos de resolución polinomiales, **se asume que el subconjunto de problemas NP-Complejos son los más difíciles de resolver**, y muy probablemente todos estos problemas sean resolubles sólo mediante el uso de algoritmos de cálculo cuyo número de pasos se exprese mediante una función exponencial de los parámetros del ejemplo.

Así pues, la **Teoría de la Complejidad sugiere que para los problemas NP-Complejos, no es probable que exista un algoritmo que de una solución óptima en tiempo razonable**, por lo que es mejor intentar obtener una solución razonablemente buena mediante algún **procedimiento heurístico**.

En este caso, puede haber más de un algoritmo disponible para resolver un determinado problema. Por ello es conveniente realizar una evaluación comparativa de los distintos algoritmos existentes para resolver un mismo problema, con el fin de elegir el método más apropiado para resolver dicho problema. A la hora de comparar distintos algoritmos se deben tener en cuenta dos importantes criterios:

- **Eficacia:** Si el valor óptimo es conocido, entonces la eficacia puede ser expresada numéricamente como un simple ratio o como una diferencia relativa. Para los problemas complejos el valor óptimo puede no ser útil, debido al elevado tiempo de cálculo para obtener ese valor óptimo. En este caso puede que sea necesario realizar un análisis de los límites o un análisis del peor de los casos.
- **Eficiencia:** La eficiencia se refiere a la necesidad de aprovechar los recursos para obtener una solución. Como la mayoría de problemas de secuenciación son problemas de optimización combinatorios difíciles, la eficiencia del algoritmo es importante para que éste pueda ser utilizado en la práctica. Las medidas que a menudo se usan para reflejar este criterio incluyen la complejidad del algoritmo y el tiempo "running". Por ejemplo, para algoritmos simples, la complejidad puede ser representada mediante expresiones matemáticas y después ser comparadas. Para algoritmos más complejos se requiere experimentación computacional.

Ambos criterios pueden considerarse con respecto al tamaño y a la estructura del problema. Se debe tener cuidado en la generación de los problemas-prueba, particularmente en lo que respecta al tamaño del problema. Por supuesto, el trabajo comparativo debería hacerse sobre el mismo grupo de problemas-prueba. La independencia y la objetividad son dos atributos muy valiosos en este tipo de estudios.

En el caso de los algoritmos que se aplican en línea para problemas dinámicos, es usual utilizar como medida de la eficacia del análisis del peor caso, donde el valor del parámetro de rendimiento del algoritmo en línea es comparado con el del algoritmo estático o fuera de línea. Esta aproximación para el caso de los algoritmos en línea fue realizada inicialmente por (Sleator, D. D. & Tarjan, R. E. 1985), quienes lo usaron para un problema de análisis de actualización de listas. El término análisis competitivo fue utilizado por (Karlin, A. R., Manasse, M. S., Rudolph, L., & Sleator, D. D. 1988). Dado un conjunto σ de trabajos, siendo $\text{coste}_A(\sigma)$ el coste incurrido por el algoritmo A sobre σ . Siendo $\text{coste}(\sigma)$ el mejor coste para el algoritmo estático sobre σ . Se dice que un algoritmo A es ρ -competitivo si $\text{coste}_A(\sigma) \leq \rho \text{coste}(\sigma)$ para todos los trabajos del conjunto σ . El ratio competitivo de A es el ínfimo conjunto de valores de ρ para los cuales A es ρ -competitivo. El objetivo es encontrar el algoritmo en línea con ratio de competitividad más pequeño.

3.6.2 Tipos de Procedimiento de Resolución del Problema de Programación de la Producción

Un problema de optimización trata de alcanzar la mejor decisión para maximizar (ganancias, velocidad, eficiencia, etc.) o minimizar (costos, tiempo, riesgo, error, etc.) un criterio determinado, teniendo en cuenta las restricciones que impiden que cualquier decisión sea posible. Existen múltiples formas de abordar dichos problemas. En este apartado se ofrece una clasificación de las mismas (ver Tabla 3-4):

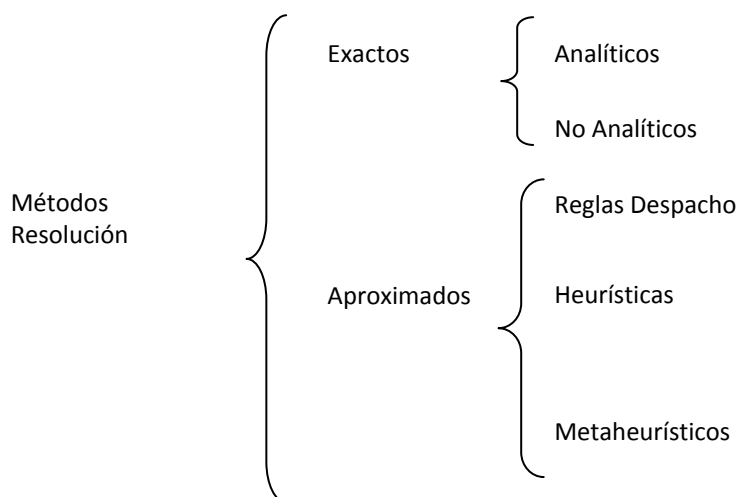


Tabla 3-4. Clasificación de los métodos de resolución de los Problemas de Programación de la Producción (elaboración Propia).

Según la clasificación propuesta de la Tabla 3-4 la primera gran división se realiza entre métodos exactos, es decir que son capaces de alcanzar al menos una de las mejores soluciones posibles para un criterio dado, y los métodos aproximados, es decir, que no garantizan la obtención de una solución óptima.

Existen diferentes métodos que permiten alcanzar una solución óptima. Es difícil ofrecer una clasificación de los métodos exactos. Sin embargo, una posible agrupación sería atendiendo a si estos son analíticos o no. Los procedimientos analíticos ofrecen un análisis de las variables en el que se establece una relación de asociación o causalidad. Entre los métodos analíticos se encuentra, por ejemplo, el método simplex para modelos de Programación Lineal, el método de Planos de Corte, o incluso el de Programación Dinámica. Existen métodos no analíticos entre los que destacan los basados en la enumeración de soluciones como es el caso del algoritmo de Ramificación y Poda (Branch and Bound).

Dentro de los métodos aproximados se ha realizado una diferenciación en base a la arquitectura del método en sí. De esta manera se distingue entre procedimientos basados en Reglas de Despacho, Heurísticos y Metaheurísticos.

Las Reglas de Despacho no constituyen un método en sí mismo, ya que sólo sirve para establecer un criterio de selección o prioridad. Estas reglas pueden ser utilizadas dentro de un procedimiento más complejo (Heurístico o Metaheurístico). Existen innumerables reglas (FIFO, LIFO, SOT, etc.). En el problema concreto de Programación de la Producción pueden jugar un papel importante, ya que en la mayoría de los algoritmos se propone un procedimiento para la selección de un trabajo, que posteriormente será secuenciado. En este punto “la selección” de un trabajo, y otra veces en “la selección” de un recurso (máquina, etc.) es donde tienen cabida las Reglas de Despacho.

Los métodos Heurísticos son procedimientos completos, es decir, que proporcionan soluciones factibles, aunque no necesariamente óptimas. Este tipo de algoritmos buscan alcanzar una solución de alta calidad en el menor tiempo posible, y se suele aplicar a problemas de alta complejidad. Su estrategia consiste en no considerar de forma deliberada algún aspecto del problema, y así poder alcanzar una solución analizando un espacio de soluciones más reducido del realmente existente. Por lo tanto, trata de algoritmos, normalmente diseñados para resolver un tipo de problema muy concreto, y que por lo tanto no tienen aplicación genérica, ya que pueden proporcionar soluciones malas o infactibles en determinadas instancias. Al principio, esta forma de resolver problemas, no fue bien vista en los círculos académicos, debido fundamentalmente a su escaso rigor matemático. Sin embargo, gracias a su interés práctico para solucionar problemas reales se fueron abriendo poco a poco las puertas de los métodos Heurísticos.

Los procedimientos Metaheurísticos son algoritmos de alto nivel que se utilizan para cohesionar un conjunto de acciones mezcla de otros procedimientos, como Heurísticos, Reglas de Despacho, métodos exactos aplicados a partes del problema, etc. Estos métodos tiene el objetivo de resolver un tipo específico de problemas, y en este aspecto son de menos particularizado que los procedimientos Heurísticos. Las Metaheurísticas proporciona una serie de pasos que el usuario debe seguir, pero también proporciona una serie de parámetros para ajustar la aplicación de los mismos, así como ámbitos donde el usuario puede insertar sus propuestas, normalmente Heurísticas. Existen múltiples propuestas, muchas de ellas basadas o inspiradas en el comportamiento de la naturaleza, otras en técnicas de Inteligencia Artificial. Algunos ejemplos son:

- La Búsqueda Tabú (Tabu Search).
- El Recocido Simulado (Simulated Annealing).
- Los Algoritmos Genéticos (Genetic Algorithms).
- Los Algoritmos Meméticos, también conocidos como Algoritmos evolutivos (Memetic Algorithms).
- La Optimización Aleatoria (Random Optimization).
- La Optimización basada en Colonias de Hormigas (Ant colony Optimization).
- Los Algoritmos Voraces (Greedy Algorithms).
- La Ascensión de Colinas (Hill-Climbing).
- La Inteligencia de Enjambre (Swarm Intelligence).
- La Búsqueda de Armonía (Harmony search).

Entre los algoritmos Metaheurísticos se resaltan el de Recocido Simulado, Búsqueda Tabú y Genético por los resultados obtenidos en el problema de Programación de la Producción.

Los algoritmos de Recocido Simulado (Simulated Annealing-SA) se basan en una analogía con el proceso de enfriamiento lento de los metales, modelado por (Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. 1953). A partir de este procedimiento (Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. 1983), resuelven un problema de optimización combinatoria que aparece en el diseño de circuitos impresos.

La Búsqueda Tabú (Tabu Search - TS) consiste, en su forma más elemental, en un algoritmo de búsqueda por entornos con una lista donde se almacenan las posiciones visitadas más recientemente. Se dota al algoritmo de una estructura en forma de lista circular (lista tabú) de tamaño L, donde se almacenan las posiciones visitadas más recientemente, prohibiendo que puedan ser revisitadas mientras se encuentran en ella, a menos que se cumpla un criterio de aspiración (usualmente es que se mejore el valor previo de la función objetivo). Con esta estrategia, se intenta desarrollar una exploración lo más exhaustiva posible del espacio de soluciones (Glover, F. 1990).

Un Algoritmo Genético (Genetic Algorithm – GA) es un procedimiento basado en la analogía con la evolución de los seres vivos. La premisa que subyace a este tipo de enfoques es, que se pueden encontrar soluciones aproximadas a problemas de gran complejidad computacional mediante un procedimiento de evolución simulada matemáticamente en un ordenador (Holland, J. H. 1975).

Particularizando el estudio al Problema relacionado con el Taller de Flujo Híbrido, en la Tabla 3-5 se puede ver la taxonomía de métodos de resolución heurística propuesta por (Quadt, D. & Kuhn, H. 2007) para dicho taller. Quadt divide los métodos de resolución en óptimos o exactos y Heurísticos. A su vez divide los métodos heurísticos en Holísticos y Por Descomposición. El autor considera que los métodos Holísticos son aquellos que afrontan el problema en su conjunto, sin divisiones o particiones del mismo. Los métodos Por Descomposición dividen el problema en subproblemas. Según el autor cuando se aborda la Programación de la Producción descomponiendo el Problema este puede seguir un enfoque que sea orientado al Trabajo, a la Etapa, o al Problema. Se considera que una descomposición Orientada al Trabajo selecciona en cada instante un Trabajo que es secuenciado en cada etapa hasta completar sus operaciones. Un problema que considera la descomposición orientada a la Etapa plantea en cada ocasión el Problema de Programación de la Producción en una etapa, es decir el Problema de Máquinas en Paralelo, que luego coordina para obtener una solución coherente en su conjunto. En la orientación por etapas los autores comentan tres métodos basados en listas. La orientación basada en el Problema considera en primer lugar el aspecto relacionado con el cálculo del tamaño de lote, posteriormente la carga del taller y finalmente la secuenciación. La orientación al problema divide el procedimiento de resolución en fases, y cada una de ellas aborda un aspecto.

Adicionalmente, algunos autores consideran otros criterios de clasificación. Un criterio bastante extendido, en los métodos aproximados, es el de agrupar los algoritmos en constructivos o de mejora. Se entiende que un algoritmo es constructivo si no se dispone de una solución desde el instante inicial sino que esta se va progresivamente configurando, y en cada iteración se añade uno o varios elementos a la solución final. Por el contrario los

algoritmos de mejora son aquellos en los que se dispone desde el principio de una solución pero en cada iteración de refina y se mejora hasta alcanzar el final del procedimiento.

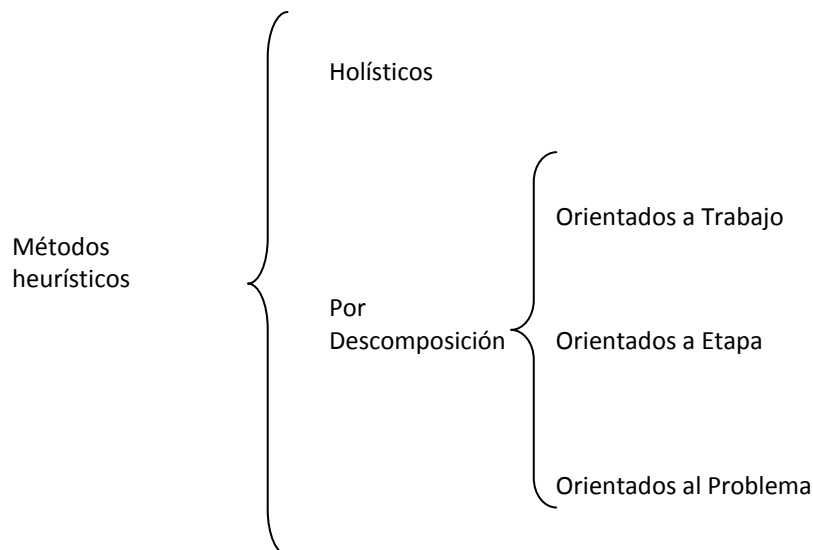


Tabla 3-5. Clasificación de los métodos de resolución de los Problemas de Programación de la Producción en un Taller de Flujo Híbrido (Quadt, D. et al. 2007).

La capacidad de agrupar o clasificar los métodos de resolución del Problema de Programación de la Producción, y en especial el que se desarrolla en el Taller de Flujo Híbrido es fundamental a la hora de analizar las aportaciones de los diferentes autores. Este conocimiento permite un mejor entendimiento y evaluación de la afinidad de las soluciones propuestas por los autores en relación con el problema que se desea abordar en este trabajo.

3.7 Métodos de Resolución de Programación de la Producción para el Taller de Flujo Híbrido: Los Modelos y Algoritmos

Es conocido que en general los problemas de secuenciación presentan una alta complejidad y que es difícil encontrar métodos y algoritmos que permitan encontrar soluciones óptimas en tiempos razonables. Dentro de la problemática relacionada con el Taller de Flujo en todas sus variantes el caso más sencillo es el del Taller de Flujo Estándar. Un breve análisis de su complejidad permitirá valorar mejor el grado de dificultad al que se enfrenta el programador en los casos del **Taller de Flujo SDST** y **HSDST**.

Según (Conway, R. W. et al. 1967) el Taller de Flujo se caracteriza por:

- Para cualquier criterio de optimización basado en una medida de eficacia regular, sólo es necesario considerar secuencias donde la entrada de los trabajos en las máquinas 1 y 2 es idéntica. Luego en general sólo es necesario considerar $(n!)^{m-1}$ secuencias (siendo n el número de trabajos y m el número de recursos o máquinas).
- En el caso concreto de minimización del “*makespan*” (C_{max}), sólo es necesario considerar secuencias donde la entrada de los trabajos en las máquinas $m - 1$ y m es idéntica. En este caso podemos tener la misma secuencia en las máquinas 1 y 2 (la minimización del “*makespan*” también es una medida de eficacia regular) y la misma secuencia en las máquinas $m - 1$ y m , por lo que sólo es necesario considerar $(n!)^{m-2}$ posibles ordenaciones.

De las características anteriores se obtienen implicaciones en cuanto a los problemas para el Taller de Flujo. Para el problema $n/2/F/Medida$, siendo “Medida” cualquier medida de eficacia regular, las secuencias de permutación^{iv} dominan a las secuencias generales, esto es, examinando sólo las secuencias de permutación tenemos la seguridad de poder encontrar la solución óptima. De esta manera, el problema $n/2/F/Medida$ sólo requiere examinar $n!$ posibles soluciones.

En el caso de los problemas $n/2/F/C_{max}$ y $n/3/F/C_{max}$ la situación es similar, con examinar $n!$ secuencias es suficiente. Ahora bien, en el caso general con m máquinas, $m > 3$, las secuencias de permutación ya no dominan a las secuencias generales y se hace necesario evaluar $(n!)^{m-2}$ secuencias. Pese a que a la no predominancia de las secuencias de permutación en el caso general la literatura trata en un porcentaje muy elevado los denominados Talleres de Flujo de Permutación. En **(Rinnooy Kan, A. H. G. 1976)** se demostró que el problema del Taller de Flujo de Permutación es NP-Completo.

Algunas revisiones literarias que permiten realizar un análisis del Problema del Taller de Flujo en su enfoque más tradicional y útil en esta tesis, la minimización del C_{max} , se pueden encontrar en las aportaciones de (Framinan, J. M., Gupta, J. N. D., & Leisten, R. 2004), **(Ruiz, R. & Maroto, C. 2005a)** y **(Hejazi, S. R. & Saghafian, S. 2005)**. **(Framinan, J. M. et al. 2004)** en las

^{iv} Una Secuencia de Permutación es aquella secuencia en la que todos los trabajos siguen el mismo orden de entrada en los recursos, es decir, el orden de proceso de los trabajos en el recurso 1 es el mismo que en el recurso 2, y así hasta el último recurso.

cuales se realiza una revisión del Problema de minimización del Cmax en un Taller de Flujo de Permutación, tanto de procedimiento exactos como aproximados. En **(Ruiz, R. et al. 2005a)** se realiza una revisión exhaustiva de las técnicas aproximadas, en concreto metaheurísticas, aplicadas al Problema del Taller de Flujo de Permutación con el criterio de minimización del Cmax o Makespan. Los autores evalúan los resultados de veinticinco métodos usando el conjunto de datos estándar de Taillard. En **(Hejazi, S. R. et al. 2005)** también se realiza una revisión en profundidad de un conjunto más amplio de referencia, ya que consideran todas las técnicas, exactas y aproximadas, para el problema de minimización del Cmax en el Taller de Flujo, sin restringir al caso de permutación.

En los tres siguientes apartados se realizará una revisión de las aportaciones más relevantes referidas al Taller de Flujo con SDST, el Taller de Flujo Híbrido, y el Taller de Flujo Híbrido con SDST.

3.7.1 El Caso del Taller de Flujo con Tiempos de Cambio de Partida Dependientes de la Secuencia.

El taller de Flujo con SDST se puede considerar intuitivamente como un caso mucho más complejo y difícil de resolver que el Taller de Flujo estándar. De hecho **(Gupta, J. N. D. 1986)**, demostró que el taller SDST es NP-Completo incluso en el caso de una sola máquina. Adicionalmente, Los autores de **(Gupta, J. N. D. & Darrow, W. P. 1986)** demostraron, en el caso de 2 máquinas, y cuando sólo existen Tiempos de Cambio de Partida en una máquina (sea la primera o la segunda), que el SDST flowshop también es NP-Completo. También en este mismo trabajo los autores demostraron que las secuencias de permutación ya no dominan en ningún caso a las secuencias generales cuando el criterio de optimización es el Cmax. Todo esto indica que el taller SDST es mucho más complejo que el taller estándar y si además tenemos en cuenta que muchas de las propiedades que se han investigado para el taller estándar no se cumplen para el taller SDST, tendremos un nuevo problema, que aunque relacionado con el taller estándar, resulta notablemente más difícil.

Por lo que se refiere a la notación, utilizando la nomenclatura de **(Conway, R. W. et al. 1967)** es difícil identificar por completo el problema del Taller de Flujo con SDST. Una posibilidad, que se basa parcialmente en la propuesta de los mencionados autores, es **n/m/F, setup-SD/"Medida"**, donde "F" indica que se trata de un Taller de Flujo, "setup-SD" que existen Tiempos de Cambio Dependientes de la Secuencia. En el caso, en el que el problema tratase Costes de Cambio Dependientes de la Secuencia se puede sustituir "setup-SD" por "setupC-SD".

La dificultad en el enfoque del Taller de Flujo SDST no sólo se cimenta en aportaciones teóricas sino que existen algunas aportaciones que bajo un prisma práctico, basado en simulaciones, corroboran esta status. En **(Wilbrecht, J. K. & Prescott, W. B. 1969)** compararon un total de siete reglas de prioridad en un problema minimización del C_{max} en un taller general con SDST, los resultados indicaron una clara ventaja de las reglas que consideraban explícitamente los Tiempos de Cambio de Partida. Por su parte **(Kim, S. C. & Bobrowski, P. M. 1994)** evaluaron cuatro reglas de prioridad también para el mismo problema, llegando a la misma conclusión.

Como consecuencia de lo indicado anteriormente se puede subrayar que es necesario tomar en consideración de forma explícita los Tiempos de Cambio de Partida en aras de obtener resultados eficaces.

A continuación se repasan algunas de las aportaciones más interesantes que se han encontrado en la literatura. Es interesante iniciar este repaso destacando brevemente algunas revisiones literarias interesantes en el ámbito de los Problemas de Programación de la Producción con Tiempos o Costes de Cambio de Partida. En el trabajo de **(Yang, W. H. & Liao, C. J. 1999)** se proporciona una visión actual del problema estático. En **(Zhu, X. Y. & Wilhelm, W. E. 2006)** se incorpora al análisis la influencia de los tamaños de lote. Sin embargo, son **(Allahverdi, A., Ng, C. T., Cheng, T. C. E., & Kovalyov, M. Y. 2008)** quienes como continuación del trabajo de **(Cheng, T. C. E., Gupta, J. N. D., & Wang, G. Q. 2000)** proporcionan la revisión literaria que constituye el análisis más actualizado y reconocido que se ha localizado. En este trabajo se establece una taxonomía para el problema de la Programación de la Producción cuando existen Tiempos o Costes de cambio de Partida en los talleres más conocidos, y se realiza un exhaustivo análisis. Los autores de **(Allahverdi, A. & Soroush, H. M. 2008)** realizan una buena presentación del problema relacionado con los Tiempos o Coste de Cambio de Partida aunque no se ajusta exclusivamente a un Taller de Flujo.

Aunque la mayoría de las aportaciones aplican procedimientos heurísticos algunos, aplicados a casos sencillos, aplican técnicas optimizadoras. Se dispone de un resumen en la Tabla 3-7 y Tabla 3-8.

Basado en Programación Dinámica (PD) **(Corwin, B. D. & Esogbue, A. O. 1974)** plantean un *algoritmo optimizador* para el caso de Taller de Flujo y minimización del C_{max} en el que los Tiempos de Ajuste son Separables y Dependientes de la Secuencia en una de las máquinas. Sin embargo el procedimiento solo es aplicable a problemas con quince trabajos como máximo.

En **(Uskup, E. & Smith, S. B. 1975)** se desarrolló un *algoritmo optimizador* de Ramificación y Poda (Branch and Bound-B&B) para el mismo problema de dos máquinas pero con la existencia de fechas de finalización. El criterio de optimización considerado es la minimización

de los *costes de Cambio de Partida* donde éstos son directamente proporcionales a los Tiempos de Cambio. El algoritmo consigue resolver problemas de hasta 30 trabajos, pero únicamente en el caso de dos máquinas. La aplicación de los métodos de B&B a problemas con SDST no se ha limitado solamente al Taller de Flujo.

<i>Referencias</i>	<i>Medida Eficiencia</i>	<i>Técnica</i>
(Corwin, B. D. et al. 1974)	Cmax	Programación Dinámica
(Uskup, E. et al. 1975)	Costes de Tiempo de cambio de Partida	Ramificación Y Poda
(Gupta, J. N. D. 1975) (Gupta, J. N. D. 1979)	Coste Oportunidad	Búsqueda Lexicográfica
(Gupta, J. N. D. et al. 1986)	Cmax	Enfoque TSP
(Srikar, B. N. & Ghosh, S. 1986)	Cmax	MLP usando agregación de trabajos
(Szwarc, W. & Gupta, J. N. D. 1987)	Cmax	Algoritmo basado en Johnson
(Stafford, E. F. & Tseng, F. T. 1990), (Tseng, F. T. & Stafford, E. F. 2001), (Stafford, E. F. & Tseng, F. T. 2002)	Cmax	Programación Lineal Entera Mixta
(Hwang, H. C. & Sun, J. U. 1997)	Cmax	Programación Dinámica
(Rios-Mercado, R. Z. & Bard, J. F. 1998a), (Rios-Mercado, R. Z. & Bard, J. F. 1998b)	Cmax	Ramificación y Poda
(Sonmez, A. I. & Baykasoglu, A. 1998)	$\sum w_j T_j$	Programación Dinámica
(Allahverdi, A. & Aldowaisan, T. 2001)	Suma Tiempos de Finalización	Criterio Optimizador
(Sun, J. U. & Hwang, H. C. 2001)	Cmax	Programación Dinámica
(Tang, L. X., Luh, P. B., Liu, J. Y., & Fang, L. 2002)	Continuidad del Proceso	Programación Dinámica basada en Relajación Lagrangiana
(Rios-Mercado, R. Z. & Bard, J. F. 2003)	Cmax	Programación Lineal Entera Mixta

Tabla 3-6. Resumen contribuciones al problema del Taller de Flujo con SDST con soluciones óptimas (elaboración propia).

<i>Referencias</i>	<i>Medida Eficiencia</i>	<i>Técnica</i>
(Gupta, J. N. D. et al. 1986)	Cmax	Algoritmo basado en Johnson
(Gupta, J. N. D. 1986)	Cmax	Enfoque TSP
(Szwarc, W. et al. 1987)	Cmax	Heurística
(Simons, J. V. 1992)	Cmax	Heurística
(Das, S. R., Gupta, J. N. D., & Khumawala, B. M. 1995) (Das, S. R., Gupta, J. N. D., & Khumawala, B. M. 2004)	Cmax	Basado en Índices de ahorro
(Kim, Y. D., Lim, H. G., & Park, M. W. 1996)	Cmax	NEH+SA
(Parthasarathy, S. & Rajendran, C. 1997)	Retraso máximo ponderado	SA
(Rajendran, C. & Ziegler, H. 1997)	$\sum f_j$	Heurística
(Rios-Mercado, R. Z. et al. 1998b)	Cmax	NEH y GRASP
(Hwang, H. C. & Sun, J. U. 1998)	Cmax	Algoritmo Genético
(Rios-Mercado, R. Z. & Bard, J. F. 1999)	Cmax	Heurística TSP
(Norman, B. A. 1999)	Cmax	Búsqueda Tabú
(Allahverdi, A. et al. 2001)	Suma Tiempos de Finalización	Heurística
(Sun, J. U. et al. 2001)	Cmax.	Algoritmo Genético
(Andres, C., Vicens, E., & Lario F.C. 2001)	Cmax	Recocido Simulado
(Tang, L. X. et al. 2002)	Continuidad del Proceso	Heurística basada en Relajación Lagrangiana
(Rajendran, C. & Ziegler, H. 2003)	$\sum w_j T_j + \sum F_j$	Algoritmo Voraz
(Maddux III, H. S. & Gupta, J. N. D. 2003)	Cmax.	Heurística

Tabla 3-7. Resumen contribuciones al problema del Taller de Flujo con SDST con Soluciones Aproximadas (elaboración propia).

Referencias	Medida Eficiencia	Técnica
(Ruiz, R., Maroto, C., & Alcaraz, J. 2005b)	Cmax.	Algoritmo Genético
(Andres, C., Tomás, J. V., García-Sabater, J. P., & Miralles, C. 2005)	Cmax+ n-1 \sum Ti.	Algoritmo Genético Multiobjetivo
(Tseng, F. T., Gupta, J. N. D., & Stafford, E. F. 2006)	Cmax.	Heurística
(Yip, Y., Cheng, C. Y., & Low, C. 2006)	Cmax	Búsqueda Tabú a partir de NEH
(Ruiz, R. & Stutzle, T. 2008b)	Cmax. $\sum w_j T_j$	Voráz+Búsqueda local

Tabla 3-8. Resumen contribuciones al problema del Taller de Flujo con SDST con Soluciones Aproximadas (continuación de Tabla 3-7)(elaboración propia).

En **(Gupta, J. N. D. 1975)** se presentó un *algoritmo de optimización* mediante Búsqueda Lexicográfica, que actualizó en **(Gupta, J. N. D. 1979)**. El método presentado es muy general y permite incorporar numerosas restricciones y condiciones en el problema, entre ellas los SDST. El criterio que se busca optimizar en este trabajo es una compleja función del *coste total de oportunidad*, de la cual se puede derivar el Cmax.

En **(Gupta, J. N. D. et al. 1986)** plantean una representación del problema mediante dos subproblemas de Viajante de Comercio (Traveling Sales Problem-TSP). Definiendo cotas inferiores y asegurando el *óptimo* de una secuencia de permutación si su makespan es igual al mayor de los recorridos de los dos subproblemas de Viajante de Comercio. Sin embargo, debido a la dificultad en obtener una solución exacta al TSP para ejemplos grandes se proponen *cuatro heurísticas*, las dos primeras basadas en el algoritmo de (Johnson, S. M. 1954) y la tercera y cuarta a partir de un post-procesamiento mediante intercambios desde las soluciones obtenidas por las dos anteriores. Los resultados experimentales confirman que las dos últimas son mejores siendo la tercera la que mejor comportamiento general presentan.

En **(Gupta, J. N. D. 1986)** se proponen cuatro teoremas para resolver el problema cuando no se permiten almacenamientos entre máquinas. Entre ellos, se demuestra que, en general cualquier problema de secuenciación en Taller de Flujo con SDST es NP-duro y, que en el caso de no permitir almacenamientos intermedios solo es necesario considerar secuencias basadas en permutaciones de los trabajos. En este caso, una *solución óptima* al problema consiste en una permutación de los trabajos. El autor propone una manera *heurística* de obtener una permutación mediante un algoritmo basado en el Problema del Viajante de Comercio sin mas

que definir una distancia entre los trabajos basada en los Tiempos de Proceso y ajuste de los mismos. Los resultados computacionales muestran que el algoritmo encuentra soluciones óptimas en una alta proporción de casos, extendiendo el estudio al caso de almacenamiento infinito entre las máquinas.

En **(Srikar, B. N. & Ghosh, S. 1986)** a partir del *modelo optimizador* basado en Programación Lineal Mixta (Mix Linear Programming-MLP) para Taller de Flujo de Permutación, con el objetivo de minimizar el Cmax, se propone un método de agregación de los trabajos como forma de mejorar la rapidez de cálculo, aunque no realizan experimentos en este último caso.

En **(Szwarc, W. et al. 1987)** proporcionan un *algoritmo óptimo* para el caso $m = 2$ y una *extensión heurística* para el caso general con m máquinas. En ambos casos para el taller de Flujo con SDST con la intención de minimizar el Cmax.

En **(Stafford, E. F. et al. 1990)** realizaron correcciones al modelo de Srikar y Ghosh de 1986 y propusieron también modelos para el problema Taller de Flujo de Permutación con SDSTF y objetivo minimizar el Cmax, entre otros. En este caso, el tamaño máximo de problema que se pudo resolver fue de 7 trabajos y 5 máquinas, necesitando más de seis horas de tiempo de CPU en un ordenador tipo PC/AT de la época. Once años después **(Tseng, F. T. et al. 2001)** propusieron dos *modelos MILP* para minimizar el Cmax en el Taller de Flujo SDST y el Taller de Flujo de Permutación SDST. En todo caso los autores pudieron resolver problemas de hasta 9 trabajos y 9 máquinas en aproximadamente cinco minutos de tiempo de CPU en un ordenador tipo PC/AT con un procesador de 800 MHz. Posteriormente, en **(Stafford, E. F. et al. 2002)** presentaron otros dos *modelos optimizadores* similares, con algunas mejoras sobre los anteriores, sin conseguir resolver problemas de un tamaño superior. Y más recientemente, **(Tseng, F. T. et al. 2006)** desarrollaron un algoritmo para resolver de forma *aproximada* el problema de minimización del Cmax en un Taller de Flujo de Permutación con SDST. El algoritmo propuesto se basa en un sistema de penalización sobre las secuencias obtenidas y se compara con una heurística basada en los Índices de Ahorro.

En la propuesta de **(Simons, J. V. 1992)** se presentó una *heurística* general. En concreto cuatro métodos, dos de ellos son codificaciones de los métodos MINIT y MICOT de **(Gupta, J. N. D. 1972)**, con adaptaciones para tener en cuenta la existencia de SDST. La tercera y cuarta heurísticas parten del trabajo de **(Stinson, J. P. & Smith, A. W. 1982)**. La tercera heurística, llamada TOTAL, utiliza una matriz de costes que la forman la suma de todos los Tiempos de Proceso y de Cambio de Partida para los trabajos. En la cuarta heurística, llamada SETUP, la matriz de costes sólo contiene la suma de los Tiempos de Cambio de Partida. Tras la evaluación, las heurísticas TOTAL y SETUP dominaron holgadamente a las adaptaciones de los métodos MINIT y MICOT.

Los autores de **(Das, S. R. et al. 1995)** desarrollaron una *heurística* basada en unos Índices de Ahorro. Estos índices seleccionan un trabajo no secuenciado para asignarlo a una posición de una secuencia parcialmente construida. El Índice de Ahorro considera tres conceptos: ahorro en el Tiempo de Cambio de Partida en la secuencia, ahorro en la duración de la secuencia a partir de los Tiempos de Proceso y ahorro en los Tiempos de Finalización de los trabajos. La eficacia de esta heurística se comprobó resolviendo un grupo de 240 problemas para los que se conocía la solución óptima y se calculó la desviación de la heurística con respecto a estos óptimos. En estos casos la heurística nunca se desvió más del 6%. (Das, S. R. et al. 2004)se realiza una versión actualizada.

En **(Kim, Y. D. et al. 1996)** se presentó un extenso estudio sobre un problema real de producción de tarjetas con circuitos impresos (PCB). El sistema productivo es peculiar, aunque se asimila en parte a un Taller de Flujo con SDST, aunque tiene otros aspectos más concretos. Los autores proponen *varias heurísticas*, para resolver el problema, entre las que están una adaptación de la heurística NEH, algoritmos tipo Recocido Simulado (Simulated Annealing-**SA**), Búsqueda Tabú (BT) y de Búsqueda Local Descendente. De entre todos, los algoritmos tipo SA parecen dominar a los demás.

En **(Parthasarathy, S. et al. 1997)** se plantea un *método metaheurístico* basado en el Recocido Simulado que tiene en consideración las restricciones temporales de los trabajos. El objetivo es la minimización del retraso máximo ponderado teniendo en cuenta las fechas de entrega límite de los trabajos. La generación de una solución vecina se realiza mediante la re inserción de un Trabajo. Los resultados de un problema real de una fábrica con 95 trabajos se representan con un tiempo de cálculo de 11.100 segundos en un ordenador HP-9000.

El problema del Taller de Flujo con SDST con el objetivo de minimizar el Tiempo de Flujo Total Ponderado fue tratado por **(Rajendran, C. et al. 1997)** donde se propuso una heurística. Los autores presentaron un esquema para mejorar la calidad de la heurística propuesta.

En **(Hwang, H. C. et al. 1997)** se trata el problema de una empresa de fabricación de camiones donde todos los trabajos necesitan realizar dos operaciones. Los tiempos de proceso de la primera máquina dominan a los de la segunda. Así mismo, el Tiempo de Cambio e Partida de los trabajos en la primera máquina depende de los dos próximos trabajos. Los autores convirtieron el problema en un $n/2/F$, setup, Preempt/ C_{max} , y propusieron un modelo de Programación Dinámica para resolverlo. En **(Hwang, H. C. et al. 1998)** se consideran el anterior problema abordándolo con un Algoritmo Genético.

Los autores de **(Rios-Mercado, R. Z. et al. 1998a)** desarrollaron un *algoritmo exacto* de tipo B&B para la minimización del C_{max} . Este algoritmo no permitió resolver problemas con más de

8 trabajos y 6 máquinas en menos de una hora de tiempo de CPU en una estación de trabajo tipo Sun. Posteriormente, en **(Rios-Mercado, R. Z. et al. 1998b)** los autores propusieron *un nuevo algoritmo de B&B* con el que mejoraron ligeramente el método anterior y consiguieron resolver de forma heurística problemas de hasta 10 trabajos con una diferencia respecto del óptimo del 1%. También se propuso *un método heurístico y otro metaheurístico*. El método heurístico, llamado NEHT-RMB, se basa en la adaptación del método de inserción NEH de **(Nawaz, M., Enscore, E. E., & Ham, I. 1983)** para la consideración de los Tiempos de Cambio de Partida, y las mejoras de Taillard. La propuesta metaheurística se basa en el método GRASP (*"Greedy Adaptive Search Procedure"*). Los dos métodos aproximados, resultaron mejores, y especialmente el método GRASP. **(Rios-Mercado, R. Z. et al. 1999)** trató el problema de minimización del Cmax en el Taller de Flujo con SDST. Los autores propusieron una heurística que transformaba una instancia del problema en una instancia del TSP, introduciendo una función de coste que penalizaba los Setups grandes y las malas secuencias de un programa dado. También recientemente, **(Rios-Mercado, R. Z. et al. 2003)** abordan la representación poliédrica de dos problemas *de MILP* sobre el Taller de Flujo con SDST, el problema del viajante de comercio y el problema basado en la contribución de **(Srikar, B. N. et al. 1986)**.

En **(Sonmez, A. I. et al. 1998)** se desarrolló un modelo de Programación Dinámica para el problema de $n/m/F, \text{setup}/\sum w_j T_j$, que se aplica a una factoría de fabricación de plásticos. El resultado es una ventaja en la reducción de tiempos de computación cuando el tamaño de los trabajos aumenta mucho frente a una ligera mejora cuando aumenta el número de máquinas.

Para un taller SDST donde existe una capacidad de almacenamiento limitada entre las máquinas **(Norman, B. A. 1999)** propuso varios métodos para un problema particular, con intención de reducir el Cmax. En este caso el autor utilizó *métodos metaheurístico* de tipo Búsqueda Tabú para el problema.

En **(Sun, J. U. et al. 2001)** se afrontó el problema $n/2/F, \text{setup}/C_{\max}$, donde los Tiempos de Cambio de Partida se daban solo en la segunda máquina y éstos dependían de los k trabajos inmediatamente anteriores ($k > 1$). Se ofrecieron soluciones mediante Programación Dinámica y Algoritmos Genéticos.

Los autores de **(Allahverdi, A. et al. 2001)** resolvieron un tipo específico de Taller de Flujo SDST donde los trabajos no pueden esperar entre las máquinas y donde el criterio de optimización considerado es la suma de los tiempos de finalización de los trabajos. Los autores discuten una serie de métodos que proporcionan la *solución óptima* en el caso en que $m = 2$ y cuando se cumplen una serie de condiciones especiales. También *proponen cinco heurísticas* diferentes para el problema de m máquinas. En este caso los Tiempos de Cambio de Partida que se consideran son del tipo aditivo.

En el trabajo de **(Andres, C. et al. 2001)** se presenta un algoritmo de Recocido Simulado para minimizar el C_{max} . En **(Andres, C. et al. 2005)** trataron el problema $n/m/F, setup/C_{max} + n - 1 \sum T_i$. Se propuso un Algoritmo Genético Multiobjetivo.

En **(Maddux III, H. S. et al. 2003)** enfocan el problema $n/2/F, Setup/C_{max}$ con almacenes con capacidad nula entre máquinas, y con trabajos con una segunda alternativa opcional. Presentaron una heurística para resolver el problema y demostraron la existencia de un límite inferior.

Los autores de **(Tang, L. X. et al. 2002)** analizaron el proceso de fundición en la industria del acero. Este tipo de industria tal y como lo plantean los autores se puede considerar un Taller de Flujo donde existen los Tiempos de Ajuste entre colada y colada son importantes. Los autores plantean un modelo de matemático sobre el cual aplican una relajación Lagrangiana. A partir del modelado proponen una solución *optimizadora* mediante Programación Dinámica y una heurística.

En **(Rajendran, C. et al. 2003)** se estudió el problema $n/m/F, setup/$ con una combinación de los objetivos **(Rajendran, C. et al. 1997)** y **(Sonmez, A. I. et al. 1998)**, es decir el Tiempo de Flujo y Retraso Total Ponderado. Los autores proponen heurísticas y las compara con algunas ya existentes y la regla random, y un algoritmo Voraz de búsqueda local.

En el trabajo de **(Ruiz, R. et al. 2005b)** se propusieron dos Algoritmos Genéticos (AG) para $n/m/F, setup/C_{max}$ y mostraron que las heurísticas propuestas mejoraban las de (Rios-Mercado, R. Z. et al. 1999) y otros autores. En **(Ruiz, R. et al. 2008b)** se presenta un algoritmo iterativo Voraz (Greedy) con búsqueda local que superar a sus propuestas de **(Ruiz, R. et al. 2005b)**. También en **(Ruiz, R. et al. 2008b)** proponen dos simples búsquedas locales basadas en un algoritmo iterativo Voraz. Para el problema $n/m/F, setup_d / \sum w_j T_j$. Los autores superan a los procedimientos de (Rajendran, C. et al. 2003) y algunas heurísticas anteriores.

En **(Yip, Y. et al. 2006)** se propone un algoritmo metaheurístico basado en Búsqueda Tabú que parte de una solución inicial obtenida con la heurística NEH. Con ello abordan el problema del Taller de Flujo con tiempo de establecimiento y ajuste separables.

El Taller de Flujo con Tiempos de Cambio de Partida Dependientes de la Secuencia se aproxima bastante al tipo de taller que se abordar en esta tesis y por lo tanto algunos de los procedimientos aplicados pueden ser útiles en la fase aplicación. Es de especial interés, en este apartado, el análisis que los diferentes autores realizan del aspecto concreto de la influencia de los Tiempos de Cambio de Partida, o en su caso de Ajuste, sobre la eficacia de los métodos propuestos. Habría que destacar el uso mayoritario de procedimientos aproximados para

problema de tamaño medio-grande y la confirmación de que las propuesta que tiene en cuenta de forma específica los Tiempos de Cambio son, en general, más eficaces que aquellas que no lo tienen. Así mismo, no se puede decir que existe un método predominante.

3.7.1 El Caso del Taller de Flujo Híbrido.

Este tipo de taller se puede identificar con la siguiente codificación: **n/m/F, k-parallel/"Medida"**. Donde los diversos campos han sido ya comentados excepto donde se indica "k-parallel" que significa que el taller es híbrido o flexible con "k" máquinas por etapa.

El problema de Programación de la Producción en un taller de flujo híbrido es especialmente complejo. En **(Gupta, J. N. D. 1988)** se analiza el caso de un taller de flujo con dos etapas aunque sólo existen máquinas múltiples en la primera, siendo el primero en acuñar el término taller de flujo híbrido. Demuestra que este tipo de problemas son NP-completos.

Existen ciertos trabajos que analizan los avances realizados en el ámbito de los talleres de flujo híbrido, también conocidos como taller de flujo flexible o multiprocesador. En **(Riane, F. 1998)** se analiza de forma exclusiva este tipo de talleres repasando los procedimientos propuestos y sus aplicaciones. También **(Vignier A. et al. 1995)** analiza el problema de Programación de la Producción en un Taller de Flujo Híbrido. Y en su trabajo más reciente **(Vignier, A., Billaut, J. C., & Proust, C. 1999)** analizan los problemas de flujo híbrido. Su trabajo se divide en dos partes. La primera parte se dedica a analizar los talleres de flujo híbrido con dos etapas. En la segunda parte se analiza el caso general con k etapas. Los autores analizan el conjunto de problemas que aun no han sido resueltos. En el trabajo de **(Kis, T. & Pesch, E. 2005)** se proporciona una revisión sobre los diferentes métodos exactos que se han desarrollado para el caso del taller de flujo híbrido en relación con las medidas de eficiencia C_{max} o makespan y tiempo medio de flujo. El trabajo se centra en las técnicas de ramificación y Poda y Propagación de Restricciones. En **(Quadt, D. et al. 2007)** se aporta una interesante clasificación de los tipos de solución que se ha aplicado al caso que se está analizando.

Aunque la mayoría de las aportaciones aplican procedimientos heurísticos algunos, aplicados a casos sencillos, aplican técnicas optimizadoras. Se dispone de un resumen en la Tabla 3-9, Tabla 3-10 y Tabla 3-11.

Uno de los primeros artículos que tratan los Talleres de Flujo híbridos se debe a **(Shen, V. Y. & Chen, Y. E. 1972)** los cuales proponen una *estrategia heurística* de Programación para el taller de dos etapas, con el objetivo de minimizar C_{max} basada en una extensión del algoritmo de **(Johnson, S. M. 1954)**.

En **(Salvador, M. S. 1973)** se presenta un procedimiento de separación y evaluación basado en Programación Dinámica, para la optimización del makespan en un Taller de Flujo Híbrido de k etapas y un número diferente de máquinas por etapa.

<i>Referencias</i>	<i>Medida Eficiencia</i>	<i>Técnica</i>
(Salvador, M. S. 1973)	Cmax	Programación Dinámica
(Gupta, J. N. D. & Tunc, E. A. 1991)	Cmax	Ramificación y Poda
(Brah, S. A. & Hunsucker, J. L. 1991)	Cmax	Ramificación y Poda
(Rajendran, C. & Chaudhuri, D. 1992)	Cmax	Ramificación y Poda
(Lee, C. Y., Cheng, T. C. E., & Lin, B. M. T. 1993)	Cmax	Ramificación y Poda
(Vignier A., Billaut, J. C., & Proust, C. 1996a)	Tmax	Procedimiento basado Ramificación y Poda
(Riane, F. et al. 1998)	Cmax	Ramificación y Poda y Programación Dinámica

Tabla 3-9. Resumen contribuciones al problema del Taller de Flujo Híbrido con Soluciones Exactas (elaboración propia).

En el trabajo de **(Paul, R. J. 1979)** se plantea una simulación de *reglas heurísticas* de un problema del Taller de Flujo Híbrido de dos etapas a partir de un caso real de una empresa de fabricación de envases. La conclusión del estudio es que la regla SPT es la que mejor funciona entre todas las estudiadas.

Dentro de una empresa real de fabricación de cableado, **(Narasimhan, S. L. & Panwalkar, S. S. 1984)** proponen la *heurística* CMD (Desviación acumulada mínima) para un Taller de Flujo Híbrido de dos etapas con el objetivo de minimizar la suma de los tiempos muertos y el trabajo en curso en la última etapa. La regla CMD es superior a otras heurísticas sencillas como SPT y LPT.

<i>Referencias</i>	<i>Medida Eficiencia</i>	<i>Técnica</i>
(Shen, V. Y. et al. 1972)	Cmax	Heurística basada en Johnson
(Paul, R. J. 1979)	Cmax	Simulación Reglas
(Narasimhan, S. L. & Panwalkar, S. S. 1984)	Suma Tiempos Muertos + WIP en última etapa	Regla CMD
(Gupta, J. N. D. 1988)	Cmax	Heurística basada en Índices
(Wittrock, R. J. 1988)	$C_{max} + \sum w_i$	Heurística basada en Balance de cargas del cuello de botella
(Sriskandarajah, C. & Sethi, S. P. 1989)	Cmax	Heurística de descomposición basada en Johnson
(Gupta, J. N. D. & Tunc, E. A. 1991),	Cmax	Heurística
(Lee, C. Y. & Vairaktarakis, G. L. 1994)	Cmax	Heurístico de lista FAM y LBM
(Hunsucker, J. L. & Shah, J. R. 1994)	Trabajos Retrasados	Simulación Reglas
(Brah, S. A. 1996)	Tmax	Simulación Reglas
(Vignier A., Billaut, J. C., Proust, C., & Tkindt, V. 1996b)	Lmax	Heurística tres fases
(Guinet, A. G. P. & Solomon, M. M. 1996)	Tmax	Heurística de dos fases basada en MLP.
(Guinet, A., Solomon, M. M., Kedia, P. K., & Dussauchoy, A. 1996)	Cmax	Varias Heurísticas
(Haouari, M. & M'Hallah, R. 1997)	Cmax	Dos heurísticas de dos fases basadas en Recocido Simulado y Búsqueda Tabú.
(Portmann, M. C., Vignier, A., Dardilhac, D., & Dezalay, D. 1998)	Cmax	Heurística basada en Ramificación y Poda con Algoritmo Genético

Tabla 3-10. Resumen contribuciones al problema del Taller de Flujo Híbrido con Soluciones Aproximadas (elaboración propia).

Para el caso de minimización del C_{max} (**Gupta, J. N. D. 1988**) analiza un taller de flujo con dos etapas aunque sólo existen máquinas múltiples en la primera. Demuestra que la resolución de este tipo de problema implica minimizar el tiempo total de inactividad en la última etapa, como en el Taller de Flujo simple de dos etapas. Plantea una *heurística* de lista basada en índices con el objetivo que la máquina asignada al trabajo en la primera etapa quede libre lo más tarde posible (regla basada en el tiempo de proceso más largo). Este procedimiento es extendido al caso de máquinas múltiples en la segunda etapa y una en la primera en (**Gupta, J. N. D. et al. 1991**), probándose que ofrece *resultados óptimos* en el caso en el que el número de máquinas en la segunda etapa sea mayor que el de trabajos. En caso contrario, se plantean dos *heurísticas* y un algoritmo de exacto de Ramificación y Poda para abordar problemas de tamaño pequeño.

En (**Wittrock, R. J. 1988**) se analiza un problema de Taller de Flujo Híbrido donde los trabajos pueden no pasar por todas las etapas. Se plantea un doble objetivo, en primer lugar minimizar el C_{max} y en segundo lugar minimizar el tiempo de espera en colas. El tiempo necesario para realizar un trabajo depende del recurso utilizado y del trabajo. Se proponen *unas heurísticas* en la que primero se asignan las máquinas en cada etapa tratando de minimizar en cada etapa la carga de la máquina cuello de botella, después se secuencian los trabajos en cada máquina y, por último, se calcula la fecha de inicio de cada trabajo para equilibrar la carga de las máquinas.

Para el problema de minimización del C_{max} en un Taller con máquinas múltiples (**Sriskandarajah, C. et al. 1989**) tratan de resolver en la primera etapa y una máquina en la segunda para, posteriormente, abordar el problema con el mismo número de máquinas en las dos etapas. Utilizan una variante del algoritmo de (**Johnson, S. M. 1954**) aplicada a la descomposición del problema en M -subproblemas de taller de flujo. Los mismos autores señalan que en el caso sin espera entre etapas se puede utilizar un algoritmo de resolución basado en el Problema del Viajante de Comercio. Además señalan que, en el caso de existencia de tiempos de montaje y/o desmontaje, si éstos no dependen de la secuencia se puede usar el resultado de (**Sule, D. R. & Huang, K. Y. 1983**).

Los autores de (**Brah, S. A. et al. 1991**) proponen un procedimiento *exacto* de Ramificación y Poda para minimizar el C_{max} para el Taller de Flujo Híbrido general. En el árbol de búsqueda cada nodo representa a un trabajo y depende de la forma se estará tomando una decisión de secuenciación o asignación. Definen dos tipos de cotas mínimas. Una basada en los trabajos y otra basada en las máquinas. El cálculo de estas cotas se obtiene generalizando los razonamientos habituales de taller de flujo al caso de los problemas de Taller de Flujo Híbrido. Los autores comparan el rendimiento del procedimiento con una serie de programas generados mediante un sencillo procedimiento de creación de programas sin retraso, obteniendo mejores resultados, sin embargo se observa como el tiempo de cálculo crece

cuando se resuelven problemas de un gran número de etapas (12 horas en un IBM XT para un problema de 5 etapas).

<i>Referencias</i>	<i>Medida Eficiencia</i>	<i>Técnica</i>
(Nowicki, E. & Smutnicki, C. 1998)	Cmax	Heurística basada en Búsqueda Tabú
(Brah, S. A. & Loo, L. L. 1999)	Cmax y Fmedio	Simulación Reglas
(Gupta, J. N. D. & Tunc, E. A. 1998)	Nº Trabajos Retrasados	Heurística basado en Moore
(Soewandi, H. & Elmaghraby, S. E. 2001)	Cmax	Heurística
(Soewandi, H. & Elmaghraby, S. E. 2003)	Cmax	Heurística
(Wardono, B. & Fathi, Y. 2004)	Cmax	Búsqueda Tabú
(Serifoglu, F. S. & Ulusoy, G. 2004)	Cmax	Algoritmo Genético
(Allahverdi, A. & Al-Anzi, F. S. 2006)	Cmax medio	Heurística tres fases
(Jin, Z. H., Yang, Z., & Ito, T. 2006)	Cmax	Heurística en dos fases basada en Recocido Simulado
(Ying, K. C. & Lin, S. W. 2006)	Cmax	Colonia Hormigas
(Gao, J., Gen, M., & Sun, L. Y. 2006)	Cmax, Carga máxima y Carga Total de las Máquinas	Algoritmo Genético
(Tavakkoli-Moghaddam, R., Rahimi-Vahed, A., & Mirzaei, A. H. 2007)	$\sum w_i * C_i + \sum w_i * T_i$	Algoritmo Inmune HMOIA
(Yang, T., Kuo, Y., & Cho, C. 2007)	Tmax	Algoritmo Genético

Tabla 3-11. Resumen contribuciones al problema del Taller de Flujo Híbrido con Soluciones Aproximadas (continuación de Tabla 3-10).

En **(Rajendran, C. & Chaudhuri, D. 1992)** se tratan dos problemas, uno de máquinas paralelas y otro para el Taller de Flujo Híbrido de dos etapas. Para ambos, se propone un procedimiento *exacto* de Ramificación y Poda similar al de en **(Brah, S. A. et al. 1991)**. La diferencia estriba en que, en este caso, los autores solamente consideran las secuencias de permutación, aunque impliquen la introducción de tiempos de inactividad en las máquinas. El algoritmo propuesto

supera con creces a una heurística de lista basada en la regla de tiempo de proceso más corto, utilizada con fines de comparación.

En el trabajo de **(Lee, C. Y. et al. 1993)** se estudia un problema de Taller de Flujo con tres etapas con intención de minimizar el C_{max} y se demuestra que se trata de un problema NP-Completo. Los autores proponen un procedimiento exacto mediante Ramificación y Poda y tres heurísticas de las cuales calculan los límites. En **(Lee, C. Y. et al. 1994)** se analiza el problema de minimización el C_{max} en el Taller de Flujo Híbrido de dos etapas con igual número de máquinas por etapa, resolviéndolo mediante un algoritmo *heurístico* de lista que asigna los trabajos en la primera etapa mediante un procedimiento denominado FAM (First Available Machine) y los trabajos a la segunda etapa son asignados mediante el procedimiento denominado LBM (Last Busy Machine). Los autores extienden estas ideas al caso de un número de etapas $k > 2$ y demuestran que esta heurística tiene una cota de error tan buena como la de otras heurísticas conocidas. Los autores desarrollan cinco cotas, para diferentes casos, para la heurística comentada que demuestra que no está lejos del óptimo.

Un problema particular de Taller de Flujo Híbrido se aborda en **(Hunsucker, J. L. et al. 1994)**. En este caso consideran que el número de trabajos que en un momento dado hay en el taller está limitado por una constante. Mediante un modelo de simulación estudian el efecto de seis *reglas de despacho* clásicas sobre el número de trabajos retrasados. Los resultados concluyen que el número máximo de trabajos en el sistema influye en el funcionamiento del mismo, siendo la regla SPT (Short Processing Time), la que mejor resultado tiende a dar. El número de máquinas y etapas no es significativo. Experimentos similares utilizando como criterio el retraso medio, ofrecen un buen resultado aplicando la regla FIFO.

En **(Brah, S. A. 1996)** se realiza un análisis de simulación de *reglas heurísticas* basadas en la fecha de entrega en un taller de flujo híbrido. En concreto analiza el impacto que tienen sobre la tardanza media diez reglas heurísticas tomadas de la literatura. Concluye que la tardanza media decrece cuando aumenta el número de etapas del taller en el caso de un gran número de trabajos a secuenciar, aumentando la tardanza con el número de trabajos para una configuración fija del taller. También se observa que el impacto de la regla es menor que el debido a las características del taller. Sin embargo se concluye que las reglas que funcionan mejor son las basadas en una combinación de holgura por trabajo restante y tiempo de trabajo más corto, menor fecha de entrega y menor fecha de entrega modificada. **(Brah, S. A. et al. 1999)** analiza el rendimiento de diversas heurísticas en un Taller de Flujo Híbrido. Analizan cinco de las reglas de despacho que mejor se han comportado en este tipo de talleres para el C_{max} y el tiempo medio de flujo. El estudio tiene en cuenta factores como el número de máquinas múltiples en cada etapa, el número de trabajos y el número de etapas. Los autores concluyen en que la estructura/configuración del problema explica muchos de los resultados

obtenidos pero que las reglas utilizadas tienen influencia y que la heurística de **(Nawaz, M. et al. 1983)** es la que mejor resultado proporciona.

A partir de un modelo mediante Programación Lineal Mixta del problema de Taller de Flujo Híbrido general con fechas de entrega y cuyo objetivo es minimizar la tardanza máxima **(Guinet, A. G. P. et al. 1996)** proponen un *procedimiento heurístico* en dos fases. En la primera se ordenan los trabajos mediante varias heurísticas tales como la regla CDS **(Campbell, H. G., Dudek, R. A., & Smith, M. L. 1970)**, la heurística NEH **(Nawaz, M. et al. 1983)**, la de **(Townsend, W. 1977)** y una heurística de lista que utiliza la mejor regla entre la SPT, EDD y MST (en todas ellas se utiliza el tiempo de proceso en cada etapa dividido por el número de máquinas de la etapa). En la segunda fase, el procedimiento asigna los trabajos a la máquina que minimiza su fecha de finalización en la última máquina o la fecha de finalización en la siguiente etapa. En el estudio se concluye que la heurística NEH es superior a las restantes en la mayoría de los problemas analizados.

En **(Vignier A. et al. 1996b)** se propone un *modelado analítico* y *dos heurísticas* para resolver el problema de Taller de Flujo Híbrido con una máquina en la segunda etapa. En la primera, se secuencian los trabajos en la segunda etapa según su fecha de entrega, condicionando las fechas de entrega en la primera etapa y minimizando el retraso máximo (L_{max}). La segunda heurística se compone de tres fases, y se basa también en la regla de fecha de entrega más temprana para generar una secuencia que definirá unas holguras entre las dos etapas del taller. Estas holguras son tenidas en cuenta para secuenciar los trabajos en la primera etapa. **(Vignier A. et al. 1996a)** estudian el problema de Taller de Flujo Híbrido general con fechas de llegada y fechas de entrega, con el objetivo de minimizar el retraso máximo. Cada máquina tiene un calendario de funcionamiento y se plantea un *algoritmo exacto* basado en el procedimiento de **(Brah, S. A. et al. 1991)**.

En **(Guinet, A., Solomon, M. M., Kedia, P. K., & Dussauchoy, A. 1996)** se plantea otro modelo de *Programación Lineal Mixta* para el problema de minimización del C_{max} con dos etapas. Se proponen varias heurísticas para su resolución.

Los autores de **(Haouari, M. et al. 1997)** analizan el Taller de Flujo Híbrido de dos etapas con el objetivo de minimizar el makespan. Desarrollan dos *métodos aproximados* en dos fases. En la primera se genera una solución mediante un algoritmo de lista basado en la regla de Mayor Trabajo Pendiente. Este programa es mejorado mediante técnicas basadas en Recocido Simulado y Búsqueda Tabú. En ambas las soluciones se codifican mediante la lista usada en la primera fase. De tal manera que se asume que los trabajos se van asignando según se quedan libres las máquinas. Los movimientos se basan en intercambio de trabajos adyacentes o no en la lista. Se realizan un estudio experimental comparado con la heurística propuesta por **(Lee, C.**

Y. et al. 1994). Demostrando que la propuesta arroja resultados mejores cuanto mayor es el tamaño del ejemplo del problema.

En **(Portmann, M. C. et al. 1998)** se analiza y proponen una mejora del procedimiento Ramificación y Poda (B&B) de **(Brah, S. A. et al. 1991)**. A partir de una definición de cotas mejoradas, los autores plantean una *metaheurística* mediante un Algoritmo Genético incluido en el procedimiento de B&B para mejorar en ciertas etapas los valores de la cota superior. Para ello en las etapas del proceso de B&B que ya han sido ejecutadas hasta el nodo se consideran fijas, usando el Algoritmo Genético para mejorar las etapas posteriores. El código de una solución consta de dos partes. En la primera, se representa la asignación de cada trabajo a una de las máquinas de cada etapa mediante una serie de cadenas. En segundo lugar se usa una matriz binaria $M \times M$ por cada etapa donde si $M(i,j)=1$ implica que i precede a j en una máquina, $M(i,j)= -1$ en caso contrario y $M(i,j)=0$ si se ejecutan en máquinas diferentes. Debido a este procedimiento de codificación, es necesario reparar la solución obtenida después de aplicar los operadores de cruce y mutación. Los autores concluyen que el algoritmo que incluye la modificación de la cota superior del procedimiento B&B es netamente superior a la versión que no lo incluye. Por lo que es un buen enfoque a tener en cuenta cuando las cotas en los algoritmos de este tipo no son muy buenas. Este algoritmo tiene en cuenta las decisiones ya tomadas a lo largo de la arborescencia para construir una serie de poblaciones constituidas por soluciones completas.

En **(Nowicki, E. et al. 1998)** se plantea un procedimiento *metaheurístico* mediante Búsqueda Tabú para resolver el Taller de Flujo Híbrido minimizando la fecha de finalización del último trabajo. Para ello, y basándose en una representación del problema mediante grafos y los trabajos que pertenecen al camino crítico del mismo en cada etapa (bloques), definen una serie de movimientos útiles que definen un entorno reducido al problema. Los movimientos tabú prohíben la vuelta del trabajo recién movido al centro del que procede. Para ejecutar un movimiento, se genera una lista de candidatos no prohibidos o prohibidos pero que mejoran el valor del makespan. A partir de ellos se elige el mejor movimiento. Si no se puede generar la lista de candidatos se relaja la restricción tabú más antigua hasta que es posible generarla. El inicio del algoritmo es heurístico, utilizando una serie de once heurísticas. Los autores proponen dos mejoras a su algoritmo; una implementación avanzada de la lista tabú y un procedimiento rápido de cálculo del makespan a partir de la solución previa. El algoritmo implementado mejora todas las heurísticas analizadas, puesto que encuentra soluciones mejores en menor tiempo de cálculo.

En el trabajo de **(Riane, F. et al. 1998)** se plantea un estudio para la minimización del makespan en un Taller de Flujo Híbrido de tres etapas a partir de un caso de una industria de fabricación de tableros, con una máquina en las etapas uno y tres y dos máquinas en las etapas intermedias en las que la asignación de los trabajos se realiza con antelación. Proponen dos

algoritmos para su resolución. El primero se basa en aplicar Programación Dinámica a una secuencia de los trabajos obtenidos, mediante la regla CDS, y subdivide el problema en dos subproblemas en función de que los trabajos se asignen a la primera o segunda máquina de la segunda etapa. El segundo enfoque se basa en un algoritmo de Ramificación y Poda. Para ello, definen cotas superiores del makespan. Este último procedimiento se revela mejor que el primero.

En **(Gupta, J. N. D. et al. 1998)** se analiza un taller de flujo híbrido de dos etapas, donde la primera consta de una única máquina y cuyo objetivo es minimizar el número de trabajos retrasados. Plantean dos procedimientos *heurísticos* basados en el planteamiento de **(Moore, J. E. 1975)**. Mediante un diseño de experimentos concluyen que el funcionamiento de las heurísticas depende de parámetros propios del proceso, tales como las fechas de entrega y tiempos de proceso de los trabajos, así como el número de máquinas en la segunda etapa.

Autores como **(Soewandi, H. et al. 2001)** se plantean el reto de minimizar el C_{max} en un Taller de Flujo con tres etapas y máquinas idénticas en paralelo. Se propone una heurística y se calculan las cotas de la misma para calcular su distancia al óptimo. En **(Soewandi, H. et al. 2003)** se propone una nueva heurística pero esta vez para un taller con dos etapas y máquinas uniformes.

En el trabajo de **(Wardono, B. et al. 2004)** se plantea una solución basada en Búsqueda Tabú para el problema del taller de Flujo Híbrido con limitación de capacidad entre etapas. El algoritmo propuesto complementa la búsqueda con una heurística que se utiliza para permutar trabajos y alimentar al procedimiento de búsqueda. EL objetivo es minimizar el makespan. Los autores realizan un estudio empírico.

Bajo un amplio análisis del problema del Taller de Flujo Híbrido en el contexto de la industria textil con el objetivo de minimizar el C_{max} **(Serifoglu, F. S. et al. 2004)** proponen un Algoritmo Genético y lo comparan con diferentes heurísticas en un banco de pruebas de más de cuatrocientas instancias resultando eficiente tanto para problemas pequeños como de gran tamaño.

(Allahverdi, A. et al. 2006) aplica las técnicas conocidas de secuenciación y Programación de la Producción a un sistema de computación distribuido. El objetivo es reducir el tiempo de respuesta medio (C_{max} medio). Los autores caracterizan este sistema como un Taller de Flujo Híbrido y proponen una nueva heurística que se ejecuta en tres fases. Los resultados comparativos indican que la propuesta es bastante mejor que otras conocidas tanto en eficacia como en consumo de tiempo de cálculo.

En **(Jin, Z. H. et al. 2006)** se afronta a la reducción del makespan en un Taller de Flujo Híbrido con máquinas idénticas en cada etapa. Los autores proponen una metaheurística que en primer lugar secuencia los trabajos y después los asigna a las máquinas contemplando parcialmente el taller en esta segunda parte. El algoritmo se basa en un Recocido Simulado con búsqueda en profundidad variable. Los resultados experimentales confirman gran eficacia.

Los autores de **(Ying, K. C. et al. 2006)** abordan el problema de minimización del Cmax en un Taller de Flujo Híbrido y analizan la importancia que tiene este planteamiento en numerosas industrias. Los autores proponen una metaheurística bio-inspirada como es la Colonia de Hormigas. Comparan los resultados obtenidos con Algoritmos Genéticos y Búsquedas Tabú obteniendo buenos resultados.

Después de un análisis del problema del Taller de Flujo Híbrido con Tiempos de no Disponibilidad de las máquinas desconocidos a priori **(Gao, J. et al. 2006)** plantean un Algoritmo Genético que con una innovadora representación del fenotipo sobre la que se aplican una búsqueda local y dos formas de selección del vecindario. Se analizan los resultados para la minimización de Cmax, la carga máxima de una máquina, y la carga total de una máquina.

En el trabajo de **(Tavakkoli-Moghaddam, R. et al. 2007)** se realiza una propuesta de Algoritmo Inmune bio-inspirado para afrontar el problema del Taller de Flujo Híbrido multi-objetivo. Los autores se plantean la minimización de la suma de la fecha de finalización ponderada y del retraso ponderado de cada trabajo. Para evaluar la solución de compara con otras propuesta con enfoque evolutivo.

En **(Yang, T. et al. 2007)** se analiza el problema de una empresa fabricante de condensadores eléctricos y la caracterizan como un Taller de Flujo Híbrido. Se trata de un taller con características especiales, ya que tiene más de diez etapas, un par de ellas son procesadas en lotes y la etapa final requiere disponibilidad de un conjunto de lotes para realizar un montaje. Su objetivo es minimizar el retaso máximo. Los autores plantean un Algoritmo Genético que demuestra ser bastante sensible a las restricciones planteadas.

El Taller de Flujo Híbrido considera parcialmente el problema de Programación de la Producción que se abordar en esta tesis y por lo tanto algunas de los procedimiento aplicados pueden ser útiles en la fase aplicación. Es de especial interés, en este apartado, el análisis que los diferentes autores realizan del aspecto concreto de la influencia las máquinas múltiples en cada etapa sobre la eficacia de los métodos propuestos. Habría que destacar el uso mayoritario de procedimientos aproximados para problema de tamaño medio-grande. Así

*mismo, se empieza a intuir un predominio de dos enfoques heurísticos como son las **Heurísticas Ad-Hoc y los Algoritmos Genéticos**.*

3.7.2 El Caso del Taller de Flujo Híbrido con Tiempos de Cambio de Partida Dependientes de la Secuencia.

El problema que se analiza en este apartado es el de mayor complejidad de los vistos en relación con el Taller de Flujo. Sin embargo, es de mayor interés en cuanto a su aplicación al caso que se analizará en esta tesis. Este tipo de taller se puede identificar con la siguiente codificación: **n/m/F, k-parallel, setup-SD/”Medida**.

Dentro de los problemas del Taller de Flujo este puede ser uno de los menos tratados históricamente, y que sin embargo más interés han generado en los últimos años. A continuación se analizan las referencias más importantes que se han encontrado, y que se corresponden con los últimos años. Hay que destacar que las propuestas presentadas por los diferentes autores están siempre basadas en métodos heurísticos o aproximados. Se puede ver el resumen en la Tabla 3-12 y Tabla 3-13.

En el trabajo de **(Kochlar, S. & Morris, R. J. T. 1987)** se denomina a este tipo de configuraciones como líneas de Flujo Flexibles y plantean *heurísticas* considerando tiempos de ajuste, almacenes finitos, bloqueo y tiempos de reparación en las máquinas. Las heurísticas planteadas tratan de minimizar los efectos de los tiempos de ajuste y el bloqueo.

En **(Kochlar, S. et al. 1987)** se aborda un caso real en una empresa de fabricación de papel modelando como un problema de dos etapas con múltiples máquinas por etapa y tiempos de cambio y proceso dependientes de la secuencia siendo la función objetivo la suma de los costes de asignación y los costes de cambio y teniendo en cuenta una restricción de equilibrado de las líneas. La resolución se basa en una reducción del problema a dos subproblemas de asignación y secuenciación en máquinas paralelas justificada por la existencia de un almacén entre etapas de gran capacidad. El procedimiento Programa hacia atrás desde la última etapa en una primera fase, asignando los trabajos a las máquinas de cada etapa en función de las disponibilidades y de las fechas de entrega, y posteriormente secuencian los trabajos en una segunda fase desde la primera etapa en adelante con el objetivo de minimizar los tiempos de cambio entre los trabajos en cada máquina. Los autores proponen además un modelo de PLM para el problema completo en cada etapa descomponiéndolo en sendos modelos PLM para los subproblemas de asignación y secuenciación.

En **(Gupta, J. N. D. & Tunc, E. A. 1994)** se estudia el problema de Taller de Flujo Híbrido de dos etapas, donde la primera etapa solo está constituida por una máquina. Entre cada trabajo aparecen tiempos de montaje y desmontaje no dependientes de la secuencia y el objetivo es minimizar el makespan. Definen dos cotas mínimas de duración y, a partir de ellas, plantean cuatro *heurísticas* basadas en los resultados de [Sule y Huang, 1983] y [Szwarc y Gupta, 1987]. Las heurísticas resuelven primero un problema de secuenciación en taller de flujo simple y a partir de esta secuencia asignan a las máquinas de la segunda etapa siguiendo un planteamiento de lanzamiento.

<i>Referencias</i>	<i>Medida Eficiencia</i>	<i>Técnica</i>
(Kochlar, S. et al. 1987)	Tiempos Ajuste	Heurística usando tiempos de ajuste/Preparación
(Kochlar, S. et al. 1987)	Suma Coste Asignación y Ajuste	Heurística dos etapas
(Gupta, J. N. D. et al. 1994)	Cmax	Heurísticas
(Aghezzaf, E. A., Artiba, A., & Elmaghraby, S. E. 1995a) y (Aghezzaf, E. A., Artiba, A., Moursli, O., & Tahon, C. 1995b)	Cmax, Fmax, Fmedio	Heurística
(Li, S. L. 1997)	Tiempos Ajuste	Varias Heurísticas
(Andres, C., Abad, R., Ros, L., & Vicens, E. 1998)	Cmax	Heurística
(Liu, C. Y. & Chang, S. C. 2000)	Tiempos y Costes de Setup	Heurística basada en Relajación Lagrangiana

Tabla 3-12. Resumen contribuciones al problema del Taller de Flujo Híbrido con SDST con Soluciones Aproximadas (elaboración propia).

En **(Aghezzaf, E. A. et al. 1995a)** y **(Aghezzaf, E. A. et al. 1995b)** plantean un algoritmo *heurístico* a dos fases para resolver el problema de Taller de Flujo Híbrido con Tiempos de Cambio Dependientes de la Secuencia que aparece en una empresa textil mediante su descomposición en varios subproblemas de máquinas paralelas. Se estudia para el caso de minimización del Cmax, Fmax y Flujo medio. En una primera fase los autores resuelven un problema de asignación y secuenciación, suponiendo fijadas los tamaños de los lotes. En una segunda fase se consideran fijados los Tiempos de Proceso unitario y las cantidades a producir, con lo que se debe determinar el tamaño de los sub-lotes a producir en un horizonte discreto.

Referencias	Medida Eficiencia	Técnica
(Kurz, M. E. & Askin, R. G. 2003)	Cmax	Heurísticas: Inserción, basadas en Johnson, y heurísticas Voraces
(Kurz, M. E. & Askin, R. G. 2004)	Cmax	4 Heurísticas + Algoritmos Genéticos Claves aleatorias
(Pugazhendhi, S., Thiagarajan, S., Rajendran, C., & Anantharaman, N. 2004)	$\sum wif_j$	Heurística no permutación a permutación
(Zandieh, M., Ghomi, S. M. T. F., & Hussein, S. M. M. 2006)	Cmax	Algoritmo Inmune
(Logendran, R., deSzoek, P., & Barnard, F. 2006)	Cmax	Búsqueda Tabú + Heurística Inicial
(Gomez-Gasquet, P., Andres, C., & García-Sabater, J. P. 2004; Stania, L., Gomez-Gasquet, P., & Andres, C. 2006)	$\sum T_j$	Reglas Despacho
(Ruiz, R. & Maroto, C. 2006)	Cmax	Algoritmo Genético
(Ruiz, R. et al. 2008b)	Cmax	Algoritmo Genético
(Jungwattanakit, J., Reodecha, M., Chaovalitwongse, P., & Werner, F. 2008)	Cmax+Nº Trabajos Retrasados	Heurísticas. Mejorada con Shift-Jobs y Algoritmos Genéticos
(Ruiz, R., Serifoglu, F. S., & Urlings, T. 2008a)	Cmax	Modelo MIP y Heurísticas

Tabla 3-13. Resumen contribuciones al problema del Taller de Flujo Híbrido con SDST con Soluciones Aproximadas (continuación de Tabla 3-12 (elaboración propia)).

El autor de (Li, S. L. 1997) analiza el problema de Taller de Flujo Híbrido con una única máquina en la primera etapa y tiempos de setup mayores y menores, independientes de la secuencia. El autor plantea dos heurísticas, hacia delante y hacia atrás (forward y backward) que tienden a reducir los tiempos de ajuste. Las familias son secuenciadas utilizando un criterio basado en los tiempos de setup mayores y la suma de los tiempos de setup menores en la familia. Mientras que los trabajos dentro de cada familia son secuenciados en progresión decreciente de la suma de los tiempos de ajuste menor y los de proceso. Los factores considerados dentro de los experimentos son el número de máquinas en la etapa 2, el número de familias, el número de lotes por familia, el tamaño de los lotes, los tiempos de ajuste mayores y menores en la

primera y segunda etapas. La conclusión de los autores es que la heurística basada en la secuenciación hacia atrás (de la última etapa a la primera) es más eficiente que la basada en la secuenciación hacia delante. Además, experimentalmente analizan sendas variantes basadas en la posibilidad de fragmentar los lotes comprobándose que esta opción mejora los resultados. Otro resultado se refiere al estudio de diferentes reglas para secuenciar en la única máquina de la primera etapa, en concreto se analizan seis, todas ellas utilizando en su definición los tiempos de ajuste mayores y menores, revelándose como la mejor de ellas la basada en el menor tiempo de proceso más ajuste medio.

En **(Andres, C. et al. 1998)** se afrontó el problema para un taller de Flujo de dos etapas con un Algoritmo Genético. Se codifican los trabajos en la primera etapa planteándose la secuenciación en la segunda mediante la asignación de los diversos trabajos a la máquina que primero queda libre.

En **(Liu, C. Y. et al. 2000)** se aborda el problema del Taller de Flujo Híbrido con Tiempos y Coste Dependientes de la Secuencia y fechas de llegada diferentes ($n/m/F$, k -parallel, setup, r_i) con el objetivo de minimizar los Tiempos y Costes de Setup. Se plantea un modelo de Programación Entera separable y se aplicó sobre el la técnica de relajación Lagrangiana con lo que se obtuvo una heurística.

El problema $n/m/F$, k -parallel, setup/ C_{max} fue analizado por **(Kurz, M. E. et al. 2003)** cuando algunos trabajos tenían la posibilidad de realizar menos de m operaciones. Propusieron y estudiaron tres tipos de heurísticas llamadas "heurísticas de Inserción", basadas en Johnsons, y heurísticas Voraces. Identificando las condiciones bajo las cuales la propuesta funcionaba correctamente. Para el problema $n/m/F$, k -parallel, setup/ C_{max} , **(Kurz, M. E. et al. 2004)** compararon cuatro heurísticas, identificando límites inferiores y comparando las misma con dichos límites, alguna basada en Algoritmos Genéticos con Claves aleatorias.

En **(Pugazhendhi, S. et al. 2004)** se afrontó el problema $n/m/F$, k -parallel/ $\sum w_i f_j$, donde algunos trabajos no realizaban m operaciones. Los autores propusieron un procedimiento heurístico para alcanzar una secuencia de no permutación a partir de una secuencia de permutación dada, demostrando su utilidad.

En la propuesta de **(Zandieh, M. et al. 2006)** se desarrolló un Algoritmo Inmune y demuestran que este algoritmo supera al Algoritmo Genético con claves aleatorias de **(Kurz, M. E. et al. 2004)**.

Con el desarrollo de tres algoritmos basados en Búsqueda Tabú los autores de **(Logendran, R. et al. 2006)** realizaron un amplio estudio estadístico relacionado con el C_{max} y los tiempos de

computación. En todos los casos se consideraron diversos procedimientos para fijar una solución ideal que facilitase el inicio de la búsqueda.

En **(Gomez-Gasquet, P. et al. 2004; Stania, L. et al. 2006)** se evaluaron diversas reglas de despacho para el problema $n/3/F, k\text{-parallel}, \text{setup}/$. En el primer caso para diez reglas y en el segundo caso trece reglas con el objetivo de minimizar el retraso promedio. En el segundo trabajo se concluyó que las reglas que contemplaban explícitamente la consideración de los Setup eran más eficaces que el resto.

El problema $n/m/F, k\text{-parallel}, \text{setup}/C_{\max}$ también fue analizado por **(Ruiz, R. et al. 2006)**, pero en este caso en un entorno de Industria Cerámica y más complejo. Las máquinas de cada etapa eran “no relacionadas” y en algunos casos ciertos trabajos no eran ejecutables en ciertas máquinas. Los autores propusieron un Algoritmo Genético en el que se discute ampliamente el proceso de calibrado así como los resultados obtenidos. En **(Ruiz, R. et al. 2008b)** se presenta un modelo matemático y algunas heurísticas para el problema $n/m/F, k\text{-parallel}, \text{setup}/C_{\max}$. Donde se consideran varios parámetros próximos a la realidad como es el caso de trabajo no elegibles en ciertas máquinas, máquinas “no relacionadas”, posibles Cambios de Partida que se pueden realizar de forma anticipada o no anticipada, etc. Los autores obtienen soluciones próximas al óptimo, incluso en instancias de gran tamaño. En **(Ruiz, R. et al. 2008a)** se realiza una aproximación al modelado práctico de problemas reales de minimización del C_{\max} en el ámbito del Taller de Flujo Híbrido desarrollando modelos de Programación Entera Mixta y heurísticas. Los autores tienen en cuenta el problema añadido de los Tiempos de cambio de Partida Dependientes de la Secuencia. Se realiza una evaluación de las propuestas.

A partir del problema del Taller de Flujo Híbrido en una industria textil con máquina no relacionadas y Tiempos de Cambio de Partida **(Jungwattanakit, J. et al. 2008)** plantean minimizar dos objetivos simultáneamente, el C_{\max} y el número de trabajos retrasados. Se plantea un modelo de Programación Entera Mixta. Dado que el problema es NP-Completo los autores desarrollan una serie de heurísticas. Inicialmente se plantean una serie de reglas de despacho en combinación con heurísticas constructivas. Posteriormente se mejoran proponiendo un método de movimiento de trabajos y otros basado en un Algoritmo Genético. La propuesta se testea comparándola con otras con cincuenta trabajos y 20 etapas.

La Programación de la Producción en un Taller de Flujo Híbrido con Tiempos de Cambio de Partida Dependientes de la Secuencia es un aspecto que en sí mismo se desea abordar en esta tesis, y por lo tanto algunas de los procedimientos recopilados pueden ser útiles en la fase aplicación.

En este apartado ha sido relevante analizar como los autores han combinado dos condiciones que incrementan bastante la dificultad de resolución, como son la disponibilidad de máquinas alternativas en cada etapa y la inclusión de los Tiempos de Cambio Dependientes de la Secuencia. En este tipo de problemas los métodos exactos casi no existen, incluso es difícil ver su aplicación a problemas sencillos. Es por lo tanto, dentro de los tipos de problema estudiados, donde más métodos aproximados se han desarrollado. Aunque no puede verse con claridad un predominio de ningún método aproximado, los procedimientos metaheurísticos tienen una importante presencia cuanto más complejo es el problema, incluso por encima de las heurísticas Ad-Hoc. También sería destacable la presencia en los últimos años de numerosos Algoritmos bio-inspirados que proporciona resultados muy competitivos.

3.8 Conclusiones

En este capítulo se ha realizado un amplio recorrido sobre las referencias literarias más significativas de cada uno de los aspectos que han sido destacados. Se ha partido de una revisión del proceso de Programación de la Producción en el apartado 2. En el apartado 3 se han subrayado aspectos relativos a la teoría general de la Programación de la Producción. En el apartado 4, se han presentado características generales del problema que se plantea en relación a la Incertidumbre y el problema de la Reprogramación. En el apartado 5, se ha realizado una introducción al Taller de Flujo con sus diferentes variantes, en el que se define sus principales características. En el apartado 6, se han analizado las características de los procedimientos de resolución de los problemas combinatorios en relación a la Programación de la Producción. En el apartado 7, se han presentado los métodos y algoritmos de resolución del Taller de Flujo con Tiempos de Cambio de Partida Dependientes de la Secuencia, del Taller de Flujo Híbrido, y del Taller de Flujo Híbrido con Tiempos de Cambio de Partida Dependientes de la Secuencia.

Esta revisión bibliográfica ha permitido observar como en la Programación/Secuenciación de la Producción **existe una preocupación por los problemas con Incertidumbre, pero con una escasa presencia en la literatura** en contraposición a los problemas deterministas. Así mismo, la literatura referencia ya desde (Graham, R. L. 1969) problemas con enfoques cambiantes, pero su **aplicación ha resultado ser muy escasa** frente a los casos estáticos. Por otro lado, se ha observado como **muchas de las referencias relevantes en el ámbito de la Reprogramación con Incertidumbre proceden de una aplicación en el ámbito de la informática**. En el campo de la Programación de la Producción se identificado un progresivo interés por los problemas de Reprogramación afectados por algún tipo de Incertidumbre.

Al finalizar este capítulo se puede decir que existe una propuesta que parece interesante en relación a la estrategia para realizar la Programación de la Producción: El enfoque Predictivo-Reactivo. Esta aproximación permitiría iniciar cada periodo productivo con un Programa establecido en un contexto determinista (a priori) y en función de los cambios producidos por la incertidumbre consustancial con el entorno aplicar revisiones del Programa para ajustarlo a la realidad. **La Propuesta Predictivo-Reactiva se ajusta bastante bien a la forma de trabajar actualmente implantada en muchas de las empresas productoras de baldosas cerámicas**, lo que permitiría abordar el problema de la incertidumbre sin proponer cambios radicales.

En cuanto a los métodos y algoritmos revisados para problemas similares al identificado, la mayoría de ellos emplean modelos matemáticos para la definición del problema, pero **son pocas las referencias que utilizan técnicas optimizadoras**, decantándose **la mayoría por algoritmos heurísticos o metaheurísticos**. Entre las técnicas no optimizadoras existen dos tendencias, que a medida que el problema se orienta más a entornos dinámicos o semi-dinámicos se acentúan, éstas son: Las heurísticas ad-hoc o las metaheurísticas con cierto mimetismo con la naturaleza o **bio-inspiradas**. Esta tendencia potencia la necesidad de examinar nuevas referencias en relación con la denominada **Teoría de la Agencia**, que introduce un paradigma nuevo a partir de concepto de Inteligencia Artificial, y que en cierta medida están orientados en la misma línea que los algoritmos bio-inspirados. Este será el objeto del próximo capítulo.

3.9 Bibliografía

Abdel-Wahab, H. M. & Kameda, T. 1978. *Scheduling to minimize maximum acumalative cost subject to series-parallel precedence constraints. Operations Research*, 26: 141-159.

Abumaizar, R. J. & Svestka, J. A. 1997. *Rescheduling job shops under random disruptions. International Journal of Production Research*, 35(7): 2065-2082.

Adam, N. R. & Surkis, J. 1980. *Priority update intervals and anomalies in dynamic ratio job shop scheduling rules. Management Science*, 26: 1127-1137.

Aggarwal, S. C. & McCart, B. A. 1974. *The development and evluation of a cost based composite scheduling rule. Naval Research Logistics*, 21: 155-169.

Aghezzaf, E. A., Artiba, A., and Elmaghraby, S. E. *Hybrid FlowShop: an LP based heuristic for planning level problems. ETFA Proceedings*, 551-559. 1995a.

Aghezzaf, E. A., Artiba, A., Moursli, O., and Tahon, C. *Hybrid Flowshops problems, a decomposition based heuristic approach. International Conference on Industrial Ingeneering and Prodcution Managemanent (IEPM'95). FUCAM/IFIP/INRIA Proceeding*, 43-56. 1995b.

Akturk, M. S. & Gorgulu, E. 1999. *Match-up scheduling under a machine breakdown*. *European Journal of Operational Research*, 112(1): 81-97.

Allahverdi, A. & Al-Anzi, F. S. 2006. *Scheduling multi-stage parallel-processor services to minimize average response time*. *Journal of the Operational Research Society*, 57(1): 101-110.

Allahverdi, A. & Aldowaisan, T. 2001. *Minimizing total completion time in a no-wait flowshop with sequence-dependent additive changeover times*. *Journal of the Operational Research Society*, 52(4): 449-462.

Allahverdi, A., Ng, C. T., Cheng, T. C. E., & Kovalyov, M. Y. 2008. *A survey of scheduling problems with setup times or costs*. *European Journal of Operational Research*, 187(3): 985-1032.

Allahverdi, A. & Soroush, H. M. 2008. *The significance of reducing setup times/setup costs*. *European Journal of Operational Research*, 187(3): 978-984.

Alvarez, E. & Diaz, F. 2004. *An application of a real-time scheduling system for turbulent manufacturing environments*. *Robotics and Computer-Integrated Manufacturing*, 20(6): 485-494.

Andrés, C. 2001. *Programación de la Producción en Talleres de Flujo Híbridos con Tiempos de Cambio de Partida dependientes de la secuencia. Modelos, Métodos y Algoritmos de Resolución. Aplicación a Empresas del Sector Cerámico*. Universidad Politécnica de Valencia.

Andres, C., Abad, R., Ros, L., and Vicens, E. *A Genetic Algorithm for Production Scheduling in a two stage Hybrid flowshop with sequence dependent setup times*. 16th European Conference on Operation Research . 1998. Belgium.

Andres, C., Tomás, J. V., García-Sabater, J. P., and Miralles, C. *A multi-objective genetic algorithm to solve the scheduling problem in flowshops with sequence dependent setup times*. *Proceedings of the International Conference on Industrial Engineering and Systems Management* , 630-635. 2005. Marrakech, Morocco.

Andres, C., Vicens, E., and Lario F.C. *Algoritmo de recocido simulado para la secuenciación en talleres de flujo híbridos con tiempos de cambio de partida dependientes de la secuencia*. XXVI Congreso nacional de estadística e investigación operativa . 2001. Úbeda (Jaen).

Anglani, A., Grieco, A., Guerriero, E., & Musmanno, R. 2005. *Robust scheduling of parallel machines with sequence-dependent set-up costs*. *European Journal of Operational Research*, 161(3): 704-720.

Apics. Diccionario. 1994. American Production and Inventory Control Society.

Artigues, C. 1997. *Ordennancement en temps reel d'ateliers avec temps de preparation des ressources*. LAAS.Université Paul Sabatier de Toulouse.

Aytug, H., Bhattacharyya, S., Koehler, G. J., & Snowdon, J. L. 1994. *A review of machine learning in scheduling*. *IEEE Transactions on Engineering Management*, 41(2): 165-171.

Aytug, H., Lawley, M. A., McKay, K., Mohan, S., & Uzsoy, R. 2005. *Executing production schedules in the face of uncertainties: A review and some future directions*. *European Journal of Operational Research*, 161(1): 86-110.

Baker, K. R. 1974. *Introduction to Sequencing and Scheduling*. New York: John Wiley and Sons.

Baptiste, P. & Favrel, J. 1993. *Taking into account the rescheduling problem during the scheduling phase*. *Production Planning & Control*, 4: 349-360.

Barber, F. 2003. *Presentacion de la Monografía: Problemas de Satisfacción de Restricciones*. *Revista Iberoamericana De Inteligencia Articial*, 3(20): 7-9.

Barroso, A. M., Leite, J. C. B., & Loques, O. G. 2002. *Treating uncertainty in distributed scheduling*. *Journal of Systems and Software*, 63(2): 129-136.

Barták, R. *Theory and practice of constraint propagation*. Figwer, J. 3rd Workshop on Constraint Programming in Decision and Control. *Constraint Programming in Decision and Control*. 2001. Poland.

Bean, J. C., Birge, J. R., Mittenthal, J., & Noon, C. E. 1991. *Machup scheduling with multiple resource release dates and disruption*. *Operations Research*, 39: 470-483.

Bellman, R. & Zadeh, L. A. 1970. *Decision-making in a fuzzy environment*. *Management Science*, 17, nr.4: 141-164.

Bemelman, R., Tharumarajah, A., Welgama, P., & Wells, A. 1999. *Application of a behaviour-based scheduling approach for distributed scheduling of an assembly shop*. *Production Planning & Control*, 10(3): 266-275.

Bhaskaran, K. & Pinedo, M. 1991. *Dispatching*. In G. Salvendy (Ed.), *Handbook of Industrial Engineering*: New York: Wiley.

Black, G. W., McKay, K. N., & Messimer, S. L. 2004. *Predictive, stochastic and dynamic extensions to aversion dynamics scheduling*. *Journal of Scheduling*, 7(4): 277-292.

Blazewicz, J., Ecker, K. H., Pesch, E., Schmidt, G., & Weglarz, J. 1996. *Scheduling computer and manufacturing processes*. Springer.

Bollapragada, R. and Sadeh, N. M. *Cost-based approaches to stochastic job shop scheduling problems*. 1996. *Research Report, Robotics Institute, Carnegie Mellon University*.

Bollapragada, R. & Sadeh, N. M. 2004. *An empirical study of policies to integrate reactive scheduling and control in just-in-time job shop environments*. *International Journal of Production Research*, 42(4): 693-718.

Brah, S. A. 1996. *A comparative analysis of due date based job sequencing rules in a flow shop with multiple processors*. *Production Planning & Control*, 7(4): 362-373.

Brah, S. A. & Hunsucker, J. L. 1991. *Branch and Bound Algorithm for the Flow-Shop with Multiple Processors*. *European Journal of Operational Research*, 51(1): 88-99.

Brah, S. A. & Loo, L. L. 1999. *Heuristics for scheduling in a flow shop with multiple processors*. *European Journal of Operational Research*, 113(1): 113-122.

Brandimarte, P., Rigodanza, M., & Roero, L. 1997. *Conceptual modeling of an object oriented scheduling architecture based on the shifting bottleneck procedure*. *International Journal of Production Research*, 35(7): 2065-2082.

Brucker, P. 2006. *Scheduling Algorithms*. Berlin: Springer-Verlag.

Burbidge, J. L. 1992. *Change to group technology: process organization is obsolete*. *International Journal of Operations Research*, 30: 1209-1219.

Burbidge, J. L. 1963. *Production flow analysis*. *Production Engineer*, 42: 742.

Campbell, H. G., Dudek, R. A., & Smith, M. L. 1970. *Heuristic Algorithm for N-Job, M-Machine Sequencing Problem*. *Management Science Series B-Application*, 16(10): B630-B637.

Carvalho, D., Protti, F., De Gregorio, M., & Franca, F. M. G. 2005. *A novel distributed scheduling algorithm for resource sharing under near-heavy load*. *Principles of Distributed Systems*, 3544: 431-442.

Chan, F. T. S., Chung, S. H., & Chan, P. L. Y. 2005. *An adaptive genetic algorithm with dominated genes for distributed scheduling problems*. *Expert Systems With Applications*, 29(2): 364-371.

Chen, C. & Yih, Y. 1996. *Identifying attributes for knowledge base development in dynamic scheduling environments*. *International Journal of Production Research*, 34(6): 1739-1755.

Chen, J. & Chen, F. F. 2003. *Adaptive scheduling in random flexible manufacturing systems subject to machine breakdowns*. *International Journal of Production Research*, 41(9): 1927-1951.

Cheng, T. C. E., Gupta, J. N. D., & Wang, G. Q. 2000. *A review of flowshop scheduling research with setup times*. *Production and Operations Management*, 9(3): 262-282.

Chrétienne, P., Coffman, J. E., Lensrta, J. K., & Liu, Z. 1995. *Scheduling theory and its applications*. Chichester: Wiley.

Chun, H. W. & Wong, R. Y. M. 2003. *N* - an agent-based negotiation algorithm for dynamic scheduling and rescheduling*. *Advanced Engineering Informatics*, 17(1): 1-22.

Church, L. K. & Uzsoy, R. 1992. *Analysis of Periodic and Event-Driven Rescheduling Policies in Dynamic Shops*. *International Journal of Computer Integrated Manufacturing*, 5(3): 153-163.

Cicirello, V. A. & Smith, S. F. 2004. *Wasp-like agents for distributed factory coordination*. *Autonomous Agents and Multi-Agent Systems*, 8(3): 237-266.

Companys, R. & Corominas, A. 1996. Dirección de operaciones. Organización de la producción II.: Barcelona.

Conway, R. W., Maxwell, W. L., & Miller, L. W. 1967. Theory of Scheduling. Addison-Wesley Publishing Company.

Corwin, B. D. & Esogbue, A. O. 1974. 2 Machine Flow Shop Scheduling Problems with Sequence Dependent Setup Times - Dynamic-Programming Approach. Naval Research Logistics, 21(3): 515-524.

Cowling, P. & Johansson, M. 2002. Using real time information for effective dynamic scheduling. European Journal of Operational Research, 139(2): 230-244.

Daniels, R. L. & Carrillo, J. E. 1997. Beta-Robust scheduling for single machine systems with uncertain processing times. IIE Transactions on Scheduling and Logistics, 29: 977-985.

Daniels, R. L. & Kouvelis, P. 1995. Robust Scheduling to Hedge Against Processing Time Uncertainty in Single-Stage Production. Management Science, 41(2): 363-376.

Das, S. R., Gupta, J. N. D., & Khumawala, B. M. 1995. A Savings Index Heuristic Algorithm for Flowshop Scheduling with Sequence-Dependent Set-Up Times. Journal of the Operational Research Society, 46(11): 1365-1373.

Das, S. R., Gupta, J. N. D., & Khumawala, B. M. 2004. A savings index heuristic algorithm for flowshop scheduling with sequence-dependent setup times (vol 46, pg 1365, 1995). Journal of the Operational Research Society, 55(12): 1369.

Deen, S. M. 1993. Cooperation Issues in Holonic Manufacturing Systems. Information Infrastructure Systems for Manufacturing, 14: 401-412.

Dempster, M. A., Lensrta, J. K., and Rinnooy, K. Deterministic and stochastic scheduling. Proceedings of the NATO Advanced Study and Research Institute on Theoretical Approaches to Scheduling Problems, 6-17. 1982. Durham (UK).

Dewan, P. & Joshi, S. 2002. Auction-based distributed scheduling in a dynamic job shop environment. International Journal of Production Research, 40(5): 1173-1191.

Dooley, K. J. & Mahmoodi, F. 1992. Identification of robust scheduling heuristics: application of Taguchi methods in simulation studies. Computers and Industrial Engineering, 22: 359-368.

Dorn, J., Kerr, R., & Thalhammer, G. 1994. Reactive Scheduling in A Fuzzy-Temporal Framework. Knowledge-Based Reactive Scheduling, 15: 39-55.

Duenas, A. & Petrovic, D. 2008. An approach to predictive-reactive scheduling of parallel machines subject to disruptions. Annals of Operations Research, 159(1): 65-82.

Duffie, N. A. & Prabhu, V. V. 1994. Real-Time Distributed Scheduling of Heterarchical Manufacturing Systems. Journal of Manufacturing Systems, 13(2): 94-107.

Dutta, A. 1990. *Reacting to Scheduling Exceptions in Fms Environments*. *Iie Transactions*, 22(4): 300-314.

Efstathiou, J. *Formalizing the repair of schedule through the knowledge acquisitions*. 9th European Knowledge Acquisition - EKAW. *Advances in Knowledge Acquisition*, 306-320. 1996.

Fang, J. & Xi, Y. G. 1997. *A rolling horizon job shop rescheduling strategy in the dynamic environment*. *International Journal of Advanced Manufacturing Technology*, 13(3): 227-232.

Fox, M. S. & Smith, S. F. 1984. *ISIS - Knowledge based system for factory scheduling*. *Expert Systems*, 1: 300-314.

Framinan, J. M., Gupta, J. N. D., & Leisten, R. 2004. *A review and classification of heuristics for permutation flow-shop scheduling with makespan objective*. *Journal of the Operational Research Society*, 55(12): 1243-1255.

Galbraith, J. 1973. *Designing complex organizations*. Massachusetts: Addison-Wesley Pub. Co.

Gao, J., Gen, M., & Sun, L. Y. 2006. *Scheduling jobs and maintenances in flexible job shop with a hybrid genetic algorithm*. *Journal of Intelligent Manufacturing*, 17(4): 493-507.

Garey, M. R. & Johnson, D. S. 1979. *Computers and Intractability: A guide to the Theory of NP-Completeness*. San Francisco: W. H. Freeman and Company.

Gargeya, V. B. & Deane, R. H. 1999. *Scheduling in the dynamic job shop under auxiliary resource constraints: a simulation study*. *International Journal of Production Research*, 37(12): 2817-2834.

Gargouri, E. & Hammadi, S. 2003. *A distributed scheduling for agro-food manufacturing problems*. *Ieee Transactions on Systems Man and Cybernetics Part C-Applications and Reviews*, 33(2): 176-185.

Garner, B. J. & Ridley, G. J. 1994. *Application of Neural-Network Process Models in Reactive Scheduling*. *Knowledge-Based Reactive Scheduling*, 15: 19-27.

Garvey, A. J. & Lesser, V. R. 1993. *Design-to-time real-time scheduling*. *IEEE Transactions on Systems, Man and Cybernetics*, 23(6): 491.

Ghosh, S. 2001. *Understanding complex, real-world systems through asynchronous, distributed decision-making algorithms*. *Journal of Systems and Software*, 58(2): 153-167.

Glover, F. 1990. *Tabu Search - A Tutorial*. *Interfaces*, 20(4): 74-94.

Gomez-Gasquet, P., Andres, C., and García-Sabater, J. P. *Dynamic hybrid flow-shop scheduling with due dates and sequence dependent setup times*. *Production Planning and Scheduling 2004 (PMS'04)*. *Proceedings*, 254-259. 2004. Nancy (France).

Graham, R. L. 1969. *Bounds on Multiprocessing Timing Anomalies*. *Siam Journal on Applied Mathematics*, 17(2): 416-&.

- Graves, C. S. 1981. *Review of production scheduling*. *Operations Research*, 29: 646-675.
- Guinet, A., Solomon, M. M., Kedia, P. K., & Dussauchoy, A. 1996. *A computational study of heuristics for two-stage flexible flowshops*. *International Journal of Production Research*, 34(5): 1399-1415.
- Guinet, A. G. P. & Solomon, M. M. 1996. *Scheduling hybrid flowshops to minimize maximum tardiness or maximum completion time*. *International Journal of Production Research*, 34(6): 1643-1654.
- Guo, B. & Nonaka, Y. 1999. *Rescheduling and optimization of schedules considering machine failures*. *International Journal of Production Economics*, 60-1: 503-513.
- Gupta, J. N. D. 1975. *A Search Algorithm for the Generalized Flowshop Scheduling Problem*. *Computers & Operations Research*, 2: 83-90.
- Gupta, J. N. D. 1979. *Improved Lexicographic Search Algorithm for the Flowshop Scheduling Problem*. *Computers & Operations Research*, 6(2): 117-120.
- Gupta, J. N. D. 1972. *Heuristic Algorithms for Multistage Flowshop Scheduling Problem*. *AIIE Transactions*, 4(1): 11-18.
- Gupta, J. N. D. 1988. *2-Stage, Hybrid Flowshop Scheduling Problem*. *Journal of the Operational Research Society*, 39(4): 359-364.
- Gupta, J. N. D. 1986. *Flowshop Schedules with Sequence Dependent Setup Times*. *Journal of the Operations Research Society of Japan*, 29(3): 206-219.
- Gupta, J. N. D. & Darrow, W. P. 1986. *The 2-Machine Sequence Dependent Flowshop Scheduling Problem*. *European Journal of Operational Research*, 24(3): 439-446.
- Gupta, J. N. D. & Tunc, E. A. 1994. *Scheduling A 2-Stage Hybrid Flowshop with Separable Setup and Removal Times*. *European Journal of Operational Research*, 77(3): 415-428.
- Gupta, J. N. D. & Tunc, E. A. 1991. *Schedules for A 2-Stage Hybrid Flowshop with Parallel Machines at the 2Nd Stage*. *International Journal of Production Research*, 29(7): 1489-1502.
- Gupta, J. N. D. & Tunc, E. A. 1998. *Minimizing tardy jobs in a two-stage hybrid flowshop*. *International Journal of Production Research*, 36(9): 2397-2417.
- Hadavi, K., Hsu, W. L., Chen, T., & Lee, C. N. 1992. *An Architecture for Real-Time Distributed Scheduling*. *Ai Magazine*, 13(3): 46-56.
- Hall, N. G. & Potts, C. N. 2004. *Rescheduling for new orders*. *Operations Research*, 52(3): 440-453.
- Hall, N. G. & Sriskandarajah, C. 1996. *A survey of machine scheduling problems with blocking and no-wait in process*. *Operations Research*, 44(3): 510-525.

Haouari, M. & M'Hallah, R. 1997. *Heuristic algorithms for the two-stage hybrid flowshop problem. Operations Research Letters, 21(1): 43-53.*

Hashiba, S. & Chang, T. C. 1991. *PCB assembly set-up reduction using Group Technology. Computers & Industrial Engineering, 21: 453-457.*

Hasle, G. & Smith, S. F. 1994. *Directing an opportunistic scheduler: an empirical investigation of reactive scenario. Artificial intelligence in reactive scheduling: 1-11. UK: Chapman and Hall.*

Hauptman, B. & Jovan, V. 2004. *An approach to process production reactive scheduling. Isa Transactions, 43(2): 305-318.*

Hax, A. C. & Meal, H. C. 1975. *Hierarchical integration of production planning and scheduling. In M. A. Geisler (Ed.), Studies in the Management Sciences: North Holland, Amsterdam.*

He, Y., Smith, M., and Dudek, R. *Effects of inaccuracy of processing time estimation on effectiveness of dispatch rules. 3rd Industrial Engineering Research Conference. 308-313. 1994.*

Hejazi, S. R. & Saghafian, S. 2005. *Flowshop-scheduling problems with makespan criterion: a review. International Journal of Production Research, 43(14): 2895-2929.*

Henning, G. P. & Cerda, J. 1995. *An expert system for predictive and reactive scheduling of multiproduct batch plants. Latin Am.App.Res., 25: 187-198.*

Herrmann, J. W. *A genetic algorithm for minimax optimization problems. Proceeding of the 1999 Congress on Evolutionary Computation, 1099-1103. 1999. Washington, D.C.*

Herrmann, J. W., Chung-Yee, L., & Snowdon, J. 1993. *A classification of static scheduling problems. Complexity in Numerical Optimization: Pardalos (eds.).*

Ho, C. 1989. *Evaluating the impact of operating environments on MRP systems nervousness. International Journal of Production Research, 27: 1115-1135.*

Hoffmann, T. R. & Scudder, G. D. 1983. *Priority scheduling with cost considerations. International Journal of Production Research, 26: 881-889.*

Holland, J. H. 1975. *Adaptation in Natural and Artificial Systems. University of Michigan, Ann Arbor.*

Hunsucker, J. L. & Shah, J. R. 1994. *Comparative Performance Analysis of Priority Rules in A Constrained Flow-Shop with Multiple Processors Environment. European Journal of Operational Research, 72(1): 102-114.*

Hwang, H. C. & Sun, J. U. 1997. *Production sequencing problem with reentrant work flows and sequence dependent setup times. Computers & Industrial Engineering, 33: 773-776.*

Hwang, H. C. & Sun, J. U. 1998. *Production sequencing problem with re-entrant work flows and sequence dependent setup times. International Journal of Production Research, 36: 2435-2450.*

- Jain, A. K and Elmaraghy, H. A. *Production scheduling/rescheduling in flexible manufacturing*. 35, 281-309. 1997. *International Journal of Production Research*.
- Jang, W. S. & Klein, C. M. 2002. *Minimizing the expected number of tardy jobs when processing times are normally distributed*. *Operations Research Letters*, 30(2): 100-106.
- Jayamohan, M. S. & Rajendran, C. 2000. *New dispatching rules for shop scheduling: a step forward*. *International Journal of Production Research*, 38(3): 563-586.
- Jensen, M. T. 2001. *Improving robustness and flexibility of tardiness and total flow-time job shops using robustness measures*. *Applied Soft Computing*, 1(1): 35-52.
- Jensen, M. T. 2003. *Generating robust and flexible job shop schedules using genetic algorithms*. *IEEE Transactions on Evolutionary Computation*, 7(3): 275-288.
- Jeong, I. J. & Leon, V. J. 2002. *A distributed scheduling methodology for a two-machine flowshop using cooperative interaction via multiple coupling agents*. *Journal of Manufacturing Systems*, 21(2): 126-139.
- Jeong, I. J. & Leon, V. J. 2005. *A single-machine distributed scheduling methodology using cooperative interaction via coupling agents*. *IEEE Transactions*, 37(2): 137-152.
- Jeong, K. C. & Kim, Y. D. 1998. *A real time scheduling mechanism for a flexible manufacturing system using simulation and dispatching rules*. *International Journal of Production Research*, 36: 2609-2626.
- Jian, H. Z., Nee, A. Y. C., Fuh, J. Y. H., & Zhang, Y. F. 2003. *A modified genetic algorithm for distributed scheduling problems*. *Journal of Intelligent Manufacturing*, 14(3-4): 351-362.
- Jin, Z. H., Yang, Z., & Ito, T. 2006. *Metaheuristic algorithms for the multistage hybrid flowshop scheduling problem*. *International Journal of Production Economics*, 100(2): 322-334.
- Johnson, S. M. 1954. *Optimal two-and three -stage production schedules with setup times included*. *Naval Research Quarterly*, 1: 61-68.
- Jones, C. H. 1973. *An economic evaluation of job shop dispatching rules*. *Management Science*, 20: 293-307.
- Jungwattanakit, J., Reodecha, M., Chaovalitwongse, P., & Werner, F. 2008. *Algorithms for flexible flow shop problems with unrelated parallel machines, setup times, and dual criteria*. *International Journal of Advanced Manufacturing Technology*, 37(3-4): 354-370.
- Kamoun, H. & Sriskandarajah, C. 1993. *The Complexity of Scheduling Jobs in Repetitive Manufacturing Systems*. *European Journal of Operational Research*, 70(3): 350-364.
- Karlin, A. R., Manasse, M. S., Rudolph, L., & Sleator, D. D. 1988. *Competitive Snoopy Caching*. *Algorithmica*, 3(1): 79-119.

Kim, K. H., Bae, J. W., Song, J. Y., & Lee, H. Y. 1996. *A distributed scheduling and shop floor control method. Computers & Industrial Engineering*, 31(3-4): 583-586.

Kim, M. H. and Kim, Y. D. *Simulation-based real-time scheduling in a flexible manufacturing system*. 13, 85-93. 1994. *J. Manufact. Syst.*

Kim, S. C. & Bobrowski, P. M. 1994. *Impact of Sequence-Dependent Setup Time on Job-Shop Scheduling Performance. International Journal of Production Research*, 32(7): 1503-1520.

Kim, Y. D., Lim, H. G., & Park, M. W. 1996. *Search heuristics for a flowshop scheduling problem in a printed circuit board assembly process. European Journal of Operational Research*, 91(1): 124-143.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. 1983. *Optimization by Simulated Annealing. Science*, 220(4598): 671-680.

Kis, T. & Pesch, E. 2005. *A review of exact solution methods for the non-preemptive multiprocessor flowshop problem. European Journal of Operational Research*, 164(3): 592-608.

Kochlar, S. & Morris, R. J. T. 1987. *Heuristic methods for flexible flow line scheduling. Journal of Manufacturing Systems*, 6: 299-314.

Kolmogoroff, A. 1950. *Foundations of probability. New York: Chelsea Pub Co; 2nd edition (June 1, 1960).*

Koopman, B. O. 1940. *The axioms and algebra of intuitive probability. Annals of Mathematics*, 41: 269-292.

Kouvelis, P. & Yu, G. 1997. *Robust discrete optimization and its applications. Kluwer Academic Publishers.*

Kumar, R., Tiwari, M. K., & Allada, V. 2004. *Modelling and rescheduling of a re-entrant wafer fabrication line involving machine unreliability. International Journal of Production Research*, 42(21): 4431-4455.

Kurz, M. E. & Askin, R. G. 2003. *Comparing scheduling rules for flexible flow lines. International Journal of Production Economics*, 85(3): 371-388.

Kurz, M. E. & Askin, R. G. 2004. *Scheduling flexible flow lines with sequence-dependent setup times. European Journal of Operational Research*, 159(1): 66-82.

Kutanoglu, E. & Wu, S. D. 2004. *Improving scheduling robustness via preprocessing and dynamic adaptation. IIE Transactions*, 36(11): 1107-1124.

Lario F.C. 1986. *Modelos en Planificación de la Producción. Servicio de Publicaciones de la Universidad Politécnica de Valencia.*

Lawrence, S. R. 1991. *Scheduling a single machine to maximise net present values. International Journal of Production Research*, 29: 1141-1160.

Lawrence, S. R. & Sewell, E. C. 1997. *Heuristic, optimal, static and dynamic schedules when processing time are uncertain. Journal of Operations Management, 15: 71-82.*

Lee, C. Y., Cheng, T. C. E., & Lin, B. M. T. 1993. *Minimizing the Makespan in the 3-Machine Assembly-Type Flowshop Scheduling Problem. Management Science, 39(5): 616-625.*

Lee, C. Y. & Vairaktarakis, G. L. 1994. *Minimizing Makespan in Hybrid Flowshops. Operations Research Letters, 16(3): 149-158.*

Lee, T. E. & Posner, M. E. 1997. *Performance measures and schedules in periodic job shops. Operations Research, 45(1): 72-91.*

Lejmi, T. & Sabuncuoglu, I. 2002. *Effect of load, processing time and due date variation on the effectiveness of scheduling rules. International Journal of Production Research, 40(4): 945-974.*

Lenstra, J. K., Rinnooykan, A. H., & Brucker, P. 1977. *Complexity of machine scheduling problems. Annals of Discrete Mathematics, 1: 343-362.*

Leon, V. J., Wu, S. D., and Storer, R. H. *Robustness measures and robust scheduling for job shops. 1990. Department of Industrial Engineering, Lehigh University.*

Leon, V. J., Wu, S. D., & Storer, R. H. 1994. *Robustness Measures and Robust Scheduling for Job Shops. IIE Transactions, 26(5): 32-43.*

Li, R. K., Shyu, Y. T., & Adiga, S. 1993. *A Heuristic Rescheduling Algorithm for Computer-Based Production Scheduling Systems. International Journal of Production Research, 31(8): 1815-1826.*

Li, S. L. 1997. *A hybrid two-stage flowshop with part family, batch production, major and minor set-ups. European Journal of Operational Research, 102(1): 142-156.*

Liu, C. Y. & Chang, S. C. 2000. *Scheduling flexible flow shops with sequence-dependent setup effects. IEEE Transactions on Robotics and Automation, 16(4): 408-419.*

Lodree, E., Jang, W. S., & Klein, C. M. 2004. *A new rule for minimizing the number of tardy jobs in dynamic flow shops. European Journal of Operational Research, 159(1): 258-263.*

Logendran, R., deSzoeko, P., & Barnard, F. 2006. *Sequence-dependent group scheduling problems in flexible flow shops. International Journal of Production Economics, 102(1): 66-86.*

Lu, S. H. & Kumar, P. R. 1991. *Distributed Scheduling Based on Due Dates and Buffer Priorities. IEEE Transactions on Automatic Control, 36(12): 1406-1416.*

Maccarthy, B. L. & Liu, J. Y. 1993. *Addressing the Gap in Scheduling Research - A Review of Optimization and Heuristic Methods in Production Scheduling. International Journal of Production Research, 31(1): 59-79.*

Maddux III, H. S. and Gupta, J. N. D. *Scheduling intermediate and finished products in a two-stage flowshop with sequence dependent setup times. Annual Meeting of the Decision Sciences Institute* , 1579-1585. 2003.

Markowitz, D. M. & Wein, L. M. 2001. *Heavy traffic analysis of dynamic cyclic policies: A unified treatment of the single machine scheduling problem. Operations Research*, 49(2): 246-270.

Martin, B. & Sanz, A. 1997. *Redes neuronales y Sistemas borrosos. RAMA*.

Mason, S. J., Jin, S., & Wessels, C. M. 2004. *Rescheduling strategies for minimizing total weighted tardiness in complex job shops. International Journal of Production Research*, 42(3): 613-628.

Matsuo, H. 1990. *Cyclic Sequencing Problems in the 2-Machine Permutation Flow-Shop - Complexity, Worst-Case, and Average-Case Analysis. Naval Research Logistics*, 37(5): 679-694.

Mckay, K., Mortn, T. E., Ramnath, P., & Wang, J. 2000. *Aversion dynamics-scheduling when the system changes. Journal of Scheduling*, 3(2): 71-88.

McKay, K. N., Buzacott, J. A., & Safayeni, F. R. 1989. *The Scheduler's knowledge of uncertainty: The missing link. Knowledge based production management systems: Amsterdam: Elsevier Science Publishers B. V.*

Mehta, S. V. & Uzsoy, R. 1999. *Predictable scheduling of a single machine subject to breakdowns. International Journal of Computer Integrated Manufacturing*, 12(1): 15-38.

Mehta, S. V. & Uzsoy, R. M. 1998. *Predictable scheduling of a job shop subject to breakdowns. IEEE Transactions on Robotics and Automation*, 14(3): 365-378.

Mendez, C. A. & Cerda, J. 2004. *An MILP framework for batch reactive scheduling with limited discrete resources. Computers & Chemical Engineering*, 28(6-7): 1059-1068.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. 1953. *Equation of State Calculations by Fast Computing Machines. Journal of Chemical Physics*, 21(6): 1087-1092.

Min, H. S. & Yih, Y. 2003. *Development of a real-time multi-objective scheduler for a semiconductor fabrication system. International Journal of Production Research*, 41(10): 2345-2364.

Miyashita, K. 1998. *CAMPS: a constraint-based architecture for multiagent planning and scheduling. Journal of Intelligent Manufacturing*, 9(2): 147-154.

Miyashita, K. 1995. *Case-Based Knowledge Acquisition for Schedule Optimization. Artificial Intelligence in Engineering*, 9(4): 277-287.

Montoya-Torres, J. R. 2003. *Competitive analysis of a better on-line algorithm to minimize total completion time on a single-machine. Journal of Global Optimization*, 27(1): 97-103.

Moore, J. E. 1975. *An algorithm for a single machine scheduling problem with sequence dependent setup times and scheduling windows*. *Management Science, AIIE Transactions*, 7: 35-43.

Muhlemann, A. P., Lockett, A. G., & Farn, C. K. 1982. *Job Shop Scheduling Heuristics and Frequency of Scheduling*. *International Journal of Production Research*, 20(2): 227-241.

Mula, J. 2004. *Modelos para la planificación de la producción bajo incertidumbre. Aplicación en una empresa del sector del automóvil*. Departamento de Organización de Empresas, Economía Financiera y Contabilidad. Universidad Politécnica de Valencia.

Nakamura, M., Nakada, N., Kinjo, H., & Onaga, K. 2001. *An autonomous distributed scheduling scheme for parallel machine problems*. *Ieice Transactions on Fundamentals of Electronics Communications and Computer Sciences*, E84A(3): 763-770.

Narasimhan, S. L., McLeavey, D. W., & Billington, P. J. 1995. *Production planning and inventory control*. USA: Prentice-Hall International Inc.

Narasimhan, S. L. & Panwalkar, S. S. 1984. *Scheduling in A 2-Stage Manufacturing Process*. *International Journal of Production Research*, 22(4): 555-564.

Nawaz, M., Ensore, E. E., & Ham, I. 1983. *A Heuristic Algorithm for the M-Machine, N-Job Flowshop Sequencing Problem*. *Omega-International Journal of Management Science*, 11(1): 91-95.

Nishi, T., Konishi, M., & Hasebe, S. 2004. *A decentralized scheduling method for flowshop problems with resource constraints*. *Electrical Engineering in Japan*, 149(1): 44-51.

Norman, B. A. 1999. *Scheduling flowshops with finite buffers and sequence dependent setup times*. *Computers & Industrial Engineering*, 36(1): 163-177.

Nowicki, E. & Smutnicki, C. 1998. *The flow shop with parallel machines: A tabu search approach*. *European Journal of Operational Research*, 106(2-3): 226-253.

O'Donovan, R., Uzsoy, R., & McKay, K. N. 1999. *Predictable scheduling of a single machine with breakdowns and sensitive jobs*. *International Journal of Production Research*, 37(18): 4217-4233.

Olumolade, M. O. & Norrie, D. H. 1996. *Reactive scheduling system for cellular manufacturing with failure-prone machines*. *International Journal of Computer Integration Manufacture*.

Ouelhadj, D. 2003. *A multi-agent system for the integrated dynamic scheduling of steel production*. Nottingham University, School of Computer Science and Information Technology.

Ovacik, I. M. & Uzsoy, R. 1994. *Rolling horizon algorithms for a single machine dynamic scheduling problems with sequence-dependent set-up times*. *International Journal of Production Research*, 32: 1243-1263.

Ovacik, I. M. & Uzsoy, R. 1995. *Rolling horizon procedures for dynamic parallel machine scheduling problems with sequence dependent set up times. International Journal of Production Research, 33: 3173-3192.*

Parthasarathy, S. & Rajendran, C. 1997. *A simulated annealing heuristic for scheduling to minimize mean weighted tardiness in a flowshop with sequence-dependent setup times of jobs - A case study. Production Planning & Control, 8(5): 475-483.*

Paul, R. J. 1979. *Production Scheduling Problem in the Glass-Container Industry. Operations Research, 27(2): 290-302.*

Petrovic, D. & Duenas, A. 2006. *A fuzzy logic based production scheduling/rescheduling in the presence of uncertain disruptions. Fuzzy Sets and Systems, 157(16): 2273-2285.*

Pinedo, M. 1995. *Scheduling. Theory, Algorithms and Systems. Englewood Cliffs : Prentice-Hall.*

Pinedo, M. 2002. *Scheduling. Theory, Algorithms and Systems. Englewood Cliffs : Prentice-Hall.*

Pinedo, M. & Chao, X. 1999. *Operations scheduling with applications in manufacturing and services. Boston: Irwin McGraw Hill.*

Pinedo, M. & Singer, M. 1999. *A shifting bottleneck heuristic for minimizing the total weighted tardiness in a job shop. Naval Research Logistics, 46: 1-17.*

Piramuthu, S., Park, S. C., Raman, N., and Shaw, M. J. *Integration of simulation modelling and inductive learning in an adaptive decision support system. Boczelc, A. and Whihston, A. Model Management Systems. 1991. IEEE Society Press.*

Portmann, M. C., Vignier, A., Dardilhac, D., & Dezalay, D. 1998. *Branch and bound crossed with GA to solve hybrid flowshops. European Journal of Operational Research, 107(2): 389-400.*

Potts, C. N. & Vanwassenhove, L. N. 1992. *Integrating Scheduling with Batching and Lot-Sizing - A Review of Algorithms and Complexity. Journal of the Operational Research Society, 43(5): 395-406.*

Pugazhendhi, S., Thiagarajan, S., Rajendran, C., & Anantharaman, N. 2004. *Generating non-permutation schedules in flowline-based manufacturing systems with sequence-dependent setup times of jobs: a heuristic approach. International Journal of Advanced Manufacturing Technology, 23(1-2): 64-78.*

Qi, J. G., Burns, G. R., & Harrison, D. K. 2000. *The application of parallel multipopulation genetic algorithms to dynamic job shop scheduling. International Journal of Advanced Manufacturing Technology, 16: 609-615.*

Quadt, D. & Kuhn, H. 2007. *A taxonomy of flexible flow line scheduling procedures. European Journal of Operational Research, 178(3): 686-698.*

Raheja, A. S. & Subramaniam, V. 2002. *Reactive recovery of job shop schedules - A review. International Journal of Advanced Manufacturing Technology*, 19: 756-763.

Rajendran, C. & Chaudhuri, D. 1992. *Scheduling in Normal-Job, Meta-Stage Flowshop with Parallel Processors to Minimize Makespan. International Journal of Production Economics*, 27(2): 137-143.

Rajendran, C. & Ziegler, H. 1997. *A heuristic for scheduling to minimize the sum of weighted flowtime of jobs in a flowshop with sequence-dependent setup times of jobs. Computers & Industrial Engineering*, 33(1-2): 281-284.

Rajendran, C. & Ziegler, H. 2003. *Scheduling to minimize the sum of weighted flowtime and weighted tardiness of jobs in a flowshop with sequence-dependent setup times. European Journal of Operational Research*, 149(3): 513-522.

Raman, N., Rachamadugu, R. V., & Talbot, F. B. 1989. *Real-Time Scheduling of An Automated Manufacturing Center. European Journal of Operational Research*, 40(2): 222-242.

Riane, F. 1998. *Scheduling Hybrid Flowshops: Algorithms and Applications. Facultés Universitaires Catholiques de Mons.*

Riane, F., Artiba, A., & Elmaghraby, S. E. 1998. *A hybrid three-stage flowshop problem: efficient heuristics to minimize makespan. European Journal of Operational Research*, (109): 321-329.

Riggs, J. L. 1987. *Production systems: planning, analysis, and control. USA: Waveland Press Inc.*

Rinnooy Kan, A. H. G. 1976. *Machine Scheduling Problems: Classification, Complexity and Computations. The Hague.*

Rios-Mercado, R. Z. & Bard, J. F. 2003. *The flow shop scheduling polyhedron with setup times. Journal of Combinatorial Optimization*, 7(3): 291-318.

Rios-Mercado, R. Z. & Bard, J. F. 1998b. *Heuristics for the flow line problem with setup costs. European Journal of Operational Research*, 110(1): 76-98.

Rios-Mercado, R. Z. & Bard, J. F. 1998a. *Computational experience with a branch-and-cut algorithm for flowshop scheduling with setups. Computers & Operations Research*, 25(5): 351-366.

Rios-Mercado, R. Z. & Bard, J. F. 1999. *An enhanced TSP-based heuristic for makespan minimization in a flow shop with setup times. Journal of Heuristics*, 5(1): 53-70.

Rohlder, T. R. & Scudder, G. D. 1993. *Comparing performance measures in dynamic job shops: economics vs. time. International Journal of Production Economics*, 32: 160-183.

Rossi, A. & Dini, G. 2000. *Dynamic scheduling of FMS using a real-time genetic algorithm. International Journal of Production Research*, 38(1): 1-20.

Roundy, R. 1992. *Cyclic Schedules for Job Shops with Identical Jobs. Mathematics of Operations Research*, 17(4): 842-865.

Rovithakis, G. A., Perrakis, S. E., & Christodoulou, M. A. 2001. *Application of a neural-network scheduler on a real manufacturing system. IEEE Transactions on Control Systems Technology*, 9(2): 261-270.

Ruiz, R. & Maroto, C. 2005a. *A comprehensive review and evaluation of permutation flowshop heuristics. European Journal of Operational Research*, 165(2): 479-494.

Ruiz, R. & Maroto, C. 2006. *A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility. European Journal of Operational Research*, 169(3): 781-800.

Ruiz, R., Maroto, C., & Alcaraz, J. 2005b. *Solving the flowshop scheduling problem with sequence dependent setup times using advanced metaheuristics - Discrete optimization. European Journal of Operational Research*, 165(1): 34-54.

Ruiz, R., Serifoglu, F. S., & Urlings, T. 2008a. *Modeling realistic hybrid flexible flowshop scheduling problems. Computers & Operations Research*, 35(4): 1151-1175.

Ruiz, R. & Stutzle, T. 2008b. *An Iterated Greedy heuristic for the sequence dependent setup times flowshop problem with makespan and weighted tardiness objectives. European Journal of Operational Research*, 187(3): 1143-1159.

Sabuncuoglu, I. & Bayiz, M. 2000. *Analysis of reactive scheduling problems in a job shop environment. European Journal of Operational Research*, 126(3): 567-586.

Sabuncuoglu, I. & Comlekci, A. 2002. *Operation-based flowtime estimation in a dynamic job shop. Omega*, 30(6): 423-442.

Sabuncuoglu, I. & Karabuk, S. 1999. *Rescheduling frequency in an FMS with uncertain processing times and unreliable machines. Journal of Manufacturing Systems*, 18(4): 268-283.

Sabuncuoglu, I. & Kizilisik, O. B. 2003. *Reactive scheduling in a dynamic and stochastic FMS environment. International Journal of Production Research*, 41(17): 4211-4231.

Sadeh, N. M., Otsuka, S., and Schnellbach, R. *Redictive and reactive scheduling with micro-boss scheduling and control system. 1993. Centre for Integrated Manufacturing Decision Systems, Carnegie Melon University.*

Salvador, M. S. *A solution to a special class of flow-shop scheduling problem. Symposium on the Theory of Scheduling and its Applications*, 83-91. 1973. Berlin (Germany), Springer-Verlag.

Schneeweiss, C. 2003. *Distributed Decision Making. Springer.*

Scudder, G. D. & Smith, D. 1989. *Application of the net present value criteriun in ramdom and flow shop scheduling. Decision Science*, 20(602): 622.

Serifoglu, F. S. & Ulusoy, G. 2004. Multiprocessor task scheduling in multistage hybrid flowshops: a genetic algorithm approach. *Journal of the Operational Research Society*, 55(5): 504-512.

Sha, L. & Goodenough, J. B. 1990. Real-time scheduling theory and ada. *IEEEComputer*, 53-62.

Shafaei, R. & Brunn, P. 2000. Workshop scheduling using practical (inaccurate) data Part 3: A framework to integrate job releasing, routing and scheduling functions to create a robust predictive schedule. *International Journal of Production Research*, 38(1): 85-99.

Shafaei, R. & Brunn, P. 1999b. Workshop scheduling using practical (inaccurate) data Part 2: An investigation of the robustness of scheduling rules in a dynamic and stochastic environment. *International Journal of Production Research*, 37(18): 4105-4117.

Shafaei, R. & Brunn, P. 1999a. Workshop scheduling using practical (inaccurate) data Part 1: The performance of heuristic scheduling rules in a dynamic job shop environment using a rolling time horizon approach. *International Journal of Production Research*, 37(18): 3913-3925.

Shen, V. Y. and Chen, Y. E. A scheduling strategy for the flowshop problem in a system with two classes of processors. *Conference on Information and Systems Science. Proceedings* , 645-649. 1972.

Shen, W. and Norrie, D. H. *Agent-Based Manufacturing Enterprise Infrastructure for Distributed Integrated Intelligent Manufacturing Systems. Proceedings of PAAM'98* . 1999. London, UK.

Simons, J. V. 1992. Heuristics in Flow-Shop Scheduling with Sequence Dependent Setup Times. *Omega-International Journal of Management Science*, 20(2): 215-225.

Singer, M. 2000. Forecasting policies for scheduling a stochastic due date job shop. *Computer and Operations Research*, 28: 193-207.

Siwamogsatham, T. & Saygin, C. 2004. Auction-based distributed scheduling and control scheme for flexible manufacturing systems. *International Journal of Production Research*, 42(3): 547-572.

Slany, W. 1996. Scheduling as a fuzzy multiple criteria optimization problem. *Fuzzy Sets and Systems*, 78(2): 197-222.

Sleator, D. D. & Tarjan, R. E. 1985. Amortized Efficiency of List Update and Paging Rules. *Communications of the Acm*, 28(2): 202-208.

Smith, S. F. A constraint-based framework for reactive management of factory schedules. *International Conference on Expert Systems and Leading Edge in Production Planning and Control* . 1987. Hilton Head, South Carolina.

Smith, S. F. A methodology and architecture for reactive scheduling. 1992. The Robotic Institute, Carnegie Mellon University.

Smith, S. F., Keng, N., and Kempf, K. *Exploiting local flexibility during execution of pre-computed schedules*. 1990. *The Robotic Institute, Carnegie Mellon University and Knowledge based Laboratory, Intel Corporation*.

Soewandi, H. & Elmaghraby, S. E. 2001. *Sequencing three-stage flexible flowshops with identical machines to minimize makespan*. *IIE Transactions*, 33(11): 985-993.

Soewandi, H. & Elmaghraby, S. E. 2003. *Sequencing on two-stage hybrid flowshops with uniform machines to minimize makespan*. *IIE Transactions*, 35(5): 467-477.

Sonmez, A. I. & Baykasoglu, A. 1998. *A new dynamic programming formulation of (n x m) flowshop sequencing problems with due dates*. *International Journal of Production Research*, 36(8): 2269-2283.

Srikar, B. N. & Ghosh, S. 1986. *A Milp Model for the N-Job, M-Stage Flowshop with Sequence Dependent Set-Up Times*. *International Journal of Production Research*, 24(6): 1459-1474.

Sriskandarajah, C. & Sethi, S. P. 1989. *Scheduling Algorithms for Flexible Flowshops - Worst and Average Case Performance*. *European Journal of Operational Research*, 43(2): 143-160.

Stafford, E. F. & Tseng, F. T. 1990. *On the Srikar-Ghosh Milp Model for the Nxm Sdst Flowshop Problem*. *International Journal of Production Research*, 28(10): 1817-1830.

Stafford, E. F. & Tseng, F. T. 2002. *Two models for a family of flowshop sequencing problems*. *European Journal of Operational Research*, 142(2): 282-293.

Stania, L., Gomez-Gasquet, P., and Andres, C. *Estudio Experimental sobre el Comportamiento de Reglas de Despacho en un Taller de Máquinas Paralelas no Relacionadas con Llegada Dinámica de Trabajos y Setup Dependiente de la Secuencia*. XXIX Congreso Nacional de Estadística e Investigación Operativa. Actas del Congreso, 475-476. 15-5-2006. Tenerife.

Stankovic, J. A., Spuri, M., Marco, D. N., & Buttazzo, G. C. 1995. *Implications of classical scheduling results for real-time systems*. *IEEE Computer*, 16-25.

Stinson, J. P. & Smith, A. W. 1982. *A Heuristic Programming Procedure for Sequencing the Static Flowshop*. *International Journal of Production Research*, 20(6): 753-764.

Stoop, P. P. M. & Weirs, V. C. S. 1996. *The complexity of scheduling in practice*. *International Journal of Operations and Production Management*, 16(10): 37-53.

Subramaniam, V., Raheja, A. S., & Reddy, K. R. B. 2005a. *Reactive repair tool for job shop schedules*. *International Journal of Production Research*, 43(1): 1-23.

Subramaniam, V., Raheja, A. S., & Reddy, K. R. B. 2005b. *Reactive repair tool for job shop schedules*. *International Journal of Production Research*, 43(1): 1-23.

Sule, D. R. & Huang, K. Y. 1983. *Sequency on 2 and 3 Machines with Setup, Processing and Removal Times Separated*. *International Journal of Production Research*, 21(5): 723-732.

Sun, D. & Lin, L. 1994. A Dynamic Job-Shop Scheduling Framework - A Backward Approach. *International Journal of Production Research*, 32(4): 967-985.

Sun, J. & Xue, D. 2001. A dynamic reactive scheduling mechanism for responding to changes of production orders and manufacturing resources. *Computers in Industry*, 46(2): 189-207.

Sun, J. U. & Hwang, H. C. 2001. Scheduling problem in a two-machine flow line with the N-step prior-job-dependent set-up times. *Journal of Systems Science*, 32: 375-385.

Suresh, V. & Chaudhuri, D. 1993. Dynamic Scheduling - A Survey of Research. *International Journal of Production Economics*, 32(1): 53-63.

Sycara, K. P. & Miyashita, K. 1994. Adaptive Schedule Repair. *Knowledge-Based Reactive Scheduling*, 15: 107-124.

Sycara, K. P., Roth, S. F., Sadeh, N., & Fox, M. S. 1991. Resource-Allocation in Distributed Factory Scheduling. *Ieee Expert-Intelligent Systems & Their Applications*, 6(1): 29-40.

Szelke, E. & Kerr, R. M. 1994. Knowledge-Based Reactive Scheduling. *Production Planning & Control*, 5(2): 124-145.

Szelke, E. & Markus, G. 1997. A learning reactive scheduler using CBR/L. *Computers in Industry*, 33(1): 31-46.

Szwarc, W. & Gupta, J. N. D. 1987. A Flowshop Problem with Sequence-Dependent Additive Setup Times. *Naval Research Logistics*, 34(5): 619-627.

Tabe, T. & Salvendy, G. 1988. Toward a hybrid intelligent system for scheduling and rescheduling of FMS. *International Journal of Computer Integration Manufacture*, 1(3): 154-164.

Tang, L. X., Luh, P. B., Liu, J. Y., & Fang, L. 2002. Steel-making process scheduling using Lagrangian relaxation. *International Journal of Production Research*, 40(1): 55-70.

Tang, L. X. & Wang, X. P. 2008. A predictive reactive scheduling method for color-coating production in steel industry. *International Journal of Advanced Manufacturing Technology*, 35(7-8): 633-645.

Tang, Z. B. & Shi, L. Y. 1992. Note on Distributed Scheduling Based on Due Dates and Buffer Priorities. *Ieee Transactions on Automatic Control*, 37(10): 1661.

Tavakkoli-Moghaddam, R., Rahimi-Vahed, A., & Mirzaei, A. H. 2007. A hybrid multi-objective immune algorithm for a flow shop scheduling problem with bi-objectives: Weighted mean completion time and weighted mean tardiness. *Information Sciences*, 177(22): 5072-5090.

Tharumarajah, A. & Bemelman, R. 1997. Approaches and issues in scheduling a distributed shop-floor environment. *Computers in Industry*, 34(1): 95-109.

Townsend, W. 1977. Sequencing N Jobs on M Machines to Minimize Maximum Tardiness - A Branch-And-Bound Solution. *Management Science*, 23(9): 1016-1019.

Tseng, F. T., Gupta, J. N. D., & Stafford, E. F. 2006. A penalty-based heuristic algorithm for the permutation flowshop scheduling problem with sequence-dependent set-up times. *Journal of the Operational Research Society*, 57(5): 541-551.

Tseng, F. T. & Stafford, E. F. 2001. Two MILP models for the N x M SDST flowshop sequencing problem. *International Journal of Production Research*, 39(8): 1777-1809.

Uskup, E. & Smith, S. B. 1975. Branch-And-Bound Algorithm for 2-Stage Production-Sequencing Problems. *Operations Research*, 23(1): 118-136.

Vicens, E. 1988. Sistema Integrado de Gestión de Producción en una Empresa de Fabricación de Muebles. *Planificación Jerárquica y de Requerimientos de Materiales*.

Vicens, E., Poler, R., Ortiz, A., Albarracín, J. M., García, J. P., Andrés, C., & Alemany, M. M. E. 1999. *Apuntes de Gestión Industrial en Sistemas de producción Inventario*. Editorial de la Universidad Politécnica de Valencia.

Vieira, G. E., Herrmann, J. W., & Lin, E. 2003. Rescheduling manufacturing systems: A framework of strategies, policies, and methods. *Journal of Scheduling*, 6(1): 39-62.

Vieira, G. E., Herrmann, J. W., & Lin, E. 2000b. Analytical models to predict the performance of a single-machine system under periodic and event-driven rescheduling strategies. *International Journal of Production Research*, 38(8): 1899-1915.

Vieira, G. E., Herrmann, J. W., & Lin, E. 2000a. Predicting the performance of rescheduling strategies for parallel machine systems. *Journal of Manufacturing Systems*, 19(4): 256-266.

Vignier A., Billaut, J. C., and Proust, C. *Les Problemes d'ordonnement de type flow shop hybride. Etat de l'art. Journées d'Etude: affectacion et ordennancement, CNRS / GdR Automatique, GT3, 7-47, Tours. 1995.*

Vignier A., Billaut, J. C., and Proust, C. *Solving k stage hybrid flowshop scheduling problems". Multiconference of Computational Engineering in Systems Applications (CESA'96) , 250-258. 1996a. Lille (France).*

Vignier A., Billaut, J. C., Proust, C., and Tkindt, V. *Resolution of some two stage hybrid flowshop scheduling problems. EEE / Systems, Man and Cybernetics (SMC'96). 2934-2941. 1996b. Beijing (China).*

Vignier, A., Billaut, J. C., & Proust, C. 1999. Hybrid flowshop scheduling problems: State of the art. *Rairo-Recherche Operationnelle-Operations Research*, 33(2): 117-183.

Wardono, B. & Fathi, Y. 2004. A tabu search algorithm for the multi-stage parallel machine problem with limited buffer capacities. *European Journal of Operational Research*, 155(2): 380-401.

Wilbrecht, J. K. & Prescott, W. B. 1969. *Influence of Setup Time on Job Shop Performance. Management Science Series B-Application*, 16(4): B274-B280.

Wilson, H. G. & Mardis, B. J. 1983. *Modifying job sequence rules for work in process inventory reduction. IIE Transactions*, 15: 320-323.

Wittrock, R. J. 1988. *An Adaptable Scheduling Algorithm for Flexible Flow Lines. Operations Research*, 36(3): 445-453.

Wu, D. S., Storer, R. H., and Chang, P. C. *A rescheduling procedure for manufacturing systems under random disruptions. Joint USA/Germany Conference on New Direction of Operations Research in Manufacturing. Proceedings of Joint USA/Germany Conference on New Direction of Operations Research in Manufacturing.* 292-306. 1991.

Wu, D. S. & Wysk, R. A. 1989. *An application of discrete-event simulation to on-line control and scheduling of flexible manufacturing. International Journal of Production Research*, 27(9).

Wu, S. D., Byeon, E. S., & Storer, R. H. 1999. *A graph-theoretic decomposition of the job shop scheduling problem to achieve scheduling robustness. Operations Research*, 47(1): 113-124.

Wu, S. D., Storer, R. H., & Chang, P. C. 1993. *One machine rescheduling heuristic with efficiency and stability as criteria. Computer and Operations Research*, 20: 1-14.

Yamamoto, M. & Nof, S. Y. 1985. *Scheduling Rescheduling in the Manufacturing Operating System Environment. International Journal of Production Research*, 23(4): 705-722.

Yang, T., Kuo, Y., & Cho, C. 2007. *A genetic algorithms simulation approach for the multi-attribute combinatorial dispatching decision problem. European Journal of Operational Research*, 176(3): 1859-1873.

Yang, W. H. & Liao, C. J. 1999. *Survey of scheduling research involving setup times. International Journal of Systems Science*, 30(2): 143-155.

Yellig, E. J. & Mackulak, G. T. 1997. *Robust deterministic scheduling in stochastic environments: the method of capacity hedge points. International Journal of Production Research*, 35: 369-379.

Ying, K. C. & Lin, S. W. 2006. *Multiprocessor task scheduling in multistage hybrid flow-shops: an ant colony system approach. International Journal of Production Research*, 44(16): 3161-3177.

Yip, Y., Cheng, C. Y., & Low, C. 2006. *Sequencing of an M machine flow shop with setup, processing and removal times separated. International Journal of Advanced Manufacturing Technology*, 30(3-4): 286-296.

Yoshida, T. & Hitomi, K. 1979. *Optimal 2-Stage Production Scheduling with Setup Times Separated. Aiie Transactions*, 11(3): 261-263.

Zandieh, M., Ghomi, S. M. T. F., & Hussein, S. M. M. 2006. *An immune algorithm approach to hybrid flow shops scheduling with sequence-dependent setup times. Applied Mathematics and Computation, 180(1): 111-127.*

Zhu, X. Y. & Wilhelm, W. E. 2006. *Scheduling and lot sizing with sequence-dependent setup: A literature review. IIE Transactions, 38(11): 987-1007.*

Zimmermann, H. J. 1996. *The fuzzy set theory and its applications. Springer; 3 edition (April 30, 1996).*

**Estado del Arte de los Sistemas
Multiagente y su Aplicación a los
Sistemas de Fabricación y su
Programación de la Producción**

Capítulo

4

Tabla de contenido

4. Capítulo 4 201

4.1 Introducción 201

4.2 Teoría de Agentes y Sistemas Multiagente 203

4.2.1 Introducción 203

4.2.2 Agentes. Conceptos básicos 206

4.2.2.1 Agentes, Sistemas Multiagente y Sistemas Basados en Agentes 207

4.2.2.2 Clasificación de los Agentes 212

4.2.3 Arquitectura de Agentes 214

4.3 Los Sistemas Multiagente: La Organización 217

4.3.1 La Organización 220

4.3.1.1 La Colaboración 224

4.3.1.2 La Coordinación 227

4.3.2 La Negociación 228

4.4 Los Sistemas Multiagente: El Intercambio de Conocimiento 230

4.4.1 La Comunicación 230

4.4.2 Ontologías 232

4.5 Análisis, Diseño e Implementación de los Sistemas Multiagente 234

4.5.1 Organizaciones para la Estandarización de los Sistemas basados en Agentes . 235

4.5.2 Metodologías y Herramientas para el Diseño de Sistemas Multiagente 238

4.5.3 Plataformas para la Implementación de Sistemas Multiagentes 239

4.6 Ventajas del Uso de Sistemas Multiagente 242

4.7 Sistemas Inteligentes de Fabricación basados en Agentes 243

4.7.1	Planificación, Programación y Control de la Producción	244
4.7.2	Integración de la Planificación y la Programación de la Producción.....	245
4.7.3	Sistemas Holónicos.....	249
4.8	Aplicaciones de los Sistemas Multiagente a la Programación de la Producción	251
4.8.1	Soluciones basadas en Teoría de Agentes	253
4.8.2	Integración de la Programación de la Producción en el SPCO.....	259
4.8.3	Habilidades Sociales en la Aplicación de los SMA	261
4.8.4	Aprendizaje en la Aplicación de los SMA	263
4.8.5	Influencia de los SMA en las Soluciones.....	264
4.9	Conclusiones	265
4.10	Bibliografía	267

Índice de figuras

Figura 4-1. Agente en relación con su entorno (elaboración propia).....	207
Figura 4-2. Clasificación de los Agentes según (Nwana, H. S. 1995).....	212
Figura 4-3. Áreas que Influyen en los Agentes según (Altmann, J. et al. 2000).	213
Figura 4-4. Arquitectura de un Agente BDI (elaboración propia).	214
Figura 4-5. Arquitectura de un Agente Reactivo (elaboración propia).....	215
Figura 4-6. Representación de una arquitectura híbrida (Muller, J. P. 1997).....	217
Figura 4-7. Organización de la Sociedad, (Corchado, J. M. et al. 2002).....	224
Figura 4-8. Estructura del protocolo Contrat-Net entre diferentes niveles de (Cardon, A. et al. 2000).....	255

Índice de tablas

Tabla 4-1. Tipos de interacciones entre agentes(Corchado, J. M. et al. 2002).	218
Tabla 4-2. Clasificación de los procedimientos para la Colaboración entre Agentes de una Organización (elaboración propia).	225
Tabla 4-3. Revisión de Plataformas/Herramientas de Desarrollo de Agentes basadas en JAVA basada en (Vrba, P. 2003).	241
Tabla 4-4. Resumen basado en (Shen, W. M., Wang, L. H., & Hao, Q. 2006) sobre proyectos de investigación que trabajan en la integración de los procesos de Planificación y Programación aplicando teoría de Agentes.	248
Tabla 4-5. Tabla comparativa Holones vs Agentes de (Giret-Boggino, A. S. 2005).....	250
Tabla 4-6. Resumen de Proyecto orientados a la Programación de la Producción utilizando Sistemas Multiagentes basado en (Shen, W. M. et al. 2006).	252

4.1 Introducción

El estudio de los comportamientos inteligentes colectivos consecuencia de la interacción de diversos elementos llamados **Agentes** es el objetivo fundamental de la Inteligencia Artificial Distribuida (AID)(Avouris, N. M. & Gasser, L. 1992). La IAD ha experimentado una evolución apreciable en su corta vida, que podemos clasificar según (Iglesias-Fernández, C. A. 1998) en tres etapas cronológicamente diferenciadas:

- La IAD “Clásica”,(Moulin, B. & Chiab-draa, B. 1996), (Huhns, M. N. & Gasser, L. 1989), empieza a estudiar la conducta colectiva en contraposición de la Inteligencia Artificial que estudia la individual.
- La IAD “Autónoma”, (Castelfranchini, C. & Conte, R. 1996), se centra en los Agentes individuales dentro de un mundo social.
- La IAD “Comercial”, (Hedberg, S. 1996), (Nwana, H. S. 1995) se centra en la aplicación de las IAD clásica y autónoma al desarrollo de Agentes, normalmente denominado Agentes Software, con características muy diferenciadas que están siendo explotados de forma comercial.

La aparición de los Agentes Software supuso la irrupción de un nuevo paradigma en el desarrollo del software que no solo influyó en las fases de conceptualización, diseño e implementación del mismo, sino también en la aplicabilidad de las soluciones propuestas.

La **Teoría de Agentes y Sistemas Multiagente (SMA)** ha permitido dar un mayor realismo al tratamiento de ciertos problemas, incorporando a los modelos ciertas características que normalmente no se tienen en cuenta, o se hace de forma parcial, por la dificultad que incorporan al proceso de resolución del mismo. **Cuando el problema que se desea solucionar incluye características como: Gran Tamaño, Alto Grado de Incertidumbre, Dinamismo, o Singularidad en la distribución de sus elementos, es una excelente opción la aplicación de Agentes para obtener grandes ventajas en el proceso de resolución.** En este sentido, el problema de la Programación de la Producción es un problema susceptible de ser abordado mediante Agentes, ya que suelen considerarse gran parte de los factores mencionados anteriormente.

Entre todas las características indicadas en el párrafo anterior, quizás sean dos las que predominen sobre el resto: Singular Distribución y Dinamismo del problema. Los problemas

Distribuidos han sido la base de la IAD. Esta propiedad ha permitido abordar tanto problemas Distribuidos por su propia naturaleza como; redes de ordenadores, distribución eléctrica, redes de semáforos, etc., como también subdividir problemas centralizados como; detección de errores ((Jennings, N. R., Corera, J. M., & Laresgoiti, I. 1995)), aplicaciones industriales ((Parunak, H. V. 1993)), procesamiento de lenguaje natural ((Sabah, G. 1990)), etc., en módulos y submódulos de forma que un individuo o un pequeño grupo de la sociedad afronta ese submódulo, pero de una forma coordinada con el resto de individuos.

Por otro lado, **la teoría de Sistemas Multiagente permite ofrecer soluciones dinámicas, a problemas dinámicos.** Esto se debe a que la mayor parte de las soluciones están soportadas en herramientas software, y desde el punto de vista de la tecnología software los Agentes se caracterizan como un entorno abierto que favorece el tratamiento variable en el tiempo de los problemas. Según (Hewitt, C. 1986) este concepto implica que la estructura del sistema es capaz de modificarse dinámicamente por sí misma. Esto puede suponer que los elementos que la forman en un instante dado no son conocidos con anterioridad o pueden cambiar a lo largo del tiempo. Normalmente en un sistema abierto los elementos suelen ser heterogéneos en su composición. De esta forma se puede decir **que la ingeniería del software basada en Agentes está relacionada con los sistemas complejos distribuidos en sistemas/entornos abiertos.** Esta tecnología no sólo puede ser utilizada para modelar una realidad existente sino también para modelar y diseñar una nueva propuesta que suponga el desarrollo de un nuevo sistema complejo.

Las técnicas de Inteligencia Artificial se utilizan en la Fabricación Inteligente desde hace más de dos décadas. Sin embargo, las recientes técnicas desarrolladas en los Sistemas Multiagente en el nuevo dominio de la Inteligencia Artificial Distribuida han proporcionado nuevas e interesantes posibilidades. Durante los últimos años los investigadores han aplicado las técnicas basadas en la Teoría de Agentes a los Sistemas Productivos; Planificación, Programación y Control, Gestión de Materiales, Gestión de la Cadena de Suministro, Logística, etc.

La Programación de la Producción ha sido ampliamente estudiada en la literatura mediante varios métodos; Heurísticas, Propagación de Restricciones, Recocido Simulado, Búsqueda Tabú, Algoritmos Genéticos, Redes Neuronales, etc. **La tecnología de Agentes se ha convertido en una nueva y reciente forma de abordar los problemas relacionados con la Toma de Decisiones en el entorno de los Procesos Productivos.**

En este capítulo se dedica un primer bloque de apartados a tratar los aspectos relacionados con la Teoría de los Agentes Software (Agencia). En el primero de los apartados de este bloque se, correspondiente con el apartado 4.2 se realiza un recorrido por los conceptos básicos y el

análisis del Agente como unidad y de forma individual. En el segundo, correspondiente con el apartado 4.3, se analizarán los aspectos colectivos relativos a la organización de lo que se ha denominado Sistemas Multiagentes. En el apartado 4.4 se consideran las cuestiones relativas al intercambio de conocimiento, es decir el lenguaje y el significado de los mensajes. En el apartado 4.5 se proporciona una visión general sobre el conjunto de elementos que intervienen y condicionan el desarrollo de un sistema basado en Agentes. En el apartado 4.6 se destacan las ventajas del uso de los Sistemas Multiagente en la búsqueda de soluciones para el cálculo de Programas Productivos. Un segundo bloque está formado por el apartado 4.7 y 4.8. En el se analiza la aplicación de los Sistemas basados en Agentes a la Gestión de la Producción en todos sus aspectos, y a la Programación de la Producción en concreto. El apartado 4.7 presenta la aplicación que se ha realizado hasta la fecha en lo que se ha denominado Fabricación Inteligente. El apartado 4.8 se realiza un profundo análisis bibliográfico sobre la aplicación de Sistemas Multiagente a la generación de Programas Productivos. Finalmente en el apartado 4.9 se realiza una reflexión a modo de conclusión de los contenidos presentados en el capítulo.

4.2 Teoría de Agentes y Sistemas Multiagente

4.2.1 Introducción

Aunque la IAD es la referencia básica que se sigue en este capítulo, ya que su producción científica es la más cercana al enfoque que se desea realizar, se debe tener en cuenta que en el estudio de la teoría relacionada con los Agentes y de los Sistemas de Agentes hay múltiples disciplinas que realizan aportaciones complementando el mismo desde diversos enfoques. En muchos casos la influencia se ha producido y se produce en un sentido bidireccional. Para (Luck, M., McBurney, P., Shehory, O., Willmott, S., & Agentlink Community 2005) algunas de las disciplinas cercanas son:

- **Economía:** La aplicación de la tecnología a los problemas de asignación recursos distribuidos. Algunos ejemplos incluye la “Teoría de Juegos”, que estudia las propiedades de las interacciones económicas formalizadas entre participantes, y mecanismos de la teoría del diseño, que considera el problema del diseño óptimo de los mecanismos de asignación de recursos. Las subastas son los mecanismos más comúnmente estudiados en economía, y la combinación de la Teoría Económica, la Investigación Operativa y la Informática ha producido una nueva disciplina: El diseño de subastas basado en ordenador.
- **Lógica:** Al igual que la Informática, los últimos años son florecientes para las aplicaciones de la Lógica Formal a la tecnología de Agentes, especialmente la Lógica

Modal y Temporal. La Lógica del conocimiento y de las creencias (Lógica Epistémica) ha sido usada para representar los estados internos de los Agentes de una forma computacional. La Lógica Deóntica ha sido usada para representar obligaciones y normas en los Sistemas de Agentes. La Lógica de los Procesos y la Dinámica ha sido usada para razonar sobre la interacción entre los Agentes.

- **Filosofía:** Un buen número de áreas de la Filosofía han tenido influencia en la Teoría y Diseño de Agentes. La Filosofía de las creencias y las intenciones, por ejemplo, lideran directamente el modelo BDI de agente software racional usado para representar los estados internos de los Agentes autónomos. La Teoría del Habla, una rama de la Filosofía del lenguaje, ha sido usada para proporcionar semántica al lenguaje comunicación FIPA (Foundation for Intelligent Physical Agents) de Agentes (FIPA 2008). De la misma forma, la Teoría de la Argumentación, que data de la época aristotélica, se usa ahora para por los diseñadores de los protocolos de interacción entre Agentes para el diseño de lenguajes capaces de soportar argumentación y razonamiento no deductivo. Los temas de veracidad y obligaciones en los Sistemas Multiagente se han basado en la Teoría Filosófica de la delegación y las normas.
- **Biología:** La metamorfosis Biológica ha sido muy influyente en la computación en las últimas tres décadas, por ejemplo, en el desarrollo de la computación Evolucionista y las Redes Neuronales. Los modelos de los Sistemas Multiagente han encontrado aplicación para la simulación de Sistemas Biológicos, tales como las poblaciones de peces en el mar del Norte.
- **Ciencias Sociales:** Aunque quizás tenga relaciones menos desarrolladas que las económicas, hay varios puntos de relación entre las Ciencias Sociales y la tecnología de los Sistemas Multiagente. Dado que los Sistemas Multiagente están enmarcados dentro de la interacción, entidades autónomas, el Diseño Organizacional y la Teoría Política son muy importantes en su diseño y evaluación. Debido a la importancia de la predicción de las relaciones de un Agente con el resto, la Sociología y las Teorías Legales y de las Normas de Comportamiento de grupos son relevantes, junto con las Teorías Psicológicas de la veracidad y la persuasión. También la Teoría del Marketing puede tener influencia debido a la relación de los Agentes con otros Agentes y con los Humanos.

La evolución definitiva de la Teoría de Agentes y Sistemas de Agentes se produjo de la mano de la IAD. Como ya se ha indicado según evolución cronológica establecida por (Iglesias-Fernández, C. A. 1998), en la IAD se establecen tres etapas: Clásica, Autónoma y Comercial.

En la Etapa Clásica, la IAD es un subcampo de la Inteligencia Artificial, ver (O'Hare, G. M. P. & Jennings, N. R. 1996). En esta etapa se consolidan los trabajos relacionados con las técnicas necesarias para la coordinación y distribución del conocimiento. Durante este periodo los términos nodo y Agente se encuentran habitualmente para referirse a entidades similares. Así mismo se pueden distinguir dos tipos de áreas de investigación relevantes, (Bond, A. H. 1988):

- Resolución (cooperativa) de Problemas Distribuidos, en el que se estudia como un conjunto de “nodos” o “módulos” cooperan para dividir el conocimiento de un problema y el desarrollo de la solución.
- **Sistemas Multiagente** (*Multiagent Systems*), que estudia la coordinación de la conducta inteligente entre un conjunto de Agentes Inteligentes Autónomos.

La diferencia fundamental entre ambas áreas estriba en que mientras en la primera existe un plan inicial, y normalmente centralizado, en los SMA el sistema es más flexible y los Agentes pueden decidir en cada momento las interacciones más adecuadas.

Durante esta Etapa Clásica se estudiaron los problemas básicos, que consolidaron la IAD y dieron pie a la Etapa Autónoma en la que toma mayor protagonismo la Teoría de Agentes. Algunos de estos problemas, expuestos por (Iglesias-Fernández, C. A. 1998) fueron:

- ¿Cómo formular, describir, descomponer y asignar y sintetizar los resultados entre un grupo de Agentes Inteligentes?
- ¿Cómo capacitar a los Agentes para que se comuniquen e interactúen?
- ¿Cómo asegurar que los Agentes actúan coherentemente al tomar decisiones o realizar acciones, como ajustar los efectos globales de las decisiones locales y prevenir decisiones no deseadas?
- ¿Cómo capacitar a los Agentes para representar y razonar sobre acciones, planes y conocimientos de otros Agentes para coordinarse?
- ¿Cómo reconocer y reconciliar puntos de vista e intenciones conflictivas entre un conjunto de Agentes para coordinar sus acciones?
- ¿Cómo utilizar técnicas ingenieriles y desarrollar sistemas con IAD. Como diseñar plataformas SMA y metodologías de desarrollo?

En la Etapa Autónoma de la IAD, toman relevancia los aspectos relacionados con los Agentes, dejando de lado las cuestiones generales, comentadas anteriormente, de la IAD clásica y centrándose en cuestiones relativas a los individuos –Agente-. Los aspectos fundamentales abordados en esta etapa fueron, (Wooldrige, M. & Jennings, N. R. 1995):

- La Teoría de Agente: Que trata de responder a la pregunta de qué es un Agente, y utilizar formalismos matemáticos para representar y razonar sobre las propiedades de los Agentes.
- Arquitecturas de Agentes: Que trata las arquitecturas hardware y software que permiten reflejar las propiedades establecidas por los teóricos.
- Lenguajes de Agentes: Que son lenguajes software para programar y experimentar con Agentes.

Durante la Etapa Autónoma la IAD estudió los Agentes y los Sistemas Multiagente, en cuyo periodo, este ámbito se empezó a considerar un campo por sí mismo.

Una vez consolidados los aspectos tratados en la Etapa Autónoma, varias casas comercializadoras de software empezaron a adquirir, o bien a desarrollarlas sí mismas, prototipos basados en Agentes, ya fuesen aplicaciones o plataformas. Este es el comienzo de la denominada Etapa Comercializadora. Durante esta etapa los Agentes suavizan ciertas propiedades y se especializan considerablemente. Algunos de los Agentes más conocidos, según (Nwana, H. S. 1995) son: Agentes Interfaz, Agentes Móviles, Agentes Internet.

En la actualidad la Teoría de los Agentes software tiene entidad propia siendo varios los congresos relevantes en los que se tratan dichas cuestiones de forma exclusiva. Entre estos se pueden destacar **ATAL** (Agent Theories, Architectures and Languages), **AAMAS** (Autonomous Agent and Multi-Agent Systems), **HoloMAS** (Holonc and Multi-Agent Systems for Manufacturing). Existen múltiples congresos en los cuales se incluyen puntos de encuentro para los aspectos relacionados con la Teoría de los Agente Software.

4.2.2 Agentes. Conceptos básicos

En este apartado se van a introducir los elementos conceptuales básicos que permitirán diferenciar entre un Agente, un Sistema basado en Agentes y un Sistema Multiagente. Cada uno de estos conceptos supone un nivel de abstracción e implementación diferente de una solución. Es importante poder diferenciar entre los diferentes enfoques, ya que en ocasiones

se utilizan con intenciones contrapuestas, y en otras de forma conjunta y complementaria, lo que hace difícil su identificación.

Es este apartado también se presentarán algunas clasificaciones alternativas de los Agentes que permitirán completar el cocimiento que se tiene sobre ellos.

4.2.2.1 Agentes, Sistemas Multiagente y Sistemas Basados en Agentes

De manera informal se entiende que un Agente es una entidad con capacidad de computación que en un determinado entorno es capaz de actuar de forma autónoma, y con cierto grado de libertad, para alcanzar sus objetivos. El Agente es consciente de su entorno, es capaz de percibir los cambios en el mismo, y de actuar sobre éste en consecuencia.

Según la definición de (Wooldrige, M. et al. 1995) podemos decir que un Agente es todo Sistema Informático que cumple las siguientes condiciones:

- **Autonomía:** Tiene la capacidad de actuar de forma directa sin intervención humana o de otros Agentes.
- **Sociabilidad:** La capacidad de relacionarse con otros Agentes utilizando para ello algún tipo de lenguaje.
- **Reactividad:** El Agente tiene vínculos con su entorno que se manifiestan en forma de estímulos y de reacción ante los mismos.
- **Iniciativa:** Un Agente no sólo debe reaccionar ante los cambios que se producen en su entorno, sino que debe tener la capacidad de tomar la iniciativa.

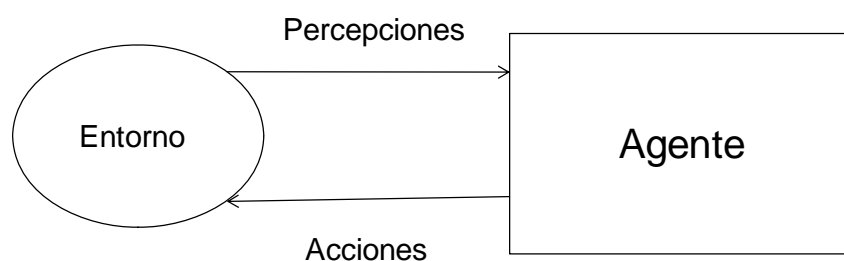


Figura 4-1. Agente en relación con su entorno (elaboración propia).

Las condiciones anteriores pueden ser satisfechas por múltiples sistemas software que sin embargo carecen de la inteligencia necesaria para denominarse Agentes. Por este motivo (Corchado, J. M. & Molina, J. M. 2002) proponen una serie de condiciones adicionales que eliminen la ambigüedad que aún permanece en la definición de (Wooldrige, M. et al. 1995).

- **Movilidad:** Habilidad de un Agente de trasladarse en una red de comunicación informática.
- **Veracidad:** Propiedad por la que un Agente no comunica información falsa intencionadamente.
- **Benevolencia:** Un Agente no tiene objetivos contradictorios y siempre intenta realizar la tarea que se le solicita.
- **Racionalidad:** Un Agente tiene unos objetivos específicos y siempre intenta llevarlos a cabo.

Según (Russell, S. & Norvig, P. 2002) la racionalidad nos lleva a vincular varios conceptos: Autonomía y Aprendizaje más Percepción. El Agente racional debe realizar todas las acciones que favorezcan la obtención de su máximo rendimiento, tomando como base las percepciones recibidas y el conocimiento obtenido. Este conocimiento implica para el autor un aprendizaje a través del ciclo de vida del Agente y un conocimiento inicial. Para (Russell, S. et al. 2002) la racionalidad de un Agente depende de:

- De la medida con la que se evalúa el grado de éxito alcanzado.
- De la secuencia de percepciones, es decir de todo aquello que el Agente ha percibido del entorno.
- Del conocimiento que el Agente posee del medio.
- De las acciones que el Agente puede llevar a cabo.

En (Shoham, Y. 1993) se define un Agente como una entidad cuyo estado está formado por componentes mentales típicos de humanos como creencias, capacidades, elecciones y compromisos. Estos estados determinan las acciones que llevan a cabo los Agentes y que están afectados por los mensajes que reciben.

Una evolución en el uso de los Agentes es el planteamiento e implementación de Sistemas Basados en Agentes. De forma intuitiva se puede decir que *un Sistema Basado en Agentes es*

aquel que utiliza los Agentes como elementos de abstracción para representar su modelo, pudiendo estar configurado por uno o varios Agentes. Por el contrario, *un Sistema Multiagente está concebido desde el primer instante como un conjunto de elementos que interactúan entre sí para alcanzar un objetivo de forma cooperativa*. En este segundo caso hay que realizar un mayor esfuerzo en la construcción de todos aquellos aspectos relacionados con la interrelación.

El estudio de Sistemas Multiagente se centra en los sistemas en los cuales varios Agentes Interactúan unos con otros. Los Agentes son considerados Entidades Autónomas. Su Interacción puede ser Cooperativa, como las hormigas de una colonia o Interesada como los agentes comerciales de un mercado. Los investigadores de SMA desarrollan ideas basadas en diversos campos fuera de la Inteligencia Artificial como la Economía, la Filosofía, los Sistemas Complejos o la Biología.

Según (Lesser, V. R. 1995), un Sistema Multiagente es un sistema computacional en el cual varios Agentes Semi-Autónomos Interactúan o trabajan juntos para realizar un conjunto de tareas o satisfacer un conjunto de objetivos. Dichos sistemas pueden implicar Agentes Homogéneos o Heterogéneos teniendo objetivos comunes o distintos. Para que los Agentes alcancen soluciones “óptimas” y compatibles a sus subproblemas, interdependientes con el mínimo uso de recursos requieren tener una visión del problema suficientemente actualizada, completa y consistente. Obtener dicha información frecuentemente no es práctico debido a:

- Ancho de banda de comunicación limitado y coste computacional del tratamiento de la información.
- La heterogeneidad de los Agentes hace difícil el intercambio de información, y la competitividad de ciertos Agentes hace que puedan desear no compartir alguna información.
- El carácter dinámico del entorno debido a los cambios en los problemas, los Agentes y los recursos, así como a la imposibilidad de predecir las acciones de los Agentes.

Según (Sycara, K. P. 1998) un sistema multiagente se caracteriza porque (1) cada Agente tiene información incompleta o no tiene capacidad para resolver el problema completo, tiene un punto de vista limitado, (2) no hay un sistema global de control, (3) los datos son descentralizados, (4) y la computación es asíncrona. Por eso según la misma autora los Sistemas Multiagentes tienen la habilidad de:

- Resolver problemas muy grandes para un solo Agente debido a que pudiera fallar por limitación de recursos, por cuellos de botella o fallos en momentos críticos.
- Permiten la interconexión de múltiples sistemas con su herencia. Para permanecer al día con las necesidades de negocio los sistemas deben de ser actualizados periódicamente.
- Proveer solución a problemas que pueden ser planteados como una sociedad de componentes-Agentes Autónomos Interactuando.
- Proporcionar soluciones que usan eficientemente información procedente de fuentes Distribuidas.
- Proporcionar soluciones en situaciones donde la experiencia está Distribuida.
- Para mejorar el rendimiento en:
 - **Eficiencia computacional**, porque se aprovecha la concurrencia.
 - **Fiabilidad**, porque se forman dinámicamente Agentes con la misma habilidad o existe una apropiada coordinación inter-Agentes.
 - **Extensibilidad**, porque el número y capacidades de los Agentes puede ser alterada.
 - **Robustez**, porque el sistema está preparado para soportar las incertidumbres mediante el intercambio adecuado de información entre Agentes.
 - **Mantenimiento**, porque el sistema está formado por componentes-Agentes que reflejan modularidad y por lo tanto un mantenimiento más sencillo.
 - **Flexibilidad**, porque Agentes con diversas capacidades pueden organizarse de forma adaptativa para resolver el problema actual.
 - **Reusabilidad**, porque la funcionalidad de los diferentes Agentes puede ser reutilizada por otros agentes o equipos de Agentes para resolver diferentes problemas.

Según (Corchado, J. M. et al. 2002) los SMA se caracterizan por una serie de propiedades: Descripción de Competencias, Modelos de Agentes, Comunicación, Comportamiento y Puntos de Interacción.

La **Descripción de Competencias** se encarga de la definición del problema en términos de tareas, subtareas y sus relaciones. Así se determina como resolver el problema, como distribuirlo entre los diferentes Agentes y las Interacciones entre éstos. (Corchado, J. M. et al.

2002) hacen referencia a una descomposición de los problemas en dos pasos. Un primer paso se realizará durante la definición del problema, a nivel de aplicación, y un segundo paso durante la fase de resolución del problema para asignar la actividad de los Agentes. Este segundo paso puede establecer asignaciones deterministas o dinámicas dando lugar a Organizaciones Adaptativas o No Adaptativas.

Los **Modelos de Agentes** conocidos son la representación que cada Agente realiza de los otros Agentes que existen en su entorno. Se trata de una representación local de la información relativa a dichos Agentes. Dicha información se organiza habitualmente como conocimiento estructurado en modelos de Agentes conocidos. Este modelo de Agentes conocidos contiene información que puede ayudar a razonar sobre otros Agentes y satisfacer las necesidades de comunicación con otros Agentes. Así mismo, permite predecir o esperar el comportamiento de otros Agentes, Planear y Coordinar actividades locales de acuerdo con una meta global, y permitir la estructura de la arquitectura de un Agente, mejorando la posibilidad de agrupar Agentes. Los autores inciden en lo deseable de la autoconfiguración dinámica de un sistema distribuido. Esta condición permite al sistema adaptar las nuevas configuraciones del grupo, o nuevos entornos, sin intervención del desarrollador, simplemente actualizando los modelos de Agentes conocidos.

La **Comunicación** se necesita para la definición del problema, su descomposición y distribución, entre los agentes. El intercambio de información o acto de comunicación se realiza por medio de un protocolo o lenguaje que puede tener diferentes niveles de complejidad.

El **Comportamiento** coherente de los agentes desde un punto de vista global es el objetivo prioritario de la formación del sistema. Dicha coherencia se puede definir como una propiedad del sistema medida por la eficiencia, calidad, claridad de una solución global y el nivel de fallos no críticos del sistema según (Bond, A. H. 1988). El comportamiento del sistema se ve influido fundamentalmente por el tipo de organización de los Agentes, que establece los distintos tipos de interacción. Los tipos de organización pueden ser Centralizados, Descentralizados o Mixtos. Dichas organizaciones se revisarán con profundidad en un apartado posterior.

Los **Puntos de Interacción** de un Agente son importantes porque establecen las ubicaciones iniciales para las actividades cooperativas entre los Agentes. Son los puntos en los que el Agente establece una relación mediante un envío o petición de información con otros Agentes de su entorno.

En este apartado se han presentado algunas ideas clave para poder distinguir entre Agentes, Sistemas basados en Agentes y Sistemas Multiagente. A continuación se ofrecen algunas alternativas para poder clasificar los tipos de Agente en función de criterios alternativos.

4.2.2.2 Clasificación de los Agentes

Según (Altmann, J., Essmayr, W., Grabner, M., Gruber, F., Klug, L., Stockner, W. et al. 2000) hay varios criterios para clasificar un agente:

- Movilidad.
- Capacidades de Interacción, y
- Atributos Ideales.

El criterio de movilidad divide a los Agentes en dos grupos: Los Agentes capaces de migrar de un entorno a otro, y los que no son capaces de trasladarse. Dichos grupos se denominan respectivamente Agentes móviles y Agentes estacionarios.

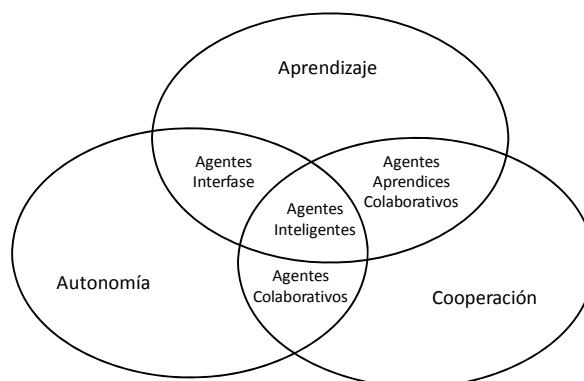


Figura 4-2. Clasificación de los Agentes según (Nwana, H. S. 1995).

El criterio basado en las capacidades de interacción también divide a los Agentes en dos grandes grupos: Los Agentes “deliberativos” o BDI con un modelo de razonamiento interno que les permite planificar y establecer contacto con otros Agentes, y los Agentes “reactivos”,

que sólo reacciona ante eventos del entorno y no disponen de un modelo de comportamiento interno.

El criterio basado en los atributos ideales necesita la definición de un conjunto mínimo de propiedades, las cuales deben de existir en todos los Agentes. (Nwana, H. S. 1995) proporciona un conjunto de atributos, explícitamente reducido, que está compuesto por: Autonomía, Aprendizaje y Cooperación.

En la Figura 4-2 se muestra la clasificación de (Nwana, H. S. 1995) intenta Asignar cada agente con un atributo dominante. Con lo que se obtienen cuatro tipos de Agente: Interfase, Colaborativos, Colaborativos y con capacidad de Aprendizaje e Inteligentes.

Otra clasificación interesante propuesta por (Altmann, J. et al. 2000) es aquella en la que los Agentes son categorizados en base al área de conocimiento que les influencia.

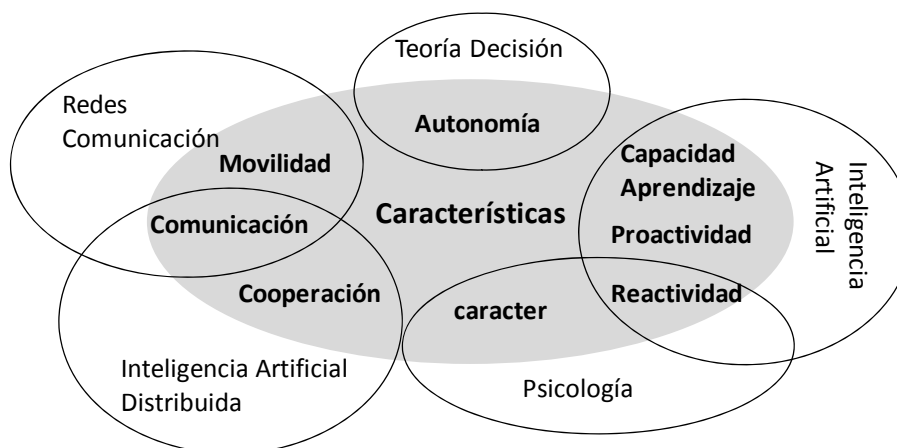


Figura 4-3. Áreas que Influyen en los Agentes según (Altmann, J. et al. 2000).

En este caso se identifican un conjunto de atributos que caracterizan a los Agentes: Reactividad, Proactividad, Aprendizaje, Autonomía, Movilidad, Comunicación, Carácter. Estos atributos no son sino características de los Agentes, algunas de las cuales ya fueron definidas anteriormente. En base a la ausencia o presencia de dichas características se puede realizar una clasificación de los Agentes.

Con las clasificaciones ofrecidas se dan por finalizado el apartado sobre conceptos básicos de la Teoría de Agentes. Antes de concluir la revisión de los conceptos relativos a los Agentes como entidades individuales se procederá a realizar un análisis sobre las arquitecturas fundamentales que se han utilizado hasta el momento para construir un Agente.

4.2.3 Arquitectura de Agentes

En un Sistema Multiagente el razonamiento individual de los Agentes debe de tener en cuenta los efectos no locales de sus acciones locales. Este aspecto ha sido ampliamente estudiado en el área de la Inteligencia Artificial. En (Muller, J. P. 1998) se realiza una revisión de las diferentes arquitecturas que pueden ser utilizadas en el diseño de un Agente. Se pueden distinguir básicamente tres tipos de Arquitecturas: Deliberativa o BDI, Reactiva e Híbrida.

En (Wooldrige, M. et al. 1995) plantearon la formalización de una lógica axiomática para el comportamiento racional de los Agentes. En los trabajos de (Shoham, Y. 1993), (Rao, A. S. & Georgeff, M. P. 1994) y (Wooldrige, M. et al. 1995) se trata la formalización de un modelo para el comportamiento de los Agentes en términos de Creencias, Deseos y objetivos o Intenciones. **Este tipo de sistemas conocidos como BDI (Belief, Desire, Intention) configuran un tipo de Arquitectura conocida como Deliberativa.** En la Figura 4-4 se muestra de forma gráfica una presentación de la arquitectura BDI.

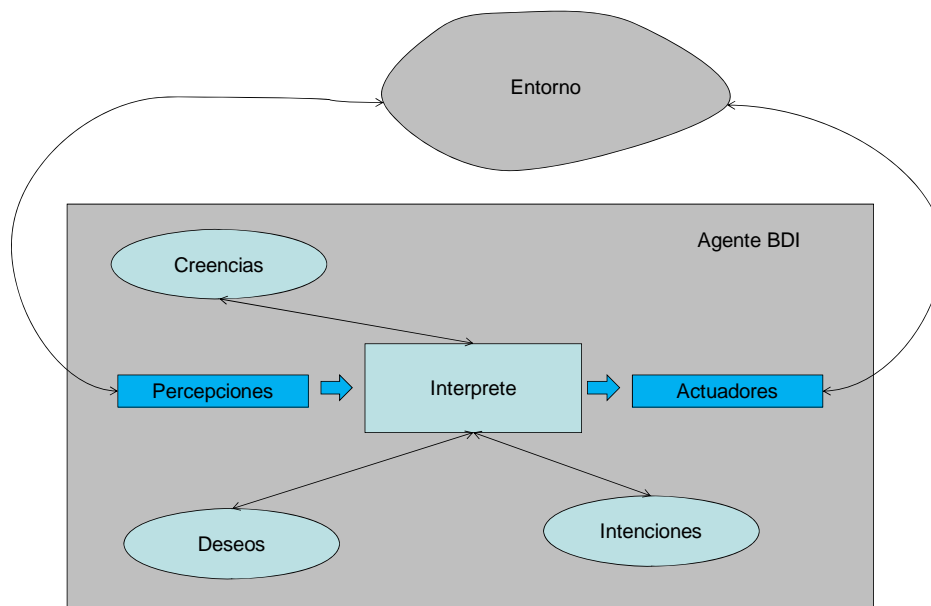


Figura 4-4. Arquitectura de un Agente BDI (elaboración propia).

Los Agentes Deliberativos mantienen una representación del mundo mediante un modelo simbólico que constituyen sus creencias. Así mismo tiene un conjunto de objetivos que deben alcanzar que son denominados deseos. Para poder alcanzar sus deseos, un Agente parte de su estado de conocimiento actual del mundo, es decir de sus creencias, y ejecuta uno o varios de sus planes posibles (intenciones). En cada momento, el Agente, en función de sus percepciones actualiza sus creencias y modifica sus intenciones en curso para adaptarse mejor

a las nuevas situaciones. Este proceso razonado puede tener un coste temporal alto que hace que en ocasiones no sea adecuado para algunas actividades, sobre todo en las que trabajan en tiempo real.

Otra aproximación es la Arquitectura Reactiva. Esta propone el desarrollo de Agentes reactivos en contraposición al comportamiento racional de los Agentes BDI. Los Agentes reactivos no tienen una representación de su entorno y actúan en forma de respuesta a estímulos. Éstos responden al estado presente del entorno y no tienen en cuenta la historia ni los planes de futuro. La mayor fuerza de los **Agentes Reactivos se basa en la idea de que interacciones simples entre Agentes pueden originar comportamientos complejos**. La simplificación de las respuestas supone un incremento importante de velocidad en las mismas y sencillez en la implementación de la solución. Este tipo de sistemas se sustenta en varias concepciones diferentes.

La primera aproximación tiene su origen en las críticas de (Brooks, R. A. 1991) a los Agentes Deliberativos y sus aseveraciones de que primero la Inteligencia es fruto de la interacción entre un Agente y su entorno, y segundo de que el comportamiento Inteligente emerge de la interacción de varios comportamientos simples organizados en niveles a través de una relación maestro-esclavo.

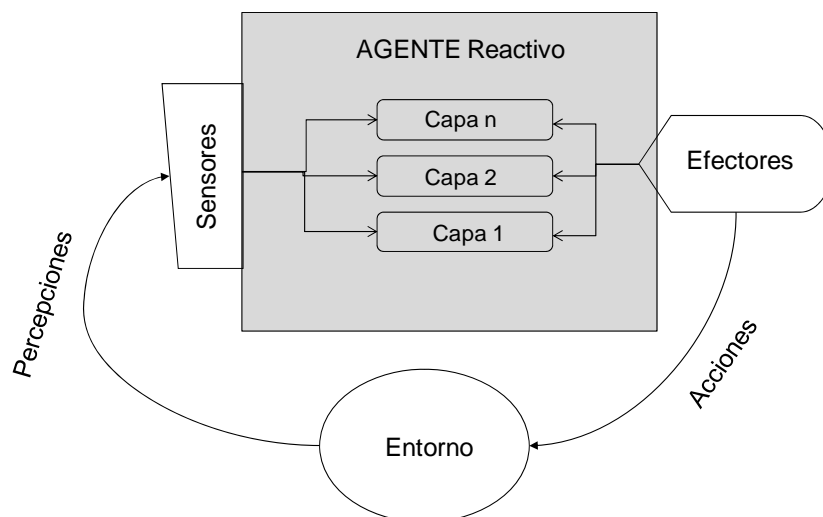


Figura 4-5. Arquitectura de un Agente Reactivo (elaboración propia).

La segunda aproximación se basa en la idea de la sociedad de la mente de (Minsky, M. 1986) a partir de la cual se considera el comportamiento de un Agente como el resultado de entidades en competencia para obtener el control sobre las acciones del Agente. Esta idea ha sido desarrollada en diferentes formas por (Maes, P. 1991), (Travers, M. 1988), (Drogoul, A. &

Ferber, J. 1992), (Ferber, J. 1996). En este caso un Agente se basa en un conjunto de tareas en conflicto donde sólo una puede ser activada simultáneamente.

En su implementación los Agentes reactivos suelen caracterizarse por estar constituidos por capas de forma que las percepciones son filtradas y en cada capa sólo se atienden las que son de interés para la misma. Se trata de simplificar el mecanismo de actuación sin dejar de considerar diversos tipos de eventos que pueden resultar de interés (ver Figura 4-5).

En cualquier caso, los Agentes puramente reactivos suelen sufrir dos tipos de limitaciones:

- La primera debida a que al ser puramente reactivos los Agentes toman las decisiones basados en información local. Los Agentes no tienen en cuenta la información no local o los efectos de sus decisiones en el comportamiento global. Lo que podría llevar al sistema a una situación inestable.
- La segunda consiste en que no existe una relación entendible entre el comportamiento local y el global. Esto hace muy complicado el proceso de diseño, que suele estar basado en el método de prueba y error.

Los mayores éxitos de los Agentes Reactivos se encuentran relacionados con su vinculación a acciones físicas como actuadores, por ejemplo en robots. En (Bonasso, R. P., Firby, J., Gat, E., Kortenkamp, D., Miller, D. P., & Slack M.G. 1997) se presenta un buen ejemplo del planteamiento y desarrollo de un arquitectura reactiva.

Una tercera alternativa a las dos arquitecturas anteriores es la Arquitectura Híbrida en la cual se potencian ciertas ventajas y/o se reducen ciertos inconvenientes para obtener una propuesta alternativa que se adapta mejor al problema en cuestión. Normalmente este tipo de arquitecturas se estructuran en un sistema de capas en las cuales las más bajas o cercanas a los puntos de contacto con el entorno son dotadas de características Reactivas, mientras que las capas superiores se caracterizan por un comportamiento más racional o BDI. Entre ambas capas pueden existir varios niveles intermedios responsables de aplicar diversos grados de Racionalidad según corresponda en los diferentes estímulos/respuestas que el Agente deba ejecutar. De esta forma se diversifica el comportamiento del Agente en función de las necesidades de cada momento aplicando Reactividad o algún grado de Racionalidad, mientras no se pierde a largo plazo el objetivo global.

En la Figura 4-6 se muestra una Arquitectura Híbrida de triple capa, donde cada capa tiene su base de conocimiento. Existen otros tipos de enfoque híbrido pero en todo caso se trata siempre de buscar el equilibrio entre reacción y razonamiento que mejor se ajuste al problema que deseamos resolver.

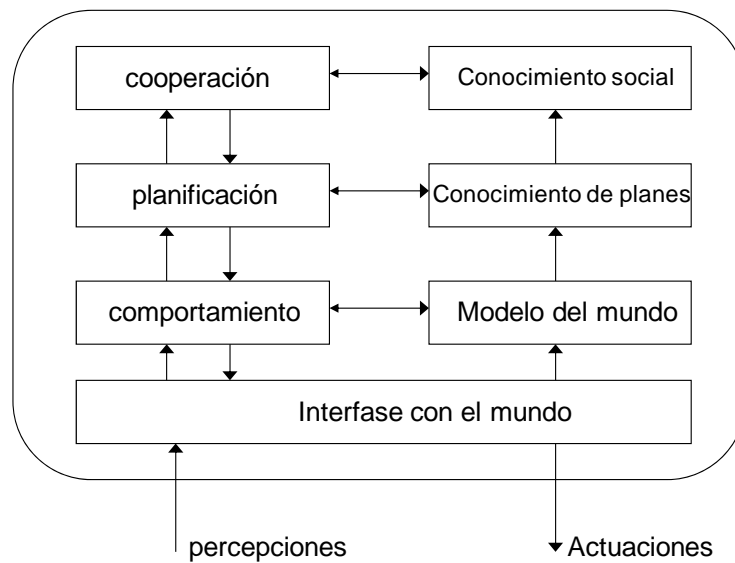


Figura 4-6. Representación de una arquitectura híbrida (Muller, J. P. 1997).

Con la revisión de las tres arquitecturas tipo que los investigadores han reconocido de forma reiterada en los últimos años se da por concluido el bloque dedicado a la revisión de los Agentes como entidades individuales. Como conclusión se debe resaltar que en el ámbito de la Programación de la Producción el autor de este trabajo considera, a tenor de lo visto, que la Arquitectura BDI puede ser la opción más adecuada. Si en algún momento se considerase la opción de incorporar elementos físicos de control, como sensores, etc., como consecuencia de una integración entre Programación y Control de la Producción quizás la Arquitectura Híbrida sea la más adecuada. Esta integración no se considera dentro del marco de esta tesis.

A continuación se realizará una revisión de los conceptos clave y las aportaciones que se han realizado en relación con los Sistemas Multiagente, es decir se consideraran los aspectos, no individuales de los Agentes, sino aquellos que hacen que se pueda hablar de un Sistema.

4.3 Los Sistemas Multiagente: La Organización

Los Agentes se agrupan en sociedades para formar sistemas y conjuntos complejos con capacidad para realizar tareas concretas sobre las cuales tienen habilidades. El estudio de estas sociedades implica el análisis de diferentes características significativas (Comunicación, Negociación, Coordinación, etc.), que en muchos casos conducen a un Sistema Organizado.

En este apartado se analizan los conceptos de **Interacción** y **Cooperación**, siguiendo el modelo de (Corchado, J. M. et al. 2002). Estos permiten establecer de forma progresiva las bases de lo que se entiende por **Organización**, que será analizada posteriormente con más detalle. Dentro del Nivel Organizativo se desarrollaran los conceptos de **Colaboración**, **Coordinación** y **Negociación** en un Sistema Multiagente Deliberativo.

El primer paso en el análisis de las relaciones entre Agentes debe ser el estudio de sus Interacciones. En (Ferber, J. 1999) se considera que una situación en la que se producen interacciones es *“aquella en la que existe una unión de comportamientos, resultado de un agrupamiento de Agentes, los cuales tienen que actuar atendiendo a: (1) sus objetivos, (2) los recursos disponibles para los Agentes y, (3) las habilidades individuales de cada Agente”*.

Tipo Interacción	Objetivos	Recursos	Habilidades
Independencia	Compatibles	Suficientes	Suficientes
Colaboración simple	Compatibles	Suficientes	Insuficientes
Obstrucción	Compatibles	Insuficientes	Suficientes
Colaboración coordinada	Compatibles	Insuficientes	Insuficientes
Competición individual absoluta	Incompatibles	Suficientes	Suficientes
Competición colectiva absoluta	Incompatibles	Suficientes	Insuficientes
Conflictos individuales sobre recursos	Incompatibles	Insuficientes	Suficientes
Conflictos colectivos sobre recursos	Incompatibles	Insuficientes	Insuficientes

Tabla 4-1. Tipos de interacciones entre agentes(Corchado, J. M. et al. 2002).

Aunque atendiendo exclusivamente a los objetivos propuestos existen tres grandes tipos de Interacción: **Cooperación**, **Indiferencia** y **Enfrentamiento**. En general, según las tres características definidas en (Ferber, J. 1999), es decir, Objetivos, Habilidades y Recursos, y en (Corchado, J. M. et al. 2002) se establece una clasificación de los tipos de interacción que pueden existir, ver Tabla 4-1.

Como se puede apreciar por el simple hecho de existir un contacto entre un conjunto de Agentes ya existe interacción, aún en los casos más extremos: Indiferencia o Enfrentamiento.

Además todo conjunto de Agentes, con el fin de satisfacer las necesidades derivadas de las interacciones, dispondrá de un Nivel Organizativo. Dicho Nivel Organizativo podrá tener un grado de influencia en el ciclo de vida del Agente que variará según las necesidades de las interacciones a las que deba hacer frente. En este sentido hay que considerar que la organización determinará que tipo de interacciones se pueden establecer, y viceversa.

Uno de los tres grandes grupos de interacciones son las que se denominan **Cooperativas**. Este grupo es de especial interés en el estudio de los Sistemas Multiagentes, ya que requieren del desarrollo de un conjunto de habilidades más complejas que el resto de grupos.

Se puede dar una situación de **Cooperación** si se cumplen ciertas condiciones que varían según diversos autores. Según (Conte, R., Miceli, M., & Castelfranchini, C. 1990) “**los Agentes Cooperan cuando coordinan sus acciones porque han identificado un objetivo común**”. Otros autores asocian esta característica a la observación externa, como (Durfee, E. H., Lesser, V. R., & Corkill, D. D. 1989) que considera que un conjunto de Agentes cooperan cuando un observador externo puede describir como cooperativas las acciones realizadas por los Agentes. En este mismo sentido (Ferber, J. 1999) estableció diversos índices para decidir si existe una situación de cooperación, que básicamente establecían (1) la existencia de un Agente especial/adicional a la sociedad, (2) la existencia de Agentes especializados en resolver o evitar conflictos.

Según (Ferber, J. 1999) la acción cooperativa se puede deber a la mejora del incremento de la capacidad de supervivencia, la mejora de las prestaciones del sistema o para reducir el número de conflictos. Este autor propone una serie de métodos que permite a los Agentes cooperar. Estas técnicas pueden ser utilizadas de forma conjunta, y con diferente peso, en el proceso de cooperación según el problema que afronte la organización. Dichos métodos son:

- **Agrupamiento y multiplicación:** Los Agentes se agrupan en una única entidad que afronta la solución del problema de forma conjunta. Cada Agente está especializado, pudiendo existir múltiples réplicas.
- **Comunicación:** Envío de información entre los diferentes Agentes con el objeto de mantener actualizado el conocimiento del entorno.
- **Especialización:** Existencia de un conjunto de Agentes con habilidades muy específicas. El tipo de Agentes diferentes es mucho más amplio que en otro tipo de sistemas, ya que para obtener un objetivo es necesaria la participación de gran número de Agentes.

- **Colaboración por la Compartición de Tareas y Recursos:** Este tipo de técnicas reparten las tareas que hay que realizar entre los diferentes Agentes así como los recursos disponibles para conseguir su fin.
- **Coordinación de Acciones:** Esta técnica permite gestionar las acciones que deben realizar los Agentes para conseguir el objetivo. La coordinación puede ser centralizada (Jerárquica, Igualitaria), o distribuida (Modelo de Agentes Conocidos, Contract Net Protocol).
- **Resolución de Conflictos mediante Arbitrio y Negociación:** El arbitrio hace referencia a un conjunto de reglas que deben cumplir todos los Agentes, con la posibilidad de la existencia de un Agente árbitro. La Negociación permite establecer un acuerdo entre Agentes en conflicto.

En general el campo de la Inteligencia Artificial Distribuida ha invertido un gran esfuerzo en el estudio de la Cooperación.

4.3.1 La Organización

El concepto de “Organización” en un Sistema Multiagente está íntimamente relacionado con el papel que cada Agente desempeña dentro del Sistema desde el punto de vista de su relación con el grupo, el comportamiento que se espera de él, y la distribución de poder o autoridad dentro del grupo.

La Inteligencia Artificial Distribuida nos proporciona varias definiciones de lo que se debe entender por Organización en un Sistema Multiagente. Así (Gasser, L. 1986) ve a una Organización como “un conjunto particular de cuestiones zanjadas y no zanjadas sobre las creencias y las acciones a través de las cuales los Agentes ven a otros Agentes”, desde este mismo punto de vista (Bond, A. H. 1988) definen una Organización como “un conjunto de Agentes con acuerdos mutuos y globales y creencias mutuas”. Según (Sycara, K. P. 1998) una Organización consiste en “un grupo de Agentes, un conjunto de actividades realizadas por los Agentes, un conjunto de conexiones entre los Agentes, y un conjunto de objetivos o criterios de evaluación”. Se suele asociar el concepto de Organización con la estructura del propio sistema, con los patrones de información que se manejan, con el control de las relaciones entre los Agentes y con la distribución de las habilidades para solucionar el problema. En (Corkill, D. D. & Lesser, V. R. 1983) se confirma que la estructura proporciona a los Agentes una visión global de como afrontar la solución de los problemas el sistema.

La definición de la Organización se complica aún más cuando se tiene en cuenta el aspecto dinámico de los Agentes. En este caso suele aparecer la función de registro de los diferentes Agentes y de sus habilidades. Esta función puede estar a cargo de uno o varios Agentes que se encargan de gestionar la información dinámicamente sobre el estado del sistema en cada momento. Esta tarea puede llegar a ser sumamente compleja, en los sistemas compuestos por un alto número de Agentes, y puede llegar a requerir un conjunto de habilidades orientadas a la propia gestión del sistema. Los autores de (Decker, K., Sycara, K. P., & Williamson, M. 1997) identificaron diferentes tipos de Agentes que pueden participar en esta tarea.

Algunos tipos de Organización descritos en la literatura, (Sycara, K. P. 1998) son:

Jerárquica: Las decisiones y el control se realiza de forma centralizado en cada nivel. Los Agentes del nivel superior ejercen el control sobre los recursos y las decisiones, estableciendo una comunicación vertical desde el nivel superior al nivel inferior subordinado.

Comunidad de Expertos: Esta organización es plana, y cada participante es especialista en un área. Cada Agente interactúa por medio de reglas y comportamiento ((Lewis, M. & Sycara, K. P. 1993), (Lander, S., Lesser, V. R., & Connell, M. E. 1991)). Los Agentes se coordinan mediante ajustes mutuos de sus soluciones para alcanzar una coherencia global.

Mercado: El control está distribuido entre los Agentes a los que compete una tarea o recurso a través de mecanismos como la subasta o el contrato. Los Agentes interactúan a través de una variable, el precio, que es usada para evaluar los servicios ((Müllen, T. & Wellman, P. 1996), (Davis, R. & Smiths, R. G. 1983), (Sandholm, T. 1993)). La coordinación de los Agentes se realiza mediante el ajuste del precio.

Comunidad Científica: Es un ejemplo de como una comunidad plural puede operar ((Kornfeld, W. A. & Hewitt, C. E. 1981)). Las soluciones a los problemas se construyen localmente, y después se comunican a otros miembros y las pueden probar y refinar.

Así mismo, (Shen, W. & Norrie, D. H. 1999), en su análisis sobre la aplicación de Agentes en la industria, propone una clasificación de la organización de los Agentes en tres categorías: Jerárquica, Federativa y Autónoma.

Hace ya algunos años que las Empresas Industriales están compuestas normalmente de unidades semi-autónomas, frecuentemente distribuidas, cada una con un grado de control sobre los recursos locales o con información diferente sobre las necesidades. No es extraño encontrar soluciones jerárquicas. Estas soluciones son normalmente criticadas por su carácter

centralizado. Algunos ejemplos de este tipo de Organización se encuentran en los proyectos Holonic Manufacturing System-HMS, que en su primera fase es descrito en (van Leeuwen, E. H. & Norrie, D. H. 1997) y en su segunda fase se describe en (Gruver, W. A., Kotak, D. B., van Leeuwen, E. H., & Norrie, D. 2003), o bien en Architecture for Distributed Dynamic Manufacturing Systems-ADDYMS descrito en (Butler, J. & Ohtsubo, H. 1992), o en la estructura de Control y Planificación de la Producción de (Fischer, K. 1994).

La propuesta federada puede contener, normalmente de forma alternativa, tres tipos de individuos: Facilitadores (Facilitators), Intermediarios (Brokers) y Mediadores (Mediators). Las estructuras federadas son capaces de coordinar la actividad Multiagente proporcionando una reducción de la sobrecarga, asegurando la estabilidad, y proporcionando escalabilidad, (Genesereth, M. R. & Ketchpel, S. P. 1994).

En la propuesta basada en Facilitadores los Agentes relacionados entre sí se agrupan y la comunicación entre ellos se realiza a través de los Facilitadores. Los Facilitadores son los responsables de proporcionar un enlace entre Agentes locales y remotos proporcionando servicios como el del enrutado de mensajes al destino adecuado, o el trasladando los mensajes de entrada a sus propios Agentes. Los proyectos como CIIMPLEX (Consortium for Integrated Intelligent

Manufacturing Planning and Execution) descrito por (Peng, Y., Finin, T., Labrou, Y., Chu, B., Long, J., Tolone, W. J. et al. 1998), PACT (Palo Alto Consortium Testbed) publicado en (Cutkosky, M. R., Englemore, R. S., Fikes, R. E., Genesereth, M. R., Gruber, F., Mark, W. S. et al. 1997), o SHADE (SHARED Dependency Engineering) resumido en (McGuire, J. G., Kuokka, D. R., Weber, J. C., Tenenbaum, J. M., & Gruber, F. 1993) utilizan esta propuesta.

Los Intermediarios son similares a los Facilitadores pero con algunas funciones adicionales como las de monitorización y notificación. La principal diferencia entre un Facilitador y un Intermediario es que el primero es responsable sólo de un determinado grupo de Agentes mientras que el Intermediario puede ser usado por cualquier Agente del sistema. Algunos proyectos que utilizan este tipo de Agente son AIMS (Agile Infrastructure for Manufacturing Systems) presentado por (Park, H., Tenenbaum, J. M., & Dove, R. 1993) y el ya nombrado CIIMPLEX de (Peng, Y. et al. 1998).

La propuesta basada en Mediadores es un tercer tipo de organización federada donde el mediador asume un papel adicional, la coordinación del sistema. Esta coordinación promueve la cooperación entre Agentes, y aprendizaje del comportamiento de los mismos. En (Gaines, B. R., Norrie, D. H., & Lapsley, A. Z. 1995) se proporciona una detallada descripción de esta propuesta. Aplicaciones de este tipo de organización se pueden encontrar en (Maturana, F. P.

& Norrie, D. H. 1996), (Shen, W., Xue, D., & Norrie, D. H. 1998) o en (Ouelhadj, D., Hanachi, C., & Bouzouia, B. 1996).

Por último la propuesta basada en Agentes Autónomos se basa en que los Agentes cumplen las siguientes características: (1) no son controlados, ni gestionados por otros Agentes o humanos, (2) pueden comunicarse o interactuar con otros Agentes del sistema y también con otros Agentes externos, (3) tienen conocimiento sobre otros Agentes y entornos, (4) tienen sus propios objetivos y sus motivaciones asociadas. Los sistemas de Agentes Autónomos proporcionan un buen conjunto para el desarrollo de Sistemas de Inteligencia Distribuida que cuenta con herramientas de ingeniería encapsuladas como Agentes, y son capaces de proporcionar servicios especiales. Este tipo de propuesta proporciona un buen entorno de desarrollo de sistemas multi-robot. Algunos ejemplos de este tipo de proyectos son DIDE (Distributed Intelligent Design Environment) descrito por (Shen, W. M. & Barthes, J. P. A. 1996) y AARIA (Autonomous Agent for Rock Island Arsenal) cuyos resultados fueron publicados en (Parunak, H. V., Baker, A. D., & Clark, S. J. 2001).

En la Figura 4-7 se muestra un esquema de los distintos problemas y mecanismos que se encuentran relacionados con los conceptos de Interacción, Cooperación y Organización. En dicha figura se indica como los Agentes implementan técnicas de Distribución de Tareas, técnicas de Coordinación de Acciones y mecanismos de Comunicación y Negociación.

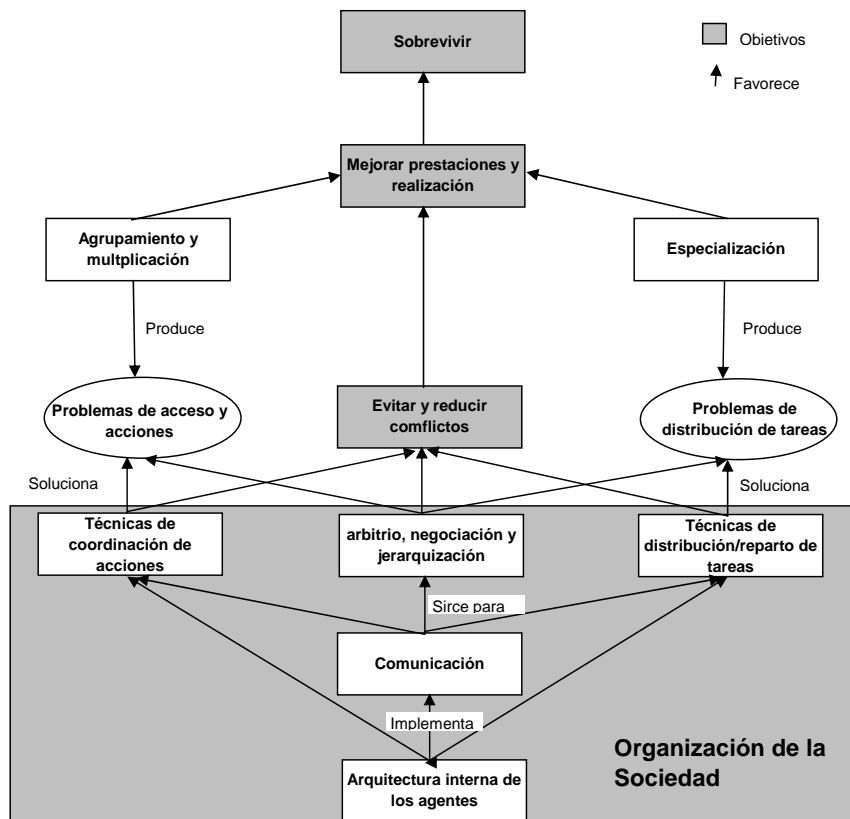


Figura 4-7. Organización de la Sociedad, (Corchado, J. M. et al. 2002).

A continuación se comentan los aspectos relacionados con la Colaboración, Coordinación y Negociación fundamentales en una Organización de Agentes.

4.3.1.1 La Colaboración

La colaboración junto a la coordinación son dos de las características fundamentales para que pueda existir cooperación en un Sistema Multiagente. Según (Ferber, J. 1999), la colaboración entre los Agentes se entiende como la función de distribuir el trabajo entre los diversos Agentes que constituyen el Sistema Multiagente. Precisamente, esta habilidad social es una de las que diferencia a un Sistema Multiagente de un Sistema Compuesto por Agentes.

La asignación de tareas es un problema estudiado en muchas otras áreas de conocimiento. Sin embargo, es importante tener en cuenta una característica de los Agentes que implica la existencia de un escenario diferente: **el compromiso**. Cuando un *Agente acuerda con otro realizar una tarea, este acuerdo pasa a ser un compromiso y un objetivo de dicho Agente.*

Se pueden clasificar los procedimientos de colaboración de la siguiente manera:

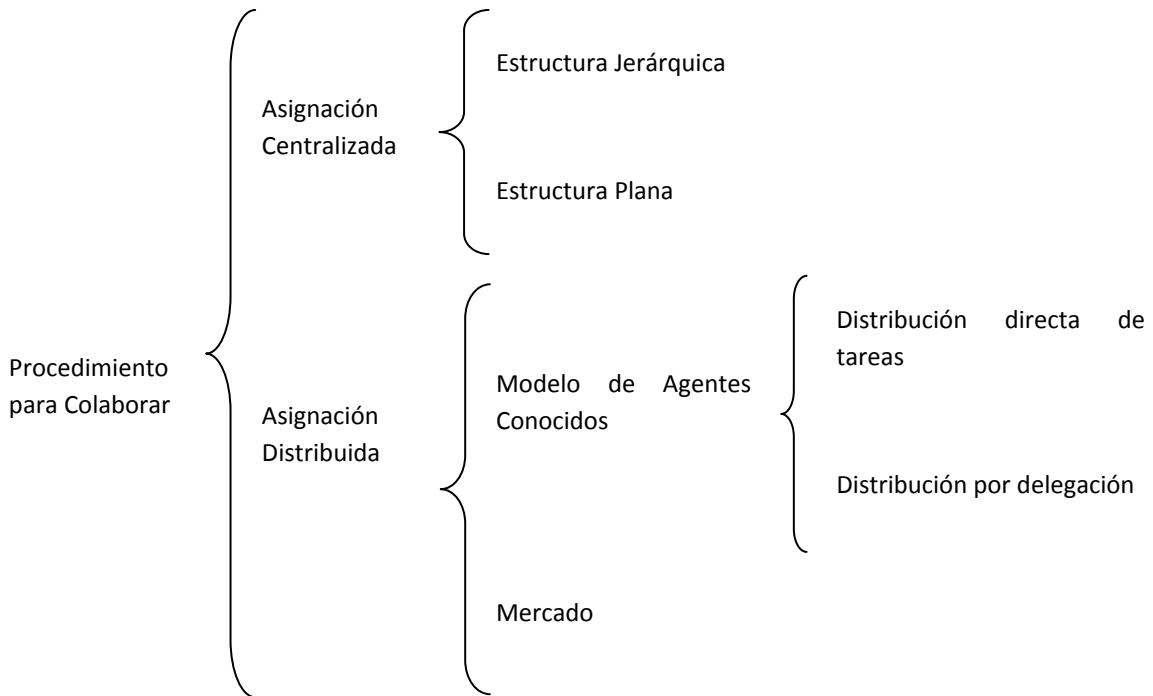


Tabla 4-2. Clasificación de los procedimientos para la Colaboración entre Agentes de una Organización (elaboración propia).

Todos los tipos de colaboración existente deben afrontar el problema de la asignación de responsabilidades y recursos a los Agentes para realizar el proceso de solución del problema. Este problema, conocido como asignación de tareas, encuentra grandes beneficios en la reducción de las interdependencias. Las interdependencias están estrechamente vinculadas al incremento de los conflictos, cuya resolución tiene un coste asociado, y al aumento en las comunicaciones entre los Agentes, lo que supone un retardo en la ejecución.

Desde muy temprano la Inteligencia Artificial Distribuida se ocupó del problema de la asignación de tareas. Las primeras propuestas confiaron al diseñador toda la responsabilidad en éste proceso. Sin embargo, con la aparición de los sistemas flexibles, incompatibles con esta propuesta, se obtuvieron los resultados en los cuales los propios Agentes asumían de una u otra forma esta responsabilidad. Una de las mejores propuestas fue el “**Contract Net Protocol**” (CNP) realizada por (Smith, R. G. 1979). En su versión más sencilla del protocolo se basa en la existencia de un Agente Subastador (auctioneer) y un conjunto de Agentes Postores (bidders). El protocolo consta de los siguientes pasos:

- El subastador necesita realizar una determinada tarea, y envía una petición para que le envíen propuestas (ofertas). En la petición se incluye una descripción de la tarea a realizar.
- Los Agentes postores que reciben la propuesta, si creen poder realizar la tarea solicitada envían una oferta al subastador.
- Cuando el subastador cierra el periodo de recepción de las ofertas, éstas son evaluadas. El subastador envía un mensaje al postor de la oferta ganadora comunicándole que se ha convertido en su contratista.
- El protocolo finaliza cuando el postor que recibe el mensaje de ganador contesta aceptando la propuesta o rechazándola. Si la rechaza el proceso se inicia de nuevo.

En este protocolo un Agente puede desempeñar cualquiera de los papeles de cliente (subastador) o proveedor (postor), incluso al mismo tiempo. En general el sistema puede soportar al mismo tiempo varios procesos de CNP por lo que pueden existir varios clientes al mismo tiempo, incluso un Agente puede ser cliente en un proceso y proveedor en otro. Al existir varios protocolos en marcha se puede producir una interferencia. El mismo autor, (Smith, R. G. 1980), propuso una modificación para incluir en el protocolo una medida de calidad y una fecha de caducidad del contrato.

Hay que destacar el alto grado de incertidumbre en el proceso. La falta de información afecta al aspecto temporal ya que ni los clientes, ni los proveedores saben cuando les puede llegar una oferta/propuesta mejor en el proceso de contratación. La incertidumbre también afecta al aspecto de las competencias, ya que ni los clientes saben nada sobre otras propuestas, ni los proveedores sobre las ofertas de otros Agentes. En referencia a la incertidumbre temporal en (Zlotkin, G. & Rosenschein, J. S. 1994) se utilizan conceptos de Mercado, funciones de Utilidad, Teoría de Juegos, etc. para abordar este asunto. En (Sandholm, T. & Lesser, V. R. 1995) se realizó una extensión en el CNP introduciendo penalizaciones por incumplimiento del acuerdo, y en (Sycara, K. P. 1997) se mezcló la teoría financiera con CNP para mejorar el proceso de contratación en entornos inciertos. En (Ferber, J. 1999) se propusieron varias modificaciones del CNP la primera consiste en guiar el protocolo desde los Agentes proveedores, ya que al ser estos quienes realmente tienen las capacidades, pueden reducir los pasos haciendo la contratación más directa, y la segunda modificación utiliza CNP en combinación con el modelo de Agentes conocidos, de tal forma que el primero se usa en las tareas complejas y el segundo a las sencillas. En (Parunak, H. V. 1987) se implementó el CNP aplicándolo a un problema de control de planta.

Otro aspecto interesante a tener en cuenta en el CNP es la política de subcontrataciones. Cuando un proveedor consigue un contrato nada impide que este realice una subcontratación parcial o total del contrato recién obtenido. Esta cadena de subcontrataciones puede darse hasta el infinito. Por otro lado, no poder subcontratar nada puede impedir que se cumplan el contrato en un plazo razonable. En definitiva, este aspecto debe ser definido antes de proceder con el CNP.

4.3.1.2 La Coordinación

La coordinación junto a la colaboración son dos de las características fundamentales para que pueda existir cooperación en un sistema multiagente.

La coordinación es una actividad para la gestión de las interdependencias entre las acciones. La coordinación entre un grupo de Agentes se basa en considerar todas las tareas a realizar, y su coordinación para no ejecutar acciones no deseables, por ejemplo:

- Comunicación de resultados redundantes.
- Distribución inapropiada de las tareas entre los Agentes.

Esta coordinación está relacionada con la planificación de acciones para la resolución de tareas, ya que estos planes permiten:

- Conocer a alto nivel y predecir el comportamiento de otros Agentes del sistema.
- Intercambiar resultados intermedios que lleven al progreso en la solución de la tarea global.
- Evitar acciones redundantes, si ellas no son deseables.

En función del ámbito en el que se realice la distribución de tareas se puede hablar de:

- **Coordinación Global:** Cuando el SMA determina y planifica globalmente las acciones de los diferentes Agentes.

- **Coordinación Individual:** Cuando el SMA le da completa autonomía a los Agentes, es decir, cada Agente decide qué hacer y resuelve localmente los conflictos que detecte con otros Agentes.

En función del objetivo que se persiga en el diseño del proceso de coordinación se puede hablar de:

- **Coordinación orientada a los problemas:** Los Agentes deben coordinar los planes de realización de acciones para prevenir interbloqueos, repetición de acciones y creación de inconsistencias.
- **Coordinación orientada a la cooperación:** Los Agentes no se coordinan en el ámbito de planes, sino en el ámbito de acciones. Esto significa que los Agentes se coordinan en el momento de ejecutar acciones.

La coordinación es por lo tanto una forma refinada de colaboración que permite evitar ciertos aspectos no deseados en la ejecución de los planes para obtener los objetivos comunes de forma eficaz. En (Durfee, E. H. et al. 1989) y (Ferber, J. 1999) se puede ampliar el conocimiento sobre diversos aspectos de la colaboración ya que los autores presentan una caracterización detallada de esta forma de actuar.

4.3.2 La Negociación

Para que los mecanismos de cooperación y coordinación sean exitosos en un sistema de Agentes que actúan interdependientemente, debe existir un mecanismo adicional, por medio del cual, los integrantes de un sistema se puedan poner de acuerdo cuando cada Agente defiende sus propios intereses. Éste debe llevar a los miembros de la sociedad a una situación que exista un beneficio común teniendo en cuenta el punto de vista de cada uno. Este mecanismo se denomina negociación.

La negociación se entiende como un medio para que los Agentes puedan alcanzar sus objetivos en un entorno de competencia. Es una forma de interactuar dentro de un grupo con el fin de alcanzar un acuerdo. Este proceso sólo tiene sentido cuando los Agentes dentro de su autonomía pueden plantearse un conjunto de opciones posibles, es decir, son flexibles en su forma de actuar. En este caso, unos Agentes pueden intentar “convencer” a otros Agentes de que actúen de una determinada manera. El proceso de negociación se puede plantear de diferentes formas sin que ello quiera decir que una sea mejor que otra, todo depende de la

situación para la que se plantee. Según se recoge en (López-Carmona, M. A. 2006) un proceso de negociación se puede basar en:

- Teoría de Juegos (Binmore, K. & Vulkan, N. 1999; Dash, R. K., Parkes, D. C., & Jennings, N. R. 2003).
- Procedimientos Heurísticos.
- Argumentación (Jennings, N. R., Faratin, P., Lomuscio, A. R., Parsons, S., & Sierra, C. 2001; Jennings, N. R., Parsons, S., Noriega, P., & Sierra, C. 1998).
- Relaciones Bilaterales basadas en Restricciones Difusas.

Entre las diferentes alternativas para enfocar el proceso negociador, quizás la más extendida y más ajustada a la naturaleza de los Sistemas Multiagente sea la que se basa en Argumentos. Se puede entender que un Argumento es una metainformación que pretende hacer más atractiva una propuesta en un proceso de negociación. La naturaleza y tipos de argumentos pueden variar enormemente sin embargo los más habituales son: amenazas, recompensas, o apelaciones. Los argumentos permiten a los Agentes refinar sus procesos de tomas de decisión, y en este sentido potencian su autonomía. Los protocolos y los lenguajes de comunicación son especialmente importantes en las negociaciones basadas en Argumentos.

En general, son tres los grandes aspectos que según los expertos deben ser analizados en un proceso de negociación:

- **Los Protocolos de Negociación**, entendidos como un conjunto de reglas que dirigen el proceso.
- **Los Objetivos de la Negociación**, entendidos como el conjunto de aspectos sobre los cuales se debe alcanzar un acuerdo.
- **Los Modelos de Razonamiento**, sobre los cuales los Agentes van a tomar las diferentes decisiones que forman parte del proceso.

Los Protocolos de Negociación deben cubrir aspectos como los estados en los que es posible que un Agente se encuentre a lo largo del proceso, los participantes que pueden formar parte de la negociación y su rol, los eventos que pueden considerarse y las acciones que son válidas dentro del proceso. Los protocolos de negociación están sumamente vinculados a la comunicación. El éxito de un protocolo se basa en que todos los participantes sean conscientes

del proceso y de los contenidos que se dan a conocer, en este sentido cabe recordar la importancia de los lenguajes y las ontologías, que son la base fundamental de la comunicación.

Los Objetivos de la Negociación deben considerar si estos son únicos o múltiples, y cuales son los márgenes existentes para alcanzar un acuerdo sobre los mismos. En este punto han sido de utilidad los enfoques tipo mercado con subastas que permiten fijar un precio sobre los servicios/productos que se demanda/ofrecen. En todo caso, un aspecto fundamental es como evaluar el cumplimiento de los objetivos. Esta evaluación podría ser individual o global. El protocolo de negociación debe estar diseñado de acuerdo a las especificaciones de los Objetivos de la Negociación. Un ejemplo es el protocolo "Contrat Net" de (Smith, R. G. 1979) aplicable a un objetivo único.

Los Modelos de Razonamiento están sumamente condicionados por el protocolo de negociación, el tipo de objetivos planteado y las acciones que se pueden realizar. En este apartado juegan un papel importante los argumentos en los procesos basados en los mismos.

Con este apartado se cierra la revisión del comportamiento de los Agentes dentro de una comunidad. Queda por analizar otro aspecto que condicionan el comportamiento como es la comunicación que se analizará en el apartado siguiente.

4.4 Los Sistemas Multiagente: El Intercambio de Conocimiento

La comunicación es el mecanismo que posibilita el intercambio de conocimiento de forma más directa entre los miembros de una comunidad. En este apartado se presentan aspectos relacionados con el acto comunicativo, los lenguajes y las ontologías.

4.4.1 La Comunicación

La comunicación es el soporte de la coordinación, la cooperación y por lo tanto de la negociación entre Agentes de un mismo sistema. En (Russell, S. et al. 2002) se establece que es la comunicación entre Agentes la actividad que permite sincronizar acciones, enviar y recibir conocimiento y resolver conflictos.

Se pueden identificar varios formatos de comunicación, (Werner.E. 1989), (Corchado, J. M. et al. 2002):

- **No comunicación.** Los Agentes interactúan sin comunicarse, infiriendo las intenciones de los otros Agentes. Este modelo consigue buenos resultados sino existe interrelación entre los Agente, y por lo tanto, no hay conflictos, (Schelling, T. C. 1969).
- **Comunicación primitiva.** La comunicación se restringe a un número de señales fijas con significados preestablecidos, (Hoare, C. A. R. 1978).
- **Arquitectura de pizarra.** Los Agentes escriben sobre una “pizarra” común a todos, los mensajes que necesitan comunicarse. Procede la de la inteligencia artificial, donde se utiliza para compartir memoria y conocimiento,(Hayes-Roth, B. 1985).
- **Intercambio de planes e información.** Dos Agentes se intercambian los planes de forma que al saber como actuará el otro puede establecer estrategias. En sistemas con un número elevado de Agentes puede no ser eficaz por su alto coste computacional.
- **Intercambio de mensajes.** Los Agentes van actuando en respuesta a los mensajes procesados, (Agha, G. & Hewitt, C. 1987).
- **Comunicación de alto nivel.** Existe un dialogo o conversación que permite la generación e interpretación de declaraciones, con el objeto de comunicar la información que el emisor conoce formada por creencias, compromisos e intenciones, para que el receptor alcance el mismo estado mental que el emisor, (Grosz, B. & Sidner, C. 1990).

Como cualquier otro software los sistemas basados en el conocimiento se enfrentan a un entorno en el cual existen diferentes plataformas hardware, diferentes medios y lenguajes de comunicación, diferentes protocolos, etc. Sin embargo, estos sistemas necesitan respetar unos fuertes requerimientos relativos a su interoperabilidad. Los Agentes de un Sistema Multiagente realizan continuos actos de intercambio de conocimiento tanto para su propia autorregulación como para alcanzar sus propios objetivos. Para esta comunicación de conocimiento se necesitan convenciones a tres niveles, (Gruber, T. R. 1993): Lenguaje de Representación del Conocimiento, Protocolos de Comunicación y Especificación del Contenido de Conocimiento Compartido.

El nivel cubierto por los lenguajes de representación del conocimiento se encarga de expresar el contenido de un mensaje en términos entendibles por los individuos que forman parte de la comunicación. Diversos investigadores han realizado propuestas de lenguajes estándares: (Fulton, J. A. 1992), (Genesereth, M. R. & Fikes, R. 1992) o (Morik, K., Causse, K., & Boswell, R. 1991). Se destacan dos lenguajes de comunicación entre Agentes:

- **KQML** (Knowledge Query and Manipulation Language), (Finin, T. 1994) es un lenguaje que puede ser usado para interactuar con un o más sistemas inteligentes que comparten conocimiento para solucionar un problema de forma cooperativa. - <http://www.cs.umbc.edu/kqml/> - .
- **KIF** (Knowledge Interchange Format) es un lenguaje para el intercambio de conocimiento entre sistemas distintos. - <http://www.cs.umbc.edu/kse/kif/> .

Según (Gruber, T. R. 1993) la comunicación va más allá de los lenguajes. Así en este nivel se incluiría a los lenguajes de comunicación de alto nivel y también a los protocolos. Los lenguajes de alto nivel harían posible que la sintaxis y la semántica de la comunicación fuesen correctas, mientras que los protocolos de interacción permiten ordenar la comunicación dando lugar a una conversación. En este ámbito se pueden destacar los esfuerzos de (Finin, T. 1994), y en general la organización FIPA.

En el nivel asociado a la especificación del contenido se establecen acuerdos sobre las asunciones que se comparten y los modelos de cada dominio. Este nivel afecta al vocabulario utilizado en la comunicación y se define mediante las llamadas “ontologías”. Algunas aportaciones importantes se pueden encontrar en (Genesereth, M. R. & Nilsson, N. J. 1987), (Newell, A. 1982) o (Gruber, T. R. 1991). Este aspecto se desarrolla más detenidamente en el apartado siguiente.

4.4.2 Ontologías

Una ontología es una especificación explícita de una conceptualización, según (Gruber, T. R. 1993). En su aplicación a los Sistemas Multiagente una ontología es una descripción, como la especificación formal de un programa, de conceptos y relaciones que pueden existir para un Agente o una comunidad de Agentes. Esta definición es consistente con el uso más general de las ontologías como un conjunto de definiciones de conceptos.

Ontologías como un mecanismo de especificación

Un cuerpo de conocimiento formalmente representado está basado en una conceptualización: los objetos, los conceptos, y otras entidades que son asumidos como ciertos en algún área de interés y de relaciones que se mantienen entre ellos, (Genesereth, M. R. et al. 1987). Una conceptualización es un resumen, una vista simplificada del mundo, que se desea representar para un determinado propósito.

El término “ontología” es un préstamo de la filosofía, donde representa una contabilidad sistemática de “Existencias”. Para la Inteligencia Artificial lo que existe es lo que se puede representar. Cuando el conocimiento de un determinado dominio es representado en un formalismo declarativo, el conjunto de objetos que pueden ser representados se llaman universo del discurso. Este conjunto de objetos y sus relaciones, se reflejan en el vocabulario representacional con el cual un programa basado en el conocimiento representa el conocimiento. En el contexto de la Inteligencia Artificial se puede describir la ontología de un programa mediante la definición de un conjunto de términos representacionales. En tal ontología, las definiciones asocian los nombres de las entidades en el universo del discurso con el texto descriptivo legible por el hombre; lo que los nombres significan, los axiomas formales que restringen la interpretación, y la formación adecuada de estos términos.

Se usan ontologías comunes para describir acuerdos ontológicos para un conjunto de Agentes de forma que se puedan comunicar sobre un dominio de un discurso sin necesidad de operar sobre la globalidad de la teoría compartida. Se dice que un Agente está de acuerdo a una ontología si sus acciones observables son consistentes con las definiciones en la ontología. La idea de los acuerdos ontológicos se basa en la perspectiva del nivel de conocimiento, (Newell, A. 1982). El nivel de conocimiento es el nivel de descripción del conocimiento de un Agente que es independiente del nivel simbólico representación usada internamente por el Agente. El conocimiento es atribuido a los Agentes mediante la observación de sus acciones; un Agente conoce algo si actúa como si tuviese la información y estuviese actuando racionalmente para conseguir sus objetivos. Las acciones de los Agentes se pueden hacer patentes a través de una su conversación, donde los clientes interactúan con un Agente haciendo aserciones lógicas (comunicados) e interrogaciones, (Levesque, H. J. 1984).

Las ontologías definen el vocabulario con el cual preguntan y realizan aserciones entre los individuos que comparten la ontología. Los acuerdos ontológicos son pactados para compartir vocabulario de forma coherente y consistente. Los Agentes que comparten vocabulario no necesitan compartir un conocimiento base. Los Agentes pueden conocer cosas que otros no conocen.

Ontologías y Conocimiento

Como se ha visto una ontología permite especificar representaciones formales del conocimiento. En el ámbito de los Sistemas Multiagente, y en general el dominio de las organizaciones y de los individuos, los datos que lo definen, y que fluyen a través del sistema se pueden considerar en términos de conocimiento. Este conocimiento afecta tanto a las acciones encaminadas a obtener los fines marcados, como al conjunto de acciones reguladoras basadas en normas de gestión interna del sistema con objeto de garantizar el funcionamiento operativo del sistema. La correcta representación del conocimiento permitirá disponer de la

base necesaria para razonar de forma eficiente en las diversas tareas del sistema. En (Steve, G., Gangemi, A., & Pisanelli, D. M. 1997) se distinguen tres tipos fundamentales de ontologías:

- **Ontologías de un dominio**, en las que se representa el conocimiento especializado pertinente de un dominio o subdominio, como la medicina, las aplicaciones militares, la cardiología o, en nuestro caso particular, la oncología.
- **Ontologías genéricas**, en las que se representan conceptos generales y fundacionales del conocimiento como las estructuras parte/todo, la cuantificación, los procesos o los tipos de objetos.
- **Ontologías representacionales**, en las que se especifican las conceptualizaciones que subyacen a los formalismos de representación del conocimiento, por lo que también se denominan meta-ontologías (meta-level o top-level ontologies).

A estos tres tipos en (Guarino, N. 1998) se añaden las ontologías que han sido creadas para una actividad o tarea específica denominadas “task ontologies”, y las ontologías creadas para una aplicación específica.

El objetivo de este apartado ha sido poner de manifiesto la posibilidad de contemplar el uso de ontologías como una forma de puesta en común de una parte del conocimiento que poseen los Agentes. Sin embargo, existen otros conceptos relacionados con el uso de ontologías que resulta interesantes tener en cuenta pero que no entran dentro del objetivo de este trabajo como son: Las técnicas de Aprendizaje que permiten ir reconociendo los elementos conceptuales de forma dinámica; El mapeado entre ontologías que permite relaciona términos de entornos diferentes; Las técnicas de evaluación de ontologías que permiten validar el diseño de una ontología para un propósito dado.

El uso de ontologías es fundamental en entornos altamente distribuidos donde los participantes (Agentes) proceden de entornos diversos, y muchas veces actúan en el de forma esporádica. Su diseño e implementación se ha extendido mucho en los últimos años porque además de su utilidad directa permite realizar un ejercicio de modelado conceptual del mundo que se desea representar.

4.5 Análisis, Diseño e Implementación de los Sistemas Multiagente

En su afán por desentrañar la complejidad creciente de la construcción de sistemas software debida, principalmente, al continuo avance tecnológico, el progreso de la Ingeniería del

Software ha caminado parejo al desarrollo de mecanismos de abstracción cada vez más potentes, y de más alto nivel. La abstracción modular, tipos abstractos de datos y, más recientemente, objetos y componentes son claros ejemplos de este tipo de abstracciones.

Los Agentes software, son un tipo de sistemas software por el que han apostado muchos investigadores y desarrolladores. Sin embargo, no se debe perder de vista, que los Agentes no son más que un mecanismo más de abstracción para un tipo de software muy concreto. Y como tal software su desarrollo puede ser auspiciado por técnicas de Ingeniería del Software (IS), como cualquier otro sistema informático.

En este apartado se indagará en las técnicas necesarias para obtener un Sistema Multiagente ajustado a las necesidades de cada caso. En primer lugar se presentan las organizaciones de mayor prestigio e influencia que marcan los estándares del software basado en Agentes. En segundo lugar se analizarán las diversas metodologías que soportan la generación de Sistemas Multiagentes y que en mayor o menor medida cubren las fases de análisis, desarrollo e implementación. Dentro de este punto se presentaran las herramientas que soportan las diferentes metodologías en caso de existir. En tercer lugar y último lugar, se presentarán y caracterizarán las plataformas software para el desarrollo y ejecución de Agentes y Sistemas Multiagente.

4.5.1 Organizaciones para la Estandarización de los Sistemas basados en Agentes

Las organizaciones que se presentan en este punto están muy vinculadas a las plataformas de Agentes en las cuales se implementa definitivamente las normas propuestas por éstas. Las plataformas proporcionan un entorno para la ejecución de Agentes software, para la gestión de dicha ejecución, para la gestión de los recursos, y a su vez garantizan la protección y la seguridad de los Agentes, y de la propia plataforma.

Estos sistemas se han desarrollado dando lugar a diferentes arquitecturas, e implementaciones. Para evitar la falta de interoperabilidad, fundamental en un sistema distribuido, se han creado diversas asociaciones y organismos. Algunos de ellos son:

- **FIPA** (Fundation for Intelligent Physical Agents)

FIPA – www.fipa.org - es una organización sin ánimo de lucro enfocada a definir estándares para la interoperación de softwares heterogéneos de Agentes. Fundada en Génova en agosto

de 1996. Mediante varias especificaciones ha ido estableciendo una arquitectura de referencia para el desarrollo de aplicaciones basadas en Agentes y Sistemas Multiagente.

FIPA presenta especificaciones agrupadas en 5 categorías: Aplicaciones, Arquitecturas Abstractas, Comunicación entre Agentes, Gestión de Agentes, Transporte de Mensajes de Agentes.

Diversas empresas han desarrollado sus plataformas de acuerdo con las normas de FIPA. Al menos cuatro de dichas implementaciones son de acceso libre: Nortel Network's FIPA-OS, Fujitsu Laboratories of America's April Agent Platform, CSELT(Italy)'s Java Agent Development Framework (JADE), y Comtec (Japan)'s Comtec Agent Platform .

- **OMG** (Object Management Group)

OMG – www.omg.org - fue fundada en 1989 por once compañías entre las que se pueden destacar 3Com, HP, Canon, Sun, Unisys y American Airlines. En la actualidad agrupa a más de 800 miembros. Su estándar más conocido es CORBA. El grupo OMG fue establecido para crear un mercado de software basado en componentes mediante la introducción de objetos software estandarizados. Ello incluye el establecimiento de guías y especificaciones para proporcionar un marco común para el desarrollo de aplicaciones.

En 1995 OMG empezó a trabajar en un estándar llamado Mobile Agent Facility (MAF) para mejorar la interoperabilidad entre plataformas de Agentes.

- **MASIF** (Mobile Agent System Interoperability Facility)

MASIF,(Milojicic, C., Breugst, M., Busse, I., Campbell, J., Covaci, S., Friedman, B. et al. 1998), busca la estandarización de las plataformas que combinan el uso de Agentes móviles y del paradigma cliente/servidor. Este proceso de estandarización ha producido lo que se conoce como fundamentos de MASIF. MASIF proporciona una arquitectura de referencia para Agentes móviles.

MASIF desarrollo para el grupo OMG una propuesta para la directiva RFP3 relacionada con la estandarización de Agentes móviles. En concreto, surge como consecuencia de la unión al grupo de trabajo sobre MAF de OMG de IBM, General Magic, The Open Group, GMD FOKUS, etc. La especificación propuesta fue aceptada en 1998.

MASIF cubre las siguientes funcionalidades: Gestión de Agentes, Seguimiento de Agentes, Transporte de Agentes (migración), Denominación de Agentes, y Sintaxis de Localización y Tipado de Agentes. MASIF usa CORBA para gestionar la seguridad de los agentes. MASIF no trata aspectos de comunicación entre Agentes.

La primera versión que se implementó con esta especificación fue IKV's Grasshopper Agent System en 1998. A su vez IBM y FOKUS se comprometieron al desarrollo de sus propias plataformas de acuerdo a MASIF. SOMA (Security Open Mobile Agents) de la Universidad de Bologna en Italia es otra de las plataformas desarrollada en sintonía con MASIF.

- **KSE** (Knowledge Sharing Effort)

Knowledge Sharing Effort - <http://www-ksl.stanford.edu/knowledge-sharing/papers/kse-overview.html> - es un consorcio, esponsorizado por ARPA (Advanced Research Projects Agency), para el desarrollo de acuerdos que faciliten compartir y reutilizar el conocimiento. Según (Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T. et al. 1991) el objetivo del consorcio es definir, desarrollar y probar la infraestructura, y el soporte tecnológico que facilita a los participantes construir un más grande y funcionalmente más amplio sistema que realice trabajo por sí mismo. El consorcio proporciona especificaciones e implementaciones de dominio público, informes, comunicaciones y artículos técnicos, y librerías software reusables de dominio público (demostraciones).

KSE está organizado en grupos de trabajo orientados a cuatro áreas. Estos grupos son: Interlingua, Knowledge Representation System Specification, External Interfaces, y Shared, Reusable Knowledge Bases. El grupo de External Interfaces se encarga de los aspectos relacionados con las interacciones en ejecución entre los sistemas basados en el conocimiento y otros sistemas, con especial interés en los lenguajes de comunicación. Algunos de los lenguajes más destacados son, los ya comentados, KQML y KIF, muy extendidos en la comunicación entre Agentes.

Además KSE ha proporcionado tecnología para construir Ontologías compartidas - <http://www.cs.umbc.edu/kse/ontology/> -, y tecnología para construir protocolos compartidos.

Como se puede observar las organizaciones comentadas anteriormente han ido proporcionando una serie de marcos de trabajo en diversos ámbitos que ha potenciado el desarrollo de los Sistemas basados en Agentes.

4.5.2 Metodologías y Herramientas para el Diseño de Sistemas Multiagente

La aplicación de la ingeniería del software al paradigma de Agentes, conocida como Ingeniería del Software Orientada a Agentes (AOSE – Agent Oriented Software Engineering) ha generado en los últimos años numerosos trabajos. Algunos de esos trabajos han sido analizados en (Iglesias, C., Garijo, M., & González, J. 1999) y en (Wooldrige, M. & Ciancarini, P. 2001). Hasta el momento, los trabajos existentes se han centrado en intentar buscar métodos de desarrollo para modelar sistemas reales complejos y con características claramente distribuidas.

El desarrollo de Sistemas Multiagente constituye hoy en día uno de los problemas más interesantes del área, ya que es un hecho que cada vez son más necesarias metodologías, métodos, técnicas y herramientas que faciliten el desarrollo de aplicaciones basadas en el paradigma de aplicaciones Multiagente. La mayoría de estos trabajos se basan en una visión de un sistema como una organización computacional, consistente en diferentes roles interactuando. Cuando hablamos de sistemas complejos, el poder identificar los diferentes subsistemas que forman parte del sistema global, y las posibles interacciones y dependencias entre ellos, es crucial a la hora de poder abordar su construcción.

Entre las aproximaciones existentes que se han considerado más interesantes están:

- La metodología para agentes BDI, esta aproximación, (Kinny, D. & Georgeff, M. 1996), trata de explorar como las técnicas de modelado OO (Orientado a Objetos) se puede extender para aplicarse a sistemas de Agentes basados en la arquitectura BDI.
- **DESIRE**, (Brazier, F., Dunin Keplicz, B., Jennings, N., & Treur, J. 1997), constituye un entorno lo suficientemente expresivo para permitir a los diseñadores de Sistemas Multiagente centrarse en el diseño conceptual y la especificación de su sistema.
- **GAIA**, (Wooldridge, M., Jennings, N. R., & Kinny, D. 1998), intenta permitir a un analista ir de forma sistemática desde un conjunto de requerimientos a un diseño que sea lo suficientemente detallado para poder ser implementado directamente.
- **MAS-CommonKADS**, (Iglesias-Fernández, C. A. 1998), la metodología está basada en CommonKADS, (Schreiber, A., Akkermans, J., & Anjewierden, A. 2000), aportando una serie de modelos para desarrollar las fases de análisis y de diseño de Sistemas Multiagente. La principal característica desarrollada es la incorporación de técnicas orientadas a objetos a CommonKADS, la cual es tomada como eje fundamental a lo largo de todo el proceso.

- **MASSIVE**, (Lind, J. 1999), este método de desarrollo de Sistemas Multiagentes (Multi-Agent SystemS Iterative View Engeneering) desarrollado en DFKI. Está constituido por un conjunto de vistas diferentes del sistema a construir donde el desarrollo que se sigue consiste en una visión iterativa del mismo.
- **MaSE** (Multiagent System Engineering), (Wood, M. F. 2000), a partir de las especificaciones de un sistema y mediante guías a seguir a través de un ciclo de vida pretende obtener un sistema basado en Agentes implementado.
- **AUML**, (Odell, J., Parunak, H., & Bauer, B. 2000), este trabajo se centra en intentar emplear herramientas de desarrollo ya existentes y que están teniendo éxito para aplicaciones industriales reales, como puede ser el caso UML, tratando de orientarlas hacia el campo de los Agentes.
- **TROPOS**, (Castro, J., Kolp, M., & Mylopoulos, J. 2002), presenta una metodología de desarrollo de software basado en agentes mediante extensiones de UML y empleando un entorno de modelado propuesto por (Yu, E. 1996).
- **MESSAGE**, (EURESCOM 2000) y (EURESCOM 2001), metodología orientada a Agentes que trata de proveer un lenguaje, un método y unas guías de aplicación, centrándose fundamentalmente en las fases de análisis y diseño. Hasta el momento sólo se han especificado los diferentes modelos que constituyen la fase de análisis.
- **INGENIAS**, (Gomez, J. 2002), metodología basada en MESSAGE, la cual presenta fundamentalmente como aspecto más interesante la existencia de herramientas que permiten un desarrollo a alto nivel de entidades transformables automáticamente en componentes software.

Aunque no son del interés de este trabajo es importante destacar que en los últimos años se ha producido un auge importante de las metodologías orientadas a los sistemas de tiempo real.

4.5.3 Plataformas para la Implementación de Sistemas Multiagentes

Las herramientas usadas para el desarrollo de Agentes se caracterizan por proporcionar al usuario un conjunto de librerías para la especificación de nuevos Agentes identificados por atributos y comportamientos personalizados. Estas herramientas, conocidas como **Plataformas de Agentes, ponen a disposición del usuario un entorno para la ejecución de aplicaciones basadas en Agentes**. Muchas de estas Plataformas facilitan al usuario

mecanismos de comunicación entre Agentes, mecanismos de gestión de Agentes, seguridad en el sistema, etc.

Se han utilizado diversos lenguajes de programación para el desarrollo de las Plataformas de Agentes (C++, JAVA, etc.). No obstante, el desarrollo de *Plataformas el lenguaje JAVA está predominando*. La portabilidad del software y la independencia del hardware de los sistemas desarrollados en JAVA son dos razones fundamentales para apoyar este predominio.

En (Vrba, P. 2003) se realiza un interesante estudio comparativo de las principales Plataformas de Agentes desarrolladas en JAVA. El estudio analiza la compatibilidad de cada plataforma con la normativa de la FIPA, así como cuestiones de seguridad, disponibilidad de código abierto y uso de la versión J2ME de JAVA para la ejecución de la plataforma.

La compatibilidad de las Plataformas con la normativa FIPA se realiza mediante el estudio de tres características claves como son: las soluciones para la Comunicación entre Agentes dentro de la Plataforma (Protocolos, Lenguajes, Servicios, etc.), Gestión de la Organización de Agentes (AMS y DF) e Interoperabilidad entre Plataformas a través del paso de mensajes (MTP).

El estudio del uso de código abierto está relacionado con la posibilidad de disponer del código de forma pública con autorización para poder modificarlo introduciendo pequeñas adaptaciones. En aspectos relacionados con el control de planta es interesante la modificación del código para el acceso de los PLC's de a través de la propia plataforma de Agentes.

El uso de la versión J2ME (JAVA 2 Micro Edition) está vinculado a la reducción del consumo de memoria, y por lo tanto a la posibilidad de incorporar al Sistema Multiagente a Agentes con recursos limitados como PDA's, etc.

La Seguridad se estudia desde el punto de vista de la Comunicación, detectando si la Plataforma dispone de servicios de autenticación, autorización, permisos, etc.

Revisión de Plataformas/Herramientas de Desarrollo de Agentes basadas en JAVA							
Plataforma	Desarrollador	Compatibilidad FIPA			Código abierto	J2ME	Seguridad
		Comunicación entre agentes	Gestión de agentes	Mensajes Inter-plataformas			
JADE	CSELT http://jade.cselt.it	✓	✓	✓	✓	✓	✓
FIPA-OS	Emorphia http://fipa-os.sourceforge.net	✓	✓	✓	✓	✓	✓
ZEUS	British Telecom www.labs.bt.com/projects/agents/zeus	✓	✓	✓	✓	✗	✓
JACK	Agent Oriented Software http://www.agent—software.com	✓	✓	✓	✗	✗	✓
GRASSHOPPER 2	IKV++ Technologies AG http://grasshopper.de	✓	✓	✓	✗	✓	✓
ADK	Trylian http://www.trylian.com	✓	✗	✗	✗	?	✓
JAS	Fujitsu, HP, IBM, SUN, ... http://java-agent.org	✓	✓	✓	✓	✗	?
AgentBuilder	Intellione Technologies http://www.agentbuilder.com	✗	✗	✗	✗	✗	?
MadKit	MadKit Team http://www.madkit.org	✗	✗	✗	✓	✗	?
Comtec Agent Platform	Communication Technologies http://ais.comtec.co.jp/ap	✓	✓	✓	✓	✗	✗
Bee-Agent	Toshiba http://www2-toshiba.co.jp/beeagent/index.htm	✗	✗	✗	✗	✗	✓
Aglets	IBM Japan http://trl.ibm.com/agents	✗	✗	✗	✓	✗	✓

Tabla 4-3. Revisión de Plataformas/Herramientas de Desarrollo de Agentes basadas en JAVA basada en (Vrba, P. 2003).

En la Tabla 4-3 se resume el cumplimiento (✓), el no cumplimiento (✗), o la duda (?) de las variables estudiadas en las diferentes plataformas de agentes basada en JAVA.

Existen otras plataformas basadas y no basadas en JAVA. En la red de excelencia europea Agentlink (www.agentlink.org) se dispone de una pequeña descripción de alrededor de una cincuentena de Plataformas.

4.6 Ventajas del Uso de Sistemas Multiagente

A partir del análisis de que un Sistema Multiagente, y en general también de un Sistema basado en Agentes, se pueden identificar algunas ventajas significativas aplicables al Proceso de Programación de la Producción. Algunas de las ventajas del Sistema Multiagente son:

- **Permite de forma natural la convivencia de diferentes roles (gestiones).** Las diversas etapas del proceso de Planificación/Programación de la Producción en la industria del pavimento/revestimiento cerámico, tiene características lógicas y técnicas diferentes, implicando un conjunto de actividades diferentes. Los Procesos tienen diferentes restricciones y podrían utilizar diferentes modelos y métodos para la resolución de sus problemas. Los sistemas distribuidos y autónomos, aunque coordinados, parecen más apropiados que los sistemas centralizados y complejos que no reflejan las particularidades de cada ámbito de actuación.
- **Permiten alcanzar valores adecuados, y en su caso óptimos, de diversas medidas de forma simultánea y adaptándose al entorno cambiante.** La propuesta de un SMA supone la existencia de un conjunto de Agentes heterogéneos que se integran obteniendo excelentes valores de programas con objetivos diferentes relacionados con proceso diferentes. Dichos Agentes son capaces de percibir los cambios en el entorno y se adaptan produciendo cambios en su comunidad para alcanzar los objetivos establecidos. Su autonomía les permite realizar tareas encaminadas a alcanzar objetivos locales. La cooperación entre los Agentes posibilita la coordinación del sistema para alcanzar los objetivos globales, permitiendo a su vez la flexibilidad necesaria para adaptarse a los cambios.
- **Son Sistemas idóneos para entornos con Incertidumbre.** Un sistema de Planificación/Programación autónomo y distribuido basado en SMA se caracteriza por mejorar la reactividad ante eventos y esta potencialmente capacitado para resolver problemas de Programación de la Producción con Incertidumbre.
- **Representan de forma natural la evolución temporal de los acontecimientos.** Los Agentes operan de forma asíncrona pudiendo representar fácilmente situaciones de actividades en paralelo y sin coordinación aparente ante un evento como son las situaciones de mercado ante una oferta, o la ejecución de operaciones en máquinas o centros de trabajos diferentes.
- **Facilitan el Diseño y la Implementación de una Solución.** Los SMA proporcionan las bases para la creación de una arquitectura que posibilita la reducción de complejidad, la flexibilidad, la escalabilidad y la tolerancia a fallos.

- Por la forma de plantear la búsqueda de soluciones los Sistemas basados en **Agentes permiten extender su aplicación al ámbito de la Cadena de Suministro**. De esta forma la Programación de la Producción podría ser considerada de forma natural dentro de este ámbito.
- **Son Sistemas Robustos**. Un SMA mantiene su funcionalidad a pesar de fallos individuales puntuales. Es fiable.
- **Son Sistemas Escalables**. Un SMA es inherentemente modular y permite una rápida respuesta ante nuevas necesidades del sistema. Un SMA es abierto permitiendo la integración de dinámica de nuevos Agentes, eliminación de los existentes o actualización de los mismos con diferentes funcionalidades o roles.
- **Usan computaciones paralelas** a través de un gran número de procesadores, los cuales pueden proveer Sistemas de secuenciación/Programación con gran eficiencia.
- **Conexión física a los dispositivos** que representan para realizar una Programación de la Producción dinámica en tiempo real, gracias a los Agentes recurso.

Estas son algunas de las razones que permiten enfocar tanto los trabajos de resolución cuantitativa del problema de Programación de la Producción, como de Gestión de dicho proceso. Las actividades de Gestión del Proceso de Programación de la Producción suponen la integración con el Proceso de Planificación y con el Proceso de Control de la Producción. Este proceso es por su propia naturaleza complejo y su correcta evolución depende del grado de colaboración y de acuerdo entre las partes. En ambos aspectos, resolución de problemas con un enfoque de Cuantitativo y Gestión, los Sistemas Multiagente puede ser una herramienta que aporte gran valor.

4.7 Sistemas Inteligentes de Fabricación basados en Agentes

Según (Shen, W. et al. 1999), las empresas fabriles del siglo XXI se encontrarán en un entorno donde los mercados serán frecuentemente cambiantes, donde habrá nuevas tecnologías emergiendo continuamente, con competidores actuando de forma globalizada. Por lo que las estrategias de los sistemas productivos deberían cambiar para soportar la competencia global, la innovación e introducción de nuevos productos, y los mercados sumamente reactivos. Estos sistemas de fabricación necesitarán satisfacer, según los autores, los siguientes requerimientos: Integración empresarial, organización distribuida, entornos heterogéneos, interoperabilidad, estructuras dinámicas y abierta, cooperación, integración de los humanos

con el software y el hardware, agilidad, escalabilidad, tolerancia a fallos. Algunos de estos nuevos requerimientos, hoy en día evidentes, son posibles soportarlos tecnológicamente con los sistemas basados en Agentes.

Las técnicas de Inteligencia Artificial (AI) se están usando en la Fabricación Inteligente hace más de veinte años. No obstante, los desarrollos en los Sistemas Multiagente en el nuevo dominio de Inteligencia Artificial Distribuida (DAI) han aportado posibilidades interesantes. Por eso, en los últimos diez años, los investigadores han aplicado la tecnología de Agentes a la Integración de la fabricación en las Empresas y la Cadena de Suministro, Planificación, Secuenciación, Control de ejecución, Gestión de Inventarios y desarrollo de nuevos tipos de sistemas de fabricación como los sistemas holónicos de fabricación.

En los sistemas de fabricación inteligente, los Agentes pueden ser usados para representar tanto recursos de fabricación como trabajadores, celdas, máquinas, herramientas, instalaciones fijas, AGV's, como productos, componentes, operaciones para facilitar la planificación de los recursos de fabricación, secuenciación y control de ejecución.

A continuación se realiza un pequeño recorrido por los aspectos relacionados con la aplicación de la Teoría de Agentes al Sistema de Planificación y Control de Operaciones, a la Integración entre la Planificación y la Programación de la Producción y al entendimiento de la propuesta de Sistemas de Fabricación Holónicos.

4.7.1 Planificación, Programación y Control de la Producción

En el Sistema de Planificación y Control de Operaciones existen tres actividades clave como son: La Planificación, la Programación y el Control de la Producción.

Por sus características los Sistemas basados en Agentes han sido utilizados para abordar todas ellas. En unas ocasiones de forma individual, y en otras ocasiones de forma conjunta. La Planificación y el Control de Planta han sido los procesos más extensamente tratados. Se debe tener en cuenta que ambos entran en el entorno de la toma de decisiones distribuidas, aunque con enfoques diferentes.

La Planificación de la Producción debe tener en cuenta un conjunto de requerimientos y decisiones distribuidas que residen en los fabricantes, proveedores, y clientes. En las últimas décadas la distribución de las decisiones se ha potenciado enormemente con la incorporación de los conceptos de Empresa Virtual, y Empresa Extendida, que junto con los enfoques de

Cadena/Red de suministro han desembocado en los planteamientos de Planificación Colaborativa.

La Programación de la Producción es un problema complejo, particularmente cuando se da en un entorno abierto y dinámico, que suelen ser todos los entornos reales. Una planta productiva, es un entorno incierto en casi todos los enfoques que se realicen. Se puede requerir al sistema para que proporcione tareas adicionales que no habían sido previstas, y algunas veces está permitido omitir ciertas tareas. Los recursos disponibles para ejecutar las tareas están sujetos a cambios. Algunos recursos pueden convertirse en no disponibles, y pueden introducirse recursos adicionales. El tiempo de comienzo y el tiempo de proceso de una operación suele estar sujeto también a variaciones. Una operación puede consumir más o menos tiempo del previsto, y las tareas pueden activarse más pronto o más tarde. Debido a sus aspectos altamente combinatorios, su naturaleza dinámica y su interés práctico para los sistemas de fabricación, el problema de Programación de la Producción ha sido ampliamente estudiado en la literatura mediante diversas técnicas: Optimizadoras, Heurísticas, Metaheurísticas, etc. Recientemente, también la Tecnología de Agentes ha sido utilizada en la resolución de este problema.

El control de planta tiene como objetivo el seguimiento de los programas propuestos y la aplicación de medidas para la corrección de posibles desviaciones. El problema de control se puede considerar en dos niveles: El primero es el que considera a los recursos de forma individual, y lo supervisa en función de las expectativas de entidades superiores. El segundo nivel considera la coordinación del conjunto de recursos individuales. La tecnología de Agentes se suele aplicar en el nivel de coordinación aunque existen ejemplos de su aplicación a nivel individual (Brennan, R., Balasubramanian, S., & Norrie, D. H. 1997) y (Wang, L., Balasubramanian, S., & Norrie, D. H. 1998). Una de las primeras propuestas que aborda de manera conjunta la Programación de la Producción y el Control de Planta es (Shaw, M. J. P. 1987) que introdujo el uso de mecanismos de subastas. Los autores de (Parunak, H. V. 1987) fueron otros de los pioneros en la utilización de sistemas basados en Agentes en este ámbito en el proyecto YAMS.

A continuación se profundiza en los aspectos de integración entre las áreas de Planificación y Programación de la Producción.

4.7.2 Integración de la Planificación y la Programación de la Producción

Aproximaciones tradicionales a la Planificación y la Programación de la Producción no consideran las restricciones de dichos dominios simultáneamente. A pesar de ser parciales

estas aproximaciones han estado en boga debido a la no disponibilidad de una estructura unificada. Las aproximaciones basadas en Agentes proporcionan una posible forma de integrar actividades de Planificación y Programación de la Producción por medio de la coordinación a nivel de empresa. En (Muscettola, N. 2008) se abordan los problemas de integración de ambas áreas desde un enfoque de Sistemas de Fabricación Inteligentes. Existen diversos ejemplos con este tipo de enfoque en la literatura. Sin embargo se desea comentar aquellos que ilustran las dos líneas más significativas. El primero es el que se corresponde con el proyecto MetaMorph y en el se resalta el esfuerzo que se está realizando en la integración de dichos procesos pero en el ámbito de la Cadena de Suministro. El segundo es el proyecto descrito por (Gu, P., Balasubramanian, S., & Norrie, D. H. 1997) en el que se propone una integración de ambos procesos resaltando la idea de integración empresarial. Si bien ambos enfoques no son en absoluto incompatibles cada uno muestra un tipo de esfuerzo diferente con unas características distintas.

El proyecto MetaMorph (Shen, W. et al. 1998) es un ejemplo de integración de Planificación y la Programación de la Producción en el ámbito de la Cadena de Suministro. Utiliza una arquitectura mediador-central híbrida basada en Agente para integrar socios, suministradores y clientes dinámicamente con la empresa principal a través de sus respectivos mediadores dentro de una red de cadenas de suministro vía internet e intranets. Los Agentes pueden ser utilizados para representar recursos (máquinas, herramientas, etc.) y piezas de fabricación, para encapsular los sistemas software existentes, para actuar como coordinadores de sistema/subsistema (mediadores), ejecutar una o más funciones de la cadena de suministro.

En general la utilización de la teoría de Agentes a la cadena de Suministro, normalmente con una aproximación de Integración Empresarial, ha proporcionado los siguientes beneficios:

- Se produce un mejor ajuste entre los requerimientos del mercado y la forma de actuar de la empresa.
- Se realiza un esfuerzo de mejora global que involucra a todos los actores de la Cadena de Suministro: fabricantes, proveedores, clientes.
- Se realiza una mejor asignación/utilización de los recursos de la cadena de Suministro.
- Se consigue una mejor gestión de los materiales e inventarios de la Cadena de Suministro al realizar un enfoque dinámico.
- Aumenta el valor de la información por lo que el flujo de la misma se mejora.

Sin embargo, los problemas de seguridad resultado de una arquitectura abierta de sistemas basados en Agentes, en concreto cuando se utiliza Internet y tecnología de agentes móviles, son evidentes, aunque estos problemas no son específicos de los Sistemas Multiagente y puede que sean mitigados con futuras investigaciones.

A partir de la idea de integración empresarial los autores de (Gu, P. et al. 1997) proponen un mecanismo de aliniamiento de los procesos de Diseño, Planificación y Programación de la Producción. Se propone un Sistema Multiagente donde las rutas y secuencias en las operaciones se logran mediante redes contractuales de ofertas. La tarea de Planificación de la Producción que incluye está distribuida entre Agentes máquina individuales. No existe un Planificador centralizado. La asignación de trabajos y selección de los mismos se lleva a cabo a través de procesos jerárquicos de ofertas entre Agentes máquina y el gestor del taller, entre Agentes máquina de distintos niveles y entre Agentes máquina y Agentes herramienta. Se configuró un modelo de costes para facilitar la toma de decisiones en el proceso de negociación.

Otro aspecto que suele incorporar la utilización de sistemas basados en Agentes es su proximidad a los entornos dinámicos. Los sistemas tradicionales alternan la Programación y Ejecución. Por ejemplo, una compañía genera cada semana un Programa Productivo en relación con sus operaciones de fabricación del siguiente periodo. Sin embargo, el mundo real tiende a cambiar de forma que invalida los Programas anticipados. Los sistemas naturales no planifican anticipadamente, sino que ajustan sus operaciones en una escala de tiempo comparable a aquella en que su entorno cambia. Los sistemas basados en Agentes propician un acercamiento a este tipo de filosofías, si bien las condiciones de cada caso pueden no permitir su aplicación radical, al menos supone la introducción a un cambio de perspectiva.

Un ejemplo ilustrativo de la integración entre la Planificación y la Programación de la Producción en un entorno dinámico son las propuestas de (McDonell, P., Smith, S., Joshi, S., & Kumara, S. R. T. 1999) y (Lim, M. K. & Zhang, D. Z. 2004). En (McDonell, P. et al. 1999) la integración de la Planificación/Programación de las operaciones en tiempo real y el control del taller se consiguen mediante procesos de pujas progresivas realizadas en paralelo entre un Agente gestor y múltiples Agentes recurso. En este acercamiento, el plan de procesos de producción completo puede ser construido, incluyendo los planes de herramientas, instalaciones y transporte. Se tiene en cuenta el proceso de control del taller.

En (Lim, M. K. et al. 2004) se afronta el reto de integrar el proceso de Planificación y Programación de la Producción para poder ser más reactivos en el caso de variaciones en los cambios súbitos en la demanda y los tipos de productos que se ofrecen en mercados muy dinámicos. Se propone un Sistema Multiagente formado por dos niveles de Agentes: Los Informadores y los Ejecutores. Los Agentes Informadores se utilizan para proveer de datos al

resto de Agentes. Los Agentes ejecutores disponen de un Agente Gestor que se encarga de coordinar las acciones del resto de Agentes (secuenciador, grupo de máquinas, programador, optimizador y un número variable de Agentes recurso y trabajo). Para la negociación entre las partes se utiliza el protocolo Contrat-Net. La plataforma se implementa en JAVA.

Debido a la naturaleza autónoma y cooperativa de los Agentes, las funciones de Programación y planificación de la Producción pueden ser integradas, y en cualquiera de los casos existen dos alternativa claras de afrontarlo, bien con un nivel elevado de Agentes funcionales trabajando por separados, o bien con un nivel bajo de Agentes recurso con un coordinador.

Proyecto	Referencia	Principales Características
CAMPS	(Miyashita, K. 1998)	Se centra en el problema de descomposición y coordinación de tareas. Se propone un prototipo que gestiona restricciones intra e inter-agentes. Se incluye una metodología de reparación del Plan y/o Programa obtenido de forma distribuida. Se utilizan coaliciones de Agentes para resolver el problema.
ProPlant	(Pechoucek, M., Marik, V., & Stepankova, O. 2000)	Enfatizan el proceso de creación de equipos de agentes utilizando métodos basados en el conocimiento. Los agentes utilizan un modelo de conocimiento denominado tribase (3bA)
HMS	(Gou, L., Hasegawa, T., Luh, P., Tamura, S., & Oblack, J. 1994) Toshiba	Propone y potencia la utilización del Sistemas Holónicos de Fabricación en la Industria para abordar aspecto de Planificación, Programación y Control.
Sin Nombre identificado	(Lim, M. K. et al. 2004) University of Exeter, UK	Desarrolla una plataforma que permita afrontar dinámicamente los ajustes de los Planes y Programas Productivos en función de las variaciones de la demanda.
Sin Nombre identificado	(Usher, J. M. 2003) University of Mississippi, USA	Propone una serie de métodos de negociación para la configuración de la ruta de los trabajos en un taller. La arquitectura del sistema mantiene información de la Planificación y de la Ejecución de los trabajos.

Tabla 4-4. Resumen basado en (Shen, W. M., Wang, L. H., & Hao, Q. 2006) sobre proyectos de investigación que trabajan en la integración de los procesos de Planificación y Programación aplicando teoría de Agentes.

Otros proyectos de investigación interesantes que han planteando algún tipo de integración entre la Planificación y la Programación de la Producción se encuentran resumidos en la Tabla 4-4.

Se ha mostrado como la idea de integrar los procesos de Planificación y de Programación de la Producción inicio su andadura hace ya algunos años. Y ha puesto de manifiesto la utilidad de las propuestas realizadas. No obstante, en muchas ocasiones de ha podido constatar la falta de soluciones, metodologías o propuestas genéricas que permitieran un uso extensivo de las arquitecturas Multiagente a múltiples tipos de problemas.

4.7.3 Sistemas Holónicos

La Fabricación Holónica es un paradigma desarrollado en el marco del programa *Intelligent Manufacturing Systems* (IMS - Sistemas de Fabricación Inteligente). La Fabricación Holónica se basa en conceptos de los “sistemas holónicos”, desarrollados por Arthur Koestler. El trabajo realizado en el programa IMS ha trasladado estos conceptos al mundo de la fabricación, considerando al sistema de fabricación como un compuesto de módulos autónomos con control distribuido.

Según (Giret-Boggino, A. S. 2005) los holones se pueden definir como paradigma con las siguientes características básicas: Autonomía, Cooperación y Reorganización. Además de estas características, que las podemos llamar “propiedades de comportamiento”, los holones tienen “características estructurales”. Una de ellas es la “recursividad”, la cual permite a los holones estar formados internamente por entidades auto-similares (holones), que a su vez pueden estar formadas por holones, y así sucesivamente (hasta que se llega a un nivel atómico en el cual nuevas sub-divisiones son imposibles o de poca utilidad para el dominio de aplicación).

Los holones pueden estar compuestos por holones los cuales 1) pueden ser holarquías, 2) pueden participar en varias holarquías simultáneamente, y 3) pueden entrar, salir, o cambiar - es decir, holones que forman holarquías dinámicas. Otra propiedad estructural importante, como ha sido definida por el consorcio (HMS 1994), es que los holones usualmente poseen una parte de procesamiento de información con una parte opcional de procesamiento físico.

En la Tabla 4-5 se pueden apreciar las características que diferencian Holones y Agentes.

Propiedad	Holón	Agente
Autonomía	Si	Si
Reactividad	Si	Si
Proactividad	Si	Si
Habilidad Social	Si. La Interfaz Humana es específica de cada holón.	Si. La Interfaz Humana se implementa generalmente por uno o varios agentes especializados.
Cooperación	Si. Los holones nunca rechazan de manera deliberada la cooperación con otro holón.	Si. El agente puede competir y cooperar.
Reorganización	Si. Holarquías.	Si. Jerarquías, organización horizontal, heterarquías, etc.
Racionalidad	Si	Si
Aprendizaje	Si	Si
Benevolencia	Si	SI
Movilidad	Los holones raramente necesitarán de movilidad para la ejecución de sus tareas.	Si
Rekursión	Si	No existe una definición formal de agente recursivo, aunque existen trabajos en el área SMA que utiliza estructuras complejas de agentes.
Procesamiento de la Información y Físico	Si. La separación es explícita, aunque la parte de Procesamiento Físico es opcional.	No existe una separación explícita.
Actitudes mentales	Si. Los holones no necesitan razonar acerca de sus propias actitudes mentales o aquellas de otras unidades de control.	Si

Tabla 4-5. Tabla comparativa Holones vs Agentes de (Giret-Boggino, A. S. 2005)

El holón entendido como entidad perteneciente a un Sistema Holónico de Fabricación debe respetar unas normas en su uso y definición, que en relación con la Programación de la Producción son básicamente:

- A cada holón se le asocia un módulo de toma de decisión local y capacidad de computación.
- Se utiliza una estrategia de interacción cooperativa la cual rige la manera en la que los holones intercambian información, y determinan soluciones aceptadas por todos.
- Un mecanismo o protocolo de intercambio que gestiona el intercambio de los tipos de mensajes necesarios para ejecutar una estrategia cooperativa.
- Un medio que asegura que los requisitos globales de la fábrica son abordados.
- Un grado de coordinación central.

Desde el punto de vista del autor de este trabajo, estas normas convierten a los sistemas holónicos en elementos demasiado rígidos como para poder plantear propuestas que se relacionan más con lo mental que con lo físico, como es el caso de los Planes y Programas de Producción. Se ha entendido que una propuesta innovadora en el ámbito de la Programación de la Producción debe contar con la mayor flexibilidad posible, y debe seguir la línea de propuestas como la de los algoritmos bio-inspirados, que en algunos ámbitos están ofreciendo buenos resultados.

En este apartado se ha querido presentar el concepto de holón y distinguir términos que en ocasiones han sido confundidos, como lo son el de Holón y el Agente. La conclusión es que son paradigmas diferentes aunque similares. Quizás su mayor diferencia no estribe en la definición del concepto holón o Agente sino en la forma de aplicarlo.

4.8 Aplicaciones de los Sistemas Multiagente a la Programación de la Producción

La aplicación de los Sistemas Multiagente al campo de la Fabricación ha ido progresivamente abarcando diversos aspectos de dicha área. La Programación de la Producción no ha sido ajena al uso de esta herramienta y a las nuevas perspectivas que desde ésta se han podido ir generando. Quizás la producción científica y sus aplicaciones han alcanzado un nivel significativo en la última década. Los autores han ido incorporando entre sus campos de

investigación aspectos de suma importancia como la negociación, aprendizaje, etc. superando el enfoque clásico centrado únicamente aspectos finalistas.

Los primeros resultados se han derivado del trabajo realizado en diversos proyectos de investigación. Mucho de ellos han servido para unificar criterios y aunar esfuerzo sobre aspectos no resueltos. Algunos de estos se han resumido en la Tabla 4-6.

Proyecto	Referencia	Principales Características
ABACUS	UCB, UMIST (McEleney, B., O'Hare, G. M. P., & Sampson, J. 1998)	Utiliza agentes funcionales; aproximación BDI para diseño de agentes.
ADDYMS	(Butler, J. et al. 1992)	Los agentes representan recursos físicos; scheduling de recursos local dinámico.
ACM	Sandia Lab. (Goldsmith, S. Y. & Interrante, L. D. 1998)	Utiliza agentes físicos: agentes máquina y agentes pieza.
CAMPS	(Miyashita, K. 1998)	Metodología basada en reparaciones junto con mecanismos basados en restricciones.
CORTES	CMU (Sadeh, N. & Fox, M. S. 1989)	Técnicas microoportunistas para resolución de problemas de scheduling.
DAS	U. of Strathclyde (Burke, P. & Prosser, P. 1994)	Arquitectura jerárquica con agentes que Representan recursos, grupos de recursos, y un proceso de scheduling completo.
IFCF	Purdue (Lin, G. Y. J. & Solberg, J. J. 1992)	Agentes recurso representan recursos físico; modelo de control tipo mercado.
LMS	(Fordyce, K. & Sullivan, G. G. 1994)	Utiliza agentes funcionales; protocolo de votación para la comunicación
MASCADA	Daimler-Benz AG, KULeuven (Brückner, S., Wyns, J., Peeters, P., & Kollingbaum, M. 1998)	Comportamiento emergente en control de fabricación; manejo proactivo de perturbaciones; agentes activables en caliente.

Tabla 4-6. Resumen de Proyecto orientados a la Programación de la Producción utilizando Sistemas Multiagentes basado en (Shen, W. M. et al. 2006).

Una vez los diversos proyectos han ido consolidando líneas de trabajo se han realizado aportaciones muy interesantes en diversas revistas y congresos de las áreas de Inteligencia Artificial, Investigación Operativa y Gestión de la Producción. A continuación se realiza una

revisión de las referencias más recientes en las que, desde distintos puntos de vista, se proponen la aplicación de Agentes, Sistemas basados en Agentes o Multiagentes para abordar la Programación de la Producción. En concreto se incide en los siguientes aspectos:

- Soluciones basadas en Teoría de Agentes para la obtención de Programas Productivos en diferentes tipos de talleres.
- Soluciones que intentan tener una visión amplia del sistema productivo, y que incluyen los aspectos de Programación de la Producción como parte de la propuesta.
- Adaptación y mejora de las habilidades sociales de los Agentes utilizados para obtener una solución de forma colaborativa.
- Adaptación y mejora del proceso de aprendizaje en los Agentes o Sistemas Multiagente utilizados en la obtención de un Programa, normalmente en entornos colaborativos.
- Análisis del impacto de las arquitecturas Multiagente en la calidad de las soluciones.

El conjunto de temas propuestos aportan una visión tanto de cómo debe plantearse el proceso/algoritmo de resolución como el conjunto de técnicas, y habilidades, que un sistema basado en Agentes o Multiagente necesita implementar para poder ser aplicado en casos reales.

4.8.1 Soluciones basadas en Teoría de Agentes

La teoría de Agentes aplicada a la Programación de la Producción ha proporcionado en su corta vida múltiples aplicaciones. Con los trabajos realizados por (Cardon, A., Galinho, T., & Vacher, J. P. 2000), (Ng, C. T., Cheng, T. C. E., & Yuan, J. J. 2006), (Csaji, B. C. & Monostori, L. 2005), (Babayan, A. & He, D. 2004), (Aydin, M. E. & Fogarty, T. C. 2004), (Yu, X. F. & Ram, B. 2006), (Wong, T. N., Leung, C. W., Mak, K. L., & Fung, R. Y. K. 2006) y (Babayan, A. et al. 2004) se pretende realizar un análisis tradicional, es decir basado en el algoritmo/procedimiento de resolución, en este caso del Sistema Multiagente empleado en diversos tipos de talleres. En dichas referencias lo relevante es como se obtiene la solución para obtener un Programa Productivo, ya sea con un enfoque Predictivo o Reactivo. Sin embargo, en los trabajos de (Sun, J. & Xue, D. 2001) y (Cowling, P. I., Ouelhadj, D., & Petrovic, S. 2004) se quiere incidir en como

se puede plantear y solventar la reprogramación de la producción debida a alguna necesidad de adaptación del Programa Productivo en curso.

Para un problema de Programación de la Producción en un taller de general con n trabajos y m máquinas con el objetivo de minimizar adelantos y retrasos sobre una fecha propuesta (Cardon, A. et al. 2000) proponen la aplicación de un Sistema Multiagente en el que se embebe un Algoritmo Genético. Los autores hacen agrupaciones de entidades en base a la proximidad de ciertos elementos sobre un determinado punto del diagrama Gantt. En dichos grupos los Agentes representan elementos como máquinas o trabajos, y también entidades de gestión como distribuidores de trabajos, o gestores. Con esta parcelación aparecen Agentes que se sitúan en diferentes niveles. Se puede hablar de meta-Agentes que utilizan meta-información y que su ámbito de actuación es el grupo (grupo de máquinas, grupo de trabajos, etc.), así como de micro-Agentes que representan entidades individuales. La idea perseguida por los autores es que los meta-Agentes sean los responsables de canalizar la colaboración de su grupo con otros grupos. Cada agente tiene un universo conocido formado por trabajos, máquinas, un Agente Distribuidor de Trabajos (ADT) y el Programa obtenido. Dentro del Universo, las máquinas o los trabajos son Agentes reactivos que sólo pueden dar conformidad a ciertas operaciones, pero no proponen acciones.

Como núcleo fundamental dentro del proceso de colaboración está la negociación entre grupos y entidades. Los autores proponen el uso de Contrat-Net como protocolo de negociación. Se propone a un Agente, el ADT, que realizará el reparto de tareas entre otros Agentes. Cuando el ADT realiza una propuesta de asignación de un trabajo a una máquina lo hace a través de un Agente que representa a la máquina en el proceso negociador, el Agente negociador (AN). Entre el ADT y el AN se inicia un proceso negociador utilizando Contrat-Net. El ADT realiza la propuesta a varias máquinas, lo que genera competencia entre ellas.

Como se puede ver en la Figura 4-8 entre el Agente AN y los Agentes que representan a las máquinas puede existir una compleja trama de Agentes. Los autores proponen el uso de un Agente intermedio que recibe la propuesta de asignar a un conjunto de trabajos, y que mediante el uso de una Agente forward, que conoce la disposición del diagrama Gantt hacia el futuro, y mediante un Agente Backward, que conoce la evolución ya realizada de los trabajos en el tiempo, son capaces de calcular las nuevas asignaciones intentando minimizar un objetivo basado en un valor económico. Los Agentes Forward y Backward se relacionan con Agentes que representan al conjunto de máquinas.

Sobre el Sistema Multiagente propuesto los autores introducen el concepto de caracterización de los Agentes como si fueren seres vivos. Mediante una combinación de 10 bits se definen las reglas que cada Agente puede usar ante ciertas situaciones. Dichas reglas se ubican en una

base de datos. El cambio de un bit implica la modificación del carácter del Agente y de su comportamiento (más agresivo, más sociable, etc.). El comportamiento de un Agente influye en su evolución en el Sistema. Mediante un Algoritmo Genético (Cardon, A. et al. 2000) identifican la función de selección, mutación, cruce y ajuste (fitness). Dado que el objetivo es minimizar adelantos y retrasos simultáneamente la función de ajuste se plantea como multiobjetivo. El conjunto de macro y micro-Agentes comentado anteriormente evoluciona mediante procedimientos de cruce y mutación durante el proceso de búsqueda de una solución. Los Agentes van ajustando su forma de actuar, modificando sus 10 bit mediante el Algoritmo Genético, hasta que se considera que han alcanzado una solución adecuada.

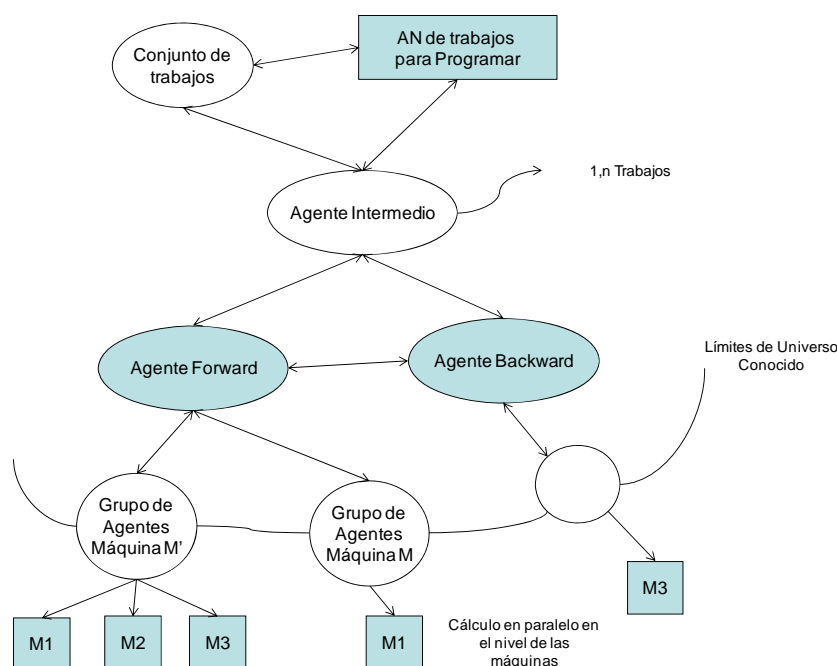


Figura 4-8. Estructura del protocolo Contrat-Net entre diferentes niveles de (Cardon, A. et al. 2000).

En el caso de una máquina, (Ng, C. T. et al. 2006) propone una resolución basada en un Agente. En concreto implementan dos enfoques diferentes. En un caso desarrollan un Agente con el objetivo de alcanzar un Programa Productivo que minimice el Cmax. En este primer caso demuestran que la dificultad del problema es NP-Hard y utilizan un sistema pseudo-polinomial para solventarlo. En el segundo caso, el objetivo es minimizar el retraso, impidiendo que exceda de un tiempo dado, mostrando que se trata también de dificultad NP-Hard.

Se ha encontrado una aplicación a un taller de máquinas paralelas no relacionadas en un entorno estocástico. En (Csaji, B. C. et al. 2005) se propone para dicho problema el uso de un Sistema Multiagente donde cada Agente implementa, basándose en programación dinámica, un procedimiento para el cálculo parcial de la solución. El sistema propuesto se regula

mediante un proceso de toma de decisiones que utiliza la teoría de cadenas de markov para alcanzar una solución próxima al óptimo.

En el trabajo de (Babayan, A. et al. 2004) se presenta un problema de taller de flujo estático de tres etapas y n trabajos. La solución al problema está basada en un algoritmo dotado de un sistema de Agentes. Estos Agentes se separan en tres grupos: secuenciables, secuenciados, y no secuenciables. Se establecen dos fases en el proceso, que denomina juego externo y juego interno. En ambas se utiliza la teoría de juegos. En el juego externo se seleccionan los Agentes secuenciables (listos en la cola de entrada) para pasar al juego interno donde realmente se establecen las secuencias sólo de los Agentes/trabajos que han entrado. En ambos casos cada Agente calcula el valor de una variable que mide su aportación al juego, y el gestor decide con que propuesta se queda. Tanto el juego interno como externo utilizan como valor de referencia lo que denominan "Shapley Value" que mide la aportación de un trabajo a un conjunto existente en función del tamaño de la coalición que se forma y del C_{max} que se obtiene. Los autores realizan un análisis experimental para treinta y cincuenta trabajos, combinan dos y tres máquinas por etapa, y establecen tres tipos de tiempos de proceso (todas las etapas el mismo tiempo, las etapas dos y tres con tiempo superior a la uno, las etapas uno y tres con tiempo superior a la dos). Los resultados obtenidos se comparan con el algoritmo de (Soewandi, H. & Elmaghraby, S. E. 2001), que dividieron el taller de flujo de tres etapas en dos subproblemas (un taller de flujo con dos etapas y un problema simple de secuenciación con máquinas en paralelo). Tanto para los resultados propios como para los de Soewandi se establece un ratio entre el valor obtenido y el mejor valor encontrado. Se mejoran los resultados para algunas combinaciones pero no para todas.

En (Kutanoglu, E. & Wu, S. D. 1999) se propone un procedimiento basado en Sistema Multiagente mediante el cual se soluciona un modelo matemático relajado mediante Lagrange utilizando como mecanismo un sistema de subastas. El autor aplica la propuesta a un taller de flujo estático con recursos distribuidos con el objetivo de minimizar el retraso.

Ya en el caso de un taller general estático (Aydin, M. E. et al. 2004) establecen un conjunto de Agentes (A-Team) que representan, cada uno, a un tipo de procedimiento diferente, y que comparten una zona de memoria común a través de la cual pueden intercambiar propuestas unos con otros. La solución al problema se obtiene mediante la computación en paralelo y de forma asíncrona de los diferentes algoritmos. Al compartir memoria, e intercambiar soluciones, el resultado del mejor Agente/algoritmo no es igual al resultado que el mismo Agente hubiese obtenido de forma individual. Se utiliza un algoritmo denominado "compose", en el cual un Agente inicializa la población y genera una solución inicial para el taller general mediante la asignación a cada máquina una cola de trabajos, que procesará mediante heurísticas (fifo, lifo o spt). Existen otros algoritmos que están basados en: Recocido Simulado, Búsqueda Tabú, Algoritmos de Escalada y Algoritmos Genéticos. En algún caso se utiliza varias

veces el mismo tipo de algoritmo modificado. Existe también un Agente destructor que mantiene la población en los niveles deseados. Los autores finalmente realizan un experimento basado en un juego de datos. Se usa en concreto el juego diez por diez, con diez trabajos y diez máquinas en un taller general. Se comparan los resultados obtenidos por el A-Team por los resultados de los diferentes algoritmos ejecutados de forma individual.

En el mismo marco del taller general, pero con un enfoque dinámico, (Yu, X. F. et al. 2006) propone un plataforma denominada RTM-DS, que se basa en la asignación de un Agente a cada máquina. Estos Agentes se coordinan entre ellos para ver que máquina realiza cada trabajo de la cola. Además disponen de toda la información del sistema. Para la realización de la secuencia se utilizan dos reglas: La regla de selección de máquinas (MSR) y la regla de secuenciación de colas (QSR): La MSR se consigue a través de la coordinación de todos los Agentes-máquina. Cuando una Agente percibe que hay un trabajo en espera, si esa máquina puede realizar el trabajo, el Agente decide si se ofrece o no a realizar el trabajo dependiendo de su tiempo mínimo de ejecución. Cuando se ofrece más de un trabajo, se utiliza un método computacional para asignar la máquina. Este proceso está inspirado en el comportamiento de los insectos. Por otra parte la QSR es implementada por cada Agente-máquina. Cuando una máquina acaba un trabajo y tiene otros en cola, el Agente-máquina selecciona el siguiente utilizando una regla específica de secuenciación de colas, propuesta por los autores. En (Wong, T. N. et al. 2006) se realiza una propuesta para el taller general dinámico y flexible.

Los autores de (Wong, T. N. et al. 2006) proponen un primer sistema llamado MAN (Multi-agent Negotiation) que se basa en la negociación entre dos tipos de Agentes; PA (part agents) y MA (machine agents), que son utilizados para representar a trabajos y máquinas respectivamente y poseen toda información de su representado. La negociación entre estos se basa en un sistema de ofertas que los MA's envían a los PA's cuando hay un trabajo en cola, el PA acepta la mejor oferta según el Critical Ratio. En el segundo modelo, HAN (Hybrid-based Agent Negotiation) se añade la figura de un Agente supervisor (SA). El objetivo de este SA es establecer un control jerárquico de la estructura SMA para gestionar las negociaciones entre los PA's y los MA's. Este Agente dispone de toda la información de los demás Agentes y dependiendo de su habilidad, se pueden considerar diferentes objetivos fijados previamente, como pueden ser minimizar el makespan, la tardanza (tardiness) de los trabajos o la carga asignable a cada máquina. Este Agente solamente interviene cuando hay conflictos entre las decisiones locales y el objetivo del sistema.

Para un taller general dinámico y con incertidumbre, (Csaji, B. C., Kadar, B., & Monostori, L. 2003) presentan un método para calcular un Programa Productivo mediante un doble nivel de adaptación. Aplican un Sistema Multiagente con arquitectura heterárquica basado en (Valckenaers, P. & Van Brussel, H. 2003). Su trabajo mejora el proceso de aprendizaje

mediante una mezcla de redes neuronales y aprendizaje reforzado, llamado Programación Neurodinámica.

Un ámbito interesante en la literatura es el que hace referencia a la utilización de los sistemas basados en Agentes para la Reprogramación de la Producción, es decir para la modificación de un Programa Productivo para adaptarlo a una nueva circunstancia. En (Sun, J. et al. 2001) se propone el uso de la técnica Match-up en colaboración con un sistema basado en Agentes para modificar parcialmente un programa en curso. El objetivo es minimizar los cambios en el Programa al mismo tiempo que se mantiene la calidad del programa. Son dos los eventos que se consideran por la intervención del usuario, la cancelación y la inserción urgente de órdenes. La estrategia reactiva es doble. Por un lado se intenta mover los trabajos hacia adelante teniendo en cuenta su fecha más tardía de entrega, y por otro lado se intenta moverlos hacia atrás teniendo en cuenta la fecha más temprana de entrega. Si no es factible incluir en el programa una orden urgente entonces se cancela alguna de las programadas. Por otro lado, los autores consideran la existencia de eventos relativos al sistema, como es el caso de rotura de máquinas o indisposición de trabajadores. En este segundo caso se combina la técnica Match-up con el Sistema de Agentes.

El Sistema de Agentes se enmarca dentro de una propuesta de integración del Sistema Productivo y del diseño del producto. La Programación de la Producción se realiza a través de un Agente mediador. Cuando el Agente mediador recibe una petición de un Agente trabajo para ser secuenciado inicia un protocolo Contrat-Net con los Agentes recurso (máquina, trabajador, etc.), seleccionando la mejor oferta y asignándole dicho trabajo al recurso correspondiente.

Mediante la aplicación de los conceptos de Programación de la producción predictiva y reactiva (Cowling, P. I. et al. 2004) propone una serie de heurísticas que permiten la adaptación de un programa en curso en función de diversos eventos en el caso de una empresa de producción de acero. Los autores proponen realizar la coordinación entre la etapa de creación de barras de acero y la de rollos de acero mediante un sistema formado por Agentes. En la etapa responsable de la producción de rollos de acero existe un Agente (HSM) que se encarga de la creación de un programa predictivo mediante un algoritmo de Búsqueda Tabú. Y un programa reactivo mediante un conjunto de ocho heurísticas. En la etapa responsable de la producción de barras de acero existen dos tipos de Agente, el gestor de las barras (SY) y el responsable del colado de acero (CC-1..CC-n). El Agente SY utiliza dos heurísticas para la obtención del programa predictivo. Los programas predictivos de ambas etapas tienen objetivos diferentes.

El Sistema Multiagente utiliza un protocolo Contrat-Net, cuyo iniciador es el Agente HSM, para negociar la disponibilidad de barras de acero según las necesidades del proceso de producción de rollos de acero y sujeto a las restricciones del proceso de producción de barras de acero. Los autores realizan un estudio experimental del sistema propuesto alcanzando resultados exitosos en cuanto a la eficacia del mismo.

Se ha podido observar como los diversos autores han realizado propuestas indicando procedimientos que tienen como base la aplicación de sistemas basados en Agentes para resolver problemas de Programación de la Producción en los talleres más representativos (monomáquina, máquinas paralelas,, taller de flujo, y taller general). Casi todas las soluciones tienen en común la utilización de un protocolo de negociación Contrat-Net, que no en vano es uno de los más utilizados en general. El análisis de las diferentes referencias ha permitido disponer de diversos enfoques en cuanto a la construcción de arquitecturas de Agentes, y como pueden los Agentes tomar diversos roles.

Tres aspectos se pueden destacar como claves en la construcción de soluciones basada en Agentes. Un aspecto es la elección del tipo de arquitectura y asignación de roles a los Agentes. Otro aspecto es el diseño del proceso negociador, es decir el protocolo y su significado. No es solo importante seleccionar un protocolo (Contrat-Net, etc.) sino el identificar el contenido de los mensajes, su interpretación dentro del proceso y el conjunto de acciones que se asocian a cada uno. Y finalmente es fundamental el diseño de los estados en los que pueden trabajar los Agentes y la evolución de los mismos. Todos estos aspectos, y algunos otros de menor relevancia, son tenidos en cuenta en las diversas metodologías existentes para del desarrollo de Sistemas Multiagente.

4.8.2 Integración de la Programación de la Producción en el SPCO.

La Programación de la Producción no solo debe ser estudiada de forma independiente y aislada del resto del Sistema de Planificación y Control de Operaciones (SPCO), y en ocasiones también del Diseño del Sistema Productivo. Esta visión integradora permite obtener conclusiones relevantes sobre el comportamiento de los Agentes en entornos colaborativos, y sobre todo como actuar ante eventos procedentes de subsistema de gestión con los que es fundamental coordinarse.

En un entorno dinámico, y con el objetivo de encontrar el mejor Programa Productivo posible (Xue, D., Sun, J., & Norrie, D. H. 2001), (Keskinocak, P., Wu, F., Goodwin, R., Murthy, S., Akkiraju, R., Kumaran, S. et al. 2002), (Dang, T. T. & Frankovic, B. 2002), (Wang, Y. H., Yin, C. W., & Zhang, Y. 2003) y (Furusho, T., Nishi, T., & Konishi, M. 2008) presentan diferentes

arquitecturas con propósitos diferentes pero que no identifican un taller concreto, sino una planta genérica.

En (Xue, D. et al. 2001) se introduce un enfoque de Sistema de Fabricación Inteligente para identificar el Programa Productivo óptimo que permita para satisfacer las restricciones del producto y del taller. Cuando se recibe una orden, se crea un Agente-orden para representar los requisitos de cliente y este Agente se asocia con el diseño del producto, que es modelado combinando las características del caso. El sistema de Programación inteligente es activado subsiguientemente para identificar el programa óptimo que satisfaga las limitaciones de diseño del producto, y los requisitos industriales que son identificados en las características, y en limitaciones industriales que son descritas como recursos disponibles. Se utilizan dos estrategias de optimización: (1) tener disponible el producto lo antes posible, (2) comenzar el producto tan tarde como posible reducir el espacio para almacenar el producto producido. En este enfoque, las limitaciones de producto son modeladas utilizando un esquema de características basado en la representación del producto. Cuando la limitación temporal de la orden de cliente no puede ser satisfecha, el cliente es informado para cambiar las características del producto ordenado.

En (Keskinocak, P. et al. 2002) describe un sistema de apoyo a la toma de decisiones para Programar la Producción de papel. Según los autores este es el primer sistema que proporciona una solución integrada de Programación de la Producción de papel, y considera las interacciones entre etapas diferentes del proceso de fabricación y distribución. Utilizando un enfoque de optimización multicriterio, se generan múltiples programas para diferentes empresas, a menudo compitiendo por los mismos objetivos. El amplio conjunto de algoritmos utilizados por el sistema es embebido en un sistema de Agentes llamado Equipo Asíncrono (el A-TEAM).

En un entorno de tiempo real, los autores de (Dang, T. T. et al. 2002) asumen la existencia de n Agentes independientes que representan a n productos, donde cada Agente tiene un conjunto de operaciones para realizar. Existen un conjunto de máquinas que pueden realizar dichas operaciones. Cada Programa Productivo tiene un conjunto de parámetros como son la fecha de entrega prevista, el coste de un determinado Programa, y el tiempo de flujo. Se establece un proceso de negociación entre los Agentes del sistema.

En el caso de un sistema de fabricación ágil donde los procesos que se realizan se pueden descomponer en múltiples actividades (Wang, Y. H. et al. 2003) propone la creación de un Sistema Multiagente que represente los roles de los diferentes actores que permita una distribución funcional. Proponen tres tipos de Agente para el entorno de Programación de la

Producción que permita descomponer el problema y utiliza un sistema de toma de decisiones basado en reglas.

En (Furusho, T. et al. 2008) se propone un sistema distribuido para la Programación de la Producción, el enrutado de sistemas AGVs, y la planificación de materiales. Existen tres Agentes que representan a cada una de las tres partes del problema. Cada Agente intercambia información y recalcula una solución hasta que se alcanza una propuesta factible. Se demuestra la utilidad de esta propuesta frente a las tradicionales.

En el marco de esta tesis la visión integradora es fundamental, ya que como se expondrá en el próximo capítulo, deberemos tener en cuenta aspectos relacionados con la Planificación de la Producción que van a influir en las tareas de Programación de la Producción. Además de los aspectos generales relacionados con el diseño e implementación de los Sistemas Multiagente en este apartado se debe destacar como los diversos autores han planteado la coordinación de acciones. Este tratamiento es fundamental en el desarrollo de la propuesta que se realizará en capítulos posteriores.

4.8.3 Habilidades Sociales en la Aplicación de los SMA

La Programación de la Producción es un problema que cuando ha sido abordado mediante la aplicación de Sistemas Multiagente o basado en Agentes ha solido tener dificultades en los aspectos relacionados con la implementación de las habilidades sociales de éstos. Como ya se analizó con anterioridad como en un entorno de colaboración aspectos como la coordinación y/o la negociación adquieren una importancia sustancial. En este sentido algunos autores han realizado propuestas que buscan mejorar la relación entre Agentes cuando el objetivo es obtener un Programa Productivo. En (Liu, J. S. & Sycara, K. P. 1997) se presenta una solución para la coordinación de Agentes en un proceso iterativo y asíncrono de cálculo de un Programa Productivo en un taller General. En esta propuesta los Agentes representan recursos y trabajos. La propuesta intenta regular de forma equilibrada la cantidad en el flujo de comunicación entre Agentes, ya que un flujo bajo implica soluciones de baja calidad y un flujo alto implica bajo rendimiento del sistema. Los autores de (Kutanoglu, E. & Wu, S. D. 2006) también centra sus esfuerzos en la coordinación pero en este caso los Agentes representan programadores de centros de trabajo diferentes y el objetivo es obtener un programa conjunto de forma colaborativa. Por lo tanto, en la resolución del problema de coordinación el autor tiene bien presente los propios requerimientos del problema y propone el uso de internet como medio de comunicación.

En un entorno dinámico (Vancza, J. & Markus, A. 2000), (Lv, C. X., Sheng, Z. Q., Li, S. F., Yin, C. W., & Zhu, Y. L. 2006), (Sousa, P. & Ramos, C. 1999) y (Shin, M. & Jung, M. 2005) profundizan en los aspectos relacionados con las técnicas de negociación. Su objetivo es ser capaces de alcanzar un acuerdo que permita la asignación y secuenciación de trabajos de formar ágil. En (Vancza, J. et al. 2000) se utiliza un modelo de mercado basado decisiones en torno al valor económico de las mismas. (Lv, C. X. et al. 2006) propone un algoritmo adaptativo que contempla aspectos la Programación de la Producción predictiva y reactiva para un taller general. (Sousa, P. et al. 1999) se ocupa en profundidad sobre lo que denomina renegociación, es decir, de alcanzar acuerdos en casos especiales en los que no es posible acordar una forma de actuar utilizando reglas generales. Aborda también casos de indecisión. Los autores tienen presente en todo momento que su mayor restricción en la toma de decisiones es la limitación de tiempo.

En el trabajo de (Shin, M. et al. 2005) se propone un protocolo de negociación denominado MANPro (Mobile agent-based negotiation process). En MANPro, las negociaciones se realizan dentro de una estructura donde los T-agents (task agent) y los R-agents (resource agent) son autónomos (tienen objetivos propios) para operar en el sistema cooperando unos con otros. Los N-agents (negotiation agent), asumen el papel de centro de la negociación y de generadores de ofertas. El sistema de negociación mediante ofertas se representa intercambiando ofertas entre un iniciador y diversos postores. El mejor postor es seleccionado por el iniciador. El sistema funciona de la siguiente manera: El N-agent genera una oferta que es evaluada por los R-agents y los T-agents, si la oferta no satisface a ambas partes, se revisa la oferta. El N-agent busca una nueva restricción para el trabajo o la máquina, si la encuentra, la tiene en cuenta para realizar una nueva oferta, si no, cierra el proceso. El proceso se repite hasta que la oferta es aceptada.

En este apartado se puede destacar la aportación de (Zhong, W. C., Liu, J., & Jiao, L. C. 2005) para la programación de un taller general en el cual se aplican conceptos sumamente interesantes sobre colaboración y coordinación basados en entornos locales. Este trabajo es consecuencia de otra aportación previa, (Zhong, W. C., Liu, J., Xue, M. Z., & Jiao, L. C. 2004), en la cual se describe con detalle en núcleo del proceso que no es otro que un algoritmo genético basado en SMA.

La consideración de las habilidades sociales en un desarrollo Multiagente es fundamental. Estas definirán la forma de colaborar entre los diferentes Agentes que forman el sistema, y por lo tanto condicionarán la arquitectura del mismo. Por este motivo, era necesario analizar como algunas de las propuestas más relevantes afrontar este aspecto. La conclusión que se ha obtenido es que existen aproximaciones muy diversas y no se han definido plantillas, ni marcos que se hayan demostrado útiles para abordar problemas tipo. Cada autor ha identificado el problema que desea resolver y ha adaptado soluciones similares existiendo convirtiéndolas en

soluciones ad-hoc. Una línea de investigación interesante puede ser el análisis de formas de colaboración (coordinación, negociación, comunicación, etc.) adecuadas para diversos tipos de problemas de Programación de la Producción de forma que se potencien facilidades que permitan superar las dificultades que caractericen a cada uno de ellos.

4.8.4 Aprendizaje en la Aplicación de los SMA

Un aspecto que debe ser tratado en profundidad en ciertas propuestas que utilizan Arquitecturas Multiagente es el del aprendizaje. Con este concepto nos referimos a como los Agentes pueden mejorar sus habilidades mediante un proceso basado en el razonamiento y en la interacción con el entorno. Se debe tener que en cuenta en muchas propuestas los Agentes disponen de un conocimiento inicial muy limitado, y que es precisamente el aprendizaje la habilidad que permite a estas entidades evolucionar y mejorar su forma de solucionar los problemas. En (Wang, Y. C. & Usher, J. M. 2004) y (Wang, Y. C. & Usher, J. M. 2005) se analiza como los Agentes son penalizados o recompensados por su entorno cada vez que realizan una acción. En concreto plantea un Agente máquina que en función de las circunstancias aplica una regla de despacho entre tres posible utilizando un algoritmo de aprendizaje Q-Learning teniendo como objetivo minimizar el retraso medio en un proceso Programación de la Producción. Los autores realizan un estudio experimental. EN (Wang, Y. C. & Usher, J. M. 2007) se propone el mismo tipo de enfoque pero para abordar el problema en un taller general y con múltiples Agentes. En un entorno Multiagente (Gabel, T. & Riedmiller, M. 2006) proponen la utilización conjunto del aprendizaje reforzado junto con el razonamiento basado en casos como forma de adquirir experiencia en un entorno de Agentes colaborativos.

Otros autores han desarrollado simuladores que permiten contrastar procedimientos de aprendizaje. Con esta filosofía (Aydin, M. E. & Oztemel, E. 2000) simularon un taller general con nueve máquinas y cinco trabajos con diferentes rutas. Dentro de este sistema de simulación (SS) actúa un Agente que consta de un "*percepción module*" que recibe toda la información del sistema actualizada, un "*cognition module*" que está compuesto por un conjunto de normas de comportamiento y toma de decisiones, y un "*action module*" que es el encargado de generar mensajes y enviarlos al SS. Este Agente es el encargado de seleccionar una heurística (SPT, COVER T o Critical Ratio) para procesar los trabajos utilizando la información disponible. Además el Agente es entrenado continuamente por un método llamado Q-III, que es un sistema de aprendizaje para el Agente (basado en prueba y error).

La consideración de las capacidades de aprendizaje en el desarrollo de Sistemas Multiagente no es siempre relevante. No todas las propuestas se basan en esta propiedad. Si se ha podido constatar con el análisis de las referencias comentadas que esta característica proporciona un elevado valor añadido a los sistemas que así lo plantean. Además se ha puesto de manifiesto

que los procesos de aprendizaje han sido estudiados en profundidad en otras áreas de la ciencia y que están bastante consolidados, o al menos existe un conjunto de propuestas bastante reiteradas entre los diferentes autores.

4.8.5 Influencia de los SMA en las Soluciones

Un aspecto poco tratado en la literatura es el estudio comparativo de las diferentes arquitecturas basadas en Sistemas Multiagente con las que se afronta la resolución de los diversos problemas de Programación de la Producción. En (Cavaliere, S., Garetti, M., Macchi, M., & Taisch, M. 2000) se realiza un análisis de dichas arquitecturas realizando una agrupación de las mismas en dos clases: Las basadas en el funcionamiento del mercado (oferta-demanda), y las basadas en una regulación mediante un supervisor. Los autores comentan las características de ambos tipos de enfoque y realizan un estudio experimental para comparar las prestaciones de ambas en diferentes situaciones. El estudio se planteó en un taller general con tres tipos de situaciones; a) Sistema Estable: sistema cargado de acuerdo a un plan previsto, todos los trabajos disponibles al inicio, el periodo considerado es la semana productiva prevista; b) Sistema con Perturbaciones: idénticas condiciones que el caso "a" pero añadiendo trabajos urgentes con fechas de entrega no previstas; c) Sistema con Fallos: Se modifican los eventos del caso "b" por fallos en máquinas. El resultado indica que los sistemas basados en mercado son más robustos y se adaptan mejor a los cambios, y que en situaciones de estabilidad los sistemas basados en mediadores obtienen mejores resultados.

En (Pendharkar, P. C. 2007) se identifican los problemas básicos de negocio que influyen en el diseño de sistemas inteligentes cooperativos, y muestran de forma práctica el diseño y la aplicación de un Sistema Multiagente para la Programación de la Producción. Los experimentos sugieren el dominio, en relación con su eficacia, de los Sistemas de Multiagente donde los Agentes coordinan sus acciones.

Los autores de (Greer, K., Stewart, J. R., & McCollum, B. 2008) comparan una plataforma de Programación de la Producción genérica y centralizada (PEGS) desarrollada por los autores a partir de una propuesta anterior de la misma plataforma desarrollada con un enfoque distribuido basado en Agentes. En los experimentos realizados con diversos test publicados se muestra como el sistema distribuido se ejecuta más rápidamente (hasta diez veces menos tiempo) y ofrece soluciones tan buenas como las publicadas y mejores que en el sistema centralizado.

Con el fin de poder actuar en un entorno real (Maheswaran, R. T., Tambe, M., Bowring, E., Pearce, J. P., & Varakantham, P. 2004) proponen un marco de trabajo denominado DiMES

(Distributed Multi-Event Scheduling) que permite aplicar la técnica de optimización de restricciones distribuidas habitual en Sistemas Multiagente, y que en otros casos es difícil de aplicar por la complejidad de la realidad. Su aplicación en casos reales es posible debido a la mejora del sistema de comunicaciones y a la precomputación de casos límite.

Este apartado ha permitido tener visibilidad sobre como influye la elección de una u otra arquitectura en la calidad de las soluciones obtenidas en función del tipo de problema y su evolución. Estas consideraciones serán tenidas en cuenta en el diseño y desarrollo de la propuesta que se realizará en esta tesis.

4.9 Conclusiones

Los Sistemas Multiagente han demostrado ser útiles para resolver el Problema de la Programación de la Producción, tanto desde un punto de vista cuantitativo del problema, como por lo que se refiere a la coordinación con otras áreas del Sistema de Planificación y Control de Operaciones. Al contrario de lo que ocurre con la aplicación de otras técnicas el número de referencias que se ha podido encontrar no ha sido muy amplio, aunque sí significativo. Esto da una idea del trabajo que aún queda por hacer en este ámbito.

Se ha podido contrastar mediante el estudio de los conceptos básicos de la teoría de Agentes, su constitución en sociedades organizadas, y su forma de intercambiar conocimiento, correspondientes a los apartado 4.2, 4.3 y 4.4 respectivamente, como la herramienta que se ha considerado en este capítulo no es sencilla de aplicar. Se han puesto de manifiesto un conjunto de dificultades inherentes a la propia herramienta, lo que implica que los usuarios deben alcanzar un nivel de destreza elevado para su aplicación. Una reflexión que se deriva de lo analizado es que **la aplicación de los Sistemas Multiagente no garantiza alcanzar buenas soluciones, es simplemente un entorno de trabajo que puede ser muy favorable en algunos problemas**. De esta reflexión se extraen dos conclusiones relevantes: (1) Es necesario analizar si las condiciones que se dan en un determinado problema de Programación de la Producción son favorables para la aplicación de Sistemas Multiagente, y (2) es necesario combinar esta herramienta con otras teorías que conduzcan al sistema a un estado que permitan alcanzar una solución buena u óptima.

Los Problemas de Programación de la Producción están, en muchas ocasiones, enfocados de una forma estática y centralizada, lo que según lo visto no parece una orientación muy adecuada para la aplicación de Sistemas Multiagente. No obstante, aún con este enfoque es posible aplicar la teoría de Agentes como método de abstracción generando un modelo basado en Agentes, aunque no sea implementado posteriormente. Sin embargo, **cuando el**

problema incluye consideraciones como la Toma de Decisiones distribuida, el dinamismo del entorno, y por lo tanto la evolución del sistema, la teoría de Agentes proporciona una herramienta excelente para abordar el problema a través de los Sistemas Multiagente. Será en este último entorno donde se plantearán las propuestas de este trabajo.

Como se expondrá en capítulos posteriores esta herramienta se aplicará por partida doble para abordar dos problemas bien diferenciados aunque interconectados. Uno de los problemas que se afrontará será el establecimiento de una estrecha colaboración entre el ámbito de la Planificación y de la Programación de la Producción. Este problema incluye las características de distribución en la toma de decisiones, y como se explicará en su momento, consideraciones dinámicas del problema. La aplicación de los Sistemas Multiagentes a este tipo de colaboración ha sido analizada en los apartados 4.7 y 4.8, donde se ha constatado que desde sus inicios los expertos consideraron que la Inteligencia Artificial Distribuida aportaba un valor añadido significativo, ya que permitía abordar problemas realista difícilmente tratables con las herramientas existentes hasta ese momento.

*Por otro lado, y como ya he indicado anteriormente, los usuarios de este tipo de herramientas no deben centrar sus esfuerzos únicamente en la propia herramienta, sino en dotar a la misma de la racionalidad que le lleve a alcanzar soluciones factibles y eficaces. En el apartado 4.8 se han revisado las aportaciones más relevantes en el desarrollo de Sistemas Multiagente para el problema de la Programación de la Producción. En concreto en el apartado 4.8.1 se ha comentado como **los autores han combinado los Sistemas Multiagente o basado en Agentes con técnicas procedentes de la investigación operativa** que han guiado al sistema hasta alcanzar su objetivo global. Muchos de los enfoques se han basado en teorías de mercado que consideran que se pueden alcanzar valores óptimos en tareas como la asignación o la secuenciación, mediante un proceso negociador que converge a un valor determinado. Otras soluciones, muy pocas, han utilizado como hilo conductor del proceso de resolución del problema los ya conocidos Algoritmos Metaheurísticos.*

Se quiere destacar la mínima existencia de propuestas que combinen los Algoritmo Bio-Inspirados y los Sistemas Multiagente. En el capítulo anterior se analizó la aplicación de los Algoritmo Bio-Inspirados a los problemas de Programación e la Producción, y se detectó que estos alcanzaban muy buenas soluciones. En este capítulo se han presentado los Sistemas Multiagente como Herramientas con las que se pueden obtener excelentes resultados. Los Sistemas Multiagente, por sus características, pueden representar el rol de seres vivos (hormigas, genes, etc.) en los que se sustentan las propuestas Bio-Inspiradas. Es posible que los diversos autores hayan encontrado problemas en adaptar algunas de las características sociales a la simulación del comportamiento de los seres vivos.

Debido a la ausencia de trabajos de investigación en este tipo de enfoques combinados (SMA y Bio-Inspirados) se puede concluir aconsejando **profundizar en el problema de Programación de Producción, con metodologías Metaheurísticas bio-inspiradas basada en Sistemas Multiagente**, lo que abre una línea de trabajo en un campo muy poco estudiado. Será precisamente en esta línea en la que se planteará y resolverá el segundo de los problemas que se pretende abordar en este trabajo.

Una vez se ha explorados modelos, métodos y algoritmos aplicados a la Programación de la Producción, así como las posibilidades que el paradigma de los Sistemas Multiagente ofrecen en dicho ámbito, y habiendo confirmado la existencia de un campo de investigación en relación con el caso identificado es necesario establecer de la forma más concreta posible el problema que se propone abordar. Este es el objetivo del siguiente capítulo

4.10 Bibliografía

Agha, G. & Hewitt, C. 1987. *Object-oriented concurrent programming*. Cambridge, MA, (USA): MIT Press.

Altmann, J., Essmayr, W., Grabner, M., Gruber, F., Klug, L., Stockner, W., and Winiwater, W. *Agent Technology: State of the Art. Deliverable SCCH-TR-0049, Project EvalAgents*. 2000. Hagenberg, Austria.

Avouris, N. M. & Gasser, L. 1992. *Distributed Artificial Intelligence: Theory and Praxis*. Kluwer.

Aydin, M. E. & Fogarty, T. C. 2004. *Teams of autonomous agents for job-shop scheduling problems: An experimental study*. *Journal of Intelligent Manufacturing*, 15(4): 455-462.

Aydin, M. E. & Oztemel, E. 2000. *Dynamic job-shop scheduling using reinforcement learning agents*. *Robotics and Autonomous Systems*, 33(2-3): 169-178.

Babayan, A. & He, D. 2004. *Solving the n-job 3-stage flexible flowshop scheduling problem using an agent-based approach*. *International Journal of Production Research*, 42(4): 777-799.

Binmore, K. & Vulkan, N. 1999. *Applying to Automated Negotiation*. *NETNOMICS*, 1(1): 1-9.

Bonasso, R. P., Firby, J., Gat, E., Kortenkamp, D., Miller, D. P., & Slack M.G. 1997. *Experiences with an Architecture for Intelligent, Reactive Agents*. *Journal of Experimental & Theoretical Artificial Intelligence*, 9(2-3): 237-256.

Bond, A. H. 1988. *Readings in Distributed Artificial Intelligence*. Morgan Kaufmann.

Brazier, F., Dunin Keplicz, B, Jennings, N., and Treur, J. *Desire:Modelling multi-agent systems in a compositional formal framework*. 1997.

Brennan, R., Balasubramanian, S., and Norrie, D. H. *Dynamic Control Architecture for Advanced Manufacturing Systems. Proceedings of International Conference on Intelligent Systems for Advanced Manufacturing*, 213-223. 1997. Pittsburgh, PA.

Brooks, R. A. 1991. *Intelligence Without Representation. Artificial Intelligence*, 47(1-3): 139-159.

Brückner, S., Wyns, J., Peeters, P., and Kollingbaum, M. *Designing Agents for the Manufacturing Process Control. AAAI Press. Proceedings of Artificial Intelligence and Manufacturing Research Planning Workshop - State of the Art & State of the Practice*, 40-46. 1998. Albuquerque, Nuevo México.

Burke, P. & Prosser, P. 1994. *The Distributed Asynchronous Scheduler. In M. Zweben & M. Fox (Eds.), Intelligent Scheduling: 309-340. San Francisco, CA (USA): Morgan-Kaufman.*

Butler, J. & Ohtsubo, H. 1992. *ADDYMS: Architecture for Distributed Dynamic Manufacturing Scheduling. Menlo, Park, California (USA): AAAI Press/The MIT Press.*

Cardon, A., Galinho, T., & Vacher, J. P. 2000. *Genetic algorithms using multi-objectives in a multi-agent system. Robotics and Autonomous Systems*, 33(2-3): 179-190.

Castelfranchini, C. & Conte, R. 1996. *Distributed Artificial Intelligence and Social Science: Critical Issue. Foundations of Distributed Artificial Intelligence: John Wiley & Sons.*

Castro, J., Kolp, M., & Mylopoulos, J. 2002. *Towards requirements-driven information systems engineering: The Tropos project. Information Systems.*

Cavaliere, S., Garetti, M., Macchi, M., & Taisch, M. 2000. *An experimental benchmarking of two multi-agent architectures for production scheduling and control. Computers in Industry*, 43(2): 139-152.

Conte, R., Miceli, M., and Castelfranchini, C. *Limits and levels of cooperation: disentangling various types of prosocial interaction. Proceedings Modelling an Autonomous Agent in a Multi-Agent World*, 152-166. 1990.

Corchado, J. M. & Molina, J. M. 2002. *Introducción a la Teoría de Agentes y Sistemas Multiagente. Salamanca: Cátedra Publicaciones.*

Corkill, D. D. and Lesser, V. R. *The Use of Metalevel Control for Coordination in a Distributed Problem-solving Network. Proceedings of the Eighth International Joint Conference on Artificial Intelligence (IJCAI-83)*, 767-770. 1983.

Cowling, P. I., Ouelhadj, D., & Petrovic, S. 2004. *Dynamic scheduling of steel casting and milling using multi-agents. Production Planning & Control*, 15(2): 178-188.

Csaji, B. C., Kadar, B., & Monostori, L. 2003. *Improving multi-agent based scheduling by neurodynamic programming. Holonic and Multi-Agent Systems for Manufacturing*, 2744: 110-123.

Csaji, B. C. & Monostori, L. 2005. *Stochastic reactive production scheduling by multi-agent based asynchronous approximate dynamic programming*. *Multi-Agent Systems and Applications Iv, Proceedings*, 3690: 388-397.

Cutkosky, M. R., Englemore, R. S., Fikes, R. E., Genesereth, M. R., Gruber, F., Mark, W. S., Tenenbaum, J. M., & Weber, J. C. 1997. *PACT: An Experiment in Integrating Concurrent Engineering Systems*. In M. N. Huhns & M. P. Singh (Eds.), *Readings in Agents*: 46-55. San Francisco, CA (USA): Morgan Kaufmann.

Dang, T. T. & Frankovic, B. 2002. *Agent-based scheduling in production systems*. *International Journal of Production Research*, 40(15): 3669-3679.

Dash, R. K., Parkes, D. C., & Jennings, N. R. 2003. *Computational Mechanism Design: A Call to Arms*. *IEEE Intelligent Systems*, 18(6): 40-47.

Davis, R. & Smiths, R. G. 1983. *Negotiation as a Metaphor for Distributed Problem Solving*. *Artificial Intelligence*, 20(1): 63-100.

Decker, K., Sycara, K. P., and Williamson, M. *Middle Agents for Internet*. *Proceedings of the fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97)*, 578-583. 1997.

Drogoul, A. and Ferber, J. *From Tom Thumb to the Dockers: Some experiments with foraging robots*. *Second International Conference on Simulation of Adaptive Behavior. Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, 451-459. 1992. Honolulu, HI, MIT Press/Bradford Books.

Durfee, E. H., Lesser, V. R., & Corkill, D. D. 1989. *Cooperative Distributed Problem Solving*. In A. Barr, P. R. Cohen, & E. A. Feigenbaum (Eds.), *The Handbook of AI*: Addison-Wesley.

EURESCOM. *MESSAGE: Methodology for engineering systems of software agents. Initial methodology*. *Technical Report P907-D1*. 2000.

EURESCOM. *MESSAGE: Methodology for engineering systems of software agents (Final)*. *Technical Report P907-TI1*. 2001.

Ferber, J. 1996. *Reactive Distributed Artificial Intelligence. Foundations of distributed artificial intelligence*: 287-314. New York: John Wiley & Sons, Inc.

Ferber, J. 1999. *Multi-Agent System: An Introduction to Distributed Artificial Intelligence*. Addison Wesley Longman.

Finin, T. 1994. *KQML: An Information and Knowledge Exchange Protocol*. In K. Fuchi & T. Yohoi (Eds.), *Knowledge Building and Knowledge Sharing: Ohmasha and IOS Press*.

FIPA Foundation for Intelligent Physical Agents. www.fipa.org.

Fischer, K. *The design of an intelligence manufacturing system - A Multi-Agent System Approach*. *Proceedings of the 2nd International Working Conference on Cooperating Knowledge-based Systems*, 83-99. 1994. University of Keele (UK).

Fordyce, K. & Sullivan, G. G. 1994. Logistics Management System (LMS): Integrating Decision Technologies for Dispatch Scheduling in Semiconductor Manufacturing. In M. Zweben & M. Fox (Eds.), Intelligent Scheduling: 473-516. San Francisco, CA (USA): Morgan-Kaufman.

Fulton, J. A. Technical report on the semantic unification meta-model. Standards working document ISO TC184/SC4/WG3 N103, IGES/PDES Organization, Dictionary/Methodology Committee. 1992.

Furusho, T., Nishi, T., & Konishi, M. 2008. Distributed optimization method for simultaneous production scheduling and transportation routing in semiconductor fabrication bays. International Journal of Innovative Computing Information and Control, 4(3): 559-575.

Gabel, T. & Riedmiller, M. 2006. Multi-agent case-based reasoning for cooperative reinforcement learners. Advances in Case-Based Reasoning, Proceedings, 4106: 32-46.

Gaines, B. R., Norrie, D. H., and Lapsley, A. Z. Mediator: an Intelligent Information System Supporting the Virtual Manufacturing Enterprise. Proceedings of 1995 IEEE International Conference on Systems, Man and Cybernetics. 964-969. 1995. New York (USA).

Gasser, L. 1986. The Integration of Computing and Routine Work. ACM Transactions on Office Information Systems, 4(3): 205-225.

Genesereth, M. R. and Fikes, R. Knowledge Interchange Format. Version 3.0 Reference Manual. Technical Report Logic-92-1. 1992. Computer Science Department, Stanford University.

Genesereth, M. R. & Ketchpel, S. P. 1994. Software Agents. Communications of the Acm, 37(7): 48-&.

Genesereth, M. R. and Nilsson, N. J. Logical Foundations of Artificial Intelligence. 1-1-1987. San Mateo, CA, Morgan Kaufmann Publishers.

Giret-Boggino, A. S. 2005. ANEMONA: Una Metodología Multiagente para Sistemas Holónicos de Fabricación. Universidad Politécnica de Valencia.

Goldsmith, S. Y. and Interrante, L. D. An Autonomous Manufacturing Collective for Job Shop Scheduling. Proceedings of AI & Manufacturing Research Planning Workshop , 69-74. 1998. The AAAI Press.

Gomez, J. 2002. Modelado de Sistemas Multiagente. , Departamento de Sistemas Informáticos y Programación, Universidad Complutense de Madrid.

Gou, L., Hasegawa, T., Luh, P., Tamura, S., and Oblack, J. Holonic planning and scheduling for a robotic assembly testbed. Proceedings of the Rensselaer's 4th International Conference on Computer Integrated Manufacturing and Automation Technology . 1994. New York (USA).

Greer, K., Stewart, J. R., & McCollum, B. 2008. Comparison of a centralised and distributed approach for a generic scheduling system. Journal of Intelligent Manufacturing, 19(1): 119-129.

Grosz, B. & Sidner, C. 1990. *Plans for Discourse*. In Cohen, Morgan, & Pollack (Eds.), *Intentions in Communications*: 417-444. MIT Press.

Gruber, T. R. *The Role of Common Ontology in Achieving Sharable, Reusable Knowledge Bases*. Allen, J. A., Fikes, R., and Sandewall, E. *Principles of Knowledge Representation and Reasoning. Proceedings of the Second International Conference*, 601-602. 1991. Cambridge, MA, Morgan Kaufmann.

Gruber, T. R. *Toward Principles for the Design of Ontologies Used for Knowledge Sharing*. Guarino, N. and Poli, R. *International Workshop on Formal Ontology. Formal Ontology in Conceptual Analysis and Knowledge Representation*. 1993. Padova, Italy, Kluwer Academic Publishers.

Gruver, W. A., Kotak, D. B., van Leeuwen, E. H., & Norrie, D. 2003. *Holonic manufacturing systems: Phase II. Holonic and Multi-Agent Systems for Manufacturing*, 2744: 1-14.

Gu, P., Balasubramanian, S., & Norrie, D. H. 1997. *Bidding-based process planning and scheduling in a multi-agent system*. *Computers & Industrial Engineering*, 32(2): 477-496.

Guarino, N. *Formal Ontology and Information Systems*. Guarino, N. *Proceedings of the 1st International Conference on Formal Ontologies in Information Systems, FOIS'98*, 3-15. 1998. Trento (Italy), IOS Press.

Hayes-Roth, B. 1985. *A Blackboard Architecture for Control*. *Artificial Intelligence*, 26(3): 251-321.

Hedberg, S. 1996. *Agents for sale: First wave of intelligent agents go commercial*. *Ieee Expert-Intelligent Systems & Their Applications*, 11(6): 16-19.

Hewitt, C. 1986. *Offices Are Open Systems*. *ACM Transactions on Office Information Systems*, 4(3): 271-287.

HMS HMS Requirements. <http://hms.ifw.uni-hannover.de/>: HMS Server.

Hoare, C. A. R. 1978. *Communicating sequential processes*.

Huhns, M. N. & Gasser, L. 1989. *Distributed Artificial Intelligence*. Hyperion Books.

Iglesias, C., Garijo, M., and González, J. *Survey of agent-oriented methodologies*. Müller, J., Singh, M., and Rao, A. *Proceedings of ATAL-98* 1555, 317-330. 1999. Heidelberg, Germany, Springer-Verlag.

Iglesias-Fernández, C. A. 1998. *Definición de una metodología para el desarrollo de sistemas multiagente*. Departamento de Ingeniería de Sistemas Telemáticos. Universidad Politécnica de Madrid.

Jennings, N. R., Corera, J. M., and Laresgoiti, I. *Developing Industrial Multiagent Systems*. *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, 423-430. 1995.

Jennings, N. R., Faratin, P., Lomuscio, A. R., Parsons, S., & Sierra, C. 2001. *Automated Negotiation: Prospects, Methods and Challenges*. *International Journal of Group Decision and Negotiation*, 10(2): 199-215.

Jennings, N. R., Parsons, S., Noriega, P., and Sierra, C. *On Argumentation-based Negotiation*. *Proceeding of International Workshop on Multi-agent Systems (IWMAS-98)*, 1-7. 1998. Dedham (USA).

Keskinocak, P., Wu, F., Goodwin, R., Murthy, S., Akkiraju, R., Kumaran, S., & Derebail, A. 2002. *Scheduling solutions for the paper industry*. *Operations Research*, 50(2): 249-259.

Kinny, D. and Georgeff, M. *Modelling and design of multi-agent systems*. Technical Report 59. 1996. Melbourne, Australia, Australian Artificial Intelligence Institute.

Kornfeld, W. A. & Hewitt, C. E. 1981. *The Scientific Community Metamorph*. *IEEE Transactions on Systems, Man and Cybernetics*, 11(1): 24-33.

Kutanoglu, E. & Wu, S. D. 2006. *Incentive compatible, collaborative production scheduling with simple communication among distributed agents*. *International Journal of Production Research*, 44(3): 421-446.

Kutanoglu, E. & Wu, S. D. 1999. *On combinatorial auction and Lagrangean relaxation for distributed resource scheduling*. *IEEE Transactions*, 31(9): 813-826.

Lander, S., Lesser, V. R., and Connell, M. E. *Conflict-Resolution Strategies for Cooperation Expert Agents*. *Proceedings of the International Working Conference on Cooperating Knowledge-Based Systems*, 183-200. 1991. New York, Springer-Verlag.

Lesser, V. R. 1995. *Multiagent Systems: An Emerging Subdiscipline of AI*. *ACM Computing Surveys*, 27(3): 340-342.

Levesque, H. J. 1984. *Foundations of a functional approach to knowledge Representation*. *Artificial Intelligence*, 23: 155-212.

Lewis, M. & Sycara, K. P. 1993. *Reaching informed agreement in multispecialist cooperation*. *Group Decision and Negotiation*, 2(3): 279-299.

Lim, M. K. & Zhang, D. Z. 2004. *An integrated agent-based approach for responsive control of manufacturing resources*. *Computers & Industrial Engineering*, 46(2): 221-232.

Lin, G. Y. J. & Solberg, J. J. 1992. *Integrated Shop Floor Control Using Autonomous Agents*. *IIE Transactions: Design and Manufacturing*, 24(3): 57-71.

Lind, J. 1999. *MASSIVE: Software Engineering for Multi-agent Systems*. DFKI.

Liu, J. S. & Sycara, K. P. 1997. *Coordination of multiple agents for production management*. *Annals of Operations Research*, 75: 235-289.

López-Carmona, M. A. 2006. *Estrategias de Negociación basadas en Restricciones Difusas sobre Sistemas Multiagente*. Departamento de Automática, Universidad de Alcalá.

Luck, M., McBurney, P., Shehory, O., Willmott, S., and Agentlink Community. *Agent Technology: Computing as Interaction A Roadmap for Agent-Based Computing*. 2005. AgentLink.

Lv, C. X., Sheng, Z. Q., Li, S. F., Yin, C. W., & Zhu, Y. L. 2006. A kind of adaptive negotiation mechanism for flexible job shop scheduling. *Computational Intelligence, Pt 2, Proceedings, 4114: 324-333*.

Maes, P. 1991. *Designing Autonomous Agents*. The MIT Press.

Maheswaran, R. T., Tambe, M., Bowring, E., Pearce, J. P., and Varakantham, P. *Taking DCOP to the Real World: Efficient Complete Solutions for Distributed Multi-Event Scheduling*. *Third International Joint Conference on Autonomous Agents and Multiagent Systems 1*, 310-317. 2004.

Maturana, F. P. & Norrie, D. H. 1996. Multi-agent Mediator architecture for distributed manufacturing. *Journal of Intelligent Manufacturing*, 7(4): 257-270.

McDonell, P., Smith, S., Joshi, S., & Kumara, S. R. T. 1999. A cascading auction protocol as a framework for integrating process planning and heterarchical shop floor control. *International Flexible Manufacturing Systems*, 11(37): 62.

McEleney, B., O'Hare, G. M. P., and Sampson, J. *An Agent Based System for Reducing Changeover Delays in a Job-Shop Factory Environment*. *Proceedings of PAAM'98*. 1998. London (uk).

McGuire, J. G., Kuokka, D. R., Weber, J. C., Tenenbaum, J. M., & Gruber, F. 1993. SHADE: Technology for knowledge-based collaborative engineering. *Journal of Concurrent Engineering: Research and Applications*, 1(3).

Milojicic, C., Breugst, M., Busse, I., Campbell, J., Covaci, S., Friedman, B., Kosaka, K., Lange, D., Ono, K., Oshima, M., Tham, C., Virdhagriswaran, S. V., and White, J. *MASIF: The OMG Mobile Agent System Interoperability Facility*. *Proceeding of 2nd International Workshop Mobile Agents (MA' 98)*, 50-60. 1998. Stuttgart (Germany), Springer. *Lecture Notes in Computer Science*, 1477. Rothemel, K. and Hohl, F.

Minsky, M. 1986. *The Society of Mind*. New York: Simon & Schuster Inc.

Miyashita, K. 1998. CAMPS: a constraint-based architecture for multiagent planning and scheduling. *Journal of Intelligent Manufacturing*, 9(2): 147-154.

Morik, K., Causse, K., and Boswell, R. *A Common Knowledge Representation Integrating Learning Tools*. Technical Report, GMD. 1991. Center for Artificial Intelligence, George Mason University.

Moulin, B. & Chiab-draa, B. 1996. *An Overview of Distributed Artificial Intelligence*. *Foundations of Distributed Artificial Intelligence: John Wiley & Sons*.

- Müllen, T. & Wellman, P. 1996. *Some Issues in the Design of Market-Oriented Agents*. In M. Wooldrige, J. P. Müller, & M. Tambe (Eds.), *Intelligent Agents {II} --- Agent Theories, Architectures, and Languages ({LNAI} 1037)*: 283-298. Heidelberg (Germany): Springer-Verlag.
- Muller, J. P. 1997. *The design of intelligence agents*. Berlin (Germany): Springer-Verlag.
- Muller, J. P. 1998. *Architectures and applications of intelligent agents: A survey*. *Knowledge Engineering Review*, 13(4): 353-380.
- Muscettola, N. 2008. *HSTS: Integrating Planing and Scheduling*. In M. Zweben & M. Fox (Eds.), *Intelligent Scheduling*: 169-212. San Francisco, CA (USA): Morgan-Kauffman Publisher.
- Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senator, T., & Swartout, W. R. 1991. *Enabling Technology for Knowledge Sharing*. *Ai Magazine*, 12(3): 36-56.
- Newell, A. 1982. *The knowledge level*. *Artificial Intelligence*, 18(1): 87-127.
- Ng, C. T., Cheng, T. C. E., & Yuan, J. J. 2006. *A note on the complexity of the problem of two-agent scheduling on a single machine*. *Journal of Combinatorial Optimization*, 12(4): 386-393.
- Nwana, H. S. 1995. *Software Agents: An Overview*. *Knowledge Engineering Review*, 11(2): 205-244.
- O'Hare, G. M. P. & Jennings, N. R. 1996. *Foundations of Distributed Artificial Intelligence*. John Wiley & Sons.
- Odell, J., Parunak, H., and Bauer, B. *Extending UML for agents*. *Proceedings of the Agent-Oriented Information Systems Workshop*, 3-17. 2000.
- Ouelhadj, D., Hanachi, C., and Bouzouia, B. *A Multi-agent system for dynamic scheduling and control in manufacturing cells*. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'1998)*, 1256-1262. 1996. Louvain (Belgium).
- Park, H., Tenenbaum, J. M., and Dove, R. *Agile infrastructure for manufacturing systems: a vision for transforming the US manufacturing base*. *Proceedings of Defense Manufacturing Conference*. 1993.
- Parunak, H. V. 1987. *Manufacturing Experience with the Contract Net*. *Ai Magazine*, 8(2): 93.
- Parunak, H. V. *MASCOT: A virtual factory for research and development in manufacturing scheduling and control*. *Proceedings of the 3rd Workshop on Intelligent Scheduling in Manufacturing*. 1993. Toronto University (Canada).
- Parunak, H. V., Baker, A. D., & Clark, S. J. 2001. *The AARIA agent architecture: From manufacturing requirements to agent-based system design*. *Integrated Computer-Aided Engineering*, 8(1): 45-58.

Pechoucek, M., Marik, V., and Stepankova, O. *Coalition formation in manufacturing multi-agent systems. Proceedings of 11th International Workshop on Database and Expert Systems Applications.* 241-246. 2000. London (UK).

Pendharkar, P. C. 2007. *The theory and experiments of designing cooperative intelligent systems. Decision Support Systems, 43(3): 1014-1030.*

Peng, Y., Finin, T., Labrou, Y., Chu, B., Long, J., Tolone, W. J., and Boughannam, A. *A Multi-Agent system for Enterprise Integration. Proceedings of the 3rd International Conference on the Practical Applications of Agents and Multi-Agent Systems (PAAM-98).* 155-169. 1998. London (UK).

Rao, A. S. and Georgeff, M. P. *Deliberation and its role in the formation of intentions. Proceedings of the seventh conference (1991) on Uncertainty in artificial intelligence , 300-307.* 1994. Los Angeles, California, United States, Morgan Kaufmann Publishers Inc. 1991.

Russell, S. & Norvig, P. 2002. *Artificial Intelligence: A Modern Approach.* Prentice-Hall.

Sabah, G. *CAMEL: A Computational Model of Natural Language Understanding using Parallel Implementation. Proceedings of Ninth European Conference on Artificial Intelligence (ECAI'90) .* 1990. Stockholm (Sweden).

Sadeh, N. and Fox, M. S. *CORTES: An Exploration into Micro-Opportunistic Job-Shop Scheduling. Proceedings of Workshop on Manufacturing Production Scheduling.IJCAI89 .* 1989.

Sandholm, T. *An Implementation of Contract Net Protocol Based on Marginal Cost Calculations. Proceedings of the Eleventh National Conference on Artificial Intelligence , 256-262.* 1993. Melon Park, California (USA).

Sandholm, T. and Lesser, V. R. *Issues in Automated Negotiation and Electronic Commerce:Extending the Contract Net Protocol.* AAAI Press. *Proceedings of the Second International Conference on Multiagent Systems , 328-335.* 1995. Melon Park, California (USA).

Schelling, T. C. 1969. *The Strategy of Conflict.* Cambridge, MA: Harvard University Press.

Schreiber, A., Akkermans, J, and Anjewierden, A. *Engineering of Knowledge and Management: The COMMONKADS Methodology.* 2000. MIT Press.

Shaw, M. J. P. 1987. *Distributed Planning in Cellular Flexible Manufacturing Systems. INFOR, 25(1): 13-25.*

Shen, W. & Norrie, D. H. 1999. *Agent-based systems for intelligent manufacturing: A state-of-the-art survey. International Journal of Knowledge and Information Systems, 1(2): 129-156.*

Shen, W., Xue, D., and Norrie, D. H. *An Agent-Based Manufacturing Enterprise Infrastructure for Distributed Integrated Intelligent Manufacturing Systems. Nwana, H. S. and Ndumu, D. T. Proceedings of the 3rd International Conference on the Practical Applications of Agents and Multi-Agent Systems (PAAM-98) , 533-548.* 1998. London, UK.

Shen, W. M. & Barthes, J. P. A. 1996. *An experimental multi-agent environment for engineering design. International Journal of Cooperative Information Systems*, 5(2-3): 131-151.

Shen, W. M., Wang, L. H., & Hao, Q. 2006. *Agent-based distributed manufacturing process planning and scheduling: A state-of-the-art survey. Ieee Transactions on Systems Man and Cybernetics Part C-Applications and Reviews*, 36(4): 563-577.

Shin, M. & Jung, M. 2005. *Bid generation and evaluation for MANPro-based real time scheduling. International Journal of Production Research*, 43(18): 3821-3836.

Shoham, Y. 1993. *Agent-Oriented Programming. Artificial Intelligence*, 60(1): 51-92.

Smith, R. G. *A frame work for Distributed Problem-Solving. Proceedings of International Joint Conferences on Artificial Intelligence (IJCAI'79) . 1979. Tokio (Japan).*

Smith, R. G. 1980. *The Contrat Net Protocol: High-level Communication and Control in a Distributed Problem Solver. IEEE Trans.On Computers*, 29(12): 1104-1113.

Soewandi, H. & Elmaghraby, S. E. 2001. *Sequencing three-stage flexible flowshops with identical machines to minimize makespan. Iie Transactions*, 33(11): 985-993.

Sousa, P. & Ramos, C. 1999. *A distributed architecture and negotiation protocol for scheduling in manufacturing systems. Computers in Industry*, 38(2): 103-113.

Steve, G., Gangemi, A., & Pisanelli, D. M. 1997. *Integrating Medical Terminologies with ONIONS Methodology. In H. Kangassalo & J. P. Charrel (Eds.), Information Modelling and Knowledge Bases VIII: Amsterdan (Netherland): IOS-Press.*

Sun, J. & Xue, D. 2001. *A dynamic reactive scheduling mechanism for responding to changes of production orders and manufacturing resources. Computers in Industry*, 46(2): 189-207.

Sycara, K. P. *Using Option Pricing to Value Commitment Flexibility in Multiagent Systems. Technical Report CMU-CS-Tr-97-169. 1997. School of Computer Science, Carmegie Mellon University.*

Sycara, K. P. 1998. *Multiagent systems. Ai Magazine*, 19(2): 79-92.

Travers, M. 1988. *Animal Construction Kit. Artificial Life*, 421-442.

Usher, J. M. 2003. *Negotiation-based routing in job shops via collaborative agents. Journal of Intelligent Manufacturing*, 14(5): 485-499.

Valckenaers, P. & Van Brussel, H. 2003. *Deadlock avoidance in flexible flow shops with loops. Journal of Intelligent Manufacturing*, 14(1): 137-144.

van Leeuwen, E. H. & Norrie, D. H. 1997. *Intelligent manufacturing: holons and holarchies. Manufacturing Engineer*, 76(2): 86-88.

Vancza, J. & Markus, A. 2000. *An agent model for incentive-based production scheduling*. *Computers in Industry*, 43(2): 173-187.

Vrba, P. *JAVA-Based Agent Platform Evaluation*. Marik V, McFarlane, D., and Valckenaers, P. *Proceeding of First International Conference on Industrial Applications of Holonic and Multi-Agent Systems (HoloMAS'03)* 2744, *Lecture Notes in Artificial Intelligence*, 47-58. 2003. Prague, Czech Republic., Springer.

Wang, L., Balasubramanian, S., and Norrie, D. H. *Agent-based Intelligent Control System Design for Real-time Distributed Manufacturing Environments*. In *Working Notes of the Agent-Based Manufacturing Workshop*, 152-159. 1998. Minneapolis, MN.

Wang, Y. C. & Usher, J. M. 2004. *Learning policies for single machine job dispatching*. *Robotics and Computer-Integrated Manufacturing*, 20(6): 553-562.

Wang, Y. C. & Usher, J. M. 2005. *Application of reinforcement learning for agent-based production scheduling*. *Engineering Applications of Artificial Intelligence*, 18(1): 73-82.

Wang, Y. C. & Usher, J. M. 2007. *A reinforcement learning approach for developing routing policies in multi-agent production scheduling*. *International Journal of Advanced Manufacturing Technology*, 33(3-4): 323-333.

Wang, Y. H., Yin, C. W., & Zhang, Y. 2003. *A multi-agent and distributed ruler based approach to production scheduling of agile manufacturing systems*. *International Journal of Computer Integrated Manufacturing*, 16(2): 81-92.

Werner, E. 1989. *Cooperating Agents: A Unified theory of communication and social structure*. *Distributed Artificial Intelligence*: 3-36. San Mateo, CA (USA): Morgan-Kaufman.

Wong, T. N., Leung, C. W., Mak, K. L., & Fung, R. Y. K. 2006. *Dynamic shopfloor scheduling in multi-agent manufacturing systems*. *Expert Systems With Applications*, 31(3): 486-494.

Wood, M. F. 2000. *Multiagent Systems Engineering: A Methodology for Analysis and Design of Multiagent Systems*. Air Force Institute of Technology.

Wooldridge, M., Jennings, N. R., & Kinny, D. 1998. *A Methodology for Agent-Oriented Analysis and Design*. Seattle, WA.

Wooldridge, M. & Ciancarini, P. 2001. *Agent-Oriented Software Engineering: The State of the Art*. *Agent-Oriented Software Engineering*: Springer-Verlag.

Wooldridge, M. & Jennings, N. R. 1995. *Intelligent Agents: Theory and Practice*. *Knowledge Engineering Review*, 10(2): 115-152.

Xue, D., Sun, J., & Norrie, D. H. 2001. *An intelligent optimal production scheduling approach using constraint-based search and agent-based collaboration*. *Computers in Industry*, 46(2): 209-231.

Yu, E. 1996. *Modelling Strategic Relationships for Process Reengineering.* , Department of Computer Science. University of Toronto.

Yu, X. F. & Ram, B. 2006. *Bio-inspired scheduling for dynamic job shops with flexible routing and sequence-dependent setups.* *International Journal of Production Research*, 44(22): 4793-4813.

Zhong, W. C., Liu, J., & Jiao, L. C. 2005. *Job-shop scheduling based on Multiagent Evolutionary Algorithm.* *Advances in Natural Computation, Pt 3, Proceedings*, 3612: 925-933.

Zhong, W. C., Liu, J., Xue, M. Z., & Jiao, L. C. 2004. *A multiagent genetic algorithm for global numerical optimization.* *Ieee Transactions on Systems Man and Cybernetics Part B-Cybernetics*, 34(2): 1128-1141.

Zlotkin, G. and Rosenschein, J. S. *Coalition, Cryptography, and Stability: Mechanisms for Coalition Formation in Task Oriented Domains.* *National Conference on Artificial Intelligence. Proceeding of the National Conference on Artificial Intelligence* , 432-437. 1994.

**Descripción y Modelado Matemático
del Problema**

Capítulo

5

Tabla de contenido

5.	Capítulo 5	285
5.1	Introducción	285
5.2	Descripción del Problema de Programación de la Producción Identificado.....	286
5.2.1	Condiciones que Afectan a la Gestión del Programa Productivo.....	287
5.2.1.1	Visión General del Dominio de Actuación de Programador de la Producción 287	
5.2.1.2	La Incertidumbre y su Impacto en el Programa Productivo.....	292
5.2.1.2.1.	La Incertidumbre en los Tiempos de Ejecución de algunas Operaciones.	292
5.2.1.2.2.	Efectos de la Incertidumbre en el Programa en Curso	294
5.2.1.2.3.	Evaluación de la Gestión del Programa de Producción	296
5.2.1.3	El Programa Productivo: La Programación Predictiva y Reactiva	296
5.2.2	Condiciones que Afectan al Cálculo del Programa Productivo	297
5.2.2.1	Consideraciones sobre la Función objetivo.....	298
5.2.2.2	Proceso Productivo en Relación con la Programación Predictiva.....	299
5.2.2.3	Hipótesis del Proceso Productivo en Relación con la Programación Reactiva 303	
5.3	Estructura del Problema Identificado	305
5.4	Formulación del Modelo Matemático para el Cálculo de Programa Productivo en un Taller de Flujo Híbrido	307
5.4.1	Modelo Predictivo Determinista	307
5.4.2	Comentarios al Modelo Predictivo Determinista.....	311
5.4.3	Modelo Determinista para la Reprogramación o Reactivo.....	314
5.4.4	Comentarios al Modelo Reactivo Determinista	321
5.4.5	Modelo con Incertidumbre	322

5.4.5.1	Modelo Predictivo Fuzzy	323
5.4.5.2	Modelo Reactivo Fuzzy.....	325
5.4.6	Comentarios al Modelo con Incertidumbre	326
5.4.7	Consideraciones sobre la Medición de la Gestión del Programa.....	329
5.4.8	Utilidad de los Modelos Matemáticos	331
5.5	Conclusiones	331
5.6	Bibliografía	333

Índice de figuras

Figura 5-1. Despliegue del Plan Maestro en Trabajos (elaboración propia).....	290
Figura 5-2. Proceso de Gestión de un Programa Productivo en un periodo (elaboración propia).	291
Figura 5-3. Taller cerámico de 3 etapas (elaboración propia).	302
Figura 5-4. Estructura de Problema Identificado (elaboración propia).	305
Figura 5-5. Relaciones entre variables para la determinación del instante de inicio del ajuste en una etapa considerando la posibilidad de solapamientos adaptado de(Andrés, C. 2001).	313
Figura 5-6. Función de pertenencia triangular de una variable x difusa (basado en (Gen, M., Tsujimura, Y., & Ida, K. 1992)).	325

5.1 Introducción

El sector cerámico es probablemente uno de los sectores productivos más destacados en España, y fundamentalmente en la Comunidad Valenciana. Probablemente debido a su alta concentración, ha sido capaz de generar una dinámica de competencia que ha proporcionado una mejora continua del sector. Este progreso se refleja finalmente en un aumento del servicio, en una gran variedad de productos y en una disminución de costes de producción.

Uno de las dificultades operativas principales de este tipo de industria es la Programación de la Producción. Tradicionalmente, este problema ha sido abordado tratando de simplificar al máximo las condiciones reales. Sin embargo estas simplificaciones van en contra de las tendencias en el sector cerámico, que es de hecho muy dinámico y asume con cierta rapidez las condiciones variables y los cambios constantes del entorno. Por lo que actualmente el sector demanda soluciones que converjan respecto a la calidad de sus soluciones y a la aplicabilidad de las mismas. La reorientación del tratamiento tradicional de Programación de la Producción practicado en el sector cerámico durante años implica no sólo plantear nuevas técnicas, sino ofrecer nuevos enfoques.

En este capítulo se realizará **una descripción del problema que se desea abordar**, y en los siguientes capítulos se ofrecerán diferentes alternativas que intentarán dar cobertura a las necesidades expuestas. Y aunque a lo largo del mismo se irá explicando su naturaleza, y sus detalles, para facilitar su comprensión de éste se debe de subrayar que **aunque el problema de la Programación de la producción es operativo, la forma de abordarlo tiene componentes de gestión y componentes de cálculo**. Hecha esta división se puede considerar que el cálculo del Programa es un aspecto puramente cuantitativo condicionado, restringido, por los aspectos de gestión. Por este motivo, a lo largo de este capítulo, y se puede avanzar que será así en el resto, el lector debe ser consciente de que se desarrollara un discurso bien ubicado en un nivel, o bien ubicado en el otro.

Los aspectos relacionados con la **Gestión del Programa**, es decir el primer nivel, se analizan mediante una revisión básicamente literaria del problema de Programación de la Producción desde el punto de vista de la **Toma de Decisiones** que conectan con la gestión de la empresa, fundamentalmente en base a su relación con la **Planificación de la Producción**. El análisis del segundo nivel, se aborda mediante un análisis preciso, empleando modelos matemáticos, fundamentalmente de los **procesos que de cálculo** de los Programas Productivos, que forman parte del núcleo operativo que se ejecutará condicionado por el tipo de gestión establecida en

el primer nivel. Los **Modelos Matemáticos permitirán realizar una definición precisa del aspecto cuantitativo** que se desea tratar, para facilitar el proceso de entendimiento del proceso de cálculo de los Programas Productivos, sin vincularlos necesariamente con el la manera de resolverlos.

El resto del capítulo está estructurado como sigue: En el apartado 5.2 se realiza una descripción completa del problema de Programación de la Producción en el entorno del Taller de Flujo Híbrido con Tiempos de Cambio de Partida dependientes de la Secuencia. En este apartado se tratarán los problemas de gestión, y de cálculo de forma diferenciada. En el apartado 5.3 se reflexionará brevemente sobre la estructura del problema identificado. En el apartado 5.4 se realizará el modelado Matemático del problema de cálculo del Programa como medio de expresión formal del mismo, tanto con un enfoque Determinista como Fuzzy. Finalmente, en el apartado 5.5 se presentan las conclusiones.

5.2 Descripción del Problema de Programación de la Producción Identificado

El problema que se desea afrontar se identifica como un caso de Problema de Programación de la Producción de un Taller de Flujo Híbrido con Tiempos de Cambio de Partida dependientes de la Secuencia (Andrés, C. 2001), en concreto dentro de la Industria de Fabricación de Pavimento y Revestimiento Cerámico.

En el ámbito de la Programación de la Producción tradicional se suele plantear un modelo determinista a partir del cual, mediante procedimiento optimizadores o heurísticos, se obtiene como resultado un Programa Productivo. En el caso que se plantea en este trabajo se aborda el problema más allá del establecimiento o cálculo propiamente dicho de un Programa, y de su correspondiente secuencia, para un periodo determinado en base a unos datos conocidos a priori. Se pretende abarcar el conjunto de decisiones relativas a la Programación de la Producción durante todo el periodo considerado teniendo en cuenta las posibles desviaciones respecto de las *Previsiones* iniciales, y por lo tanto los posibles reajustes.

En este apartado se van a describir de forma literaria todos los aspectos que ayudarán a la especificación preliminar del caso al cual se dará cobertura. En el apartado 5.2.1 se realiza una aproximación del entorno en el cual debe trabajar el programador los procesos que se consideran y la relación entre ellos. Dentro del este apartado se revisan algunos aspectos relacionados con la incertidumbre, con el objeto de justificar su influencia y determinar en que medida puede afectar, así como aspectos que ayuden a diferenciar entre la acción de establecer un Programa inicial y la de modificarlo cuando éste se encuentra en plena

ejecución. En segundo, y último lugar, en apartado 5.2.2 se detallarán los aspectos ligados a las consideraciones que desde el cálculo de un Programa Productivo inicial o modificado se deben tener en cuenta respecto del Proceso Productivo.

5.2.1 Condiciones que Afectan a la Gestión del Programa Productivo

Una forma de introducir el problema que se desea resolver es iniciar su descripción “de arriba a abajo” o de más general a más detalle. En la actividad relacionada con la Programación de la Producción, el plano de lo más general se encuentra en la frontera con otros dominios de actuación que acaban por condicionar de forma determinante la forma de calcular el Programa Productivo. En este caso se parte de la relación entre Programación y Planificación de la Producción, y poco a poco, se irán determinando ciertas condiciones que posteriormente deben ser tenidas en cuenta.

Al final de este apartado se debería detectar la existencia de dos tipos de problemas diferenciados. Un tipo de problema es el tradicional problema de Secuenciación/Programación de la Producción, que ciertamente está restringido en parte por aspectos relacionados con la gestión que se realiza del proceso de establecimiento y/o modificación del Programa. El otro problema, que es el que se refiere a la forma de conectar las decisiones de establecimiento y modificación del Programa Productivo, y como actuar en cada caso. En definitiva, algunos de los aspectos que se tratan en este apartado se incluyen como restricciones del problema de Secuenciación/Programación, y otros aspectos forman parte de las restricciones que se deberán tener en cuenta en la Gestión del Programa Productivo.

5.2.1.1 Visión General del Dominio de Actuación de Programador de la Producción

Dentro del Sistema de Planificación y Control de Operaciones (SPCO)ⁱ se diferencian claramente las funciones del Planificador y Programador. A lo largo de este capítulo se insiste en la diferenciación funcional de dichos roles, pero se refuerza la relación que ambos deben tener para cumplir adecuadamente las actividades que le son propias, y alcanzar los objetivos asignados al SPCO.

ⁱ El Sistema de Planificación y Control de Operaciones (SPCO) ha sido analizado en el capítulo tres de este documento tanto desde un punto de vista genérico como particularizado al caso de la Industria Cerámica. Los elementos que componen el SPCO, así como la relación entre ellos, han sido definidos en dicho capítulo.

Se ha identificado que en muchas ocasiones la relación entre planificador y programador es escasa, y mal definida. El único vínculo formal y robusto que existe entre ambos es la comunicación unidireccional que tiene como origen al planificadorⁱⁱ y destino al programador, que no cuenta con retroalimentación, o ésta es débil o está poco formalizada. A pesar de ello, parece lógico que debe existir una comunicación formal en ambos sentidos. Así no sólo sería el programador quien recibiese información relativa a los trabajos a realizar procedente del planificador, sino que el planificador podría incorporar en su proceso de toma de decisiones, de forma perfectamente definida, en tiempo y forma, datos relativos a cumplimiento del Plan Maestro de Producción en función de la evolución del Programa de Producción elaborado y gestionado por el programador.

Debido a las pocas evidencias de comunicación bidireccional, fluida y formalizada entre planificador y programador que se han detectado en el ámbito de aplicación analizado, y a las características del Proceso Productivo, **se considera que en las Empresas del Sector Cerámico la potenciación del trabajo Colaborativo entre Planificadores y Programadores pueden suponer una mejora importante en el rendimiento del Sistema Productivo.**

A lo largo de este capítulo se incide sobre aspectos relacionados con el proceso de obtención de los trabajos que componen una Orden de Fabricación a partir del despliegue del Plan Maestro de Producción. En este sentido hay que recordar, que según se comentó en el capítulo 3, una vez elaborado el Plan Maestro de Producción, se calculan los Requerimientos de Materiales o MRP (Material Requirement Planning), del que se obtienen dos resultados. Por un lado, el conjunto de trabajos a fabricar, y por otro las necesidades de materias primas que son imprescindibles comprar. El primer resultado se ha identificado con el nombre de Lista de Órdenes de Fabricación y el segundo con el de Lista de Compras. Cada actividad (compras, chequeos de materias primas, producción de piezas, etc.) que se deriva del MRP tiene asignada una fecha máxima de finalización que garantiza que el producto que ha originado dichas actividades podrá estar disponible en la fecha que se marcó en el Plan Maestro. Dado que el MRP no tiene en cuenta la capacidad se pueden dar situaciones de no factibilidad de la propuesta resultante por falta de recursos, por eso se ha denominado "Lista" y no "Plan" a las propuestas. Por este motivo, sobre la Lista de Órdenes se suele aplicar un análisis de los Requerimientos de Capacidad o CRP (Capacity Requirement Planning) que puede implicar un alisamiento de las cargas, y en general una redistribución de las actividades, para garantizar la disponibilidad de recursos suficientes para realizar las operaciones. Como resultado del CRP se obtiene un Plan de Fabricación y un Plan de Compras. Ambos elementos son resultado de

ⁱⁱ En muchas ocasiones el planificador es un rol con una fuerte componente de gestión y el programador es un rol con un componente altamente tecnológico.

refinar las Listas de Órdenes de Fabricación y de Compras para ajustarlas a los recursos existentes. En general, este proceso implica que un lote del Plan Maestro puede suponer múltiples trabajos en el Plan de Fabricación, por ejemplo un trabajo para cada pieza o componente, que finalmente se utilizan en un trabajo de montaje o elaboración final que da lugar al producto comercial.

En la Industria Cerámica, y en concreto en el caso que se está abordando, al no disponer de piezas o componentes, existe una relación relativamente simple entre lotes y trabajos. En la Figura 5-1 se puede ver un ejemplo de despliegue del Plan Maestro en trabajos en una empresa de fabricación de baldosas cerámicas. Obviamente, debido al proceso seguido, las fechas de disponibilidad de los lotes del Plan Maestro serán siempre posteriores a las fechas de finalización de la última operación del correspondiente trabajo. Así en el ejemplo de la Figura 5-1 se puede observar como el “Lote 102/42 Singapur 30X30 Azul”, ubicado en el periodo del 20/04/2009 al 26/04/2009, se identifica como trabajo de la Lista de Fabricación en el periodo comprendido entre el 13/04/2009 al 19/04/2009, una vez calculado el MRP. Para enriquecer más el ejemplo se ha supuesto que se necesitan siete días adicionales, una vez terminada la fabricación, antes de disponer del producto. También se puede observar como en dicho trabajo está previsto fabricar 2.000 m² frente a los 3.000 m² del lote, ya que se ha supuesto que se dispone en almacén de 1.000 m². También se puede ver, como una vez calculado el CRP, dicho trabajo se ha adelantado un periodo por falta de recursos en el periodo previsto inicialmente por el MRP.

El Plan de Fabricación contendría la información básica que el programador recibiría para iniciar su actividad, es decir la elaboración del Programa de Producción.

Analizando más detalladamente la relación planificador-programador, considerando siempre las particularidades expuestas, se puede ver como en el SPCO se establece que la elaboración de un Programa de Producción es el último paso antes de la ejecución en planta de las órdenes de producción. La Programación de la Producción está fuertemente condicionada por el Plan Maestro en el cual se configuran los lotes, identificados básicamente por el tipo de artículo, cantidad a fabricar y fecha de finalización. Por este motivo, el programador, además de las restricciones tecnológicas propias de su ámbito, está condicionado por el Plan Maestro de Producción establecido por el planificador ya que condiciona el Plan de Fabricación, información que es utilizada posteriormente por el programador.

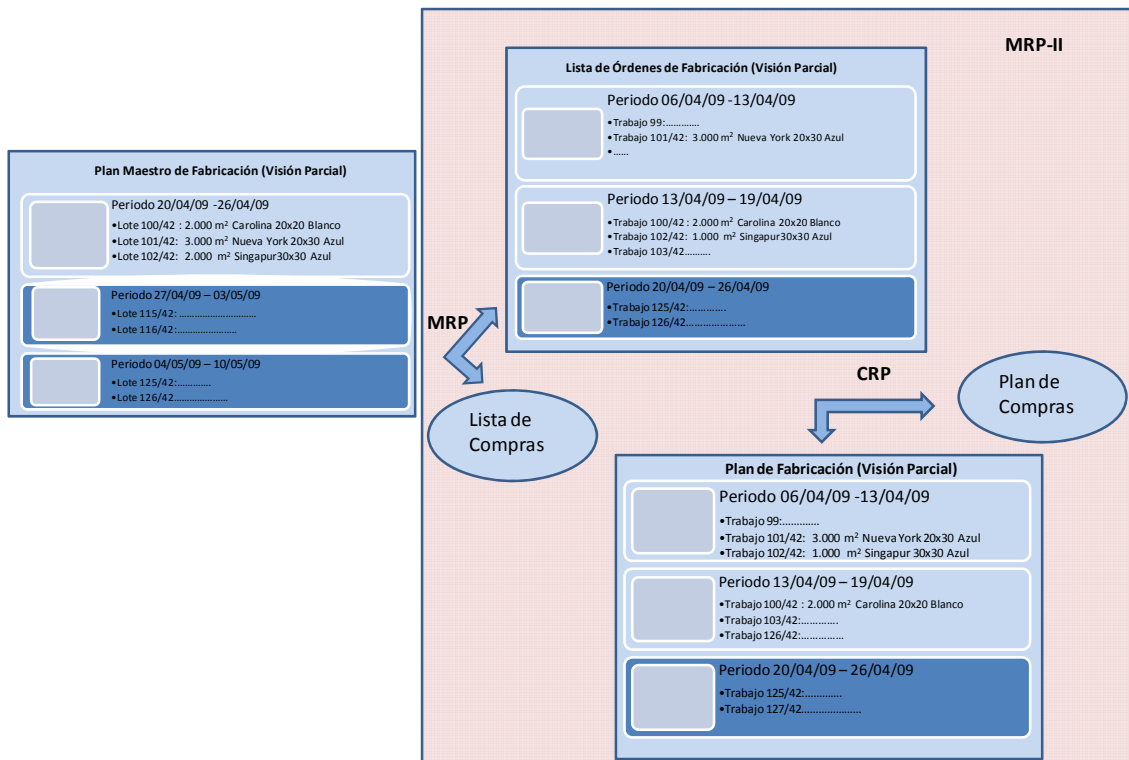


Figura 5-1. Despliegue del Plan Maestro en Trabajos (elaboración propia).

En el caso que se analiza, el objetivo global que se persigue en el Proceso Productivo es alcanzar la mayor *tasa de productividad promedio posible*. Por lo que respecta al planificador, este objetivo se va a reflejar fundamentalmente durante la elaboración del Plan Maestro. Las restricciones del planificador son básicamente la capacidad, la materia prima y la necesidad de satisfacer la demanda en las fechas acordadas en los compromisos comerciales adquiridos.

Fruto de la alineación de los objetivos globales del proceso productivo, el programador se verá forzado a usar adecuadamente los recursos para hacer frente a una propuesta de Plan Maestro que persiga cotas de producción elevadas. Pero no hay que perder de vista que en el ámbito de la Planificación el análisis de capacidades se realiza de forma poco precisaⁱⁱⁱ, y que el planificador se basa en estimaciones que normalmente tenderán a ser conservadoras para evitar que se produzcan Programas no factibles. Es necesario anticipar que la existencia de Tiempos de Cambio de Partida dependientes de la Secuencia o de Tiempos de Proceso

ⁱⁱⁱ Hay que recordar que el análisis de Capacidades que se realiza sobre el Plan Maestro se denomina aproximada, precisamente porque no es posible realizar un análisis exhaustivo.

incierto complica mucho una predicción sobre la Ocupación del taller^{iv}. Será durante el proceso de elaboración del Programa de Producción o incluso durante la ejecución del mismo cuando se pueda reducir la imprecisión y ajustar las expectativas del planificador (Plan Maestro y Plan de Fabricación) en base al Programa Productivo propuesto.

En el caso que se plantea, el hilo conductor de la actividad del programador es el proceso de Gestión del Programa Productivo (nivel jerárquico superior), que se representa en la Figura 5-2. Como se puede ver en la figura la **Gestión del Programa consiste en Establecer un Programa Productivo, y a partir de éste, Supervisar continuamente su cumplimiento (nivel jerárquico inferior), y en todo caso Modificar el Programa Establecido inicialmente si se considera necesario**. Este proceso se realizará tantas veces como sea necesario hasta el final de la ejecución del Programa Productivo. Este proceso es cíclico, comienza de nuevo al inicio de cada periodo, y concluye al final del mismo.

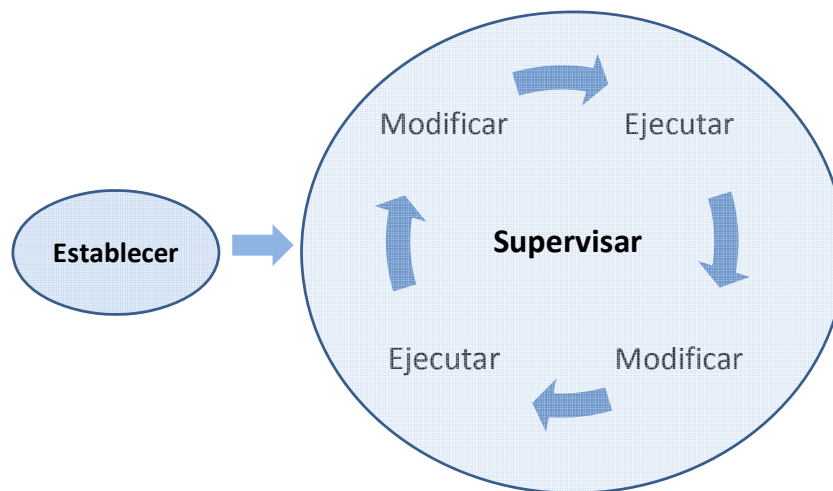


Figura 5-2. Proceso de Gestión de un Programa Productivo en un periodo (elaboración propia).

En este trabajo se habla de *Gestión del Programa Productivo*, ya que en el problema que se aborda se dan las circunstancias suficientes como para considerar necesario ir más allá de la idea tradicional consistente en focalizar el esfuerzo del programador, casi de forma exclusiva, en la elaboración o cálculo de un único Programa de Producción inicial. Programa que se supone válido para todo el periodo considerado. La necesidad de tener en cuenta el

^{iv} En el marco de este trabajo se considera que es planificador que asigna las Órdenes de Fabricación a las diferentes plantas de que puede constar un determinado Sistema Productivo. El trabajo del programador se desarrolla en el ámbito de una planta o taller.

establecimiento de un Programa Productivo, pero también su supervisión y posible modificación viene dada, en buena medida, debido a que se asume una cierta incertidumbre. Ésta está asociada de forma directa o indirecta a los Tiempos de Cambio de Partida dependientes de la secuencia y a Tiempos de Proceso que pueden ser variables. En todo caso, es necesario subrayar que cuando se habla de “supervisar” se hace referencia a un conjunto de actividades propias de lo que tradicionalmente se entiende por “control”. No obstante, dichas actividades de supervisión o control no constituyen un foco de investigación fundamental en este trabajo, y serán empleadas como elementos de necesarios para el proceso de modificación que se plantea como consecuencia de las mismas.

5.2.1.2 La Incertidumbre y su Impacto en el Programa Productivo

En este apartado se subrayará cual es el tipo de Incertidumbre que tiene mayor impacto sobre el sistema productivo que se está analizando, y cual es su origen. Con ello se quiere clarificar cual es el alcance de este factor y especificar en que parte del proceso se produce. La Incertidumbre es el elemento que puede impedir el cumplimiento del Programa Productivo en curso, y que por lo tanto, justifica la necesidad de una **gestión** (ver Figura 5-2) del mismo durante todo el periodo.

Otro resultado de este apartado es la obtención de una clasificación de los diversos impactos que la incertidumbre puede producir en un Programa Productivo. De esta forma se podrán identificar los tipos de desviaciones de un Programa Productivo más allá de su origen.

Finalmente, se diferenciará entre la evaluación de cada uno de los Programas que se generan a lo largo de un periodo y el proceso de *Gestión de la Programación*.

5.2.1.2.1. La Incertidumbre en los Tiempos de Ejecución de algunas Operaciones

En este punto se va a poner de manifiesto la necesidad de considerar la Incertidumbre en los Tiempos de Proceso y de Cambio de Partida.

Por un lado, hay que considerar que la Fabricación de las Baldosas Cerámicas es un proceso complejo que tiene como característica básica la transformación, fundamentalmente mediante la manipulación física, termodinámica y química de las materias primas. Éste es especialmente sensible el proceso de tratamiento químico, ya que es muy dependiente de las condiciones que se propician en cada momento, tanto de los materiales como del entorno. Por este

motivo, es difícil poder repetir los resultados obtenidos en dos lotes diferentes del mismo producto. Esto hace que en muchas ocasiones las operaciones deban ir ajustándose sobre la marcha lo que implica la *existencia de tiempos de proceso similares, pero no iguales, para la misma operación y el mismo producto*.

Así mismo, hay que tener en cuenta las condiciones de cada planta o taller. En la actualidad las fábricas de baldosas cerámicas están altamente automatizadas. No obstante, en los talleres de fabricación suelen existir recursos con la misma funcionalidad (máquinas, utillaje, etc.) que por diversos motivos tienen prestaciones diferentes. Estas diferencias se traducen en la existencia de velocidades diferentes para la realización del mismo tipo de operación. Se puede considerar, ya que en este trabajo no se abordan cuestiones relativas al control, que dentro de las condiciones del taller que afectan a los tiempos de proceso puede estar la no disponibilidad esporádica de algún recurso (fallos, averías, etc.). En definitiva, independientemente de la causa, los Tiempos de Proceso pueden ser variables.

Según lo expuesto anteriormente ***parece razonable considerar los Tiempos de Proceso como valores con un cierto grado de Incertidumbre***.

Por otro lado, se debe tener en cuenta que los *Tiempos de Cambio de Partida afectan a todas las etapas del proceso productivo y dependen de la secuencia*. El hecho de que el tiempo asociado al proceso de ajuste del sistema productivo entre un lote y el siguiente dependa de la secuencia implica que:

- El mismo lote puede necesitar tiempos diferentes para realizar dicha operación en función del lote que se realice con anterioridad a éste.
- El tiempo estimado para realizar dicha operación no será conocido hasta que se establezca el Programa y sea conocida la secuencia. Por lo tanto el planificador no conoce este dato en el momento de realizar el Plan Maestro.

Además, el grado de impacto no es el mismo en todas las operaciones, pudiendo invertirse horas o minutos dependiendo de la tarea a realizar. A groso modo, el Tiempo de Cambio puede rondar entre la hora y las ocho horas, y afecta a casi todas las etapas productivas aunque en diversa medida.

Como ya se comentó en el capítulo 2, los cambios de partida condicionan diversos aspectos fabriles (costes, productividad, flexibilidad de operaciones, el tamaño de lote, la capacidad de los almacenes, etc.). El mayor o menor tiempo empleado en la realización de los Cambios de Partida condiciona la Programación de la Producción, puesto que en función de lo que se esté fabricando y de lo que queda por fabricar, se incurrirán en unos Tiempos de Ajuste más o

menos elevados en función de la decisión que se tome y, por lo tanto la fecha de finalización de los trabajos se verá afectada.

El tiempo empleado en la realización de cada tipo de Cambio de Partida es sólo una estimación. Este puede aumentar o disminuir en función de la experiencia y composición del equipo humano, y de las dificultades técnicas de cada momento. En este sentido, ***parece razonable considerar los Tiempos de Cambio de Partida como valores con un cierto grado de Incertidumbre.***

5.2.1.2.2. Efectos de la Incertidumbre en el Programa en Curso

En el punto anterior se han expuesto una serie de situaciones sujetas a Incertidumbre. En este punto se va a analizar como influyen las perturbaciones que causa la Incertidumbre en el Programa Productivo en curso. Se va realizar una clasificación de la casuística que se puede producir.

En el ámbito del seguimiento del Programa es imprescindible conocer que supone una perturbación para poder responder a las dos cuestiones básicas asociadas a la Modificación del Programa Productivo, conocido tradicionalmente como Reprogramación: ¿Cuándo se realiza? ¿Qué se hace?

En el ámbito industrial existen múltiples eventos que pueden afectar a la producción en curso, y por lo tanto, al Programa Productivo. En el caso que se está analizando los eventos característicos han sido indicados en el punto anterior. En todo caso, lo que se desea destacar en este punto es que finalmente dichas perturbaciones se traducirán en una modificación de las fechas previstas de inicio o de fin de los trabajos en relación al Programa Productivo vigente. Esta modificación afecta al cumplimiento del Programa Productivo, y a la Tasa de Productividad prevista.

En este trabajo:

- Se entiende que el Programa se cumple si todos los trabajos se pueden realizar dentro de las fechas de finalización establecidas en el mismo.
- Se adopta el concepto tradicional de tasa de productividad entendido como el número de metros cuadrados fabricados por periodo productivo.

Se ha realizado una catalogación de los efectos que se pueden observar en el problema que se está analizando. Es la siguiente:

- *Desplazamiento a Derechas poco Significativo.* Uno o varios trabajos se desplazan hacia la derecha (se incrementa la fecha de inicio y/o fin), pero el conjunto de trabajos se puede realizar en su totalidad antes de la fecha de finalización prevista por el planificador, y dentro del periodo analizado.
- *Desplazamiento a Derechas Significativo.* Al desplazar el calendario previsto algún trabajo se sitúa fuera de la fecha de finalización indicada por el planificador y/o del periodo analizado.
- *Desplazamiento a Izquierdas.* Uno o varios trabajos se desplaza a la izquierda (se inicia y/o finaliza antes de lo previsto). No se contempla la posibilidad de cancelaciones de trabajos en el ámbito del programador, por lo que se trata exclusivamente modificaciones en los tiempos, ya sea de cambio o de ejecución. No afecta a la ejecución de los trabajos previstos. Como consecuencia se puede contemplar la inclusión de algún trabajo adicional. En todo caso, no se puede valorar esta opción sin cálculos previos.

Cada tipo de desplazamiento del Programa en curso puede tener implicaciones diferentes respecto al cumplimiento de las fechas de inicio o fin y a la tasa de productividad. Se va a realizar un análisis de cada caso:

- *Desplazamiento a Derechas poco Significativo:* No tiene implicaciones en relación al cumplimiento del Programa ni en cuanto a la tasa de productividad. Si se asume que el Programa inicial maximiza la tasa de productividad en el periodo productivo correspondiente, se debe suponer que un retraso no permite disponer de mayor capacidad y por lo tanto el Programa no es susceptible de ser mejorado en dichas condiciones.
- *Desplazamiento a Derechas Significativo:* Tiene implicaciones con respecto al cumplimiento del Programa y como consecuencia puede producir una reducción en la tasa de productividad. Si algún trabajo no se puede realizar antes de la fecha de finalización prevista por el planificador o del periodo es posible que haya que realizarlo en otro periodo, y en ese caso se producirá una reducción de los m^2 fabricados en el periodo analizado.
- *Desplazamiento a Izquierdas:* Puede tener implicaciones con respecto a la tasa de productividad. Puede darse el caso de que se puede ganar el tiempo suficiente como para que se pueda realizar algún trabajo más en el mismo periodo. Este aumento de

las tasa de productividad puede alcanzarse con una modificación sustancial del Programa inicialmente previsto o simplemente con la inserción en algún hueco de los nuevos trabajos.

La casuística que se puede llegar a dar durante la evolución del Programa Productivo se tendrá en cuenta en el proceso de modelado y posterior búsqueda de una solución para el reajuste del Programa en curso o Reprogramación.

5.2.1.2.3. Evaluación de la Gestión del Programa de Producción

Una vez identificada la necesidad de considerar un proceso de *Gestión del Programa Productivo*, motivado por la Incertidumbre, que se basa en dos actos clave, establecer y revisar/modificar el Programa Productivo, es necesario identificar la necesidad de evaluación del mismo. **Para evaluar la calidad de dicha gestión se deberá utilizar una medida o parámetro de eficiencia.** Estas medidas pueden estar basadas en términos temporales, económicos u otros. En la visión tradicional del modelado y resolución de problemas de Programación de la Producción se suele evaluar la calidad de un Programa frente otros obtenidos para la misma instancia de problema. Sin embargo, no es habitual la evaluación de las revisiones que sufre un Programa frente al establecido inicialmente, es decir la evaluación de su evolución.

Evaluar la calidad de la evolución de un Programa es evaluar la calidad de la gestión que se hace del mismo. La identificación de un medible asociado a la evolución del Programa Productivo permitirá identificar las variables que el programador debe controlar para alcanzar sus objetivos. Así mismo, permitirá comparar las diversas formas de gestionar el Programa, y por lo tanto seleccionar mejor técnicas utilizadas en dicho proceso. Por este motivo **se considera fundamental asignar a este proceso un medible diferente de los que en su momento se identificarán para la tarea concreta de Calcular un Programa**, en la fase de Establecimiento o de Modificación. Dicho medible puede ser aprovechado para la Gestión Global de la Empresa dentro del marco de un Sistema para la Medición de Rendimiento (Alfaro, J. J. 2003;Rodríguez, R. 2007), aunque este aspecto queda fuera del ámbito de este trabajo.

5.2.1.3 El Programa Productivo: La Programación Predictiva y Reactiva

El cálculo del Programa Productivo varía sensiblemente según se trate de un proceso de *Establecimiento o de Modificación* del mismo. Por este motivo es conveniente introducir la diferencia terminológica y conceptual que existe entre ambos enfoques.

Desde el punto de vista temporal, el problema que se afronta es cíclico, y contiene al menos un par de hitos conocidos a priori por el programador y el planificador, que son la fecha de inicio y de fin del periodo. Pueden existir otros hitos relevantes (avería, retrasos, etc.) dentro de un periodo considerado, pero éstos no serán conocidos a priori ni por el programador ni por el planificador. Los hitos que se han considerado son los que afectan al Ciclo de Vida del Programa Productivo y que suponen, por lo tanto, el inicio del proceso de Establecimiento o Modificación, es decir reactivan la comunicación entre planificador y programador. La naturaleza que causa los hitos es diferente. Las causas de los hitos producidos durante el transcurso del periodo son debidas a un desajuste importante del Programa en curso con la realidad producido por alguna de las casuísticas establecidas en el apartado 5.2.1.2. Dependiendo de la naturaleza que causa el Cálculo del Programa se puede distinguir en dos casos diferenciados, aunque no independientes; La Programación Predictiva y la Programación Reactiva (Vieira, G. E., Herrmann, J. W., & Lin, E. 2003). El primer caso se define en el Proceso de Establecimiento, y se corresponde con el desarrollo de un Programa que en todo caso se concreta con anterioridad al inicio del periodo programado, y sin vinculación con el anterior. El segundo caso se realiza en el Proceso de Modificación, y se encarga de analizar la forma de abordar un cambio en un Programa Predictivo mediante una Reprogramación del mismo, o incluso en un Programa Modificado previamente.

Hasta este momento se ha planteado la conveniencia de plantear Programas Reactivos, pero es necesario aclarar si es posible la aplicación de esta técnica analizando el tiempo de reacción que ello implica. El tiempo disponible para que un Programas sustituya a otro en curso, es decir para Tomar una Decisión, es aquel que transcurre desde que se detecta el evento problemático hasta que se produce un hito como puede ser el final de una operación. Si no se está en disposición de “reaccionar” en ese tiempo la propuesta que se realice se quedará obsoleta. En el caso de la industria cerámica donde los Tiempos de Proceso son relativamente altos, normalmente medidos en horas, se puede reaccionar bastante bien ante un cambio en el Programa de Producción. Existen otras restricciones que hay que tener en cuenta antes de plantear el proceso de cambio, como es el caso de los recursos disponibles. Este aspecto se comentará durante la definición del Modelo Reactivo.

Por la naturaleza del problema que abordan el Cálculo de los Programas Predictivo y Reactivo son actividades muy diferentes. Por este motivo, los modelos que pudieran llegar a plantearse de cada problema deben tener en cuenta diferentes datos, parámetros, variables y funciones objetivo. Una descripción de ambos procesos se realiza en el apartado 5.2.2.

5.2.2 Condiciones que Afectan al Cálculo del Programa Productivo

Hasta el momento se han ido exponiendo con detalle el conjunto de aspectos relacionados con el entorno a partir del cual se Establece o Modifica un Programa Productivo, y que afecta sobre todo a la *Gestión del Programa*, aunque no exclusivamente. En este punto se detallarán los aspectos que condicionarán la tarea de Cálculo del Programa Predictivo y del Programa Reactivo en sí misma. Se trata de introducir aspectos cuantitativos aunque sin emplear aún un lenguaje más propio de este ámbito como es el matemático.

La identificación del conjunto de condiciones que afectan al cálculo de los Programas Productivos es fundamental para entender en conjunto el problema que se plantea. Una vez analizado este aspecto se realizará una revisión del problema es su conjunto con el fin de estructurarlo y disponer de una visión más completa del mismo.

Este apartado trata en primer lugar ciertas consideraciones sobre los objetivos de una futura función objetivo, para introducir en segundo lugar las condiciones exigibles al cálculo del Programa Predictivo, y en tercer lugar las condiciones exigibles al cálculo del Programa Reactivo.

5.2.2.1 Consideraciones sobre la Función objetivo

En el problema que se está identificando ya se ha definido un objetivo asociado al Sistema Productivo, claramente establecido, como es alcanzar la máxima producción posible en cada periodo productivo. En el ámbito de trabajo del Programador éste está ligado a la *Gestión del Programa Productivo*. No se debe perder de vista que como parte de la *Gestión del Programa* se encuentran las tareas embebidas de Cálculo del Programa Predictivo o Reactivo, y en que cada una de ellas se debe establecer, a su vez, un objetivo u objetivos que favorezcan la consecución del objetivo principal. Existe por lo tanto, una relación jerárquica entre objetivos.

En base al conocimiento que se tiene del problema y a la experiencia se ha determinado que el Programa Predictivo esté vinculado a un único objetivo, mientras que para el Programa Reactivo se plantea un equilibrio entre dos objetivos. Para ambos tipos de Programas se ha definido un objetivo común que está relacionado con la *Maximización de la Tasa de*

Productividad, pero en el caso del Programa Reactivo se ha incorporado un objetivo adicional ligado a la *Estabilidad^v del Programa*.

Los Programas Predictivo y/o Reactivos no pueden manejar en su ámbito conceptos como los de la Tasa de Productividad, ya que según las hipótesis que más tarde se establecerán en los apartados 5.2.2.2 y 5.2.2.3, en la tarea de cálculo no existe la opción de modificar el número de trabajos ni la cantidad a fabricar, ni tampoco la duración del periodo. Sin embargo, sí es posible proponer un objetivo que favorezca la obtención de unos resultados que puedan ser aprovechados en el *Proceso de Gestión del Programa* para ir proponiendo cambios en el Plan Maestro o/y en el Plan de Fabricación. Una buena forma de alcanzar el objetivo de obtener el mayor valor posible de la Tasa de Productividad en el ámbito de la *Gestión del Programa* es la de utilizar como objetivo en el ámbito del Programa la minimización de la fecha de finalización de la última operación del último trabajo en terminar, también conocida como *makespan*. Esta medida de eficiencia ha sido vinculada por múltiples autores a la utilización eficiente de los recursos, y por lo tanto se ajusta a la idea de acabar lo antes posible^{vi} respetando las restricciones que se determinen en relación con la precedencia de operaciones, respeto de las fecha de entrega, etc. Si la tarea de Ajuste funciona adecuadamente se podrían realizar más órdenes. El *makespan* fuerza a los trabajos a ocupar un espacio temporal lo más cercano posible al instante de inicio (parte izquierda de un diagrama Gantt) dejando, si existe, el espacio temporal no utilizado al final del periodo. Esto permite ver con facilidad si se dispone de tiempo sobrante o no para intentar incluir algún trabajo adicional.

Los objetivos comentados serán expresados en forma de Función Objetivo cuando se planteen los Modelos Matemáticos Correspondientes. El aspecto vinculado al objetivo de *Estabilidad del Programa* en la tarea de Cálculo del Programa Reactivo será detallado y justificado, en su conjunto, en el momento de realizar el Modelo correspondiente.

5.2.2.2 Proceso Productivo en Relación con la Programación Predictiva

A continuación se van a describir las hipótesis relativas al funcionamiento del sistema productivo que se considera con el objeto de acotar inicialmente el problema que se desea

^v Se considera que un Programa Productivo es totalmente estable si durante su ejecución no presenta ningún tipo de modificación. A medida que se van realizando cambios aumenta la inestabilidad o nerviosismo del mismo.

^{vi} Como se explicará en los apartados 5.2.2.2 y 5.2.2.3 no se han identificado necesidades específicas que hayan hecho aconsejable establecer otros criterios de eficiencia relacionados con la gestión de las fechas de entrega, costes, etc.

abordar. En dichas hipótesis se hace referencia a las condiciones y restricciones necesarias para elaborar un Programa de Producción Predictivo. En ningún caso se comentan los aspectos relacionados con las causas de una Reprogramación, ni las restricciones y objetivos de la misma, ya que se expondrán en el apartado 5.2.2.3.

Se considerarán las siguientes hipótesis:

- Todos los trabajos son objeto del mismo tipo de operaciones básicas, y en el mismo orden. Para realizar cada una de las operaciones se dispone de un conjunto de máquinas alternativas, no necesariamente idénticas que se agrupan en etapas. Es decir, existe un flujo en que los trabajos pasan primero por la etapa 1 donde realizan la primera operación en alguna de las máquinas disponibles, después pasan por la etapa 2 donde realizan su segunda operación, y así sucesivamente, hasta realizar la última operación.
- Entre cada dos etapas del sistema existe un almacén de capacidad ilimitada. En este tipo de empresas, el ritmo de producción, está regulado de tal forma que no se suelen saturar los almacenes intermedios aunque se mantienen a un nivel alto de productos (sobre todo entre líneas y hornos, para evitar interrupciones de proceso en estos últimos), por lo que es razonable considerar que no existen problemas de capacidad. Existe un número suficiente de sistemas de transporte (AGV) utilizados entre etapas y los productos pueden esperar a ser transportados puesto que hay unos pequeños pulmones en la zona de carga de AGV.
- No se considera la posibilidad de que el mismo trabajo esté siendo procesado a la vez en diferentes etapas (debido a que el lote es muy grande y las primeras unidades pueden estar empezando a procesarse en una etapa mientras se finalizan las últimas unidades en la etapa anterior). La fecha más temprana de inicio del procesamiento de la primera pieza de un lote en un recurso, siempre será mayor o igual que la de finalización de la última pieza del mismo lote en la etapa anterior, lo que equivale a considerar cada lote completo como un trabajo único, cuyo tiempo de proceso total será el producto del tamaño de lote por el inverso de la capacidad del recurso (h/m^2). Esta suposición, no es muy restrictiva, puesto que los almacenes intermedios del sistema tienen una gran capacidad y se suele acumular trabajo en curso en ellos.
- No existirá la posibilidad de fraccionamiento de los trabajos en varios recursos, sino que se supone que esta opción ya habrá sido tomada en cuenta a la hora de definir el Plan Maestro de Fabricación. Esta restricción facilita enormemente la resolución de los modelos aunque hay que reconocer que aleja el mismo de la realidad. No obstante, en la propuesta los resultados se analizarán desde un punto de vista relativo o comparativo (con lo que hace la empresa u otros algoritmos similares), lo que implica

que se el criterio se aplica en todos los casos puede facilita el modelado sin quitar validez a los resultados.

- En el momento en que un lote se empiece a procesar en un recurso no se podrá interrumpir la operación hasta no haber terminado con las unidades que componen ese lote, ya que esto supondría el incremento de los costes derivados de los tiempos de cambio.
- Aquellos recursos en los que no exista posibilidad de alterar la secuencia de trabajo debido a que están conectados entre sí por una cinta transportadora, se supondrán uno solo.
- El abastecimiento de materias primas no condiciona el Programa, puesto que este problema ya se supone resuelto en el ámbito de la Planificación de Producción mediante procedimientos específicos de gestión de materiales.
- Desde el punto de vista de la secuenciación no existen restricciones de mano de obra por el motivo comentado en el punto anterior.
- Todos los recursos que pertenecen a una etapa no se suponen idénticos. En general las empresas instalan y renuevan la maquinaria según sus necesidades y posibilidades.
- Las fechas más tardías de finalización de los trabajos se suponen conocidas antes de calcular el Programa de Producción ya que han sido prefijadas dentro del procedimiento de Planificación de Producción. Se supone que no se modificarán durante el mismo.
- Los trabajos a realizar son conocidos antes de elaborar el Programa de Producción. Están incluidos en el Plan de Fabricación.
- No se permite que ningún trabajo finalice su última operación con posterioridad a la fecha de finalización prefijada. En caso de no poder satisfacer esta restricción se debe establecer una nueva propuesta de trabajos a realizar.

En la Industria Cerámica existe una amplia gama de modelos tanto de Gestión como de Diseño del Sistema Productivo. Por ejemplo, en algunas empresas la disponibilidad de AGVs puede ser baja y puede aparecer un problema de almacenamiento intermedio, o puede que la gestión de los trabajos les lleve a que parte de un trabajo esté realizando su primera operación y otra parte su segunda operación. Por este motivo, es imposible definir, lo que se puede denominar, un conjunto estándar de hipótesis. Sin embargo, las limitaciones establecidas se representan bastante bien al problema real de muchas empresas.

Por las características del tipo de problema que se desea afrontar, un problema de Programación de la Producción, hay aspectos del Diseño del Sistema Productivo que no resulta relevante. Se propone que el sistema productivo se *simplifique de la siguiente forma*:

- En primer lugar se realiza la agregación en una única entidad del conjunto formado por prensas, secaderos, volteadoras y líneas de engobe y esmaltado. Debido a que todas estas máquinas están unidas por un sistema de cinta transportadora y, por lo tanto, el orden de paso de los productos fabricados es el mismo en todas las máquinas, a efectos de Planificación y Programación de Producción pueden considerarse como un único Recurso.
- En segundo lugar, se considerará a cada horno como una máquina o Recurso.
- Las secciones de clasificado y embalado están unidas por una cinta transportadora, por lo que serán consideradas como una máquina o Recurso único.
- Entre la sección de líneas y hornos y entre hornos y clasificación existen unas amplias zonas de almacenamiento que, a efectos de la investigación, se supondrán de capacidad infinita.

De acuerdo a las simplificaciones^{vii} establecidas el sistema productivo, que da definido como un taller de tres etapas (ver **Figura 5-3**): *Prensas-esmaltado, hornos y clasificación-embalaje*.

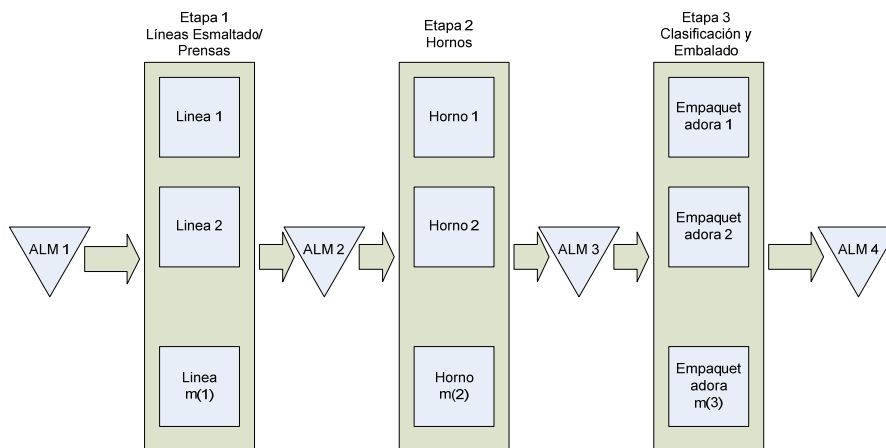


Figura 5-3. Taller cerámico de 3 etapas (elaboración propia).

^{vii} Esta disposición se considera válida para un amplio conjunto de empresas del Sector Cerámico, no obstante pueden existir distribuciones alternativas.

Basándose en la notación establecida por (Conway, R. W., Maxwell, W. L., & Miller, L. W. 1967) para poder identificar los problemas de secuenciación, el problema predictivo determinista que se pretende resolver podría ser representado como $n/r/F/k_1,..k_r$ -parallel/setup/Cmax. Es decir se trata de un taller de flujo con r etapas, y k_1 máquina en paralelo en la primera etapa, k_2 en la segunda y k_r en la última, con tiempos de setup dependientes de la secuencia, cuya función objetivo es minimizar el Cmax o makespan.

5.2.2.3 Hipótesis del Proceso Productivo en Relación con la Programación Reactiva

De la misma forma que se ha hecho para el caso del proceso de Programación Predictivo a continuación se van a describir las hipótesis relativas al sistema productivo que se consideran necesarias para elaborar un Programa de Producción Reactivo o Reprogramación.

La Programación Reactiva se produce cuando se dispone de un Programa Productivo en curso y el programador decide realizar un nuevo Programa que sustituya a éste. Ese proceso de cambio suscita una serie de cuestiones relativas a la transición entre ambos Programas, y a las condiciones en la que debe ser construido el nuevo Programa Productivo.

El Programa de Producción obtenido después de una Reprogramación debe ajustarse a las condiciones de un sistema productivo muy similares a un Programa Predictivo. Por este motivo, en general, las hipótesis del punto anterior siguen siendo válidas en este punto. No obstante se considera importante reflexionar sobre las siguientes restricciones:

- No existirá la posibilidad de fraccionamiento de los trabajos en varios recursos, sino que se supone que esta opción ya habrá sido tomada en cuenta a la hora de definir el Plan Maestro de Fabricación.
- En el momento en que un trabajo se empieza a procesar en un recurso no se podrá interrumpir la operación hasta no haber terminado con las unidades que componen ese lote, ya que esto supondría el incremento de los costes derivados de los tiempos de cambio.

Ambas condiciones son válidas por lo que se refiere al periodo de ejecución del Programa Reactivo que se genere. Sin embargo, entre el Programa en curso y el nuevo Programa Productivo propuesto se podrían realizar interrupciones y fraccionamiento de trabajos. Este caso se corresponderá exclusivamente con una situación de no disponibilidad de un recurso. En ese caso posiblemente se deberían cambiar los trabajos en curso afectados, y realizados parcialmente, a otro recurso disponible, reflejándolo así en el nuevo Programa Productivo.

Así mismo, tampoco se cumplirá la siguiente condición:

- El abastecimiento de materias primas no condiciona el Programa, puesto que este problema ya se supone resuelto al ámbito de Planificación de Producción mediante procedimientos específicos de gestión de materiales.

Como se explicará posteriormente en el cálculo del nuevo Programa Productivo se intentará evitar cierto tipo de adelantos en los trabajos con respecto a las fechas de inicio de las operaciones previstas en el Programa en curso. Esta condición se establece en base a las relaciones que actualmente se dan entre los proveedores de materia prima y las empresas cerámicas.

Adicionalmente se han de considerar otra serie de hipótesis para el problema propuesto, que son:

- Existirá un conjunto trabajos programados en el Programa nuevo que puede ser diferente, en número y composición, al conjunto de trabajos programados en el Programa en curso.
- Cada uno de los trabajos del conjunto correspondiente al nuevo Programa puede iniciarse en una operación que no necesariamente será la de la primera etapa como ocurría en el Programa determinista Predictivo.
- La cantidad a producir de los trabajos debe reducirse en aquellas operaciones que se han realizado parcialmente, en el que será menor del que se consideró en el Programa Productivo Determinista Predictivo. Se considerará que todos los trabajos que estaban procesando una operación en el instante de Reprogramar tienen pendiente de realizar parcialmente dicha operación.
- El nuevo Programa deberá garantizar, dentro de sus posibilidades, que las operaciones en curso en el instante de Reprogramación siguen procesándose tal y como lo estaban haciendo. Esto significa que los trabajos con una operación iniciada en el Programa en curso continuarán sin interrupción de ningún tipo en la misma máquina que lo estaban haciendo cuando el nuevo Programa se inicie siempre que dicho recurso esté disponible.
- La fecha de disponibilidad de cada trabajo será la del inicio del nuevo Programa.

Es difícil poder definir el problema basándose en la notación establecida por (Conway, R. W. et al. 1967) para poder identificar los problemas de secuenciación. Por este motivo se ha

realizado una incorporación de códigos nuevos respetando en todo caso la estructura propuesta. El Problema Reactivo Determinista que se pretende resolver podría ser representado como $n/r/F/k_1,..k_r$ -parallel/setup,preasig/B. Es decir se trata de un Taller de Flujo con r etapas, y k_1 máquina en paralelo en la primera etapa, k_2 en la segunda y k_r en la última. Donde existe un Tiempo de Cambio de Partida o Setup dependiente de la Secuencia y una serie de órdenes pre-asignadas a máquinas (preasig), cuya función objetivo es B (balance), un balance entre el makespan y el adelanto de las fechas de inicio de ciertos trabajos.

5.3 Estructura del Problema Identificado

Una vez se ha realizado la descripción del problema que se desea resolver, en la que se han analizado diversos matices y enfoques del mismo, es conveniente realizar un resumen del mismo que permita contar con una estructura que facilite su comprensión y futura resolución.

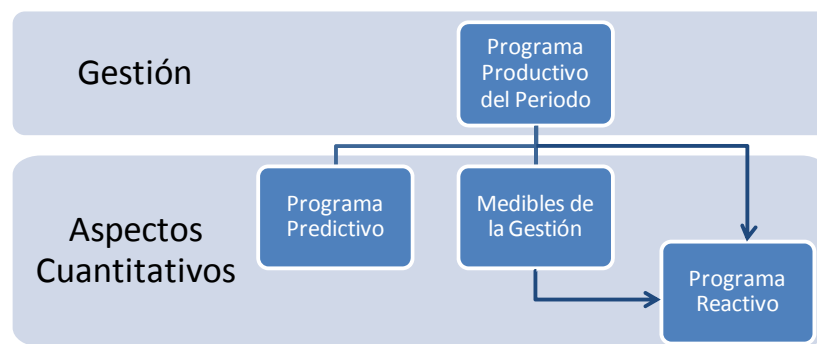


Figura 5-4. Estructura de Problema Identificado (elaboración propia).

Como ya se avanzó en la introducción del este capítulo, y como queda reflejado de una manera gráfica en la Figura 5-4 existen problemas a dos niveles. **Se puede decir que en el ámbito de la Gestión de las Operaciones de un taller existe un problema de Toma de Decisiones, que para alcanzar sus objetivos, necesita balancear un conjunto de actividades, de las cuales algunas se pueden identificar como cualitativas y otras cuantitativas.**

En el nivel de gestión el problema consiste en como coordinar los procesos de cálculo del Programa Predictivo y de los sucesivos Programas Reactivos que pudieran darse. Ello significa básicamente disponer de información y de criterios que permitan tomar las mejores decisiones, y además hacerlo en el instante más adecuado. En este nivel habría que tomar decisiones, por ejemplo, como la de determinar cuando un Programa ya no es válido, y decidir cuando se lanza el cálculo de un nuevo Programa y bajo que condiciones, o por ejemplo, decidir que se hace cuando los Programas Productivos que se calculan son infactibles por falta de capacidad (por avería de una máquina, por ejemplo).

En el nivel cuantitativo el problema se centra básicamente en realizar un cálculo, cuyo resultado será normalmente el equivalente a un diagrama Gantt con una programación, que pueda ser empleado en el nivel de gestión. En este segundo nivel se han identificado dos problemas diferentes, el cálculo del Programa Predictivo, y el cálculo del Programa Reactivo. Además en el nivel cuantitativo se han incluido comentarios sobre los medibles que permitirán evaluar la Gestión del Programa.

Como se ha podido apreciar en los problemas de cálculo de la Programación Predictiva y cálculo de la Programación Reactiva se conocen muchos más detalles que cómo conectar ambos para obtener unos resultados adecuados en el ámbito de la programación durante todo el periodo temporal que se desee considerar. Esto es así porque de hecho en la realidad que se ha observado ocurre lo mismo.

En la realidad observada se ha podido identificar en que condiciones se afronta el cálculo del Programa Predictivo, porque se hace con asiduidad, y se ha podido identificar en que condiciones se debería de calcular el Programa Reactivo, aunque de manera formal este problema no se había planteado en la mayoría de los casos observados. La formalización sobre cuando es necesario revisar un Programa, cuando es necesario modificarlo o como se deberían de hacer los cambios, que son aspectos relacionados con la Gestión del Programa, no se ha visto reflejada en ninguna experiencia práctica. Por este motivo, muchos de los aspectos que forman parte del detalle de la Gestión del Programa serán considerados en el desarrollo de la propuesta reduciéndose la fase de identificación de sus aspectos más generales y al motivo que los causa. No obstante, aunque como se llevaría a cabo la gestión de este proceso no está demasiado claro en las empresas analizadas, si que por lo general han podido identificar con más claridad los objetivos que persiguen con dicha gestión, lo que ha permitido avanzar algún comentario al respecto.

A partir de los dos niveles de problemas detectados parece razonable proponer una definición de los problemas que se incluyen en cada uno de ellos de forma precisa y pormenorizada. Pero dado que en este caso, el problema de gestión se detallará formará parte de la propuesta, no se ampliará más este nivel. Sin embargo, los problemas de cálculo de los Programas, se volverán a plantear empleando en esta ocasión el lenguaje matemático para su modelado. También se incluirán en este nivel a algunas reflexiones sobre los medibles que se deberían aplicar a la Gestión del Programa. Se debe de dejar claro, que el hecho de plantear modelos matemáticos no implica que el problema finalmente se solucione empleando dichos modelos. Se ha seleccionado el modelado matemático como forma de representación del problema, siendo por su parte uno de los más empleados en los problemas que se tratan en este nivel.

5.4 Formulación del Modelo Matemático para el Cálculo de Programa Productivo en un Taller de Flujo Híbrido

En el apartado 5.2 se presentó de forma descriptiva el problema que se desea resolver, concretamente el conjunto de hipótesis que se establecen para el modelo de Taller de Flujo Híbrido. Con el objeto de mostrar formalmente el modelo del Taller de Flujo Híbrido se ha procedido a la definición de un Modelo Matemático que se muestra en los siguientes subapartados.

Para ajustar de forma progresiva el modelo a las características del problema expuesto se va a iniciar el modelado con un enfoque Determinista Predictivo tradicional, en un segundo paso se realizará una modificación de este para introducir los aspectos relacionados con el problema de la Reprogramación de la Producción o Modelo Reactivo, y finalmente se realizará el modelado de la Incertidumbre. Siguiendo la nomenclatura de (Vieira, G. E. et al. 2003) se ha denominado Modelo Predictivo al que contemplan las condiciones a priori y objetivos del programador, y que da lugar al Programa que se empieza a ejecutar. Y se ha denominado Modelo Reactivo al que tiene en cuenta las condiciones y objetivos del programador en el momento que se decide modificar un Programa Productivo en curso, y da lugar al Programa Reactivo. Una vez introducidos los modelos Predictivo y Reactivo, se analiza cual debería ser el medible del proceso conjunto, es decir, de la Gestión del Programa.

5.4.1 Modelo Predictivo Determinista

El modelo que se expone a continuación está basado en el trabajo de (Andrés, C. 2001), y se considera válido para cualquier instante de tiempo dentro del periodo que se considera. Como ya se ha visto, si alguno de los eventos o distorsiones, que el sistema es capaz de detectar, provocasen un cambio que implique invalidar el Programa en curso se realizará una Reprogramación, y por lo tanto el cálculo de un nuevo Programa. Por este motivo el modelo no se expresa en función del tiempo, ya que no es necesario.

La nomenclatura utilizada ha sido la siguiente:

Conjuntos

R: *Conjunto de etapas. Tiene asociado un índice $r=1...RX$.*

M: *Conjunto de recursos (máquinas). Tiene asociado un índice $j=1...mx(r)$. Tal que mx es depende de la etapa.*

N: *Conjunto de trabajos a programar. Tiene asociado unos índices i , i_1 e i_2 . Todos los índices van de 1 hasta NX sino se especifica lo contrario.*

Si $i=0$ se entiende como el trabajo que estaba inicialmente en la máquina o estado de la máquina antes de empezar la secuencia planteada, si $i= NX +1$ se entiende como que se ha acabado el trabajo NX , y la máquina está en su estado final.

Restricciones Tecnológicas

$p(i,j,r)$: *Tiempo de proceso del trabajo i en la máquina j de la etapa r .*

$st(i_1,i_2, j, r)$: *Tiempo de ajuste de la máquina j de la etapa r para pasar de realizar el trabajo i_1 al trabajo i_2 .*

Fechas de Finalización Previstas

$d(i)$: *Fecha de finalización del trabajo i indicada en el Plan Maestro.*

Variables de Decisión

Continuas

$s(i,r)$: *Fecha máxima de inicio de las operaciones sobre el trabajo i en la etapa r . Las operaciones incluyen el tiempo de ajuste y cambio, y el tiempo de proceso del lote. El índice i puede tomar valores entre 0 y $NX+1$. Representa $(NX+2) * RX$ variables.*

$c(i,r)$: *Fecha de finalización del trabajo i en la etapa r . Representa $NX * RX$ variables.*

Binarias

$y(i, j, r)$: *Relacionada con la asignación de los trabajos en las máquinas.*

$\left\{ \begin{array}{l} =1 \text{ Si el trabajo } i \text{ es procesado por la máquina } j \text{ en la etapa } r. \\ =0 \text{ En otro caso} \end{array} \right.$

Este tipo de variables representan $\sum_{r=1}^{RX} NX \cdot mx(r)$ variables.

$x(i_1, i_2, j, r)$: Relacionada con la secuencia de los trabajos en las máquinas.

$$\left\{ \begin{array}{l} =1 \text{ Si el trabajo } i_1 \text{ es procesado inmediatamente antes que el} \\ \text{trabajo } i_2 \text{ en la máquina } j \text{ en la etapa } r. \\ =0 \text{ En otro caso} \end{array} \right.$$

Tanto i_1 toman valores entre 0 y NX , e i_2 toman valores entre 1 y $NX+1$. En un modelo representan $\sum_{r=1}^{RX} (NX + 1) \cdot (NX + 1) \cdot mx(r)$ variables.

Para este tipo de variables se definen los siguientes casos particulares:

$$x(0, i_2, j, r) \left\{ \begin{array}{l} =1 \text{ Si el trabajo } i_2 \text{ es el primero en ser procesado en la máquina } j \\ \text{en la etapa } r. \\ =0 \text{ En otro caso} \end{array} \right.$$

$$x(i_1, NX + 1, j, r) \left\{ \begin{array}{l} =1 \text{ Si el trabajo } i_1 \text{ es el último en ser procesado en la máquina } j \\ \text{en la etapa } r. \\ =0 \text{ En otro caso} \end{array} \right.$$

El modelo es el siguiente:

$$[5.1] \quad \text{Minimizar } Z = \max_{i=1}^{NX} \{c(i, RX)\}$$

Sujeto a:

$$[5.2] \quad \sum_{j=1}^{mx(r)} y(i, j, r) = 1 \quad \forall i, \forall r$$

$$[5.3] \quad \sum_{i_1=0}^{NX} x(i_1, i_2, j, r) - y(i_2, j, r) = 0 \quad i_1 \neq i_2, \forall i_2, \forall j, \forall r$$

$$[5.4] \quad \sum_{i_1=1}^{NX+1} x(i_2, i_1, j, r) - y(i_2, j, r) = 0 \quad i_1 \neq i_2, \forall i_2, \forall j, \forall r$$

$$[5.5] \quad \sum_{i=1}^{NX+1} x(0, i, j, r) = 1 \quad \forall j, \forall r$$

$$[5.6] \quad \sum_{i=0}^{NX} x(i, NX+1, j, r) = 1 \quad \forall j, \forall r$$

[5.7]

$$s(i_2, r) \geq s(i_1, r) + \sum_{j=1}^{mx(r)} \sum_{i_3=0}^{NX} (p(i_1, j, r) + (st(i_3, i_1, j, r) \cdot x(i_3, i_1, j, r))) + \Omega \cdot \left(\sum_{j=1}^{mx(r)} x(i_1, i_2, j, r) - 1 \right)$$

$i_1 \neq i_3, i_2 \neq i_3, \forall i_1 = 0..NX,$

$\forall i_2 = 1..NX+1, \forall r$

Ω es un valor elevado aunque lo más ajustado posible en función de los datos.

[5.8]

$$\begin{aligned} s(i_2, r) + \sum_{i_1=0}^{NX} \sum_{j=1}^{mx(r)} \{ [st(i_1, i_2, j, r)] \cdot x(i_1, i_2, j, r) \} &\geq \\ \geq s(i_2, r-1) + \left(\sum_{i_3=0}^{NX} \sum_{j=1}^{mx(r-1)} \{ p(i_2, j, r-1) + [st(i_3, i_2, j, r-1)] \cdot x(i_3, i_2, j, r-1) \} \right) & \end{aligned}$$

$i_1 \neq i_2, \forall i_2, \forall r > 1$

[5.9]

$$c(i_2, RX) \geq s(i_2, RX) + \left(\sum_{i_1=0}^{NX} \sum_{j=1}^{mx(RX)} (p(i_2, j, RX) + (st(i_1, i_2, j, RX) \cdot x(i_1, i_2, j, RX))) \right)$$

$$i_1 \neq i_2, \forall i_2$$

[5.10]

$$d(i) \geq c(i, RX)$$

$$\forall i$$

5.4.2 Comentarios al Modelo Predictivo Determinista

El modelo permite minimizar la fecha de entrega del último trabajo ya que así se plantea en la función objetivo. El modelo introduce una serie de restricciones que se pueden agrupar en 3 tipos:

- Restricciones relacionadas con la asignación y la secuencia: [5.2][5.3][5.4][5.5][5.6].
- Restricciones relacionadas con la asignación y la secuencia, en base a las fechas de inicio y fin de las operaciones:[5.7][5.8][5.9].
- Restricciones relacionadas con la fecha de entrega: [5.10].

A continuación se comentan una a una con cierto detalle.

La restricción ligada a la ecuación [5.2] obliga a asignar cada trabajo a una máquina de cada etapa. Existiendo $NX \cdot RX$ restricciones.

La restricción ligada a la ecuación [5.3] implica que, una vez asignado el trabajo a una máquina, esta tenga un trabajo predecesor, que bien puede ser cualquiera de los otros trabajos o la situación cero de la máquina (situación inicial antes de las secuenciación). En el modelo se

deben definir $\sum_{r=1}^{RX} NX \cdot mx(r)$ restricciones.

La restricción ligada a la ecuación [5.4] es similar a la anterior, implica que una vez asignado un trabajo a una máquina, este tenga una pieza sucesora, que bien puede ser cualquiera de los otros trabajos o la situación final de la máquina de la máquina (situación final después de las secuenciación). También se consideran $\sum_{r=1}^{RX} NX \cdot mx(r)$ restricciones.

En este modelo, se supone que cada máquina parte de una asignación inicial de trabajos (modelada mediante el trabajo 0) y llega a una última asignación (modelada mediante un trabajo final $NX+1$). Así, la restricción ligada a la ecuación [5.5] obliga que exista un trabajo secuenciado en primer lugar por cada máquina de cada etapa. También puede darse el caso que la máquina no se utilice (con lo que todas las variables $x(0,i,j,r)$ serían iguales a 0 $\forall i=1,\dots,NX$, salvo $x(0,NX+1,j,r)=1$). Representa $\sum_{r=1}^{RX} mx(r)$ restricciones.

Mientras que la restricción ligada a la ecuación [5.6] es parecida a la anterior salvo que obliga a que exista al menos un trabajo antes de finalizar la secuencia en cada máquina de cada etapa. También, de igual forma a ecuación [5.4] se puede dar el caso que alguna máquina no procese ningún trabajo. También representa $\sum_{r=1}^{RX} mx(r)$ restricciones en el modelo.

Las siguientes restricciones se utilizan para calcular las fechas de inicio de cada trabajo en cada máquina. Teniendo en cuenta que dos trabajos asignados a la misma máquina deben conservar una relación de precedencia o sucesión entre ellos. Así, existe una serie de relaciones que marcan el momento en el que puede empezar un trabajo en una máquina que ha procesado con anterioridad otro trabajo (restricciones ligadas a la ecuación [5.7]). En este caso se aprovecha las variables $x(i_1,i_2,j,r)$ para establecer el orden en cada máquina. En efecto, si un trabajo i_1 es secuenciado antes que otro i_2 en la máquina j de la etapa r , el término $\sum_{j=1}^{mx(r)} x(i_1,i_2,j,r) - 1$ será 0 puesto que del todo sumatorio todas las variables serán nulas salvo $x(i_1,i_2,j,r)$ que valdrá 1. En ese caso la restricción plantea la relación entre las fechas de inicio de ambos trabajos en la máquina (la del trabajo i_2 , será la del trabajo i_1 más el tiempo de ajuste entre i_1 y el trabajo secuenciado con anterioridad a i_1 en la máquina, más el tiempo de proceso de i_1). En este caso se representa con el índice $i_1=0$ a la situación previa a la realización del primer trabajo, y con el índice $i_2= NX+1$ a la situación posterior al último trabajo. En este caso son $RX \cdot (NX + 1) \cdot (NX + 1)$ restricciones.

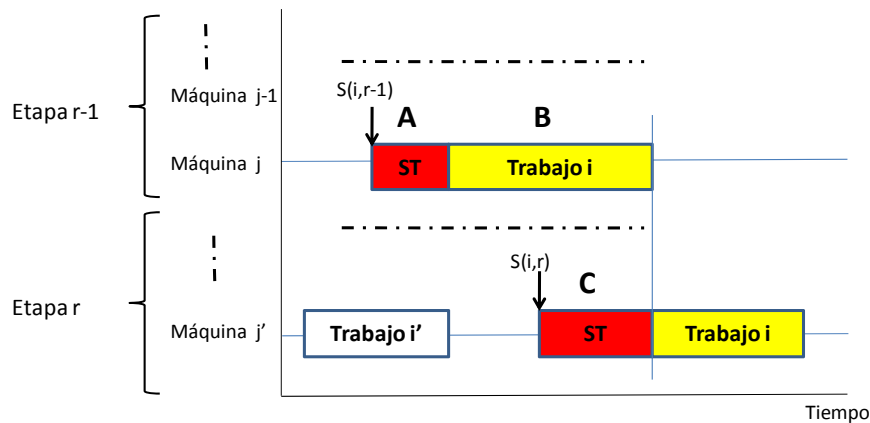
Además se debe considerar la relación que se da para las operaciones que debe sufrir un trabajo determinado en las diferentes máquinas de cada etapa del taller de flujo. En la

restricción [5.9] se indica que para el mismo trabajo, su fecha de inicio más temprana en una etapa r , será la mayor entre la fecha de disponibilidad de la máquina correspondiente de la etapa r y la fecha de fin del trabajo en la etapa $r-1$ menos el tiempo de cambio de partida necesario para el etapa r . Es decir, no es infrecuente comenzar a realizar el ajuste en una etapa mientras el trabajo no ha finalizado su proceso en la etapa anterior para ir ganando tiempo.

La explicación se de esta restricción se muestra en la figura siguiente. Si se considera a $s(i,r)$ la fecha en que se inicia el ajuste en una máquina de la etapa r , se relaciona con la fecha de inicio del ajuste en la etapa anterior mediante la expresión:

$$[5.11] \quad s(i, r) + C \geq s(i, r - 1) + A + B$$

Donde los términos A , B y C , mostrados en la formula desarrollada se corresponden con bloques temporales cuyo significado se marca en el diagrama de Gantt siguiente:



ST: Tiempo de Cambio de Partida

Figura 5-5. Relaciones entre variables para la determinación del instante de inicio del ajuste en una etapa considerando la posibilidad de solapamientos adaptado de(Andrés, C. 2001).

Esto representa $NX \cdot (RX - 1)$ restricciones.

La restricción ligada a la ecuación [5.9] permite calcular la fecha de finalización de cada trabajo en la última etapa. Similarmente al caso anterior representan NX restricciones.

La restricción ligada a la ecuación [5.10] permite garantizar que ningún trabajo terminará después de la fecha prevista. Esto representa NX restricciones.

Aunque el objeto de este apartado no era la resolución exacta del modelo es interesante tener en cuenta que de lo dicho anteriormente se puede observar que en un problema del tipo 20/3/F/1,2,3-parallel/setup/Cmax el número de restricciones se elevaría a $3*20+8*20+8*20+2*8+3*21*21+2*20+20+20 = 1.799$ restricciones, y el de variables a 3.814. Es difícil poder proporcionar un dato sobre cuanto tiempo podría necesitarse para solucionar dicho modelo, ya que depende absolutamente del equipamiento informático disponible. Tanto del software (solver), como del hardware (procesador, etc.). No obstante, se podría dar una horquilla de entre 30 minutos y varias horas.

Pero, independientemente de si es abordable o inabordable para un problema de tamaño real, la resolución del Modelo Matemático puede considerarse poco eficiente dentro del contexto que se plantea. Efectivamente, el proceso de cálculo se enmarca dentro del proceso de *Gestión del Programa*, lo que implica que es posible que sea necesario repetir el cálculo del Programa en múltiples ocasiones, con diferentes parámetros, hasta llegar a dar por concluido el proceso de Establecimiento o Modificación de un determinado Programa.

Además, aplicando criterios prácticos, se puede considerar que el resultado obtenido podría ser adaptado manualmente por el Programador, en base a su experiencia, para incluir aspectos difícilmente modelables. Es decir, si bien a priori, parece muy atractivo disponer de una solución óptima en la práctica puede ser más adecuada la utilización de soluciones sub-óptimas que puedan ser alcanzadas mediante procedimientos que requieran menos tiempo de cálculo.

5.4.3 Modelo Determinista para la Reprogramación o Reactivo

Tal y como se ha comentado anteriormente existen ciertas situaciones en las cuales el Programa Productivo establecido debe ser modificado. En dicho caso, el modelo planteado con anterioridad debe ser revisado para adaptarse a la nueva realidad. El modelo que se formule será utilizado para configurar un nuevo Programa y debe tener en cuenta esencialmente que éste depende en buena medida del anterior y del Estado del Sistema en el instante de tomar la decisión.

En este sentido, hay dos aspectos que varían sensiblemente con respecto al Modelo Predictivo:

- Se plantea una nueva Función Objetivo.
- Pueden existir trabajos en curso en el instante de Reprogramar, y por lo tanto recursos ocupados y trabajos ya asignados.

El Modelo Matemático que se propone se ajusta a las restricciones comentadas en el punto 5.2.2.3. La Función Objetivo que se formule debe de:

- Seguir considerando la correcta utilización de los recursos. Como ocurre con el cálculo del Programa Inicial puede favorecer la consecución de una Tasa de Productividad elevada.
- Incorporar un término, que este caso se ha denominado “*Adelanto*”, que se empleará para evitar el excesivo *Nerviosismo del Sistema*. En este caso, se desea evitar la modificación de trabajos cuya fecha de inicio esté próxima al instante de Reprogramación. A continuación se describe con precisión el concepto de Adelanto, y se justifica su utilización.

Cuando se plantea la posibilidad de realizar modificaciones en un Programa Productivo en curso como medio para hacer frente a posibles perturbaciones hay que considerar los efectos que pueden llegar a tener dichos cambios en el Sistema Productivo. Se trata de no solucionar unos problemas para crear otros nuevos. El punto más crítico frente a los cambios suele ser la variación en la necesidad de recursos (máquinas, materia prima, etc.), quizás necesidades superiores a las que se planteó inicialmente. Tal y como se ha enfocado el problema, los aspectos relativos a la utilización de los recursos operativos quedan englobados dentro del análisis de capacidad. No obstante, también se debe prestar atención al aprovisionamiento de materias primas en relación con los cambios de Programa Productivo. Este aspecto que se ha considerado resuelto en el enfoque Predictivo puede suponer restricciones adicionales en el planteamiento Reactivo.

En las Empresas Cerámicas existen múltiples materias primas necesarias para realizar un trabajo, pero se puede decir que hay un conjunto básico, desde el punto de vista de la caracterización del producto. Se trata de las aplicaciones cerámicas (esmaltes, serigrafías, granillas, etc.). Este tipo de materia es la que más restricciones presenta desde el punto de vista del aprovisionamiento, y por esta razón ha condicionado el planteamiento realizado en este punto. Los proveedores de aplicaciones cerámicas suelen estar muy orientados al cliente y dar un excelente servicio. No obstante, como todo fabricante necesita un tiempo mínimo para reaccionar ante un pedido. El retraso de un trabajo dentro del mismo periodo productivo no suele suponer un inconveniente ya que este tipo de producto no caduca a corto y medio plazo, y no suelen existir restricciones para su almacenamiento.

El “*Adelanto*” es un parámetro que mide el grado de anticipación de la fecha de inicio de los trabajos un Programa modificado frente al Programa Productivo en curso. Situar esta medida como parte de la Función Objetivo pretende reducir las opciones de que un cambio en un Programa implique el avance de algún trabajo, situándolo en una franja en la que el proveedor

tenga grandes dificultades para servir la materia prima necesaria, lo que puede conllevar una gestión adicional y en el peor de los casos la falta de existencias. Se puede considerar que un buen Sistema de Toma de Decisiones Automatizado, es imprescindible para detectar este tipo de situaciones en tiempo real. El soporte de las tecnologías de la Información es fundamental para poner en marcha cualquier medida relacionada con un aumento significativo de la reactividad del Sistema Productivo.

En concreto se define el “Adelanto” como la suma del número de trabajos cuya fecha de inicio de la primera operación^{viii} $s(i,1)$, en el nuevo Programa productivo S_v después de tener en cuenta el conjunto de informaciones $E=\{E_1, E_2, \dots\}$ en el instante t , se adelanta con respecto a la fecha de inicio de dicha operación en el Programa anterior S_{v-1} antes de tener en cuenta dicha información. Siendo que adicionalmente dichos trabajos deben cumplir la condición de que su primera operación $s(i,1)$ en S_v este situada dentro de una franja temporal restringida situada entre el instante t y $t+VA$, donde VA (Ventana de Avance) es una constante determinada por el programador. Formalmente se puede definir así:

$$[5.12] \quad \text{Adelanto}(S_{v-1}, S_v, E) = \sum_{i \in VA(i)} 1$$

Donde se suma uno por cada trabajo que pertenece al conjunto de trabajos cumplen las condiciones de penalización expresadas anteriormente en el instante “ t ”.

En el Modelo Matemático el adelanto modifica el valor del *makespan* obtenido, de forma que si se producen adelantos en el Programa propuesto se adiciona una cantidad al valor de *makespan*, es decir se penaliza con un cierto tiempo adicional.

Una vez descritas las bases del problema de Reprogramación o Reactivo se presenta el Modelo Matemático que lo define formalmente.

La nomenclatura utilizada tiene el mismo significado que en el modelo determinista de la Programación Predictiva excepto las que se comentan a continuación.

^{viii} La primera operación en el nuevo Programa Productivo, es su primera operación pendiente de realiza, y por lo tanto, no tiene porque coincidir con la primera operación del proceso productivo global.

Constantes

- K: *Es una constante que representa unidades de tiempo y es establecida en función de la duración media de los trabajos en el problema considerado. Su valor depende de la experiencia del programador y del peso que se quiera dar a los adelantos.*

Conjuntos

- R: *Conjunto de etapas. Tiene asociado un índice $r=1...RX$ que permite recorrer todas las operaciones existentes y un índice $e=\min(i)..RX$ que permite recorrer las operaciones pendientes de realizar del trabajo i en el Programa S_v .*
- N_v : *Conjunto de trabajos programados en la versión actual del Programa S_v . Se utilizan los índices indicados en el Programa determinista para la programación Predictiva.*
- N_{v-1} : *Conjunto de trabajos programados en la versión S_{v-1} . Tiene asociado el índice i' . $i'=1,...,NX'$;*
- NC: *Conjunto de trabajos en proceso (procesados parcialmente) en el instante t en el Programa S_{v-1} . Los trabajos cuyo recurso no esté disponible (averías, etc.) no figurarán en este conjunto aunque hayan sido interrumpidos. Tiene asociado el índice h . $h=1,...,NCX$;*

Restricciones Tecnológicas

- $pu(i,j,r)$: *Tiempo de proceso unitario (por m^2) del trabajo i en la máquina j de la etapa r .*

Estado en el Programa anterior

- $cj(i)$: *Proporciona el valor del índice i' en el Programa S_{v-1} de un trabajo i del Programa actual S_v . Devuelve valor nulo si no existe dicho trabajo en el Programa anterior.*
- $\min(i)$: *Proporciona el valor de la primera operación pendiente de realizarse del trabajo i para el Programa S_v .*
- $yd(i',j,r)$: *Vale uno si el trabajo i' fue asignado a la máquina j en la etapa r del Programa S_{v-1} .*

$sd(i',r)$: Representa la fecha de inicio de las operaciones sobre el trabajo i' en el Programa S_{v-1} en la etapa r si dicho trabajo también forma parte del Programa S_v . Se considera que toma el valor 0 si el trabajo i' ya no forma parte del Programa S_v .

VARIABLES DE DECISIÓN

Binarias

$at(i)$: Relacionada con el avance de la fecha de inicio de la primera operación con respecto al Programa anterior.

$\left\{ \begin{array}{l} =1 \text{ Si el trabajo } i \text{ tiene una fecha de inicio de la primera} \\ \text{operación en el Programa } S_v \text{ anterior a la que tenía dicha} \\ \text{operación en el Programa } S_{v-1}. \\ =0 \text{ En otro caso} \end{array} \right.$

$af(i)$: Relacionada con el inicio de la primera operación en la Ventana de Avance.

$\left\{ \begin{array}{l} =1 \text{ Si el trabajo } i \text{ tiene una fecha de inicio de la primera} \\ \text{operación en el Programa } S_v \text{ anterior al instante } t \text{ más la} \\ \text{Ventana de Avance (VA)}. \\ =0 \text{ En otro caso} \end{array} \right.$

El Modelo Matemático es el siguiente:

$$[5.13] \text{ Minimizar } Z = \left\{ \max_i^{NX} \{c(i, RX)\} + \kappa \cdot \left(\sum_i^{NX} (at(i) \cdot af(i)) \right) \right\}$$

Como se puede ver en la Función Objetivo se considera que cada trabajo adelantado tiene que suponer una mejora del makespan igual o superior al valor de K . La expresión

$\left(\sum_i^{NX} (at(i) \cdot af(i)) \right)$ mide el número de trabajos que denominaremos “penalizables”.

Sujeto a:

$$[5.14] \quad \sum_{j=1}^{mx(e)} y(i, j, e) = 1 \quad \forall i, \forall e = \min(i) \dots RX$$

$$[5.15] \quad \sum_{i_1=0}^{NX} x(i_1, i_2, j, e) - y(i_2, j, e) = 0 \quad i_1 \neq i_2, \forall i_2, \forall j, \forall e = \min(i_2) \dots RX$$

$$[5.16] \quad \sum_{i_1=1}^{NX+1} x(i_2, i_1, j, e) - y(i_2, j, e) = 0 \quad i_1 \neq i_2, \forall i_2, \forall j, \forall e = \min(i_2) \dots RX$$

$$[5.17] \quad \sum_{i=1}^{NX+1} x(0, i, j, r) = 1 \quad \forall j, \forall r$$

$$[5.18] \quad \sum_{i=0}^{NX} x(i, NX+1, j, r) = 1 \quad \forall j, \forall r$$

[5.19]

$$s(i_2, e) \geq s(i_1, e) + \sum_{j=1}^{mx(e)} \sum_{i_3=0}^{NX} (\text{pu}(i_1, j, e) \cdot l(i_1) + (\text{st}(i_3, i_1, j, e) \cdot x(i_3, i_1, j, e))) + \Omega \cdot \left(\sum_{j=1}^{mx(e)} x(i_1, i_2, j, e) - 1 \right)$$

$i_1 \neq i_3, i_2 \neq i_3, \forall i_1 = 0 \dots NX,$
 $\forall i_2 = 1 \dots NX+1, \forall r$
 Ω es un número arbitrariamente grande

[5.20]

$$s(i_2, e) + \sum_{i_1=0}^{NX} \sum_{j=1}^{mx(e)} \{[\text{st}(i_1, i_2, j, e)] \cdot x(i_1, i_2, j, e)\} \geq$$

$$\geq s(i_2, e-1) + \left(\sum_{i_3=0}^{NX} \sum_{j=1}^{mx(e-1)} \{(\text{pu}(i_2, j, e-1) \cdot l(i_2)) + [\text{st}(i_3, i_2, j, e-1) \cdot x(i_3, i_2, j, e-1)]\} \right)$$

$i_1 \neq i_2, \forall i_2, \forall e > \min(i_2)$

[5.21]

$$c(i_2, RX) \geq s(i_2, RX) + \left(\sum_{i_1=0}^{NX} \sum_{j=1}^{mx(RX)} (pu(i_2, j, RX) \cdot l(i_2) + (st(i_1, i_2, j, RX) \cdot x(i_1, i_2, j, RX))) \right)$$

$i_1 \neq i_2, \forall i_2$

[5.22] $d(i) \geq c(i, RX)$

$\forall i$

[5.23] $s(i,1) \cdot at(i) < sd(cj(i),1)$

$\forall i$

[5.24] $s(i,1) \geq sd(cj(i),1)(1 - at(i))$

$\forall i$

[5.25] $s(i,1) \cdot af(i) < t + VA$

$\forall i$

[5.26] $s(i,1) \geq (t + VA) \cdot (1 - af(i))$

$\forall i$

[5.27] $s(i, e) \geq t$

$\forall i, \forall e = \min(i)..RX$

[5.28] $\sum_{j=1}^{mx(\min(h))} y(h, j, \min(h)) \cdot yd(cj(h), j, \min(h)) = 1$

$\forall h$

[5.29] $s(h, \min(h)) = t$

$\forall h$

5.4.4 Comentarios al Modelo Reactivo Determinista

El modelo permite equilibrar los aspectos relacionados con la tasa de productividad y la modificación del Programa Productivo en función del criterio del programador mediante el parámetro K . El modelo introduce una serie de restricciones que se pueden agrupar en 3 tipos:

- Como en el Modelo Predictivo existe una serie de restricciones relacionadas con la Asignación y la Secuencia:[5.14][5.15][5.16][5.17][5.18].
- Como en el Modelo Predictivo existe una serie de restricciones relacionadas con la Asignación y la Secuencia, en base a las fechas de inicio y fin de las operaciones:[5.19][5.20][5.21].
- Igual que en el Modelo Predictivo existe una restricción relacionada con la fecha de entrega:[5.22].
- Existe un conjunto de restricciones que permiten contabilizar el número de trabajos que han adelantado su primera operación y que están situados en la ventana t a $t+VA$.: [5.23][5.24][5.25][5.26].
- Se incluye un conjunto de restricciones que garantiza la continuidad entre el Programa anterior y el actual:[5.27][5.28][5.29].

A continuación se comentan aquellas Restricciones que suponen una diferencia con el Modelo Predictivo con cierto detalle.

Las restricciones ligadas a las ecuaciones [5.14][5.15] [5.16] [5.19][5.20] y [5.21] tienen el mismo significado que las ecuaciones [5.2][5.3][5.4][5.7][5.8] y [5.9] respectivamente del Modelo Predictivo determinista pero a diferencia de éste no se contemplan todas las etapas, sólo se tienen en cuenta aquellas en las que los trabajos tienen operaciones pendientes.

Las restricciones [5.17][5.18] y [5.22] son idénticas a las ecuaciones [5.5][5.6] y [5.10] respectivamente del Modelo Predictivo.

Las restricciones [5.23] y [5.24] fuerzan a que el valor de $at(i)$ sea uno si el trabajo i se inicia en el Programa Productivo S_v con anterioridad al su correspondiente trabajo $c_j(i)=i'$ en el Programa S_{v-1} . De la misma forma las restricciones [5.25] y [5.26] fuerzan a que $at(i)$ sea uno si el trabajo i del Programa Productivo S_v que se está calculando inicia su primera operación antes del instante $t+VA$. La combinación de las variables $a(i)*at(i)$ será uno cuando coincidan

ambas circunstancias simultáneamente. De esta forma se pueden contabilizar en la Función Objetivo los trabajos que cumplen ambas condiciones.

La restricción [5.27] fuerza a que la fecha de inicio de cualquier trabajo sea siempre posterior a la fecha de puesta en marcha del Programa Reactivo.

Las restricciones [5.28] y [5.29] se utilizan para forzar a que los trabajos en ejecución en el instante de la Reprogramación no sean modificados y pueden seguir ejecutándose en la misma máquina (recurso) en la que lo hacían en el Programa S_{v-1} . Así, la restricción [5.29] obliga a que los trabajos con operaciones en curso (interrumpidos) se inicien justo en el momento de iniciarse el Programa Reactivo. Y la restricción [5.28] fuerza a que todos los trabajos en curso (interrumpidos) continúen ejecutándose en la máquina donde se están realizando en el Programa anterior S_{v-1} .

Como se puede observar con facilidad el número de restricciones y variables ha aumentado respecto del Modelo Predictivo determinista por lo que todos los comentarios realizados respecto al cálculo de una solución óptima del Modelo Predictivo son válidos en caso del Modelo Reactivo.

5.4.5 Modelo con Incertidumbre

En los puntos 5.4.1 y 5.4.3 se han definido los Modelos Deterministas para el cálculo del Programa de Producción Predictivos y Reactivos respectivamente. Lo que implica que en dichos modelos se asume certidumbre en todos los aspectos del problema. En este apartado se van a formular Modelos de Programación Matemática Fuzzy que tratan de superar la rigidez del modelo determinista y adaptarse mejor a la realidad del problema.

Como ya se comentó en el punto 5.2.1.2.1 las fuentes de incertidumbre que se consideran son los Tiempos de Ejecución de las operaciones y los Tiempos de Cambio de partida dependientes de la Secuencia. Por lo tanto, la Incertidumbre afecta a las restricciones del Modelo Matemático. (Zimmermann, H. J. 1976) propuso la utilización de conjunto difusos para poder hacer más relajados, “borrosos”, los requerimientos estrictos de la Función Objetivo y de las restricciones. No se trataba nada más que de la utilización de los conceptos de borrosidad propuestos por (Bellman, R. & Zadeh, L. 1970) unos años antes.

En este caso se plantearán dos modelos, uno para el caso de la Programación Predictiva y otro para el caso de la Programación Reactiva. En ambos casos, la Función Objetivo será Determinista, y serán las restricciones las que incorporarán coeficientes fuzzy.

A continuación se introducirán las modificaciones oportunas a los modelos deterministas para la Programación Predictiva y Reactiva que permitirán disponer de la formulación fuzzy de dichos modelos.

5.4.5.1 Modelo Predictivo Fuzzy

A partir del modelo determinista para la Programación Predictiva se formula el modelo fuzzy sustituyendo las restricciones [5.7],[5.8] y [5.9] por las siguientes ecuaciones:

[5.30]

$$s(i_2, r) \geq s(i_1, r) + \sum_{j=1}^{mx(r)} \sum_{i_3=0}^{NX} \left(\bar{p}(i_1, j, r) + \left(\overline{st}(i_3, i_1, j, r) \cdot x(i_3, i_1, j, r) \right) \right) + \Omega \cdot \left(\sum_{j=1}^{mx(r)} x(i_1, i_2, j, r) - 1 \right)$$

$i_1 \neq i_3, i_2 \neq i_3, \forall i_1 = 0..NX,$
 $\forall i_2 = 1..NX+1, \forall r$
 Ω es un número arbitrariamente grande

[5.31]

$$s(i_2, r) + \sum_{i_1=0}^{NX} \sum_{j=1}^{mx(r)} \left\{ \overline{st}(i_1, i_2, j, r) \cdot x(i_1, i_2, j, r) \right\} \geq$$

$$\geq s(i_2, r-1) + \left(\sum_{i_3=0}^{NX} \sum_{j=1}^{mx(r-1)} \left\{ \bar{p}(i_2, j, r-1) + \left\langle \overline{st}(i_3, i_2, j, r-1) \cdot x(i_3, i_2, j, r-1) \right\rangle \right\} \right)$$

$i_1 \neq i_2, \forall i_2, \forall r > 1$

[5.32]

$$c(i_2, \mathbf{RX}) \geq s(i_2, \mathbf{RX}) + \left(\sum_{i_1=0}^{NX} \sum_{j=1}^{mx(RX)} \left\langle \bar{p}(i_2, j, \mathbf{RX}) + (\bar{st}(i_1, i_2, j, \mathbf{RX}) \cdot x(i_1, i_2, j, \mathbf{RX})) \right\rangle \right)$$

$$i_1 \neq i_2, \forall i_2$$

Los parámetros ciertos $p(i, j, r)$ y $st(i_1, i_2, j, r)$ del Modelo Predictivo determinista se han sustituido por los parámetros inciertos $\bar{p}(i, j, r)$ y $\bar{st}(i_1, i_2, j, r)$. Los parámetros inciertos indicados son Conjunto Fuzzy normalizados cuya función de pertenencia podría adoptar diferentes formas (triangular, trapezoidal, curva, etc.). Quizás la representación más simple sea la triangular. Esta representación asume un compromiso intermedio entre la simplicidad y el modelado de la realidad. En ese caso se podría definir de la siguiente forma:

[5.33]

$$f(x) = \begin{cases} (x-a)/(c-a) & \text{si } a \leq x \leq c \\ (b-x)/(b-c) & \text{si } c \leq x \leq b \\ 0 & \text{en otros casos.} \end{cases}$$

Donde x , a , b y c son números reales de tal forma que si $a=b=c$ no serían números fuzzy. En la Figura 5-6 se puede observar una representación gráfica de la variable difusa. En el caso que nos ocupa, quedaría por determinar los valores a , b , c para el Tiempo de Proceso Incierto y el Tiempo de Cambio de Partida Incierto. Obviamente, igual cada parámetro determinista es matricial, los parámetros difusos también los serán. En el caso, del Tiempo de Proceso se definirán unos valores a , b , y c para cada combinación de trabajo, etapa, y recursos, y en el caso de los Tiempos de Cambio de Partida se establecerán unos valores a , b , y c para cada combinación de trabajo inicial, trabajo siguiente, etapa y recurso.

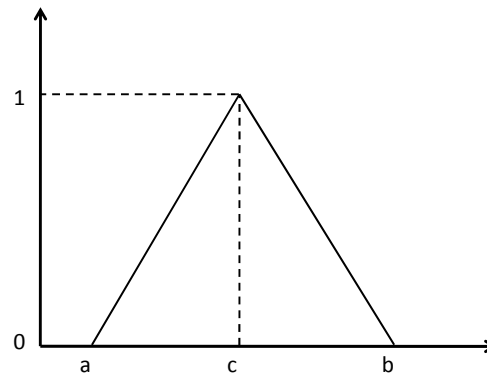


Figura 5-6. Función de pertenencia triangular de una variable x difusa (basado en (Gen, M., Tsujimura, Y., & Ida, K. 1992)).

5.4.5.2 Modelo Reactivo Fuzzy

A partir del modelo determinista para la Programación Reactiva se formula el modelo fuzzy sustituyendo las restricciones [5.19], [5.20] y [5.21] por las siguientes ecuaciones:

[5.34]

$$s(i_2, e) \geq s(i_1, e) + \sum_{j=1}^{mx(e)} \sum_{i_3=0}^{NX} (\overline{pu}(i_1, j, e) \cdot l(i_1) + \langle \overline{st}(i_3, i_1, j, e) \cdot x(i_3, i_1, j, e) \rangle) + \Omega \cdot \left(\sum_{j=1}^{mx(e)} x(i_1, i_2, j, e) - 1 \right)$$

$i_1 \neq i_3, i_2 \neq i_3, \forall i_1 = 0..NX,$
 $\forall i_2 = 1..NX+1, \forall r$
 Ω es un número arbitrariamente grande

[5.35]

$$s(i_2, e) + \sum_{i_1=0}^{NX} \sum_{j=1}^{mx(e)} \langle \overline{st}(i_1, i_2, j, e) \cdot x(i_1, i_2, j, e) \rangle \geq s(i_2, e-1) + \left(\sum_{i_3=0}^{NX} \sum_{j=1}^{mx(e-1)} \langle \overline{pu}(i_2, j, e-1) \cdot l(i_2) + \langle \overline{st}(i_3, i_2, j, e-1) \cdot x(i_3, i_2, j, e-1) \rangle \rangle \right)$$

$i_1 \neq i_2, \forall i_2, \forall e > \min(i_2)$

[5.36]

$$c(i_2, \text{RX}) \geq s(i_2, \text{RX}) + \left(\sum_{i_1=0}^{NX} \sum_{j=1}^{m \times (\text{RX})} \left(\overline{pu}(i_2, j, \text{RX}) \cdot l(i_2) + \overline{st}(i_1, i_2, j, \text{RX}) \cdot x(i_1, i_2, j, \text{RX}) \right) \right)$$

$$i_1 \neq i_2, \forall i_2$$

De la misma forma que el modelo anterior los parámetros ciertos $p(i, j, r)$ y $st(i_1, i_2, j, r)$ del Modelo Determinista Reactivo se han sustituido por los parámetros inciertos $\overline{p}(i, j, r)$ y $\overline{st}(i_1, i_2, j, r)$. Los parámetros inciertos serán Conjuntos Fuzzy Normalizados, definidos de la misma forma que se hizo para el caso de Programación Predictiva con Incertidumbre.

5.4.6 Comentarios al Modelo con Incertidumbre

La función objetivo y las restricciones desempeñan papeles diferentes, las restricciones definen el espacio de la solución de forma imprecisa o fuzzy, y la función objetivo induce el orden de alternativas de decisión. En (Mula, J. 2004) se indica que para este tipo de problemas pueden utilizarse dos posibles enfoques:

- La determinación de un conjunto difuso de la “decisión”.
- La determinación de una “decisión de maximización” precisa agregando la función objetivo después de la transformación apropiada de las restricciones. Se trata de encontrar un punto que satisfaga las restricciones y el objetivo con el grado máximo. Este concepto es adaptado a los problemas de Programación Matemática por (Tanaka, H., Okuda, T., & Asai, K. 1984).

En cualquiera de los casos, primero *se transforma cada problema fuzzy en un problema determinista equivalente*, que se denomina *Modelo Paramétrico*. Existen múltiples propuestas para el desarrollo de un Modelo Paramétrico a partir del modelado fuzzy. En (Mula, J. 2004; Peidro Payá, D. 2006) se realiza un profundo análisis y aplicación de algunas de las técnicas más relevantes al Modelado del Problema de Planificación.

En base a la propuesta de (Carlsson, C. & Korhonen, P. 1986) se va a realizar una aproximación teórica a la transformación de un modelo fuzzy en un modelo paramétrico para considerar cual es la complejidad que introduce sobre la propuesta determinista inicial. El trabajo de (Carlsson, C. et al. 1986) considera el siguiente Modelo Matemático:

$$[5.37] \quad \text{Max } \bar{z} = \bar{c} x$$

$$[5.38] \quad \text{Sujeto a } x \in X = \{x / -\bar{A}x \leq \bar{b}, x \geq 0\}$$

Los parámetros de este modelo están definidos mediante intervalos fuzzy de la forma $[c^0, c^1]$, $[A^0, A^1]$, $[b^0, b^1]$. El primer valor, o límite inferior, representa los valores seguros en el sentido de que pueden obtenerse posibles soluciones con alto grado de certidumbre. El segundo valor, o límite superior, representa los valores imposibles o irreales donde la solución obtenida puede no ser implementable. Desplazarse de valores seguros a valores imposibles supone pasar de soluciones con alto grado de implementabilidad a soluciones bajo grado de implementabilidad. El objetivo radica en obtener valores dentro de estos límites como una función de grados de imprecisión de parámetros (Carlsson, C. et al. 1986).

La precisión de una solución óptima, μ_s , se define mediante la intersección de las funciones de pertenencia relacionadas con los parámetros imprecisos definidos como μ_c , μ_A , μ_b .

$$[5.39] \quad \mu_s = (\mu_{c_j} \wedge \mu_{A_j} \wedge \mu_{b_j}) \quad j=1..n$$

La expresión significa que la precisión heredada en la solución óptima es igual a la precisión del parámetro más "arriesgado". Para el modelo [5.39] el mejor valor para la función objetivo para un nivel fijado de μ_s se alcanza cuando:

$$[5.40] \quad \mu_s = (\mu_{c_j} = \mu_{A_j} = \mu_{b_j}) \quad j=1..n$$

Aunque el modelo tiende a usar los valores más "arriesgados" de los parámetros, la expresión anterior significa que el mejor valor de la función objetivo, a un nivel dado de precisión, puede encontrarse utilizando valores para los parámetros en el mismo nivel de precisión (Carlsson, C. et al. 1986).

Asumiendo que los parámetros definidos como $p \in [p^0, p^1]$ disponen de funciones de pertenencia lineales monótonamente decrecientes:

$$[5.41] \quad \mu_p = \left[\frac{(p - p^1)}{(p^0 - p^1)} \right]$$

, donde $p \in [p^0, p^1]$ y $\mu_p = 1$ si $p < p^0$, $\mu_p = 0$ si $p > p^1$

El parámetro p se evalúa como $p = p^1 + \mu_p (p^0 - p^1)$. Usando esta estructura el modelo [5.39] quedaría:

$$[5.42] \quad \begin{aligned} \text{Max } z &= [c^1 + \mu(c^0 - c^1)]x \\ \text{sujeto a } & - [A^1 + \mu(A^0 - A^1)]x \leq b^1 + \mu(b^0 - b^1), \quad x \geq 0 \\ \text{donde } & \mu_c = \mu_A = \mu_b = \mu \end{aligned}$$

Se ha asumido en el modelo [5.42] que todas las funciones de pertenencia son idénticas y lineales para todos los parámetros. Por supuesto esto no tiene porque ser así, pudiendo disponer de combinaciones de funciones de pertenencia lineales, lineales a trozos (triangulares, trapezoidales, etc.), exponenciales, hiperbólicas que pueden adaptarse mejor a situaciones reales.

Una vez obtenido el Modelo Paramétrico quedaría por establecer un Conjunto Difuso de la decisión, y en todo caso posteriormente establecer el óptimo del conjunto. Para la determinación del Conjunto Difuso de la Solución, (Orlovsky, S. A. 1977) sugiere calcular, para todos los α -niveles del espacio de soluciones, los valores óptimos correspondientes de la función objetivo y considerar como el Conjunto Difuso de la solución los valores óptimos de la función objetivo, con el grado de pertenencia igual al α -nivel del espacio de solución. Es decir, se trata de establecer múltiples valores para μ .

En (Werners, B. 1987) se definen los conjuntos de α -niveles del espacio de soluciones como $R_\alpha = \{x / x \in X, \mu_R(x) \geq \alpha\}$ y el conjunto de soluciones óptimas para cada conjunto del α -nivel como $N(\alpha) = \{x / x \in R_\alpha, f(x) = \sup_{x \in R_\alpha} f(x')\}$. El Conjunto Difuso de la solución se define por la función de pertenencia:

$$[5.43] \quad \mu_{\text{OPT}}(x) = \begin{cases} \sup \alpha & \text{si } x \in N_{\alpha > 0}(\alpha) \\ 0 & \text{en otros casos.} \end{cases}$$

La función de pertenencia del Conjunto Difuso de *los valores óptimos de la función objetivo* se define como:

$$[5.44] \quad \mu_i(r) = \begin{cases} \sup \mu_{\text{OPT}}(x) & \text{si } r \in R \wedge f^{-1}(r) \neq \emptyset \\ 0 & \text{en otros casos.} \end{cases}$$

donde $f(x)$ es la función objetivo con valores funcionales r . En el caso de la programación lineal, la determinación de los valores r y $\mu_{\text{OPT}}(x)$ se puede obtener por *programación paramétrica*.

Por lo tanto, el cálculo de las soluciones para un *modelo fuzzy* dado implican la resolución de múltiples Modelos Deterministas asociados, tantos como niveles se consideren. Los modelos deterministas asociados un *modelo fuzzy*, denominados Modelos Paramétricos, son de complejidad similar al Modelo Determinista, y por lo tanto el tiempo de cálculo es de un orden de magnitud similar. Lo que hace que las técnicas optimizadoras aplicada a este tipo de modelos aumenten la magnitud de la complejidad de cálculo de un Problema ya de por sí difícil de resolver óptimamente en su versión Determinista. **La incorporación de la “borrosidad” en el problema nos empuja con mayor fuerza hacia la búsqueda de métodos de resolución alternativos a los optimizadores.**

5.4.7 Consideraciones sobre la Medición de la Gestión del Programa

Se ha considerado, y lo expuesto hasta este apartado así lo confirma, que un Programa Productivo estará bien gestionado si se aprovecha la ocasión originada por algún evento para situar la ocupación del taller lo más próxima posible a la Capacidad Disponible en ese instante. Para alcanzar este propósito es necesario adaptar en cada ocasión el Programa Productivo en curso, aplicando el algoritmo de reprogramación incluido en un procedimiento, que menor número de trabajos penalizables introduzca (ver la función objetivo del modelo de Programación Reactiva).

Parece razonable plantear que contra mayor sea el número de cambios que se produzcan, más fácilmente se podrá reducir la distancia entre Capacidad Disponible y Ocupación del taller, pero más probablemente se incrementaran las alternaciones no deseadas, es decir los trabajos penalizables. Por lo tanto, se trata de equilibrar ambas situaciones para lo cual se deben

formular una serie de indicadores clave que permitan guiar al programador en la *Gestión del Programa*.

Se ha establecido que la variable fundamental en la medida de la Gestión del Programa sea la Tasa de Productividad Promedio alcanzada durante un periodo, es decir, del número de m^2 producidos por semana, por mes, o por el periodo que se elija en cada momento. Se debe subrayar que el objetivo se aplica en un entorno decisional muy concreto, y que en ningún momento se pretende que las empresas busquen producir el máximo posible a toda costa sin relacionar esta dato con la demanda existente. En el entorno del Programador se entiende que los recursos disponibles han sido previamente ajustados de forma aproximada a las capacidades que en cada momento se necesitan por decisiones de tipo táctico, por ejemplo cerrando alguna línea si las necesidades de una serie de periodos se reducen. Por lo tanto, la tasa de productividad se debe entender como un concepto relacionado con el máximo aprovechamiento de los recursos existentes.

Dos formas de gestionar diferentes establecerán combinaciones cálculo de los Programas Predictivos y Reactivos diferentes, lo que supondrá una Tasa de Productividad promedio diferente. El concepto de Tasa de Productividad promedio como elemento comparativo entre diferentes formas de gestionar un Programa sólo podrá considerarse válido cuando éstos están sometidos a las mismas condiciones. Es obvio, que ante situaciones productivas diferentes como serían diversos grados de incidencias (algunos Programas sometidos a fuertes adelantos y/o retrasos en las fechas de finalización de las operaciones y otros sin incidencias, por ejemplo), o diversos grados de carga en las instalaciones motivados por demandas elevadas en algunos casos, y reducidas para otros Programas Productivos (esta situación puede suponer que en algún caso sea imposible adelantar trabajos por no existir ninguno disponible por baja demanda, por ejemplo) no sería coherente utilizar la Tasa de Productividad promedio como elemento comparativo.

Dentro de la variable Tasa de Productividad promedio se embeben los efectos, tanto de la Programación Predictiva, como de la Reactiva, ya que el número de m^2 final producido depende de la forma de abordar ambas situaciones. No obstante, puede ser interesante disponer de datos que informen sobre la tasa de productividad inicial debida a la aplicación de la correspondiente técnica de Programación Predictiva, también de datos que informen sobre el incremento o decremento de la tasa de productividad durante la ejecución del Programa Productivo. Esta última información puede ayudar a valorar como influye la forma de afrontar la Reprogramación en diversas circunstancias.

Además de la Tasa de Productividad promedio se considera que es necesario incluir algún indicador adicional que mida el esfuerzo que implica alcanzar una determinada Tasa de Productividad promedio, y el margen de mejora posible.

Dado que no se ha encontrado ningún medible definido para el entorno de la industria cerámica en relación al problema planteado en la propuesta de solución del problema se planteará aquel o aquellos que se crea que se ajustan más a las consideraciones realizadas en este apartado.

5.4.8 Utilidad de los Modelos Matemáticos

Por lo que hace referencia al proceso de cálculo de los Programas Productivos, tal y como se planteó desde un principio la elaboración de los Modelos Matemáticos ha aportado una comprensión bastante detallada de cada uno de los problemas planteados inicialmente. Desafortunadamente, en ninguno de los casos analizados se pueden obtener resultados de forma ágil mediante técnicas de resolución optimizadoras, y menos si se tiene en cuenta la Incertidumbre. La resolución de cada modelo tiene un coste computacional elevado utilizando ordenadores con unas características habituales en la oferta comercial.

Dado que el cálculo de un Programa Predictivo o Reactivo no es más que una parte del procedimiento, y que dicha actividad puede llegar a repetirse en varias ocasiones antes de validar un Programa, hay que destacar que el coste computacional global del procedimiento de Establecimiento o Modificación de un Programa Productivo puede llegar a ser mucho más elevado del que costaría el, ya de por sí elevado, Cálculo del Programa.

La solución a los problemas de cálculo, así como al problema global, se abordará en la propuesta de resolución. No obstante, ya se avanza que ésta considerará la exploración de técnicas no basadas en la optimización, más próximas a la visión metaheurística, que fundamentalmente permitan conjugar el problema de gestión y el de cálculo con unos resultados al menos razonables.

5.5 Conclusiones

En este capítulo se han identificado un problema de Programación de la Producción en el marco de la Industria Cerámica que en lo sucesivo será tratado con el fin de encontrar una solución adecuada al mismo.

Desde un primer momento se detecto la complejidad que el mismo planteaba y la necesidad de realizar diversos modelos que permitieran su completa definición. Se ha realizado una aproximación progresiva al problema utilizando en primer lugar el **lenguaje natural** o literario para establecer las bases del mismo. Y la posterior introducción del **Modelado Matemático** para definir con precisión aquellos aspectos que así lo requerían, ha permitido disponer de una visión del problema con el nivel de detalle adecuado en cada momento.

La descripción del problema (apartado 5.2) y el Modelado Matemático de los procesos de cálculo (apartado 5.4) han permitido entender el problema y han proporcionado al mismo tiempo una base importante para poder plantear soluciones al mismo.

Desde un análisis sintético del problema expuesto se puede concretar que se han identificado dos grandes dificultades:

- La primera de ellas estaría asociada a lo que se ha denominado **“Gestión del Programa Productivo”**.
- la segunda de ellas estaría relacionada con las capacidades de obtención de los **Programas Predictivo y Reactivo** en sí mismos.

La **“Gestión del Programa Productivo”** se ha planteado al considerar la necesidad de realizar cambios en el Programa en curso si se dan determinadas condiciones. Este proceso se implica la necesidad de **realizar un seguimiento continuo del Estado del Programa**, Valorar diversas **situaciones, y Tomar Decisiones sobre la posibilidad de realizar ciertos cambios**, y todo ello sin perder de vista el objetivo asociado a la Eficiencia del Proceso. Se trata por lo tanto de un **proceso complejo y difícil de realizar con éxito**, ya que éste no depende exclusivamente de la propia Gestión sino también de los eventos que vayan produciéndose. Los procesos de cálculo de los Programas Predictivo y Reactivo son funcionalidades que son demandadas por el proceso de Gestión del Programa, y que en este sentido dependen de él.

En la bibliografía revisada no se han encontrado muchos ejemplos en los que se plantee la resolución de problema en los que se combinen aspectos cualitativos y cuantitativos. La mayoría de los problemas de Programación de la Producción se enfocan con una visión cuantitativa, lo que implica la existencia de un amplio margen para la investigación. Además aunque en la literatura se han encontrado múltiples problemas enfocados como Programación Predictiva, y algunos, aunque muchos menos, con un enfoque de Programación Reactiva, **es casi imposible encontrar casos en los que ambos se hayan combinado en el mismo problema el enfoque Predictivo y Reactivo**. Por lo tanto, el problema identificado es novedoso tanto por la doble combinación, en el **ámbito de la Toma de Decisiones** (gestión y cálculo) y en el

ámbito de los tipos de Problemas de cálculo de la Programación de la Producción detectados, y además se enmarca en un entorno en el que no se conocen planteamientos similares: La industria cerámica.

En el próximo capítulo se realizará una propuesta para completar la definición del problema y abordar la solución al caso que se ha planteado. La propuesta contendrá la Metodología, Arquitectura y Herramientas que se proponen emplear, y con ella se dispondrá de una visión estructurada del trabajo que queda pendiente de realizar.

5.6 Bibliografía

Alfaro, J. J. 2003. *Sistema de Medición del Rendimiento en el Marco de la Propuesta IE-GIP. Aplicación a una PYME. Universidad Politécnica de Valencia.*

Andrés, C. 2001. *Programación de la Producción en Talleres de Flujo Híbridos con Tiempos de Cambio de Partida dependientes de la secuencia. Modelos, Métodos y Algoritmos de Resolución. Aplicación a Empresas del Sector Cerámico. Universidad Politécnica de Valencia.*

Bellman, R. & Zadeh, L. 1970. *Decision making in a fuzzy environment. Management Science, 17(4): 141-164.*

Carlsson, C. & Korhonen, P. 1986. *A parametric approach to fuzzy linear programming. Fuzzy Sets and Systems, 20(1): 17-30.*

Conway, R. W., Maxwell, W. L., & Miller, L. W. 1967. *Theory of Scheduling. Addison-Wesley Publishing Company.*

Gen, M., Tsujimura, Y., & Ida, K. 1992. *Method for Solving Multiobjective Aggregate Production Planning Problem with Fuzzy Parameters. Computers & Industrial Engineering, 23(1-4): 117-120.*

Mula, J. 2004. *Modelos para la planificación de la producción bajo incertidumbre. Aplicación en una empresa del sector del automóvil. Departamento de Organización de Empresas, Economía Financiera y Contabilidad. Universidad Politécnica de Valencia.*

Orlovsky, S. A. 1977. *On programming with fuzzy constraint sets. Kybernetes, 6: 197-201.*

Peidro Payá, D. 2006. *Modelos para la planificación táctica centralizada de una cadena de suministro bajo incertidumbre. Aplicación en una cadena de suministro del sector del automóvil. Universidad Politécnica de Valencia.*

Rodríguez, R. 2007. *Desarrollo de una Metodología y Arquitectura para la Identificación y Análisis de Relaciones entre Indicadores de Rendimiento de una Organización. Aplicación a una Empresa del Sector Cerámico. Universidad Politécnica de Valencia.*

Tanaka, H., Okuda, T., & Asai, K. 1984. On fuzzy mathematical programming. Journal of Cybernetics, 3: 37-46.

Vieira, G. E., Herrmann, J. W., & Lin, E. 2003. Rescheduling manufacturing systems: A framework of strategies, policies, and methods. Journal of Scheduling, 6(1): 39-62.

Werners, B. 1987. An interactive fuzzy programming system. Fuzzy Sets and Systems, 23: 131-147.

Zimmermann, H. J. 1976. Description and optimization of fuzzy systems. International Journal General System, 2: 209-215.

**Enfoque Empleado para la Elaboración
de la Propuesta**

Capítulo

6

Tabla de contenido

6.	Capítulo 6	339
6.1	Introducción	339
6.2	Esbozo de la Propuesta	341
6.3	Metodología para la Elaboración/Ejecución de la Propuesta	343
6.4	Arquitectura	350
6.5	Herramientas.....	354
6.6	Visión Global del enfoque M.A.T.....	357
6.7	Conclusiones	359
6.8	Bibliografía	360

Índice de figuras

Figura 6-1. Representación gráfica de la combinación de elementos de la aproximación M.A.T. (elaborada en la descripción del trabajo del proyecto ECOSELL).....	340
Figura 6-2. Relación entre la estructura del problema y la estructura de la propuesta de solución (elaboración propia).....	342
Figura 6-3. Fases de la metodología propuesta (elaboración propia).	344
Figura 6-4. Agrupación de los elementos constructivos empleados (elaboración propia) para la propuesta de la Plataforma.	351
Figura 6-5. Relación de herramientas empleadas y generadas en la propuesta de solución (elaboración propia).	355

Índice de Tablas

Tabla 6-1. Relación entre los 3 tipos de elementos del enfoque M.A.T. (elaboración propia).	358
---	-----

6.1 Introducción

Con este capítulo se inicia la segunda fase de la tesis, en la cual, se elaborará una propuesta que tiene como fin dar cobertura al problema identificado en los capítulos anteriores. Como se ha podido contrastar **el caso que se propone resolver presenta una gran complejidad tanto por los niveles decisionales a los que afecta, como por la variedad de problemas de cálculo que se han identificado**. Por este motivo se ha considerado necesario incluir una descripción del camino que se va a seguir para afrontar la fase de resolución que conformará la propuesta.

Inicialmente se consideró simplemente la necesidad de formalizar el conjunto de pasos que se pretende realizar y de resultados que se espera obtener. Como consecuencia de dicho planteamiento se abordaron aspectos metodológicos. Sin embargo, casi de inmediato, **se detectó que dicha formalización de una metodología no era suficiente para alcanzar el objetivo del capítulo**. Por ello, se tomó la decisión de que la formalización que se necesitaba se fundamentará en la aplicación de una aproximación que ya había dado buenos resultados en diferentes proyectos (V-CHAINⁱ, ECOSELLⁱⁱ). Se trata de la aproximación M.A.T., cuyo acrónimo procede precisamente de la combinación de los 3 elementos que la caracterizan: **M**etodología, **A**rquitectura y **T**ecnología/Herramientas. Debido a su amplio enfoque M.A.T. cubre mejor las necesidades identificadas.

La aproximación M.A.T., cuya representación gráfica se muestra en la Figura 6-1, se origina inicialmente durante la elaboración de la propuesta IE-GIP en el ámbito de **la Integración Empresarial** contenida en (Ortiz Bas, A. 1998). En este ámbito se identificó como era altamente recomendable contar con los 3 elementos ya indicados. La aportación de cada uno de los elementos, adaptados a las necesidades de esta tesis, debe ser la siguiente:

- **Metodología**: Explicita los pasos para abordar y desarrollar de una forma adecuada la resolución del caso. Presenta una descripción precisa de las tareas, establece el detalle y la cantidad de información necesaria, etc.

ⁱ El proyecto V-CHAIN (Virtual Enterprise for Supply Chain Management), GRD1-2000-25881, fue financiado por la Comisión Europea dentro del V Programa Marco, y ejecutado 2000 a 2003.

ⁱⁱ El proyecto ECOSELL (Extended Collaborative Selling Chain), GRD1-2001-40692, fue financiado por la Comisión Europea dentro del V Programa Marco, y ejecutado 2003 a 2005.

- Arquitectura: Proporciona los elementos conceptuales que permiten la estructuración de la propuesta, mostrando la interrelación de todas las partes y/o funciones del procedimiento seguido.
- Tecnologías/Herramientas: Dan soporte al desarrollo de los conceptos arquitectónicos y metodológicos de la propuesta.

El enfoque M.A.T. proporciona una estructura cuya ventaja fundamental es la de hacer reflexionar al constructor de la propuesta sobre cual es la **interacción de los diferentes elementos entre sí** para cada caso concreto. La mayoría de los enfoques abarcan exclusivamente la parte metodológica, pero esa visión parcial es insuficiente porque no permite al lector disponer desde el primer momento de la información relativa a los componentes (bloques constructivos) sobre los que se asienta el desarrollo de la metodología. Además esa visión parcial impide conocer los instrumentos, normalmente tecnológicos, que se deben aplicar. La ausencia de estos elementos en el caso de la propuesta que se plantea en los sucesivos capítulos impediría la comprensión de la misma.

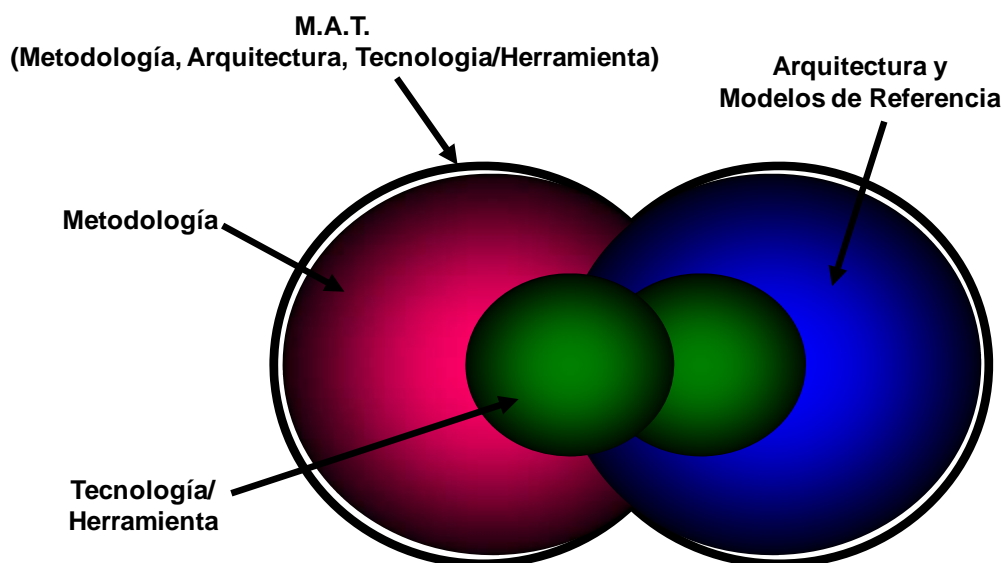


Figura 6-1. Representación gráfica de la combinación de elementos de la aproximación M.A.T. (elaborada en la descripción del trabajo del proyecto ECOSELL).

Se debe subrayar que el enfoque M.A.T. va a ser empleado para abordar la resolución de un problema concreto previamente identificado, y que en este sentido, **ni la metodología, ni la arquitectura, ni las herramientas que se desarrollen están pensadas con carácter general**. En otras palabras no se pretende alcanzar una propuesta para la resolución del problema de Programación Predictivo-Reactivo de la Producción en general, ya que no ha realizado un

análisis de los tipos de subproblemas que se pueden encontrar, ni tampoco se ha validado contra diferentes configuraciones. Sólo se trata de la estructuración de un conjunto de elementos metodológicos, arquitectónicos y tecnológicos que han resultado válidos para el caso identificado en el capítulo 5.

El resto del capítulo está organizado de la siguiente forma: En primer lugar, en el apartado 6.2, se realizará un sencillo esbozo de la propuesta que facilite la comprensión del enfoque M.A.T. En el resto de apartados se realizará una descripción de cada uno de los 3 elementos por separado, lo que se corresponde con los apartados 6.3, 6.4, y 6.5. Una vez descritos uno a uno los 3 elementos, en el apartado 6.6, se analiza la relación existente entre los ellos y su posterior desarrollo. Y finalmente en el apartado 6.7 se realiza un resumen del capítulo a modo de conclusión del mismo.

6.2 Esbozo de la Propuesta

Además de conocer como se construirá la propuesta es importante transmitir cual será la forma que ésta puede llegar a tomar. Se trata simplemente de un sencillo esbozo que permita entender el porque de algunos puntos que pueden llegar a proponer.

Hasta el momento sólo se ha identificado el problema que se desea abordar, ahora bien la propuesta de solución puede tomar muchas formas, y en función de la que se desee dar a ésta, la combinación de metodología, elementos arquitectónicos y tecnología debe ser diferente.

En el capítulo 5 se determinó la estructura del problema identificado, consistiendo este en un “nivel de gestión” que proporcionaría los resultados finalistas después de un proceso de balanceo de diversas opciones, y un “nivel de aspectos cuantitativos” ligados al cálculo de Programas Productivos Predictivos o Reactivos que alimentan al nivel de gestión. En base a esta visión se ha decidido establecer un gran continente que denominaremos “Plataforma” que estará asociada al nivel de gestión, y también existirán, ubicados dentro de la Plataforma, una serie de algoritmos para el cálculo de los Programas Productivos. Esta asociación se representa gráficamente en la Figura 6-2.

Para alcanzar un resultado adecuado durante el proceso de creación de la propuesta se manejarán diferentes conceptos que darán lugar a algoritmos, métodos y modelos capaces de mostrar como alcanzar soluciones para el problema identificado. Un paso más allá sería la implementación del conjunto de elementos que formen parte de la propuesta de manera que se pudiera automatizar el proceso de obtención del resultado. Este aspecto, la automatización, marcará la forma de elaborar o construir la propuesta. Por este motivo a la Plataforma se le

debería denominar de forma más adecuada “Plataforma Software”, ya que finalmente se podrá disponer de un elemento software. Como se verá posteriormente la metodología, la arquitectura y las herramientas que se van a emplear están muy relacionadas con el desarrollo de software.

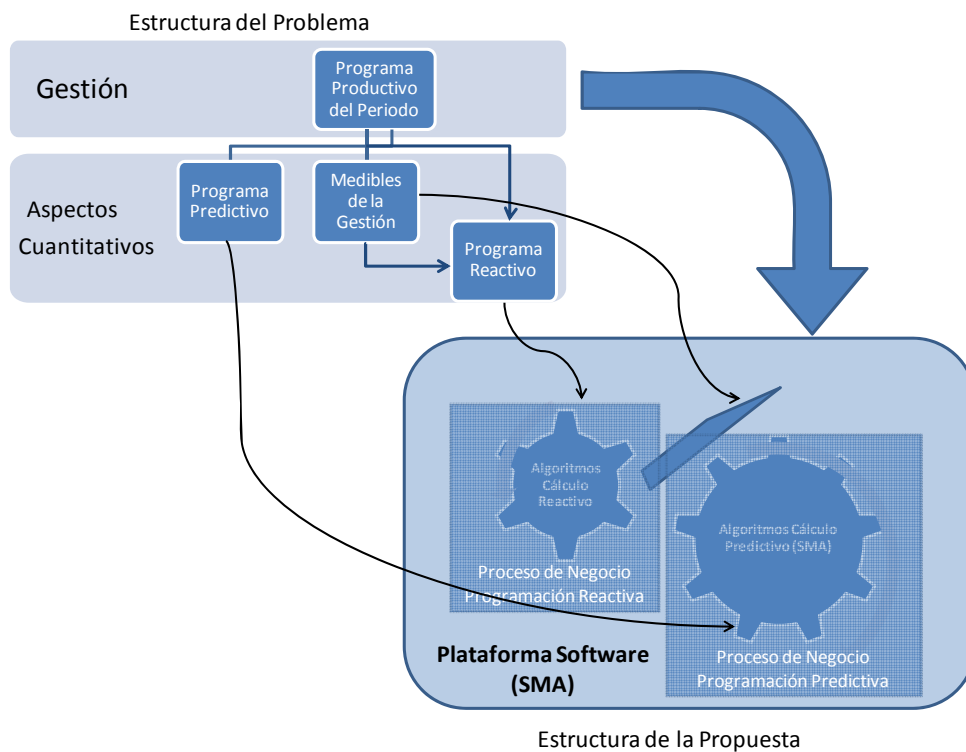


Figura 6-2. Relación entre la estructura del problema y la estructura de la propuesta de solución (elaboración propia).

Desde el punto de vista tecnológico, y en relación con la mencionada plataforma software, hay que señalar que su diseño e implementación estará basado en el paradigma de Sistemas Multiagentes. Ello significa que una vez se identifique como se desea alcanzar la solución al problema, en el proceso de diseño de la Plataforma, se seguirá un enfoque en el que se considerará la existencia de un conjunto de Agentes. Cada uno de los Agentes dispondrá de un rol determinado a través del cual deberán relacionarse entre ellos para ejecutar una propuesta que permita gestionar adecuadamente los Programas Productivos a lo largo de un periodo. El mencionado Sistema Multiagente conformará la estructura de la Plataforma Software, y dará cobertura al “nivel de gestión”. Los Sistemas Multiagente emplearán ciertas habilidades para alcanzar sus objetivos, algunas de estas habilidades no son más que procedimientos o funciones con los que éstos obtienen resultados. Entre éstos se encuentran los procedimientos de cálculo de los Programas Productivos Predictivos y Reactivos, que se encuentran ubicados en el “nivel de aspectos cuantitativos”.

No hay nada que impida que exista un Sistema Multiagente dentro de otro, trabajando en niveles diferentes. Pues bien, así es como se ha ideado la propuesta que se pretende desarrollar. La plataforma será diseñada e implementada como un Sistema Multiagente, y el procedimiento de cálculo del Programa Productivo Predictivo se diseñará e implementará con otro Sistema Multiagente. El procedimiento de cálculo del Programa Productivo Reactivo se diseñará e implementará mediante el procedimiento de desarrollo de software tradicional, en este caso orientado a objetos. Por este motivo en la Figura 6-2 se ha incluido el acrónimo de los Sistema Multiagente, SMA, en los dos elementos señalados.

Por último, es conveniente remarcar que aunque se ha desarrollado una metodología general, para alcanzar el objetivo final se emplearán metodologías complementarias, incluidas en la primera con el fin de realizar tareas específicas. Entre estas se destaca la metodología empleada para el desarrollo de experimentos, entre otras que posteriormente se indicarán.

A partir del próximo apartado de desarrollan sucesivamente los 3 elementos básicos del enfoque M.A.T.

6.3 Metodología para la Elaboración/Ejecución de la Propuesta

Dentro del elemento metodológico se propone la utilización de 6 grandes pasos, tal y como se puede ver en la Figura 6-3.

La metodología parte de la hipótesis de que el problema general ya ha sido identificado, de una manera similar al trabajo realizado en el capítulo 5. Además la metodología propuesta parte de la idea de que debe dar cobertura al desarrollo de una solución con un soporte tecnológico elevado. En concreto se ha asumido como reto alcanzar una solución que sea fácilmente utilizable en un entorno real, y en consecuencia como parte de los resultados esperados se debe elaborar una herramienta software. Este no es el único resultado que se espera obtener, pero si es uno de los que más influye en las fases establecidas.

La metodología considera tanto la construcción de una propuesta como la evaluación de la misma.

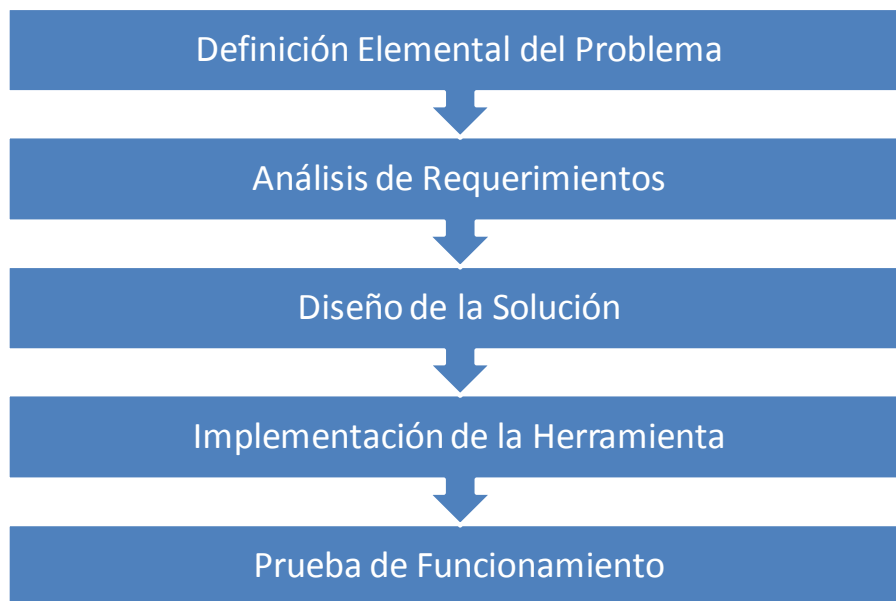


Figura 6-3. Fases de la metodología propuesta (elaboración propia).

Para cada una de las fases se han establecido hasta 4 aspectos que la definen. Un “objetivo” que establece el fin de la misma, un “desarrollo” en el que se detallan las tareas de que consta, unos “resultados” que se espera obtener en dicha fase, y en su caso, unas “restricciones” en el desarrollo de la fase correspondiente. A continuación se detallan todas las fases:

1. Definición Elemental del Problema

Bajo la consideración de que cada área de la empresa ofrece un servicio a las demás y tiene unos objetivos de funcionamiento de la misma alineados con el resto de áreas, se trata de identificar los ítems que caracterizan dicho servicio y su relación con el resto de forma genérica.

a. *Objetivo*: El objetivo es establecer cuales son los input que el “servicio de Programación de la Producción” analizado recibe de otras áreas, establecer de la misma forma cuales son los outputs que ofrece, e identificar cual es el criterio que mide la mejora. También se consideran dentro del objetivo la identificación de los “procesos” de que consta el servicio para alcanzar sus objetivos a nivel macro.

b. *Desarrollo*: Dado que se trata de un nivel macro se trata de:

- i. Identificar el objetivo del programador: Este se obtendrá a partir del problema identificado. En todo caso tiene que responder al interés de su área y del sistema productivo en conjunto.
 - ii. Identificar los “macro procesos” que el programador deberá lanzar para alcanzar su objetivo:
 - iii. Identificar “que” recursos deben ser aportados por otras áreas para que los “macro procesos” puedan operar: Estos recursos pueden ser físicos o de información.
 - iv. Identificar “cuando” se debe disponer de los recursos necesarios previamente identificados.
 - v. Revisar la coherencia del conjunto de elementos identificados y plasmarlos en un documento (escrito y gráfico).
- c. *Resultados*: Una Visión General de la problemática y análisis sobre procesos de toma de decisiones que condicionen los requerimientos.

2. Análisis Requerimientos

Una vez se ha realizado una aproximación “macro” al entorno del problema, se trata de definir detalladamente como debe de ser el “servicio” definido en el paso anterior. Este servicio quedará identificado en este paso como un “Dominio” que interactúa con otros “Dominios” con determinado objetivo.

En esta fase ya se puede intuir que existe una cierta ordenación de la misma con el objetivo de utilizar los resultados en la posterior elaboración de una herramienta software. No obstante, no se debe pensar de que por el hecho de que exista una orientación en el enfoque, que posteriormente permita aprovechar los resultados en la construcción de una herramienta, esto es lo único que se desea. La consecuencia más relevante de esta fase debe ser la obtención de una “imagen” de lo que se desea obtener en el futuro.

- a. *Objetivo*: Identificar los elementos organizativos, los recursos (físicos e información), y las tareas que forman parte o interactúan en los “macro procesos” identificados en el área de Programación de la Producción en el

paso anterior. Y determinar cuales son las restricciones que éstos imponen a un futuro Sistema de Programación de la Producción.

b. *Desarrollo:*

- i. Identificar el conjunto de requerimientos del proceso completo de Programación de la Producción para el caso identificado. Para ello se aplicará la metodología CIMOSAⁱⁱⁱ (ESPRIT Consortium AMICE 1993;Kosanke, K. 1995;M.Zelm, F.Vernadat, & K.Kosanke 1995).
 - ii. Identificar los requerimientos en el nivel correspondiente a los problemas de cálculo del Programa Productivo para definirlo. Aplicar Modelado Matemático, que puede ser la confirmación o la modificación de los modelos ya existentes, si los hay.
- c. *Resultados:* Por un lado, se obtendrá como mínimo una descripción de lo que en terminología CIMOSA es el “Dominio”, los “Procesos de Dominio”, los “procesos de Negocio”, y las “Actividades Empresariales”, las “Vistas de Objetos”, los “Objetos de Empresa”, y los “Conjuntos de Capacidades”. Todo ello expresado con el formato y contenido especificado en la metodología CIMOSA. Por otro lado, se obtendrá un modelo matemático para el cálculo del Programa Productivo Predictivo y Reactivo.
- d. *Restricción:* Dado que se ha identificado que el problema que se aborda está formado por una parte cualitativa asociada a la “Gestión”, y otra cuantitativa asociada al “Calculo del Programa” Predictivo o Reactivo, se ha considerado que la parte correspondiente al cálculo debe quedar encapsulada dentro de lo que CIMOSA denomina “Actividad Empresarial”.

La consecuencia inmediata de la restricción impuesta en esta fase es que los aspectos relacionados con la “gestión” de la Programación de la Producción en función del problema que se desea resolver toma un papel protagonista en el análisis de requerimientos. Los cálculos quedan en un segundo plano de momento.

ⁱⁱⁱ La metodología CIMOSA será ampliamente comentada en el próximo capítulo.

3. Diseño de la Solución

Cuando se alcance esta fase ya se habrá definido con detalle que es lo que se espera obtener, los requisitos previstos en la solución. Ahora falta por dar forma a la estructura que permitirá dar solución al problema respetando las condiciones previamente determinadas.

a. *Objetivo:* En este caso, se ha decidido que “dar solución” debe ser entendido por “como un modo de configurar Programas Productivos de forma automatizada con los requisitos previamente establecidos”. Por lo tanto, no sólo se busca construir el proceso decisional y/o de cálculo que corresponda sino que este se debe proporcionar como parte de una herramienta que permita su rápida, y en cierto modo, sencilla ejecución.

b. *Desarrollo:*

i. Planear y proyectar el un modo de solucionar de una forma automatizable la Gestión del Programa de Producción que permita establecer el marco adecuado para la coordinación entre la Programación Predictiva y Reactiva a lo largo del periodo, satisfaciendo los requisitos establecidos previamente. Para ello se aplica la metodología INGENIAS^{iv} (Pavon, J. & Gomez-Sanz, J. 2003).

ii. Construir los procesos detalladamente. Se incluye en esta fase cualquier procedimiento que en el paso anterior se indicará en un nivel macro. En este paso se deben elaborar Algoritmos y/o Métodos para el cálculo de Programas Productivos Predictivos o Reactivos. Para lo cual hay que realizar los siguientes pasos:

1. Selección o Creación de uno o varios Algoritmos.

^{iv} La metodología INGENIAS será ampliamente comentada en el próximo capítulo.

2. Diseño de Experimentos^v para analizar el comportamiento de los algoritmos propuestos.

3. Establecimiento de las Prestaciones Esperadas.

c. *Resultados*: Se espera obtener tres tipos de resultados:

i. Los modelos que definen la plataforma (varios modelos basados en los 5 meta-modelos de la metodología INGENIAS) que definan el sistema que soporte la Gestión del Programa Productivo.

ii. Procedimientos. Que pueden ser de tipo funcional o que como en el caso de los procedimientos de cálculo del Programa Predictivo o Reactivo deben incluir: Los Algoritmos y la Parametrización recomendada en función de las condiciones de empleo de los mismos.

iii. El análisis comparativo de los Algoritmos de Cálculo: Con el objeto de garantizar la calidad del proceso de cálculo se realizará un análisis comparativo de los algoritmos empleados,

4. Implementación de la Solución

a. *Objetivo*: Transformar el diseño conceptual en un producto software al que denominaremos Plataforma, que pueda ser ejecutado en una empresa para alcanzar los resultados esperados.

b. *Desarrollo*: Básicamente consiste en la codificación en un lenguaje de programación adecuado del sistema informático diseñado. Se han definido 2 pasos:

i. Programación de la Plataforma como Demostrador.

^v El Diseño de Experimentos puede ser considerado como una metodología que será comentada en los capítulos en los que haga uso de ella.

ii. Evaluación de la Plataforma.

c. *Resultados*: La Plataforma Software.

5. Prueba de Funcionamiento

Una vez se ha alcanzado esta fase se dispondrá de un sistema, la plataforma creada y ejecutable, que será capaz de obtener soluciones en lo que se ha denominado entorno de Demostración. En este entorno aún se necesita concretar y afinar algunos aspectos para poderse emplear como prototipo. Este cambio de demostrador a prototipo implica la elaboración de algunos procedimientos ajustados a unas necesidades concretas (tratamiento de datos de entras/salida, parametrización, etc.). En esta fase se ha planteado como una prueba sobre un caso concreto para verificar la eficacia de los planteamientos propuestos (conceptuales, algorítmicos y software) y analizar la eficiencia de los mismos.

a. *Objetivo*: Realizar un análisis comparativo entre el sistema actual de una empresa representativa del entorno estudiado y el propuesto en la resolución del problema.

b. *Desarrollo*:

i. Preparar la Simulación del Sistema en un entorno real: Preparar datos reales e identificar el sistema productivo particular y sus condiciones de trabajo.

ii. Dar valor a los parámetros de la Plataforma: Se necesita restringir y dirigir su forma de proceder mediante un conjunto de parámetros que habrán sido identificados en la fase de diseño que ajuste su funcionamiento a una instancia particular del problema.

iii. Diseñar Experimentos que permitan analizar la eficiencia de la propuesta en el entorno elegido.

iv. Establecimiento de las Prestaciones Esperadas.

c. *Resultados*: Plataforma particularizada y Análisis comparativo entre la solución propuesta y la solución actual.

Como se ve se trata de un sencilla metodología que partiendo de una identificación básica de los elementos de deben ser considerados en la solución evoluciona hasta alcanzar una plataforma parametrizable capaz de generar soluciones. Y que finalmente permite analizar cual ha sido el nivel de eficiencia de la propuesta realizada. Se trata por lo tanto una metodología para la construcción y análisis de una propuesta.

Aunque pudiera dar la sensación de que los aspectos cuantitativos no tiene un peso importante, se debe subrayar que esto no es así, lo que quedará de manifiesto cuando en el apartado 6.6 se indique como se distribuye el contenido de cada fase.

En el siguiente apartado se identificarán los elementos que son necesarios para construir la solución, y se justificará dicha necesidad. En algunos casos ya han sido comentados de forma explícita en este apartado.

6.4 Arquitectura

El segundo elemento del enfoque M.A.T. que se va a considerar es el arquitectónico. La Arquitectura fundamentalmente debe identificar los elementos constructivos empleados en la solución y la relación entre ellos. Por lo tanto el objetivo de este apartado es realizar una revisión de la metodología e identificar los bloques constructivos que serán necesarios en las diferentes fases. De esta forma cada elemento quedará vinculado con el resto dentro del contexto de la metodología.

Como se pondrá de manifiesto durante el desarrollo de este apartado una de las características más recurrente de los elementos constructivos empleados es la “abstracción”. En muchos casos se manejan ideas generales, como por ejemplo la idea de proceso, o la idea de agente software, que después de un periodo de maduración y adaptación a una realidad son particularizadas. Esto es común en los procesos creativos en los que se parte de elementos de referencia que son adaptados siguiendo un camino determinado. También es normal, como en el caso actual, que se empleen elementos abstractos en una determinada fase y el mismo tipo de elemento, pero particularizado, en otra fase de la metodología. Un ejemplo de este último concepto sería el empleo de la idea de Agente en una fase y la de un Agente concreto (por ejemplo un Agente que represente un subastador) en otra fase. En todo caso, en este apartado no se hará referencia al proceso por el cual se transforma los conceptos, que se ha considerado como parte de la metodología, sino a los conceptos o elementos en sí mismos.

La Figura 6-4 muestra un resumen de los elementos constructivos empleados durante la ejecución de la metodología y agrupados en 3 bloques diferentes. Por un lado, se encuentran

los elementos empleados en las tareas de identificación elemental, y posterior especificación de los requisitos de la solución, por otro lado se agrupan los elementos empleados en las tareas creativas, en la que se conceptualiza y da forma a la solución, y en el último grupo se encuentran los elementos empleados en las tareas de análisis experimental. Aunque existe una cierta vinculación entre agrupaciones de elementos constructivos y su utilización en pasos concretos de la metodología esto no es así en todos los casos.

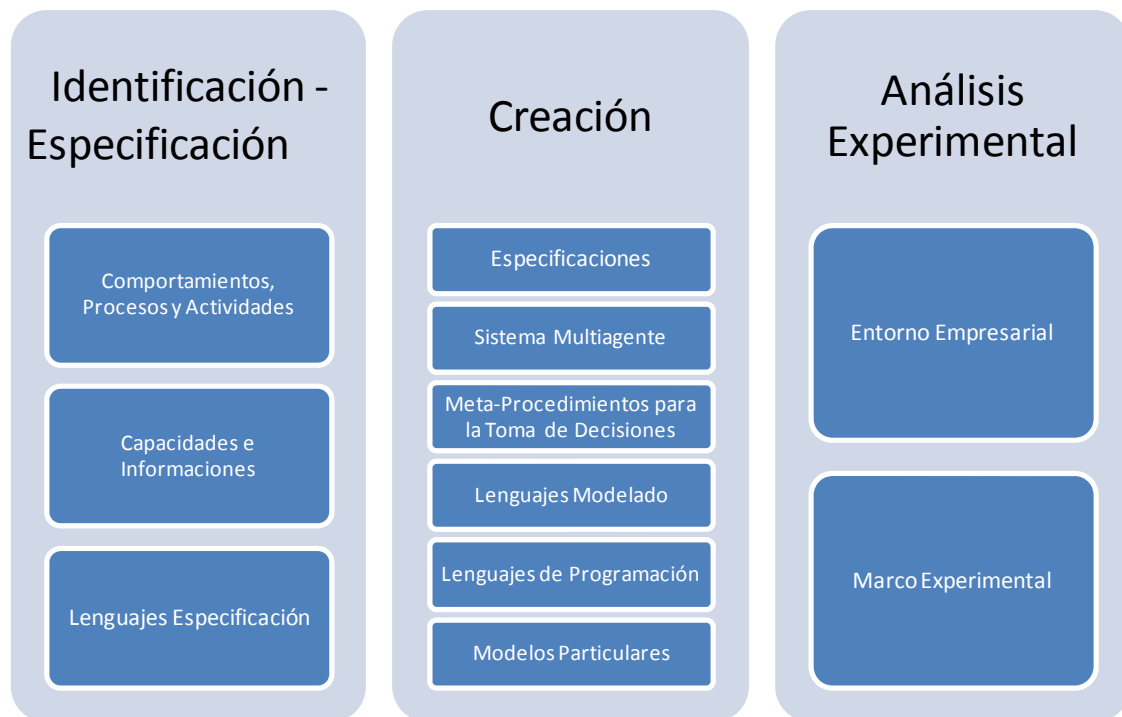


Figura 6-4. Agrupación de los elementos constructivos empleados (elaboración propia) para la propuesta de la Plataforma.

En el bloque de “Identificación – Especificación” se encuentran los elementos que permiten abstraer los aspectos relevantes del problema y su relación, y expresarlos en un lenguaje adecuado. Estos elementos se emplean únicamente en el paso 1, “Identificación Elemental del Problema”, y el paso 2, “Análisis de Requerimientos” de la metodología propuesta en el apartado anterior. Los elementos constructivos que se han destacado son 3:

- **Comportamientos, Procesos y Actividades:** Son el conjunto de abstracciones que se emplean en el paso 1 y 2 de la metodología, y se concretan en “especificaciones” que contienen las idealizaciones de lo que requiere el aspecto funcional y conductual. En concreto en el paso 1 de la metodología se necesitan los elementos funcionales más básicos, mientras que en el paso 2 de la metodología empleada, se hace uso de los funcionales y conductuales, que aplicando la metodología CIMOSA y el Modelado Matemático, permitirán obtener modelos de especificaciones

concretos. La forma de expresión requiere, a su vez, del empleo de lenguajes de especificación.

- **Capacidades e Informaciones:** Se emplean en el paso 2 de la metodología, y se concretan en “especificaciones” complementarias basadas en idealizaciones de los recursos necesarios entendidos como capacidades/cualidades para hacer algo, y elementos de información que son empleados para afrontar el problema. Estos elementos son empleados en el marco de la metodología CIMOSA, y se necesita de un lenguaje para plasmarlas en elementos concretos.
- **Lenguajes de Especificación:** Son la forma de comunicar el conjunto de comportamientos, procesos, actividades, capacidades e informaciones previamente ideados. En base a la metodología empleada, en concreto en el paso 2, se debe entender que el lenguaje empleado es por un lado el propuesto por CIMOSA, y por otro lado, el Matemático.

En el bloque de “Creación” se encuentran los elementos que permiten dar forma a la solución del problema expresados en un lenguaje adecuado. Estos elementos se emplean con mayor intensidad en el paso 3, “Diseño de la Solución”, de la metodología propuesta en el apartado anterior, aunque también son empleados en el apartado 4, “Implementación de la Herramienta”. Los elementos constructivos que se han destacado son 6:

- **Especificaciones:** Este elemento es el formado por el conjunto de requerimientos y restricciones elaborados en el paso 2 de la metodología para el caso particular que se planteó en el capítulo 5, y deben de servir de referencia en el proceso creativo. Se emplean fundamentalmente en el paso 3 para el diseño de la solución, aunque también se tienen en cuenta en el paso 4.
- **Sistema Multiagente:** Se trata de la estructura del Sistema sobre el cual está soportada la solución, y que una vez materializada dará lugar a lo que se ha denominado Plataforma. Hace referencia al conjunto de teorías previamente seleccionadas que determinan que es un Agente, como se caracteriza y como puede formar sistemas (coordinación, interacción, etc.). Con estos elementos y la aplicación de la metodología INGENIAS, exclusivamente en el paso 3 de la metodología, “Diseño de la Solución”, se obtiene el modelo de la plataforma. Para reflejar esos modelos es necesario aplicar un lenguaje determinado que en esta caso viene definido por INGENIAS.
- **Meta-Procedimientos para la Toma de Decisiones:** Se trata de estructuras procedimentales predefinidas que permiten la rápida implementación de procesos

de decisión y/o cálculo. En este caso se podría poner el ejemplo de la estructura de los Algoritmos Genéticos (Metaheurísticas) o incluso de los simples procedimientos de cálculo del trabajo más pequeño de una lista. Estos elementos son particularizados dando lugar a los algoritmos empleados en el cálculo de los Programas Predictivos y Reactivos, y también a los pequeños procesos empleados por los Agentes para tomar una decisión dentro de la plataforma. Para plasmar la particularización es necesario emplear lenguajes de modelado. Fundamentalmente se emplean en el paso 3.

- **Lenguajes de Modelado:** Son la forma de comunicar el conjunto de particularizaciones realizadas en la fase creativa del proceso de definición de la solución, en concreto en el paso 3 de la metodología. Se deben emplear diferentes lenguajes. En concreto se propone la utilización del lenguaje natural y basado en pseudo-código, para expresar algoritmos y pequeños procedimientos de decisión, y de una aproximación del lenguaje UML (Unified Modeling Language) propuesta por INGENIAS para los modelos del sistema multiagente.
- **Lenguajes de Programación:** Son la forma de realizar la implementación de los modelos particularizados para ser posteriormente ejecutados en sistemas informáticos. En este caso se ha propuesto la utilización del lenguaje de programación JAVA por las razones que en capítulo dedicado a la implementación se esgrimirán. Fundamentalmente se emplean en el paso 4.
- **Modelos Particulares:** En este elemento se hace referencia a al conjunto de diseños que se elaboran para ser posteriormente implementados. Se elaboran en el paso 3 de la metodología y se emplean en el paso 4.

En el bloque de “Análisis Experimental” se encuentran los elementos que permiten analizar de una forma científica la eficiencia de la solución del problema propuesta de forma parcial y global. Estos elementos se emplean en el paso 3, “Diseño de la Solución” de la metodología propuesta para realizar un análisis parcial en relación a los algoritmos de cálculo de los Programas Predictivos y Reactivos. También se emplean en el paso 5, “Prueba de Funcionamiento” para un análisis global de la eficiencia de la solución. Los elementos constructivos que se han destacado son 2:

- **Entorno Empresarial:** En este elemento se resume el conocimiento concreto de los procesos de la empresa sobre la que se desea realizar la prueba de funcionamiento, que permitirá el ajuste fino del funcionamiento de la Plataforma mediante Parámetros. También se incluyen en este elemento los Datos de la Empresa que

alimentar a la plataforma para que esta pueda operar en función del caso que se analice en el paso 5 de la metodología, por ejemplo los tiempos de proceso o cambio de partida.

- **Marco Experimental:** Por tales entendemos las instrucciones que como consecuencia de la aplicación de la metodología de Diseño de Experimentos científicos nos indican que se debe hacer para testear de una forma estructurada, bien la solución en su conjunto (plataforma) en el paso 5 de la metodología, o bien los algoritmos de cálculo Predictivo y Reactivo en el paso 3 de la metodología.

El conjunto de elementos contenidos en los tres bloques identificados por su afinidad constituyen la base arquitectónica que se necesita para construir la propuesta.

6.5 Herramientas

Una vez revisado el aspecto metodológico y arquitectónico se debe conocer como se va a avanzar en la construcción de la solución, y en concreto cuales son los útiles que se deben emplear para cimentarla. En este apartado se dan a conocer las herramientas que se van a emplear y las que se espera generar durante el proceso.

De la misma manera que se ha planteado con los elementos arquitectónicos, las herramientas se van a relacionar con los pasos en las que se van a emplear dentro de la metodología. Como se puede ver en la Figura 6-5 las herramientas han dividido en 2 bloques: La empleadas y las generadas. Ambas se van a comentar brevemente.

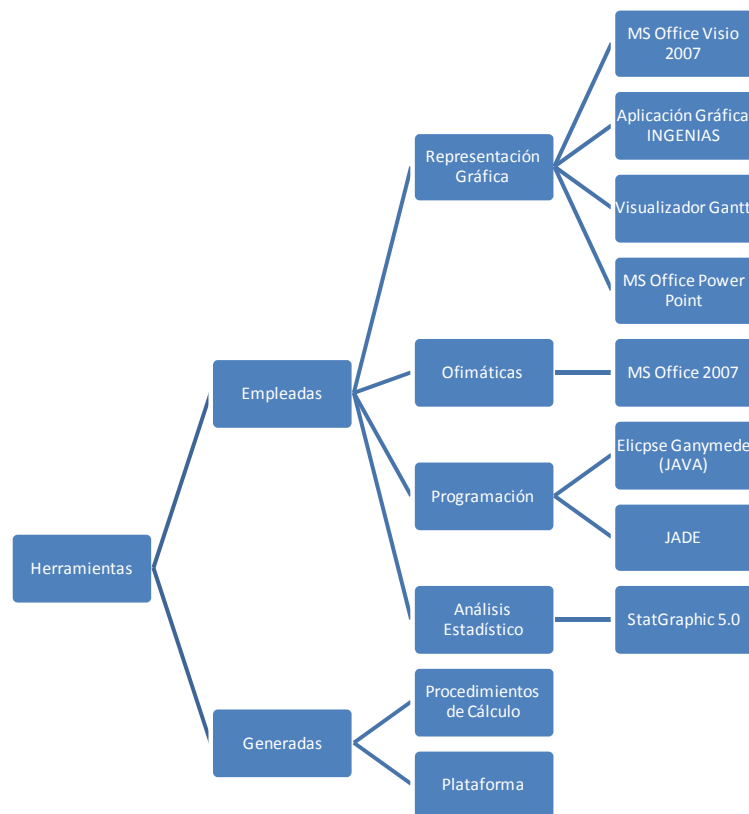


Figura 6-5. Relación de herramientas empleadas y generadas en la propuesta de solución (elaboración propia).

Las herramientas empleadas:

- Representación Gráfica (Específica):** Se ha denominado herramienta de representación gráfica al conjunto de utilidades que han permitido representar visualmente elementos específicos de algún área. En este aspecto se debe mencionar la herramienta “Microsoft Office VISIO 2007” que se ha empleado para representar elementos UML o elementos CIMOSA, y la propia herramienta proporcionada por la metodología INGENIAS, sin denominación específica, que se ha empleado para dibujar los modelos propuestos por dicha metodología. Ambas herramientas se han empleado en el paso 2 y 3 de la metodología para el análisis y diseño. También se ha empleado la herramienta desarrollada de forma específica (mediante desarrollo no comercial) para representar diagramas GANTT que permitirá visualizar Programas Productivos en el paso 3, para el diseño de algoritmos Predictivos o Reactivos, y en el paso 5, para las pruebas de funcionamiento. Se han empleado otras herramientas de representación gráfica de carácter generalista como “Microsoft PowerPoint”, pero que se ha incluido dentro de las herramientas ofimáticas.

- **Ofimáticas:** En este conjunto se hace referencia exclusiva a las herramientas incluidas en “Microsoft Office 2007”. En concreto al Word, Excel y PowerPoint que se han empleado en todos los pasos de la metodología para elaborar textos, gráficos diversos o recolección de datos para su posterior análisis estadístico.
- **Programación:** En el paso 3 de la metodología, durante la implementación de los algoritmos de cálculo de los Programas Predictivos y Reactivos, así como en el paso 4 durante la construcción de la Plataforma software que los contendrá, se emplean herramientas de programación. En concreto se utiliza el entorno Eclipse GANYMEDE para desarrollar software basado en el lenguaje de programa JAVA, y el entorno JADE (Java Agent DEvelopment framework) versión 3.5, que proporciona un conjunto de facilidades ya disponible para trabajar y desarrollar Sistemas de Agentes que soportan el estándar FIPA^{vi}.
- **Análisis Estadístico:** Para el análisis de los datos obtenidos en los diferentes experimentos que serán planteados se ha empleado siempre la herramienta Statgraphics Plus para Windows versión 5.0. En concreto se emplea en el paso 3 y 5 de la metodología, es decir en el diseño de la solución y en la fase de pruebas de funcionamiento.

Las herramientas generadas:

- **Procedimientos de Cálculo:** Se entiende por tal utilidad a las instancias ejecutables del software que contienen los algoritmos Predictivos y Reactivos realizados para generar Programas Productivos. Estos algoritmos valorados en este caso sólo por su capacidad de transformación de datos son considerados herramientas por sí mismos, aunque también pueden ser vistos como elementos constructivos si se valora el conocimiento que aportan. Esta herramienta ya está disponible en la fase de diseño de la solución.
- **Plataforma:** De la misma forma que los algoritmos, la Plataforma Software resultante, puede ser considerada como una herramienta en manos del programador de la producción que le ayuda a tomar decisiones, aunque si se analiza su contenido se podrían identificar los elementos arquitectónicos y tecnológicos que ya se han

^{vi} FIPA (Foundation for Intelligent Physical Agent) es un estándar comentado en capítulos anteriores.

especificado con anterioridad. Esta herramienta está disponible al final la fase de implementación.

El conjunto de herramientas, ya sean empleadas o generadas, que se han comentado constituyen la base instrumental que se necesita para construir la propuesta.

6.6 Visión Global del enfoque M.A.T.

Una vez se dispone de una idea bastante precisa del conjunto de pasos que se van a realizar para elaborar la propuesta, así como de los elementos y herramientas con los que se va a construir sólo resta disponer de una imagen como la de la Tabla 6-1 para tener una visión de conjunto del enfoque M.A.T. aplicado al caso concreto que se está tratando.

Metodología	Resultados	Arquitectura y Herramientas	
Paso 1: Definición Elemental del Problema	<ul style="list-style-type: none"> • Visión General (Gráfica) 	Arquitectura	<ul style="list-style-type: none"> • Comportamientos, Procesos y Actividades.
		Herramientas	<ul style="list-style-type: none"> • Ofimáticas
Paso 2: Análisis de requerimientos	<ul style="list-style-type: none"> • Especificación CIMOSA • Modelos Matemáticos 	Arquitectura	<ul style="list-style-type: none"> • Comportamientos, Procesos y Actividades. • Capacidades e Informaciones • Lenguajes de Especificación
		Herramientas	<ul style="list-style-type: none"> • Representación Gráfica • Ofimáticas
Paso 3: Diseño de la Solución	<ul style="list-style-type: none"> • Modelo INGENIAS • Procedimientos de Cálculo (Predictivo y Reactivo) • Análisis Comparativo de Algoritmos 	Arquitectura	<ul style="list-style-type: none"> • Especificaciones • Sistemas Multiagente • Lenguajes de Modelado • Meta Procedimientos de Toma de Decisiones • Lenguajes de Programación • Marco Experimental
		Herramientas	<ul style="list-style-type: none"> • Representación Gráfica • Programación • Análisis Estadístico • Ofimáticas
Paso 4: Implementación de la Herramienta	<ul style="list-style-type: none"> • Plataforma Software 	Arquitectura	<ul style="list-style-type: none"> • Modelos Particulares • Lenguajes de Especificación
		Herramientas	<ul style="list-style-type: none"> • Programación • Ofimáticas

Paso 5: Prueba de Funcionamiento	<ul style="list-style-type: none"> • Plataforma Software Particularizada • Análisis Comparativo entre Solución Actual y Propuesta 	Arquitectura	<ul style="list-style-type: none"> • Entorno Empresarial • Marco Experimental
		Herramientas	<ul style="list-style-type: none"> • Representación Gráfica • Análisis Estadístico • Ofimáticas

Tabla 6-1. Relación entre los 3 tipos de elementos del enfoque M.A.T. (elaboración propia).

En la mencionada Tabla 6-1 se puede observar como dentro de cada uno de los cuadros que representan a cada uno de los pasos metodológicos se han incluido a su vez dos cuadros. El cuadro de la izquierda contiene los elementos arquitectónicos que se emplearán en dicho paso, mientras que el cuadro de la derecha contiene las herramientas empleadas. También se ha incluido un recuadro en la parte derecha de cada paso en el que se contienen los resultados que se esperan obtener una vez concluido el paso correspondiente.

Como se puede ver el conjunto de elementos y herramientas empleado en cada paso es bastante amplio, lo que da una idea de la complejidad del proceso que se desarrollará en los próximos capítulos. Precisamente, en relación con el desarrollo literario de la propuesta, hay que señalar, que con el objeto de facilitar la conexión entre aspectos con fuerte relación entre ellos finalmente no se ha dedicado un capítulo a cada paso de la metodología. Se ha seguido una agrupación de los contenidos ligeramente diferente a la de los pasos metodológicos, que se comenta a continuación:

- **Capítulo 7:** Se denominará “Análisis, Diseño e Implementación de una Plataforma Software basada en Multiagentes para la Programación de la Producción Predictivo-Reactiva”. En el se tratan los aspectos relativos a la plataforma con un nivel de abstracción elevado, y evitando entrar en el estudio de procesos detallados, como es el caso de los algoritmos de cálculo. En este capítulo se desarrollan los 1, 2 y 4 completos. Del paso 3 se ha excluido la segunda parte correspondiente a la “Construcción los procesos a nivel de detalle”, en concreto lo relativo a los algoritmos de cálculo Predictivo y Reactivo.
- **Capítulo 8:** Se denomina “Algoritmos para la Programación Predictiva de la Producción en un Taller de Flujo Híbrido”. En este capítulo de desarrollan por completo los aspectos que tienen que ver con la selección, y en su caso diseño y desarrollo, de los algoritmos para el cálculo Predictivo de la producción. Este capítulo se enmarca dentro de la segunda parte del paso 3 que no se trató en el capítulo 7.
- **Capítulo 9:** Se denomina “Algoritmos para la Programación Reactiva de la Producción en un Taller de Flujo Híbrido”. En este capítulo de desarrollan por completo los

aspectos que tienen que ver con la selección de los algoritmos para el cálculo Reactivo de la producción. Este capítulo se enmarca dentro de la segunda parte del paso 3 que no se trató en el capítulo 7.

- **Capítulo 10:** Se denomina “Aplicación de la Propuesta de Programación de la Producción Predictivo/Reactiva a una Empresa Cerámica”. Este capítulo coincide por completo con el último paso de la metodología.

Después del recorrido realizado en este capítulo se debe estar en disposición de tener una idea de conjunto de lo que se plantea realizar en los próximos capítulos, que elementos constructivos y herramientas se van a emplear, y como se va a dividir el trabajo.

6.7 Conclusiones

Como conclusión más relevante del capítulo hay que señalar que éste debe haber servido para dar a conocer la estructura de las acciones futuras y la forma de redactarlas para darlas a conocer. Se ha tratado por lo tanto de un capítulo elaborado en aras de facilitar al lector el entendimiento del trabajo que se exponga en el futuro, y que posiblemente sea necesario recordar al final de cada uno los capítulos siguientes.

Otros aspectos destacables son que como consecuencia de la necesidad de ordenar y estructurar el trabajo se ha generado una metodología y se han identificado diversos elementos constructivos y herramientas tecnológicas que se emplearan en los diferentes pasos de la misma. Todo **este conjunto, Metodología, Arquitectura y Tecnología se ha combinado adaptándolo y particularizándolo al caso que nos ocupa**, que no es otro que el problema detectado en el capítulo 5, y bajo la denominación de aproximación M.A.T.

Aunque el trabajo desarrollado en este capítulo no ha tenido en ningún caso la intención de crear ni una metodología general, ni una arquitectura de referencia, ni tampoco ningún tipo de recomendación general para el uso de herramientas vinculadas a los pasos descritos sí que **se puede destacar lo interesante que ha resultado el trabajo de cara a su utilización futura**. Por este motivo, se considera que puede ser útil plantearse la transformación del trabajo realizado en este capítulo a un conjunto de elementos de carácter más general que pudieran ser utilizados de forma más general.

Con el bagaje adquirido se inicia el próximo capítulo en el que como ya se ha indicado se procederá a describir el Análisis, Diseño e Implementación de la Plataforma Software basada en un Sistema Multiagente para la Programación de la Producción Predictivo-Reactiva.

6.8 Bibliografía

ESPRIT Consortium AMICE 1993. CIMOSA - Open System Architecture for CIM.

Kosanke, K. 1995. Cimoso - Overview and Status. Computers in Industry, 27(2): 101-109.

M.Zelm, F.Vernadat, & K.Kosanke 1995. The CIMOSA business modelling process. Computers in Industry, 21(2).

Ortiz Bas, A. 1998. Propuesta para el desarrollo de programas de integración empresarial en empresas industriales. Aplicación a una empresa del sector cerámico. Universidad Politécnica de Valencia.

Pavon, J. & Gomez-Sanz, J. 2003. Agent oriented software engineering with INGENIAS. Multi-Agent Systems and Applications II, Proceedings, 2691: 394-403.

**Análisis, Diseño e Implementación de
una Plataforma Software basada en
Multiagentes para la Programación de
la Producción Predictivo-Reactiva**

Capítulo

7

Tabla de contenido

7.	Capítulo 7	369
7.1	Introducción	369
7.2	Definición Elemental del Problema	370
7.2.1	La Gestión Coordinada del Programa Productivo	372
7.2.2	Consideraciones Generales sobre los Medibles	386
7.2.3	Una Visión General del Proceso	387
7.3	Análisis de los Requerimientos de la Plataforma IPSU-MAS	389
7.3.1	La arquitectura CIMOSA	389
7.3.1.1	La Vista Funcional	391
7.3.1.2	La Vista Informacional	392
7.3.1.3	La Vista de Recursos	392
7.3.1.4	La Vista Organizacional	393
7.3.2	Análisis de Requerimientos	394
7.3.2.1	Establecimiento del Dominio	395
7.3.2.2	Análisis del Comportamiento	397
7.3.2.3	Análisis Operacional	406
7.3.2.4	Análisis de la Información	409
7.3.2.5	Análisis de los Recursos	412
7.3.2.6	Verificación de la Consistencia del Modelo	413
7.3.3	Resumen del Proceso de Modelado de Requerimientos	414
7.4	Diseño de la Plataforma IPSU-MAS	414

7.4.1	Metodología para el Diseño de la Plataforma IPSU-MAS	416
7.4.2	Metodología INGENIAS	417
7.4.2.1	Metodología	417
7.4.2.2	Meta Modelos	418
7.4.3	Diseño de la Plataforma IPSU-MAS aplicando INGENIAS.....	419
7.4.3.1	Esquema General	421
7.4.3.2	Modelo de Casos de Uso	422
7.4.3.3	Modelo de Organización	425
7.4.3.4	Modelo de Entorno	428
7.4.3.5	Modelo de Objetivo y Tareas	429
7.4.3.6	Modelo de Agentes	435
7.4.3.7	Modelo de Interacciones.....	442
7.4.4	Relación entre la Metodología de Análisis y de Diseño	452
7.5	Implementación de la Plataforma IPSU-MAS.....	456
7.6	Conclusiones	457
7.7	Bibliografía	459

Índice de figuras

Figura 7-1. Proceso para la definición (el establecimiento o modificación) de un Programa productivo (elaboración propia).....	371
Figura 7-2. Dominio de actuación del programador de la Producción (elaboración propia). ..	371
Figura 7-3. Ejemplo de Programa (Superior) sin respetar la fecha de finalización” D” y Programa (Inferior) que respeta “D” (elaboración propia).....	378
Figura 7-4. Ejemplo de Programa (Superior) con capacidad sobrante y Programa (Inferior) ocupación similar a la CPC (elaboración propia).	379
Figura 7-5. Ejemplo de Programa (Superior) con ocupación similar a la CPC y Programa (Inferior) superior a la CPC pero mayor tasa de productividad (m2/periodo) (elaboración propia).	380
Figura 7-6. Estados en la configuración del Programa de la Producción (elaboración propia). ..	382
Figura 7-7. Control del proceso de Gestión del Programa (elaboración propia)	388
Figura 7-8. Marco de modelado CIMOSA(Kosanke, K. 1995).	391
Figura 7-9. Principales conceptos CIMOSA y su relación.(RGCP 1999)	393
Figura 7-10. Relación entre las vistas del modelo de requerimientos CIMOSA (RGCP 1999). ..	395
Figura 7-11. Diagrama de Funcional del Proceso de Negocio de “Programación Predictiva” (elaboración propia).	399
Figura 7-12. Diagrama de Funcional del Proceso de Negocio de “Programación Reactiva” (elaboración propia).	402
Figura 7-13. Detalle de la función operativa “Configuración del CT” (elaboración propia).	405
Figura 7-14. Relación entre los elementos de la Vista Funcional del Modelo de Requerimientos de un sistema de Programación de la Producción (elaboración propia).	406
Figura 7-15. Vista Informacional del Modelo de Requerimientos (elaboración propia).	411
Figura 7-16. Relación entre las Vistas del Modelo de Requerimientos (elaboración propia)...	415
Figura 7-17. Meta-modelos para el Diseño del Sistema Multiagente (Pavon, J. et al. 2003) ..	419
Figura 7-18. Modelo de casos de uso del Planificador (elaboración propia).	424
Figura 7-19. Modelo de casos de uso del Coordinador (elaboración propia).....	424
Figura 7-20. Modelo de casos de uso del Operario (elaboración propia).....	424
Figura 7-21. Modelo de casos de uso del Programador (elaboración propia).....	425
Figura 7-22. Modelo de Organización (elaboración propia).	427
Figura 7-23. Modelo de entorno (elaboración propia).	429
Figura 7-24. Modelo de tareas y objetivos de la Producción Predictiva (elaboración propia). ..	432

Figura 7-25. Modelo de tareas y objetivos de la Producción Reactiva (elaboración propia). ..	434
Figura 7-26. Agente planificador (elaboración propia).	436
Figura 7-27. Agente Programador (elaboración propia).	438
Figura 7-28. Agente Operario (elaboración propia).	439
Figura 7-29. Modificación de estados mentales por Notificar Planificación Periódica (elaboración propia).	440
Figura 7-30. Modificación de estados mentales por definir parámetros (elaboración propia).	440
Figura 7-31. Modificación de estados mentales por reajustes de la Producción (elaboración propia).	441
Figura 7-32. Modificación de estados mentales por establecimiento de Producción Predictiva (elaboración propia).	441
Figura 7-33. Modelo INT: Obtener Planificación Periódica (elaboración propia).	443
Figura 7-34. Protocolo Obtener Planificación Periódica (elaboración propia).	444
Figura 7-35. Modelo INT: Notificar Inconsistencias (elaboración propia).	445
Figura 7-36. Protocolo Notificar Inconsistencias (elaboración propia).	445
Figura 7-37. Modelo INT: Modificar Planificación (elaboración propia).	446
Figura 7-38. Protocolo Modificar Planificación (elaboración propia).	448
Figura 7-39. Modelo INT: Notificar Programación de Producción Revisada (elaboración propia).	449
Figura 7-40. Protocolo Notificar Programación de Producción Revisada (elaboración propia).	449
Figura 7-41. Modelo INT: Notificar Problema (elaboración propia).	450
Figura 7-42. Protocolo Notificar Problema (elaboración propia).	451
Figura 7-43. Condiciones iniciadoras del Ajuste de la Producción (elaboración propia).	452
Figura 7-44. Visión de los componentes fundamentales de la Plataforma IPSU-MAS (elaboración propia).	456

Índice de tablas

Tabla 7-1. Cambios en el Sistema de Información del Planificador (elaboración propia).	384
Tabla 7-2. Flujos de trabajo en la metodología INGENIAS (Pavon, J. et al. 2003)	418
Tabla 7-3. Resumen de gráficos correspondientes a cada modelos empleado en el diseño (elaboración propia).	421
Tabla 7-4. Resumen de los casos de uso planteados en la Plataforma IPSU-MAS (elaboración propia).	422
Tabla 7-5. Notación empleada en las figuras que representan los Casos de Uso y los Modelos.	423
Tabla 7-6. Resumen de las entidades del modelo de organización planteados en la Plataforma IPSU-MAS (elaboración propia).	426
Tabla 7-7. Resumen de las entidades del modelo de entorno planteados en la Plataforma IPSU-MAS (elaboración propia).	428
Tabla 7-8. Resumen de las interacciones del modelo de objetivos y tareas (Predictivos) planteados en la Plataforma IPSU-MAS (elaboración propia).	430
Tabla 7-9. Resumen de las entidades del modelo de objetivos y tareas (Reactivo) planteados en la Plataforma IPSU-MAS (elaboración propia).	433
Tabla 7-10. Resumen de los componentes del agente planificador (elaboración propia).	436
Tabla 7-11. Resumen de los componentes del agente programador (elaboración propia).	438
Tabla 7-12. Resumen de los componentes del agente operario (elaboración propia).	440
Tabla 7-13. Resumen de las entidades del protocolo “Obtener Planificación Periódica” planteado en la Plataforma IPSU-MAS (elaboración propia).	444
Tabla 7-14. Resumen de las entidades del protocolo “notificar inconsistencias” planteado en la Plataforma IPSU-MAS (elaboración propia).	445
Tabla 7-15. Resumen de las entidades del protocolo “Modificar Planificación” planteado en la Plataforma IPSU-MAS (elaboración propia).	447
Tabla 7-16. Resumen de las entidades del protocolo “Notificar Programación de Producción Revisada” planteado en la Plataforma IPSU-MAS (elaboración propia).	450
Tabla 7-17. Resumen de las entidades del protocolo “Notificar Problema” planteado en la Plataforma IPSU-MAS (elaboración propia).	451
Tabla 7-18. Aportaciones básicas para la construcción de los modelos INGENIAS (elaboración propia).	454

BORRADOR

7.1 Introducción

El este capítulo se abordan los aspectos relacionados con el análisis, diseño e implementación de la estructura básica de la Plataforma software que servirá de nexo de unión y soporte a la propuesta más conceptual que se desarrollará en paralelo tanto en este como en los siguientes capítulos.

La primera parte de este capítulo se dedicará a realizar un exhaustivo análisis de los requerimientos que definen el problema. Este análisis se dividirá en dos partes que se corresponden con el primer y el segundo paso de la metodología indicada en el capítulo 6. En primer lugar se realizará una **definición elemental del problema** en donde se marcarán las líneas fundamentales del mismo. Y en segundo lugar se realizará **el análisis de requerimientos detallado**. Entre ambos pasos el segundo es el que más peso específico tiene. Para su desarrollo se ha empleado la propuesta **CIMOSA** que posibilita el análisis de los requisitos del sistema considerando simultáneamente diversos puntos de vista, complementarios y necesarios, como son el **funcional**, el **organizativo**, el de la **información** y el de los **recursos**.

La segunda parte del capítulo se dedicará al diseño de la Plataforma que se iniciará una vez se ha completado el análisis de los requerimientos que se desea que sean considerados en la misma. Este bloque resumirá las tareas realizadas en el tercer paso de la metodología. **La Plataforma debe ser considerada por el lector como el elemento estructural sobre el cual serán colocados posteriormente los algoritmos para la Programación Predictiva y Reactiva.** Aunque al analizar los resultados finales, es decir los Programas productivos, los efectos más visibles serán sin duda causados por los algoritmos empleados, la aportación de la Plataforma no es menor, ya que es la que **garantiza el funcionamiento conjunto de diversos procesos que en otras propuestas operan de forma aislada e individual. La coordinación funcional basada en la cooperación y la comunicación es el objetivo fundamental que se trata de obtener con este elemento software.**

En cuanto a su enfoque técnico la Plataforma empleará para su diseño el paradigma de los **Sistemas Multiagente**, y en base a este elemento distintivo, y a su propia funcionalidad se le ha denominado **“IPSU-MAS” (Integrated Platform for Scheduling under Uncertainty based on Multi-Agent Systems)**. Para el desarrollo de las tareas de diseño se ha empleado la propuesta **INGENIAS** que ya que ésta facilita el desarrollo de Sistemas basados en Multiagentes. La parte del capítulo dedicada al diseño está sumamente vinculada a la tecnología de los sistemas de

información, pero en cualquier caso la visión que se proporciona en relación al diseño de la Plataforma se corresponde con los niveles más conceptuales de la metodología empleada.

La tercera parte del capítulo se dedicará a realizar unos breves comentarios sobre la implementación de la Plataforma. En este punto se resumirán los aspectos más relevantes relacionados con el cuarto paso de la metodología.

El desarrollo del capítulo se ordena de la siguiente forma: En el apartado 7.2 se realizará la definición elemental del problema. En el apartado 7.3 se realizará el análisis de requerimientos, de forma que en el apartado 7.3.1 se comentará brevemente el enfoque CIMOSA, en el 7.3.2 se desarrollará el análisis en sí, y en el 7.3.3 se realizará una breve reflexión sobre el proceso realizado. En el apartado 7.4 se desarrolla el diseño de la Plataforma, de forma que en el apartado 7.4.1 se realiza un breve resumen de las diferentes metodologías consideradas, en el apartado 7.4.2 se comentan aspectos básicos que se proponen en la metodología INGENIAS, en el apartado 7.4.3 se realiza el diseño de la Plataforma empleando los modelos propuestos por INGENIAS, y en el apartado 7.5 se realizará una breve reflexión sobre el proceso de diseño. El apartado 7.6 se dedica a la implementación de la Plataforma. Y por último, en el apartado 7.6 se realizará una breve reflexión a modo de conclusión del capítulo.

7.2 Definición Elemental del Problema

Para que el Programa Productivo alcance con éxito el objetivo de obtener la mayor tasa de productividad posible comentado en el capítulo 5, **no es suficiente con un esfuerzo de mejora centrado únicamente en el cálculo del Programa**, sino un enfoque en un sentido amplio del proceso mediante lo que se ha denominado *Gestión del Programa Productivo*. Hay que destacar que si no existe **una coordinación bilateral ágil entre el planificador y el programador** de la Producción los posibles beneficios obtenidos de la obtención de un buen uso de los recursos no serán aprovechados. La interactuación entre ambos responsables debe producirse tanto en la fase de Establecimiento del Programa como en la fase de ejecución del mismo. En caso de no existir dicha relación el esfuerzo se traducirá en un tiempo ocioso, que en el mejor de los casos será invertido, por ejemplo, en tareas de mantenimiento que no supondrán un aumento de la tasa de productividad. En la práctica se ha demostrado que esta coordinación es tremendamente complicada, y que en muchos casos acaba por no producirse. Obviamente, la coordinación puede llegar a implicar a más responsables y no quedarse sólo en las tareas de Planificación y Programación. No obstante, debido al enfoque de esta tesis sólo se tendrán en cuenta a los dos responsables mencionados.

Aunque para potenciar la coordinación planificador-programador se propone la *Gestión del Programa Productivo*, no se debe perder de vista el proceso de Cálculo del Programa que se encuentra embebido en el mismo. Se trata del proceso de definición de un Programa Productivo consensuado entre ambos responsables. En la Figura 7-1 se esboza el proceso para la definición de la Programación de la Producción. Este proceso se debería de plantear antes de la entrada en vigor de cualquier Programa productivo, ya sea el que se desea Establecer inicialmente o una Modificación del mismo.



Figura 7-1. Proceso para la definición (el establecimiento o modificación) de un Programa productivo (elaboración propia).

El proceso de definición de un Programación Productivo tiene como objetivo, no sólo ofrecer un Programa, sino alcanzar con éste la máxima conciliación posible entre los objetivos del planificador y el programador, para lograr alcanzar como objetivo general, mediante una tarea de ajuste, la mayor tasa de productividad posible.

Para resumir la visión general del dominio de trabajo del programador se puede decir que el problema que se desea afrontar se centra en las actividades integras de éste durante un periodo, lo que comprende el proceso de *Gestión del Programa Productivo*, en el cual se integra, embebido, el proceso de definición del Programación de la Producción propiamente dicho, como se puede ver en la Figura 7-2.

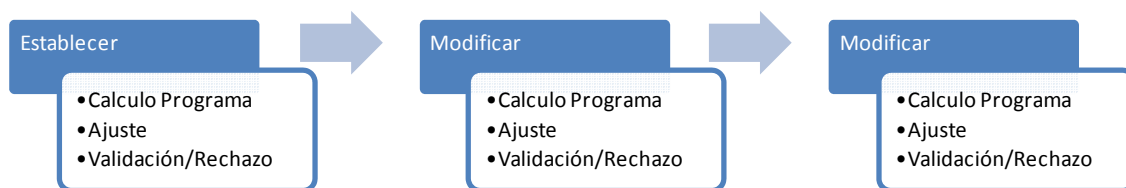


Figura 7-2. Dominio de actuación del programador de la Producción (elaboración propia).

La visión del marco de trabajo que se ha presentado se debe concretar en unos requerimientos a partir de los cuales se pueda Diseñar e Implementar una solución. En este sentido es necesario profundizar algo más en algunas cuestiones relativas a *las bases que soportarán la coordinación entre planificador y programador* en relación a la ejecución del Plan Maestro y del Programa Productivo.

La coordinación entre el área de Planificación y de Programación debe ser constante durante el periodo productivo. Esta coordinación puede entenderse como una relación bidireccional donde cada uno de los responsables realiza propuestas dentro de su ámbito, que son analizadas por la otra parte hasta definir un Programa Productivo. El objeto de realizar esta comunicación es el de poder ir refinando el Programa Productivo para alcanzar el objetivo propuesto de *obtener la mayor tasa de productividad* utilizando, en cada iteración, el conocimiento detallado disponible por parte de cada responsable. En este punto se van a cuestionar aspectos como; ¿cuándo revisar un Programa?, ¿quién tiene la iniciativa en la Modificación?, ¿cuándo es necesario coordinarse?, o ¿por qué estados puede pasar un Programa?

7.2.1 La Gestión Coordinada del Programa Productivo

La coordinación debe realizarse bajo una serie de protocolos que permitan alcanzar siempre una solución eficiente en un tiempo limitado. Estos protocolos deben ser identificados como parte de la solución del problema. No obstante, se puede decir que la pauta general debe ser aquella que permita al programador recibir la información necesaria del ámbito de la Planificación para concretar un Programa Productivo. El programador analizará los resultados alcanzados y determinará si es posible cumplir con la producción prevista por el Plan de Fabricación obtenido a partir del Cálculo de la Necesidades de Materiales y Capacidad que cuenta como entrada principal el Plan Maestro establecido por el planificador, o si es posible aumentar la producción prevista. En función del análisis se puede iniciar un reajuste de la propuesta del planificador que suponga la posterior generación de un nuevo Programaⁱ de Producción. Este proceso seguirá hasta alcanzar un final que puede ser obtenido en base a una cota de tiempo definida o alguna otra limitación de cálculo o por haber alcanzado una propuesta que se considere adecuada.

A continuación se van a realizar una serie de razonamientos relacionados con la tarea de Planificación. No obstante, no se pretende intervenir en dicha tarea, tan sólo se quiere establecer de forma sólida el vínculo entre Programación y Planificación. La tarea de Planificación utiliza un dato fundamental como es la Capacidad del Taller para realizar la

ⁱ Cabe recordar que en el caso que nos ocupa se puede considerar que no se identifican piezas ni partes dentro los ítems que forman parte de las Órdenes de Producción. Las Órdenes de Producción están compuestas por trabajos que se corresponden con algunos de los productos identificados en el Plan Maestro (los que el MRP-II identifica que son necesarios producir).

asignación de trabajos a las diversas plantas o talleres. Dicha Capacidad se puede medir de diversas formas, no obstante, es fundamental utilizar aquella que permita relacionar este concepto con los el Tiempo de Proceso y Tiempo de Cambio de Partida establecidos para cada trabajo del Programa Productivo.

Dada la capacidad disponible (CP) de una etapaⁱⁱ de un taller para un periodo dado medida en horas, se ha denominado *CPS* o Cota Máxima de Capacidad al valor obtenido en la etapa con menor CP. *CPS* representa la suma de las horas disponibles que proporcionan los recursos de la etapa cuello de botella. En una análisis más detallado se debe tener en cuenta que pueden existir restricciones en la asignación recurso-trabajo debido a que no todos los recursos son capaces de realizar las operaciones necesarias para producir cualquier artículo. Dado que la composición de los trabajos utilizados influye sustancialmente en este dato, éste se obtiene como promedio, mediante la observación de un número significativo de periodos, en condiciones de máximo rendimiento del taller y bajo el supuesto de la existencia de una programación eficiente. Además se asume que:

- Existe un acuerdo en dejar un “Margen Operativo (MO)”, es decir una capacidad de reserva no computable a priori. Este Margen Operativo permite a los responsables del taller solventar ciertas eventualidades sin incurrir en incumplimientos en las fechas de entrega de los trabajos. A efectos prácticos se puede introducir el MO como una disminución de capacidad.
- En cualquier cálculo de capacidad que se realice en Planificación se utiliza una estimación agregada, y poco precisa que suele tender a ser superior a la real, para el cálculo de los Tiempos de Cambio de Partida en que incurrirán cada trabajo durante todas sus operaciones, que será denominada Tiempo de Cambio de Partida a priori, STA.
- En cualquier cálculo de capacidad que se realice en Planificación se utiliza una estimación agregada, y poco precisa que suele tender a ser superior a la real, para el cálculo del Tiempo de Proceso en que cada trabajo incurrirá en todas sus operaciones, que aquí será denominada Tiempo de Proceso a priori, PA.

ⁱⁱ En la mayoría de las empresas de fabricación de baldosas cerámicas las operaciones que se realizan para obtener los productos finales se aplican de forma secuencial y se pueden agrupar por afinidad de recursos en diversas etapas o fases. Posteriormente, se identifica una división en etapas para el caso que se desea abordar

Por lo tanto, se puede decir que el planificador considera a efectos de calcular la composición de trabajos a realizar en un periodo:

- a) Una **Capacidad Disponible** igual a la cota superior de la etapa cuello de botella menos el margen operativo, que llamaremos “Capacidad Considerada (CPC)”, ver [7.1].
- b) La **Capacidad Considerada** por el planificador será inicialmente igual o superior a la Ocupación A Priori, es decir, a la suma de las sumas de los Tiempos de Proceso a priori y de Cambio de Partida a priori de todos los trabajos de cada familia de productos considerados en las operaciones que afectan a la etapa cuello de botella, ver [7.2].
- c) Existe una Capacidad Considerada para cada familiaⁱⁱⁱ de productos, que se corresponde con la capacidad que se obtiene teniendo en cuenta sólo aquellos recursos capaces de realizar las operaciones de cada familia. La suma de los Tiempos de Proceso a priori y de los Tiempos de Cambio de Partida a priori de los trabajos de una determinada familia no pueden superar la Capacidad Considerada para dicha familia en la etapa cuello de botella.

$$[7.1] \quad CPC = \min_{t=1}^T (CP_t) - MO = CPS - MO$$

Donde el índice “t” recorre todas las etapas existentes desde 1 hasta T.

[7.2]

$$CPC \geq \left[\sum_{i \in F_1}^n (PA_i^t + STA_i^t) + \sum_{i \in F_2}^n (PA_i^t + STA_i^t) + \dots + \sum_{i \in F_f}^n (PA_i^t + STA_i^t) \right] = O_{APRIORI}$$

Donde “t” representa a la etapa cuello de botella, y el índice “i” recorre los “n” trabajos programados seleccionando sólo aquellos que pertenecen a la familia de productos F_1 , F_2 o F_f .

ⁱⁱⁱ Se entiende por familia de productos, según el concepto tradicional, a una agrupación de referencias que se asemejan en su proceso productivo y características finales.

$$[7.3] \quad CPC_{F_f} \geq \sum_{i \in F_f}^n (PA_i^t + STA_i^t) \quad \forall f$$

A través del análisis de capacidades aproximado que permite valorar la disponibilidad de recursos antes de validar un determinado Plan Maestro, el planificador intentará incluir aquella combinación de trabajos, dentro de las necesidades existentes, que permita situarse cerca de la *CPC*. No obstante, puede existir una diferencia entre Capacidad Considerada en la Planificación y la Ocupación Planificada. Así mismo, se debe tener en cuenta que los tiempos a priori suelen sufrir una doble modificación antes de llegar a alcanzar su valor final. Estas modificaciones son:

- En el momento de validar el Programa Productivo, antes de su ejecución, los trabajos son asignados a máquinas y secuenciados, es decir existe una desagregación de los tiempos. Esto implica que los Tiempos de Proceso y de Cambio de Partida teóricos, que aquí se denominarán *P* y *ST* respectivamente, se pueden identificar, y sustituir a los tiempos a priori. Con dichos valores se puede calcular la Ocupación teórica, ver [7.4], es decir con datos deterministas.
- Durante de la ejecución del Programa productivo, cuando las operaciones de los trabajos van finalizando se conocen los tiempos reales, que pueden haber sufrido alguna variación respecto de lo tiempos teóricos (falta de calidad en el producto que obligan a realizar un lote más grande, problemas en el montaje de algún molde, etc.). En este caso el Tiempo de Proceso real aquí se denominará *PR* y el Tiempo de Cambio de Partida se denominará *STR*. De con estos valores se puede calcular la ocupación real, ver [7.6].
- Dado que la *CPC* se debió de establecer teniendo en cuenta una distribución en el tiempo de las operaciones de cada etapa que permitiera disponer de los productos en la fecha prevista una vez desagregados los tiempos pueden observarse desajustes en este sentido. Puede llegar a existir un pequeño retraso desde la fecha en la que los recursos de una etapa están reservados para realizar las operaciones de un determinado conjunto de trabajos pertenecientes a un Programa productivo y la fecha en la que éstos se inician. Esto puede ser debido por ejemplo a cualquier retraso en las operaciones anteriores. En todo caso dicho retraso debe ser contabilizado como un consumo de recursos, que denominaremos *R*. En caso de que existiese un adelanto, si los recursos están ocupados se debería esperar a la fecha de inicio asignada.

La diferencia entre los tiempos a priori y los teóricos, y los tiempos teóricos y los reales, medidos siempre sobre la etapa que se ha identificado como cuello de botella, puede suponer un aumento o una disminución de la Ocupación final. En el primer caso, dicha variación puede

ser tomada en cuenta durante el Procedimiento de Establecimiento del Programa. En todo caso, se puede realizar un proceso de ajuste, que supondrá el aumento o reducción del número de trabajos. Se intentará de esta forma que la Ocupación Teórica se aproxime a la Capacidad Considerada por el planificador. La relación entre los tiempos a priori y los teóricos se puede ver [7.5].

$$[7.4] \quad CPC \geq R + \left[\sum_{i \in F_1}^n (P_i^t + ST_i^t) + \sum_{i \in F_2}^n (P_i^t + ST_i^t) + \dots + \sum_{i \in F_f}^n (P_i^t + ST_i^t) \right] = O_{TEORICA}$$

$$[7.5] \quad O_{APRIORI} = O_{TEORICA} \pm \beta \quad \text{Donde } \beta \text{ es un número real.}$$

β será un número positivo si los tiempos teóricos son menores que los considerados a priori y no existe tiempo ocioso. Y β será negativo en caso de que los tiempos a priori sean superiores a los teóricos.

En el segundo caso, la variación debería ser tomada en cuenta en tiempo real para proceder en su caso a un Ajuste del Programa en el procedimiento de Modificación. El proceso de Modificación, podrá suponer el aumento o reducción del número de trabajos, e intentará que la Ocupación real se aproxime a la Capacidad Considerada o incluso a la Cota Superior de Capacidad en función de las condiciones de evolución del Programa Productivo y la experiencia del programador. La relación entre los tiempos teóricos y los reales se puede ver en [7.7].

[7.6]

$$CPC \geq R + \left[\sum_{i \in F_1}^n (PR_i^t + STR_i^t) + \sum_{i \in F_2}^n (PR_i^t + STR_i^t) + \dots + \sum_{i \in F_f}^n (PR_i^t + STR_i^t) \right] = O_{REAL}$$

$$[7.7] \quad O_{TEORICA} = O_{REAL} \pm \gamma \quad \text{Donde } \gamma \text{ es un número real.}$$

Según lo expuesto anteriormente se puede deducir que en el problema considerado **existe una necesidad plausible de coordinación entre el planificador y el programador de la Producción durante la fase de Establecimiento y Modificación del Programa de Producción, mediante un protocolo que suponga un dialogo que lleve a ambos a un acuerdo conveniente.**

El acuerdo debe concretar cuales son los trabajos que hay que incluir en el Programa Productivo en un determinado instante dentro del periodo analizado. En este sentido se consideran las siguientes situaciones^{iv}:

- **Situación A1:** El Plan Maestro deberá ser modificado si no es posible encontrar ningún Programa Productivo que suponga una Ocupación $O_{Teorica}$, que respete la *CPC considerada*, y que sea capaz de finalizar todos los trabajos en la fecha prevista en el *Plan de Fabricación*. En este caso, a instancias del programador, el planificador debería actuar considerando una modificación del Plan Maestro actual. Por ejemplo un Plan con menos trabajos para dicho periodo, o con menos cantidades en determinados trabajos, etc. Se realiza una ilustración del caso en la Figura 7-3 (taller de Flujo con las máquinas 11 y 12 en la etapa 1, la máquina 21 en la etapa 2, y la máquina 31 en la etapa 3). En la parte superior de dicha figura se supone que la etapa crítica es la 3, y que existe una CPC de 21 uds., 22 uds de CPS menos 1 ud. de MO, que se corresponden con el periodo 5-26. Una vez realizado el Programa (teórico) se observa como la Ocupación es de 23 uds. que se corresponden con el periodo 5-28 (el periodo 5-8 se corresponde con un retraso sobre lo previsto). Ya en la parte inferior de la figura se dispone de Programa Productivo corregido mediante la eliminación del trabajo A.
- **Situación A2:** Puede ocurrir que por un Desplazamiento a Derechas Significativo^v durante la ejecución del Programa se tenga que realizar un ajuste del mismo. En ese caso, se podría plantear una situación en la que sea imposible encontrar un Programa alternativo que determine una Ocupación O_{Real} que respete la CPC establecida, y que sea capaz de finalizar los trabajos en la fecha Prevista en el Programa Productivo anterior. Se puede emplear como ejemplo de este caso Figura 7-3, si se supone que se llega a la situación correspondiente al Programa de la parte superior debido a una duración superior a la prevista en una operación.

^{iv} Es imprescindible tener en cuenta que en el caso que se está analizando se ha determinado que todos los trabajos incluidos en un periodo tiene como fecha máxima de finalización la fecha de fin del periodo.

^v Concepto descrito en el capítulo 5.

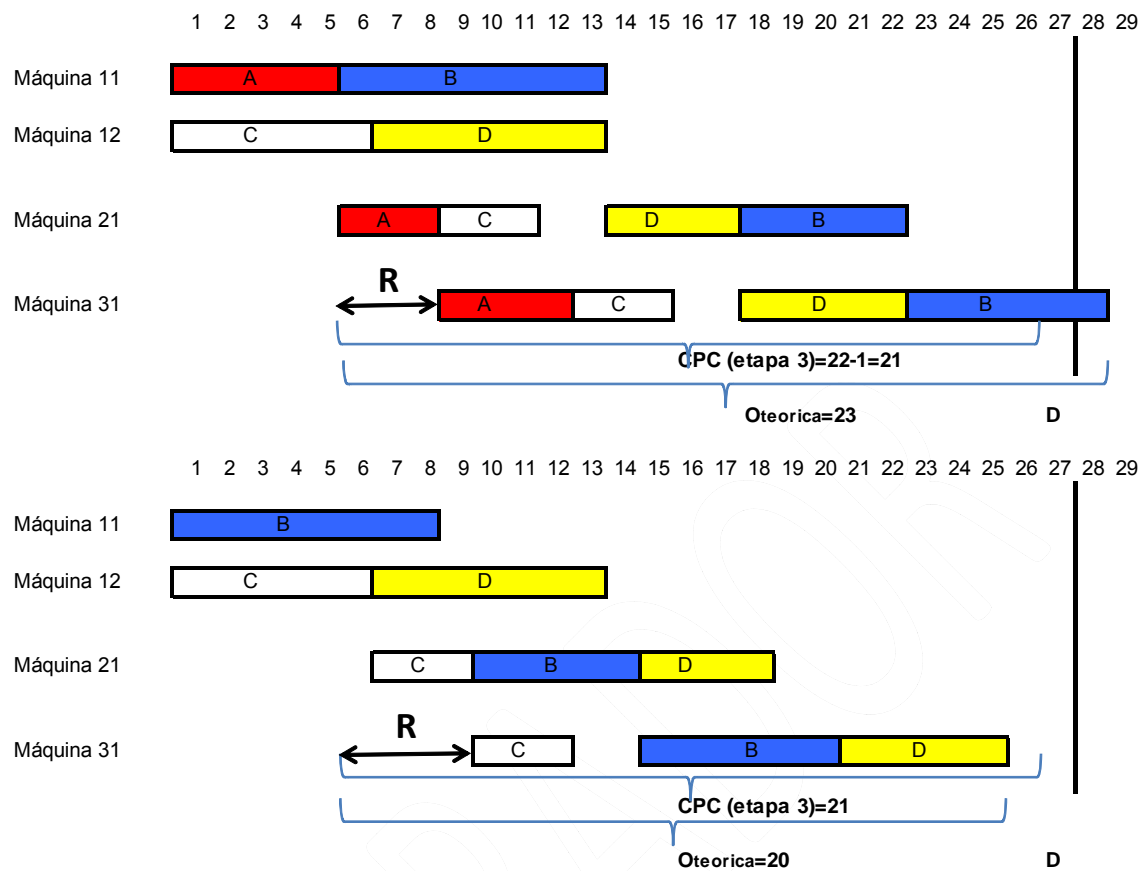


Figura 7-3. Ejemplo de Programa (Superior) sin respetar la fecha de finalización "D" y Programa (Inferior) que respeta "D" (elaboración propia).

- Situación B:** El Programa deberá ser modificado si la Ocupación que se ha alcanzado, $O_{\text{Teórica}}$ ó O_{Real} , es significativamente inferior a la CPC , es decir, si se prevee que pueden realizarse más trabajos de los propuestos por el planificador en el Plan de Fabricación. Se acompaña el caso con un ejemplo en la Figura 7-4. En la parte superior de dicha figura se indica como la etapa crítica es la 3, y que existe una CPC de 21 uds., 22 uds de CPS menos 1 ud. de MO, que se corresponden con el periodo 7-28. Una vez realizado el Programa (teórico) se observa como la Ocupación es de 18 uds. que se corresponden con el periodo 7-25 (el periodo 7-9 se corresponde con un retraso sobre lo previsto). Se observa que existe la posibilidad de aumentar la Ocupación en 3 uds más. Ya en la parte inferior de la figura se dispone de Programa productivo corregido mediante la incorporación del trabajo del trabajo D.

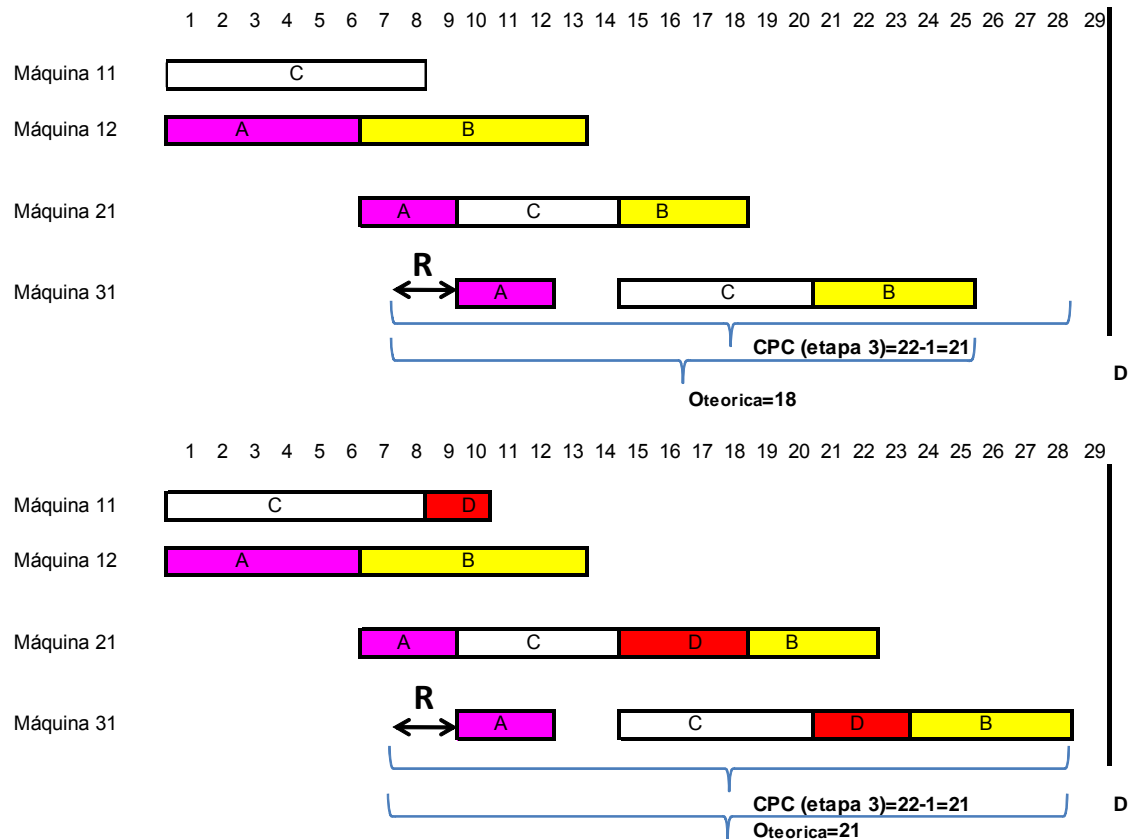


Figura 7-4. Ejemplo de Programa (Superior) con capacidad sobrante y Programa (Inferior) ocupación similar a la CPC (elaboración propia).

- Situación C:** El Programa debe ser modificado si aún siendo la Ocupación que se ha obtenido, O_{Teorica} ó O_{Real} , similar a CPC si el programador considera que se pueden realizar algunos trabajos adicionales a costa de reducir el margen operativo establecido en función del estado de ejecución del Programa y/o de la experiencia. En todo caso, se debe de respetar siempre la fecha de finalización prevista, c_i , de todos los trabajos del Programa propuesto y la Cota Superior de Capacidad. Se acompaña el caso con un ejemplo en la Figura 7-5. En la parte superior de dicha figura se indica como la etapa crítica es la 3, y que existe una CPC de 21 uds., 22 uds de CPS menos 1 ud. de MO, que se corresponden con el periodo 7-28. Una vez realizado el Programa (teórico) se observa como la Ocupación es de 19 uds. que se corresponden con el periodo 7-26 (el periodo 7-9 se corresponde con un retraso sobre lo previsto). Se observa que existe la posibilidad de aumentar la Ocupación en 2 uds más. Ya en la parte inferior de la figura se dispone de Programa productivo modificado mediante la incorporación del trabajo del trabajo D con el cual se supera la CPC en una unidad, es decir se consume el MO disponible, pero respetando al fecha de entrega D en todo caso.

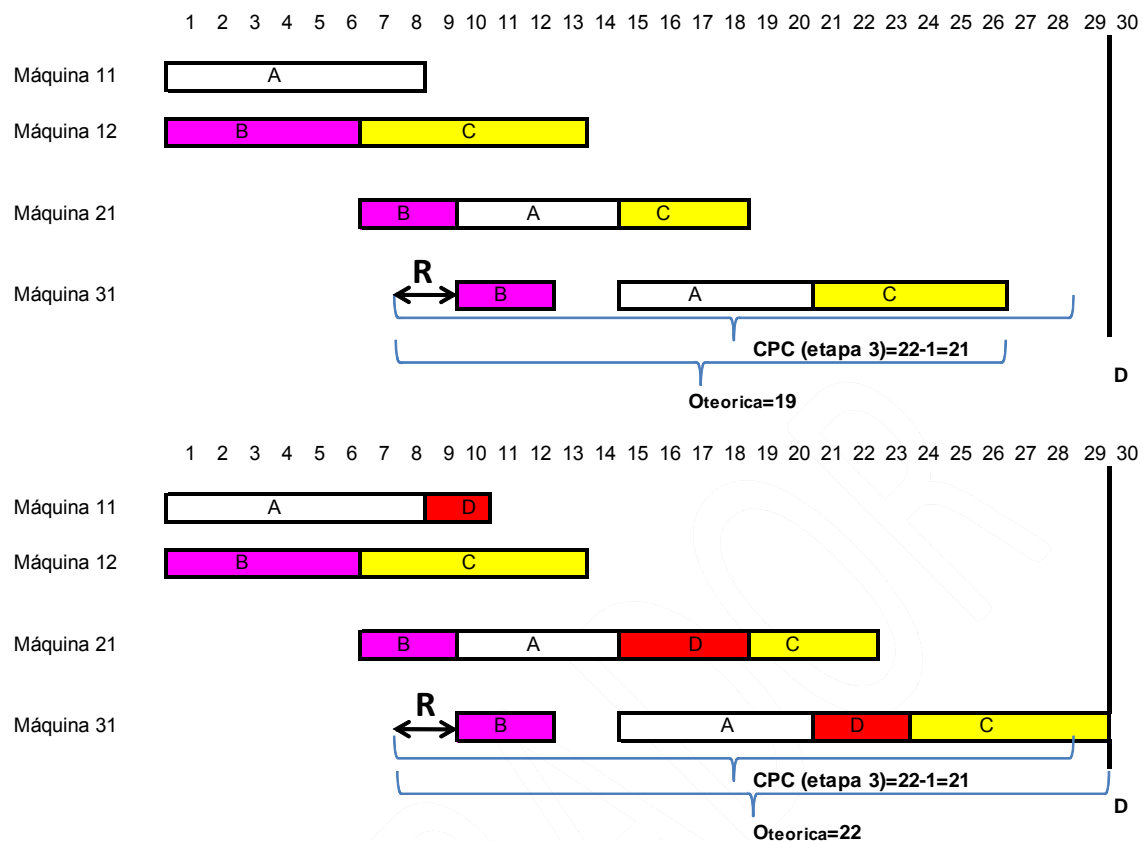


Figura 7-5. Ejemplo de Programa (Superior) con ocupación similar a la CPC y Programa (Inferior) superior a la CPC pero mayor tasa de productividad (m²/periodo) (elaboración propia).

Una vez detectadas las situaciones que se pueden producir hay que destacar el papel de los dos roles que se están analizando. En la Situación A1 y A2, la incidencia se detecta en el ámbito del programador, pero sea decidido que sea el planificador quien decide que modificaciones hay que hacer. Esto se debe a que se entiende que cual cambio en dicha situación tiene una afección comercial difícil de valorar desde la visión del programador.

En la Situación B y C, la incidencia se detecta desde el ámbito del programador, pero en este caso, se ha establecido que sea él mismo quien realice el Ajuste. En base a la información de periodos futuros contiguos al actual decide cuales son los trabajos que se adelantan. Se decide que esto sea así debido a que adelantar un trabajo^{vi} nunca implica un incumplimiento de la

^{vi} En todo caso, si el número de trabajos adelantados es muy elevado se deben considerar aspectos relativos a la gestión de almacenes, y costes del stock.

fecha de finalización prevista, y al mismo tiempo con dicha modificación se aumenta la tasa de productividad ($m^2/\text{periodo}$) de un periodo. En definitiva se respetan las restricciones de trabajo del planificador y se favorece la mejora del objetivo propuesto. Para este supuesto se desea suponer que los trabajos han sido de alguna manera priorizados por el planificador para facilitar la Toma de Decisiones en el ámbito del programador.

Como ya se ha dicho, se propone el establecimiento de un flujo bidireccional de información entre el planificador y el programador. Que ahora se puede concretar en los siguientes:

- El Programador recibirá un Plan de Fabricación del planificador, así como una priorización de trabajos.
- El planificador recibe del programador información sobre el Programa calculado y su evolución. Que en la Situación A1 y A2 implican una solicitud de cambio a la que el planificador debe responder, y que en las Situaciones C y D implica la autorización para que el programador realice una modificación que en caso de consolidarse debe ser posteriormente reconocida y asumida en el Plan Maestro de Producción.

Hasta este momento se ha realizado un análisis de las situaciones que se pueden dar durante la elaboración y ejecución de un Programa, es decir, se ha reflexionado sobre el “que puede ocurrir”, y también sobre quien debe intervenir. Sin embargo, en el proceso de Toma de Decisiones es relevante tener en cuenta “cuando se producen los acontecimientos”. En este sentido es fundamental realizar una reflexión sobre el instante en el que se modifica un Programa de Producción. Como se ha expuesto el programador está en continua observación y, si es necesario deberá plantear una revisión del Programa, con apoyo del planificador. Sin embargo, es importante diferencia entre dos tipos de Ajustes:

- Aquellos que se realizan antes de que el Programa se ponga en funcionamiento con el objetivo de reducir la diferencia entre CPC y $O_{teorica}$, es decir de aprovechar al máximo la capacidad el taller o planta.
- Aquellas que se realizan una vez el Programa se ha iniciado y existen trabajos en curso. En este caso, cada Programa que se realiza se basa en el que está en curso. Se trata de una Reprogramación.

En referencia a las Situaciones planteadas anteriormente, se puede decir en la fase de Establecimiento se puede producir la Situación A1, B y C. Si bien, la Situación A1 sólo se puede producir en esta fase, y las Situaciones B y C también se pueden producir también durante la ejecución del Programa. Si se produce la Situación B durante la ejecución será por un Desplazamiento a Izquierdas. La Situación A2 sólo se puede producir durante la ejecución del Programa.

En la Figura 7-6 se dispone de un Grafico de Estados en la configuración del Programa Productivo atendiendo a las características del problema planteado.

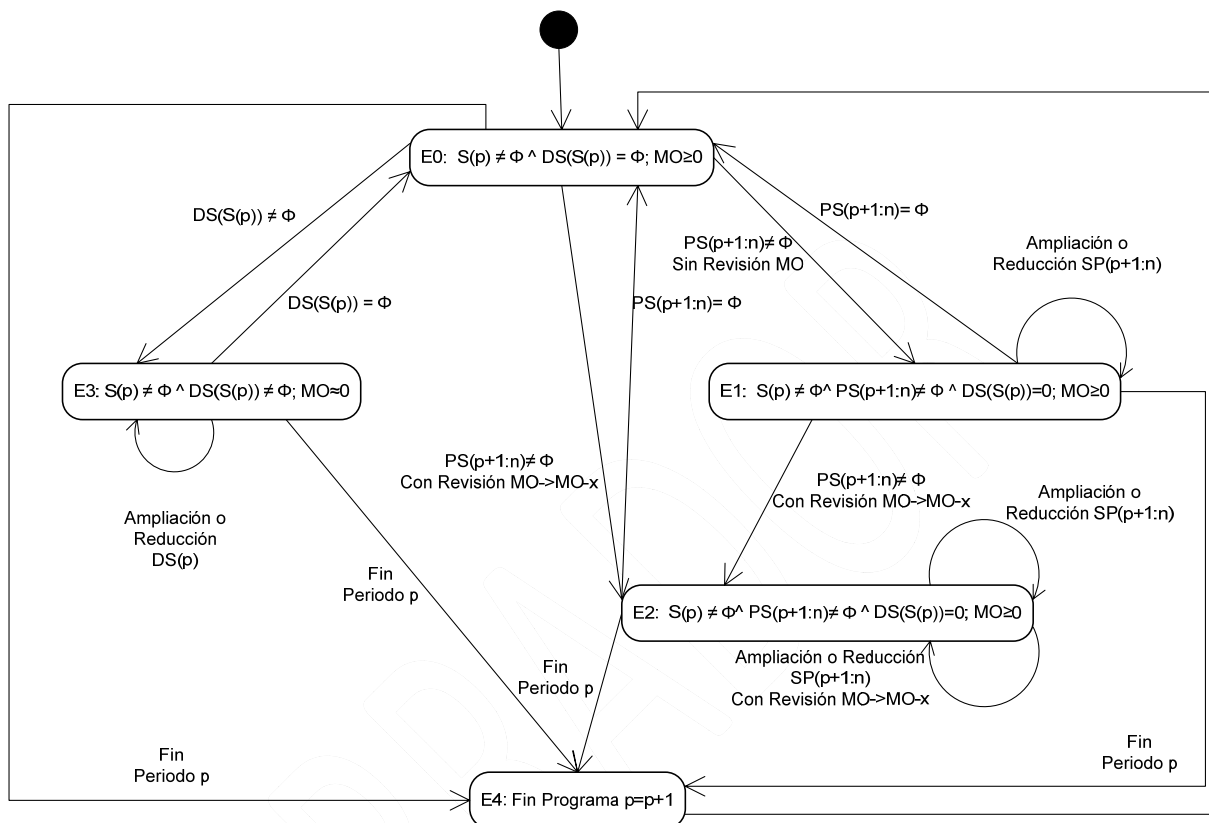


Figura 7-6. Estados en la configuración del Programa de la Producción (elaboración propia).

Como consecuencia de las Situaciones planteadas los responsables de Planificación y Programación realizarán una serie de modificaciones en el Programa y este sufrirá una evolución. Se ha considerado sólo las situaciones estables. En este sentido se ha supuesto que aquella situación en la que el Programa no respeta las fechas de finalización previstas en el mismo de algún trabajo es inestable y temporal, y se soluciona mediante un procedimiento que forma parte de la solución propuesta, hasta alcanzar un estado válido.

El Gráfico contiene cinco posibles estados identificados como E0, E1, E2, E3 y E4. La nomenclatura utilizada es la siguiente:

- $S(p)$: Conjunto de todos los trabajos que se han incluido dentro del periodo "p" en el Plan de Fabricación.

- PS (p+1:n): Subconjunto de trabajos que se han incluido dentro de algún periodo entre el “p+1” (siguiente al periodo en curso) y el periodo “n” (último periodo del horizonte) del Plan de Fabricación.
- DS (U): Subconjunto de trabajos de un conjunto cualquiera “U” que están retrasados en la finalización de su última operación con respecto a la fecha prevista de finalización establecida en el Programa de Producción.

Para entender el diagrama de estados hay que tener en cuenta que aunque inicialmente se parte de una situación en la que el programador intenta realizar un Programa Productivo exclusivamente con los trabajos incluidos para un determinado periodo del Plan de Fabricación posteriormente se pueden ir incluyendo trabajos de periodos posteriores, denominados trabajos adicionales. La fecha de máxima de finalización de esos trabajos adicionales en el ámbito del *Cálculo del Programa* es la del periodo en el cual han sido incluidos, es decir, el periodo adelantado. No obstante, en el ámbito de la *Gestión del Programa* se tiene consciencia de la fecha máxima de finalización establecida en el Plan de Fabricación, y por lo tanto existe una holgura para poder realizarlo.

Los Estados son los siguientes:

- Se parte de un Estado “E0” en el cual existe un conjunto de trabajos que se corresponden con el conjunto previsto en el Plan de Fabricación para el periodo “p” en curso, y no existe ningún trabajo cuya fecha de finalización supere a la programada. Si debido a una situación de tipo B o de tipo C se produjese un aumento del número de trabajos con respecto a lo previsto en el Plan de Fabricación se alcanzaría un Estado “E1” o “E2” (E2 sólo se modifica el MO), y si debido a una situación de tipo A2 hubiese que reducirlos se llegaría a un Estado “E3”.
- En el Estado “E1” se encuentran el estado cuando se tienen en cuenta los trabajos previstos del periodo p, en curso, más algún trabajo adicional de periodos siguientes, normalmente del contiguo. Además se considera que no existen trabajos cuya fecha de fin supere a la fecha establecida en el Programa de Producción. Si el número de trabajos aumenta, siempre que se mantenga la condición de no incurrir en retrasos el Programa, se permanece en el mismo estado. Si el número de trabajos disminuye el Programa permanece en el mismo estado, siempre que se tengan en cuenta trabajos de periodos sucesivos, en otro caso, si se diese una situación de tipo A2 se volvería al Estado “E0”. La situación A2 puede implicar un recorte en el Margen Operativo para poder hacer frente a los trabajos en curso, ya sean del periodo actual o de otro periodo. Sólo en caso de no disponer de MO se procede a cancelar trabajos iniciados que pertenecían inicialmente a otros periodos. También es posible pasar al estado “E2” si el programador decide revisar y reducir la cantidad de tiempo establecida para

el Margen Operativo. Esta situación se dará cuando se haya producido una evolución del Programa productivo, y se considere que en función del Programa pendiente de realizar el Margen Operativo actual es superior al riesgo que existe. Obviamente la reducción del Margen Operativo implica mayor disponibilidad de capacidad y la inmediata recuperación de los trabajos que hubieran sido cancelados con anterioridad en dicho periodo, si la nueva capacidad lo hace posible.

- El estado “E2” es similar al “E1”, ya que la composición de los trabajos es la misma. Pero en este estado se considera que ha habido una reducción del Margen Operativo, lo que implica una menor capacidad de hacer frente a los riesgos que puedan afectar a la ejecución del Programa productivo en curso. Se ha considerado que el Margen Operativo una vez reducido no se puede ampliar, ya que existirán nuevos trabajos que han ocupado dicho espacio y que no se pueden cancelar.
- En el Estado “E3” se encuentra aquel Programa que contiene sólo un subconjunto de los trabajos previstos inicialmente en el Plan de Fabricación. Además se considera que no existen trabajos cuya fecha de fin supere a la prevista en el Programa Productivo. Si se diese una situación de tipo B se llegaría a un estado “E0”.
- El Estado “E4” se alcanza cuando el Programa finaliza.

En función de cuando se modifique el Programa Productivo y que situación se produzca, dentro de las establecidas anteriormente, se han identificado una evolución bien diferenciada en referencia a la información que utiliza planificador (ver Tabla 7-1).

Establecimiento del Programa Productivo		Ejecución del Programa Productivo
Situación A1	Modificar el Plan Maestro.	No Aplica.
Situación A2	No Aplica.	Una opción entre: a) Modificar el Plan Maestro b) Modificar el Plan de Fabricación, y si se consolida el Plan Maestro.
Situación C ó D	Modificar el Plan de Fabricación, y si se consolida el Plan Maestro.	Modificar el Plan de Fabricación, y si se consolida el Plan Maestro.

Tabla 7-1. Cambios en el Sistema de Información del Planificador (elaboración propia).

Se distingue:

- En caso de que en la fase establecimiento se produzca una Situación A1 el Plan Maestro debe ser modificado. Ya se ha dicho que esta responsabilidad es del planificador. En este caso, toda la información que se deriva desde éste también cambiaría.
- En caso de que durante la ejecución del Programa se produzca una Situación A2 son varias las alternativas:
 - Si se trata de un Programa que se encuentra en un Estado “E0” el Plan Maestro de Producción debe ser modificado antes de continuar. Lo que implica una evolución al Estado “E3”. En todo caso, los trabajos cancelados serán considerados prioritarios a nivel de programación, y en caso de poder recuperar capacidad serán los primeros en incluir en el Programa sin consulta previa al planificador.
 - Si se trata de un Programa que se encuentra en un estado “E1” o “E2” debe evolucionar a un:
 - Estado “E1” con menos trabajos en cuyo caso se deberían consolidar los trabajos adelantos y modificar el Plan de Fabricación. Si los cambios se verifican una vez finalizado el Programa se debe cambiar el Plan Maestro de Producción.
 - Estado “E0” en cuyo caso no hay que modificar ninguna información de Planificación.
- En el caso de que se produzca una Situación “C” o “B” en fase de establecimiento o durante la ejecución se realiza una modificación del Plan de Fabricación, y si definitivamente se verifican los cambios cuando el Programa finalice, hay que realizar una modificación del Plan Maestro de Producción.

En este apartado se ha identificado “qué ocurre” en el transcurso del Ciclo de Vida de un Programa, “quien es responsable de actuar”, “cuando pueden ocurrir” las incidencias, y “que información se modifica”. No obstante, a lo largo de la propuesta se irán proponiendo algunos procesos decisionales adicionales para complementar la forma de actuar del programador de la Producción.

7.2.2 Consideraciones Generales sobre los Medibles

Hasta el momento la Gestión del Programa se asienta en un único medible que es la Tasa de Productividad promedio alcanzada. Este se basa en una correcta ejecución de los procesos que se ejecutan como soporte, es decir el proceso de Programación Predictiva y Reactiva.

La Programación Predictiva y la Programación Reactiva fueron definidas en el capítulo 5, mediante un modelo matemático, y se estableció un objetivo para cada una de ellas. La Programación Predictiva está centrada en el uso adecuado de los recursos, y se plantea como función objetivo (ver ecuación [5.1]) la minimización del makespan. La Programación Reactiva además del aprovechamiento de los recursos intenta evitar. No obstante, es importante reforzar la necesidad de incluir algunas medidas adicionales para evitar la realización de cambios en el Programa Productivo que puedan comprometer compromisos inmediatos con proveedores. En este sentido una manera de evitar dicho comportamiento es ligar la función objetivo relacionada con el proceso de Programación Reactiva de forma que el cálculo del Programa Reactivo penalice este tipo de opciones. Debido a la relación jerárquica que existe entre el proceso de Programación Reactiva y la Gestión del Programa evitando dicho efecto en el nivel decisional inferior se repercutirá positivamente en el nivel superior.

Como se introdujo en el punto 5.4.3 del capítulo 5, relativo al Modelo Reactivo, dicho impacto se puede medir en cada Programa mediante el concepto de "Adelanto". En este sentido, se debe hacer referencia a la ecuación [5.13] que posibilita que el programador pueda ajustar el factor k sobre el Modelo de Programación Reactiva con el objeto de controlar el grado de penalización sobre los trabajos adelantados.

No obstante existen una acción claramente indeseable como es el incremento de la tasa de trabajos cancelados en un periodo productivo. Los trabajos cancelados promedio dan una idea de cual es el grado de estrés al que está sujeto el proceso de Programación de la Producción. Un aspecto que no ha sido valorado hasta el momento, pero que debe ser considerado, es que el programador puede verse obligado a cancelar algunos trabajos debido a que no de tiempo a finalizarlos dentro del periodo correspondiente al Programa en curso. La cancelación es consecuencia de la aparición de un evento no previsto que impida que incluso consumiendo todo el Margen Operativo existente sea imposible acabar a tiempo. La cancelación no ha sido valorada porque se puede ver más bien de una consecuencia, y no una decisión. Sin embargo, esto sólo es cierto si la cancelación se produce cuando el Margen Operativo no ha sido modificado, es decir que es consecuencia únicamente de un exceso de retrasos para el Margen Operativo considerado inicialmente. No obstante, también puede ser consecuencia de una excesiva reducción del Margen Operativo con el objeto de ampliar el número de trabajos en curso, y la posterior aparición de algún retraso que transformen dicha modificación en demasiado optimista. Por este motivo se propone el uso de un indicador que informe sobre la

cancelación de trabajos correspondientes al Programa inicial, y otro que informe sobre cancelaciones de trabajos adicionales. Este indicador se podría emplear como complemento del dato correspondiente a la Tasa de Productividad promedio.

7.2.3 Una Visión General del Proceso

En este punto se han expuesto las líneas principales de lo que debe configurar el marco de trabajo del programador en lo referente a la *Gestión del Programa*.

En este apartado se han expuesto las líneas generales relativas al marco de trabajo del programador. El objetivo del programador debería ser mantener la Tasa de Productividad promedio con el valor más alto posible a través de un conjunto de acciones que se engloban dentro del proceso de Gestión del Programa. En un escenario básico el programador inicia su gestión con un Programa Predictivo con el que puede alcanzar una elevada cota en la utilización de los recursos. A partir de ese instante el programador debe revisar el Margen Operativo existente para abordar los posibles cambios.

La evolución del Programa puede propiciar dos tipos de situaciones. La primera situación es la que obliga a modificar el Programa para hacer frente a un evento no considerado inicialmente, lo que posiblemente obligue a recurrir al consumo de Margen Operativo para evitar cancelaciones. El segundo escenario es la reducción del Margen Operativo por voluntad propia, adaptarse al estado evolutivo del Programa y una posible reducción del riesgo existente de no cumplir el Programa en curso. En ambos el programador deberá adoptar una adecuada política de adición de trabajos al Programa en curso o de cancelación de trabajos en el Programa en curso.

En cada Reprogramación, si la ocasión es propicia para ello, el programador debe intentar incluir aquellos trabajos con los que mantenga más tiempo ocupadas las máquinas si con ello alcanza una mayor cantidad producida. Se trata por lo tanto, de seleccionar aquellos trabajos que más “rentabilidad” ofrecen entre el conjunto de trabajos elegibles, entendida en función de la Tasa de Productividad promedio. Dependiendo de la ocasión el conjunto de trabajos elegible ofrecerá mayor o menor libertad de selección para el programador.

Como se acaba de comentar un aspecto básico con el que el programador debe contar es con el Margen Operativo. Se debe considerar que a medida que el Programa avanza en el tiempo el Margen Operativo puede reducirse progresivamente, y que este se pueden ajustar a las eventualidades que se pudieran dar en el tiempo restante del periodo en curso. De esta forma el programador puede ir ajustando la Capacidad Considerada (CPC) si con ello puede

incrementar el número de m^2 efectivos realizados sin incurrir en riesgos graves de incumplimiento del Programa.

No se puede considerar, ni se ha pretendido tampoco, que el método de gestión haya quedado definido. En cada caso habrá que establecer los mecanismos pertinentes, y en el caso de este trabajo esto se realizará como parte de la definición de la Plataforma software que soportará las tareas de *Gestión del Programa*.

En la Figura 7-7 se identifica que en líneas generales la Gestión del Programa Productivo a lo largo de un determinado periodo es proceso en el cual a partir de un conjunto de órdenes seleccionables va proponiendo diversos Programas Productivos. Este proceso estará condicionado por un determinado criterio de selección de órdenes que se empleará si es necesario en los diversos procesos de reprogramación, la existencia de un Margen Operativo que condiciona la capacidad de reacción frente a posibles eventos, y un cierto valor de K que penaliza cierto tipo de modificaciones en los Programas Productivos.

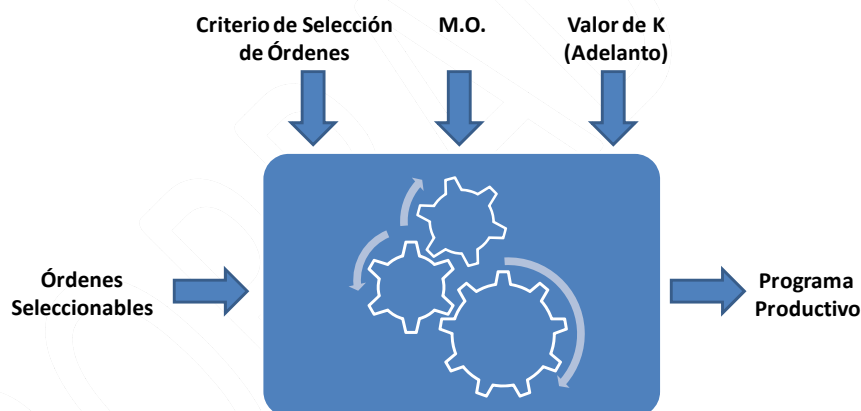


Figura 7-7. Control del proceso de Gestión del Programa (elaboración propia)

El desarrollo del contenido del proceso, que en la Figura 7-7 se representa como una caja negra, estará condicionado por el objetivo expresado tanto en el capítulo 5 como en este apartado. Dichos contenidos estarán constituidos por un conjunto de procesos que coordinados entre sí permitirán realizar la Programación Predictiva, y en su caso Reactiva que dan lugar a los diversos Programas Productivos.

En el próximo apartado se realizará el Análisis de Requerimientos necesario para abordar el desarrollo de una Plataforma Software capaz de automatizar la Gestión de la Producción teniendo en cuenta la formulación del problema que se ha realizado. Se trata facilitar la

coordinación entre planificador y programador implementando las diversas actividades descritas entre las que destacamos el Cálculo y Ajuste de los Programas Predictivos y Reactivos.

7.3 Análisis de los Requerimientos de la Plataforma IPSU-MAS

En este apartado **se identifican los requisitos del sistema que se pretende modelar**. Para ello se van a detallar los Procesos del Sistema de Fabricación, el alcance del sistema que desea cubrir, condiciones de operación, y objetivos. Se corresponde con la fase 2 de la metodología propuesta.

El modelo de definición de requerimientos describe “que” y “como” se deben hacer las cosas. Para proceder en la fase de Modelado de Requerimientos se ha seleccionado la metodología CIMOSA (ESPRIT Consortium AMICE 1993;K.Konsanke 1995;M.Zelm, F.Vernadat, & K.Konsanke 1995) ya que es una especificación que se considera presenta el enfoque más estructurado y completo. CIMOSA se desarrolló a partir de un proyecto tipo ESPRIT financiado por la Comisión Europea y pretende proporcionar una arquitectura para sistemas abiertos bajo la filosofía CIM. A continuación se realiza una breve introducción.

7.3.1 La arquitectura CIMOSA

Con el objetivo de permitir la integración empresarial los principios arquitectónicos de CIMOSA se basan en la generalización del concepto de diferenciación:

- Diferenciación entre la Representación del Usuario y la Representación del Sistema. Lo que restringe el impacto de los cambios y proporciona flexibilidad a la organización.
- Diferenciación entre Control y Funciones, haciendo posible la revisión del comportamiento para identificar circunstancias cambiantes sin alternar la funcionalidad implementada.
- Diferenciación entre Funciones e Información, para facilitar la Integración, Aplicación, Portabilidad y Mantenimiento.

La estructura de modelado CIMOSA, ver Figura 7-8, se basa en:

- Una dimensión genérica, en la cual se encuentra el:

- *Nivel genérico*: Un catálogo de bloques constructivos genéricos CIMOSA.
- *Nivel parcial*: Una librería de modelos parciales aplicables a clases de propósitos particulares.
- *Nivel particular*: Contiene el modelo particular de una empresa construido de bloques básicos y modelos parciales.
- Una dimensión de modelado, que consta de:
 - *Nivel de modelado de la definición de Requerimientos* para la recopilación de los Requerimientos de Negocio.
 - *Nivel de modelado del diseño de especificaciones* para la especificación optimizada y la representación los os Requerimientos de Negocio.
 - *Nivel de la descripción de la implementación* para describir completamente un sistema CIM y todos los módulos implementados, seleccionados e especificados.
- Una dimensión de vistas para describir los diferentes aspectos de la empresa. Dentro de ésta se dispone de:
 - La vista funcional que describe la estructura funcional y su estructura de control (reglas) las cuales definen el flujo de la acción dentro de la empresa.
 - La vista informacional la cual describe la información requerida.
 - La vista de recursos que describe los Requerimientos sobre los recursos.
 - La vista organizacional que es la descripción de las estructuras organizacionales de la empresa.
- En este trabajo se ha considerado especialmente interesante la utilización de las cuatro vistas en la arquitectura CIMOSA. Estas vistas permiten ir filtrando la totalidad del modelo para analizar cada uno de los enfoques por separado. De esta forma se reduce la complejidad del proceso y proporciona un modelo más fácilmente entendible.

Aunque en este apartado sólo se desarrollará el Modelo de Requerimientos hay que resaltar que CIMOSA cubre otros aspectos de la vida del software como son el Modelado de las Especificaciones de Diseño y el Modelo de Implementación. Estos otros modelos no se consideran en esta tesis ya que este aspecto es cubierto ampliamente y de forma más

adecuada en relación a la solución que se propone por la metodología INGENIAS (Pavon, J. & Gomez-Sanz, J. 2003) para el Modelado de Sistemas Multiagente.

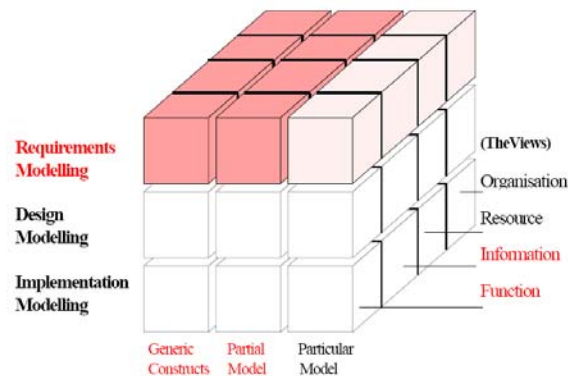


Figura 7-8. Marco de modelado CIMOSA(Kosanke, K. 1995).

Antes de proceder a la elaboración del Modelado de Requerimientos es necesario identificar con cierto detalle los conceptos clave de la arquitectura CIMOSA, para posteriormente poder interpretar correctamente el modelo. La dimensión de de la vistas es la que contiene la mayor riqueza conceptual.

7.3.1.1 La Vista Funcional

Se utiliza para organizar la empresa como un conjunto de Dominios Funcionales que interactúan entre sí, y describir los procesos actuales, las condiciones de disparo de los eventos y su ejecución.

Para una entidad sujeta a Modelado se plantean inicialmente un conjunto disjunto de Dominios de Empresas (DM-Enterprise Domain). Cada DM está compuesto por procesos completos llamados Dominios de Proceso (DP-Domain Process) que se relacionan con otros similares. Así mismo existen eventos que son los responsables de los procesos de cambio en el sistema. Los eventos pueden ser generados por elementos internos de la empresa como recursos o actividades, por temporizadores, o por elementos externos. Los eventos pueden ser esperados (deterministas o programados) o pueden ser inesperados (perturbaciones, disfunciones, etc.).

Cada DP puede incluir otros procesos (sub-procesos), así como Actividades Empresariales (EA-Enterprise Activities) en el nivel de descomposición funcional más bajo. Las EA identifican los pasos elementales de un proceso. El proceso lógico se describe en forma de red de actividades definidas mediante lo que se denomina Conjunto de Reglas de Comportamiento (BRS-Behavioral Rule Set). En CIMOSA las reglas se definen mediante operadores.

Cada EA puede descomponerse en un paso posterior en acciones denominadas Operaciones Funcionales (FO-Functional Operation). Una FO debe ser ejecutada por exactamente un actor o recurso activo de la empresa. Cada EA transforma un estado de entrada en un estado de salida mediante la activación de las FO. Cuando se completa se genera un estado final (ES-Ending Status). Cada actividad es definida por un conjunto de habilidades o Capacidades necesarias para su ejecución (Capabilities Set).

7.3.1.2 La Vista Informacional

Se utiliza para describir las entidades usadas por actividades y procesos. La información y los objetos fluyen con sus correspondientes restricciones como un modelo de datos conceptual para el soporte del proceso de integración. Desde el punto de vista informacional los Estados de Entrada y Salida en las actividades en curso son definidos en términos de Vista de Objetos (OV-Object View). Un OV representa un estado particular de uno o más Objetos de Empresa (EO-Enterprise Objects).

Un EO denota cualquier clase de objeto utilizable por la empresa (órdenes, listas de materiales, etc.). En el modelo una EO se define por su lista de propiedades denominada Elementos de Información (IE-Information Elements) y por mecanismos de abstracción.

Los IE son propiedades de los datos que están sujetas a restricciones de integridad que permiten expresar y validar la viabilidad de los datos.

7.3.1.3 La Vista de Recursos

Se usa para declarar y definir los objetos que pueden jugar el papel de recurso en la ejecución de actividades. CIMOSA define un constructor genérico denominado "Recurso" que puede ser ajustado a dos clases fundamentales: Componentes y Entidades Funcionales (FE-Functional Entities). Los primeros son reservados para elementos pasivos (herramientas, etc.), mientras

los segundos son utilizados con recursos activos que pueden ejecutar OF. Las FE pueden ser divididas a su vez en tres entidades principales: Humanos, Máquinas y Aplicaciones.

Todas las clases de recursos se pueden caracterizar por su Conjunto de Capacidades (CS-Capability Set), que representan el conjunto de habilidades que pueden proporcionar.

7.3.1.4 La Vista Organizacional

Se utiliza para describir la organización y la estructura decisional de la Entidad de Negocio, y asignar responsabilidades y autoridades a los componentes de una arquitectura particular. CIMOSA proporciona dos constructores principales para esta vista: la Unidad Organizacional (OU-Organizational Unit) y la Celda Organizacional (OC-Organizational Cell). La primera define un centro de decisión elemental y la segunda permite organizar la OU en celdas. Las OC se pueden agrupar en diversos niveles de celdas.

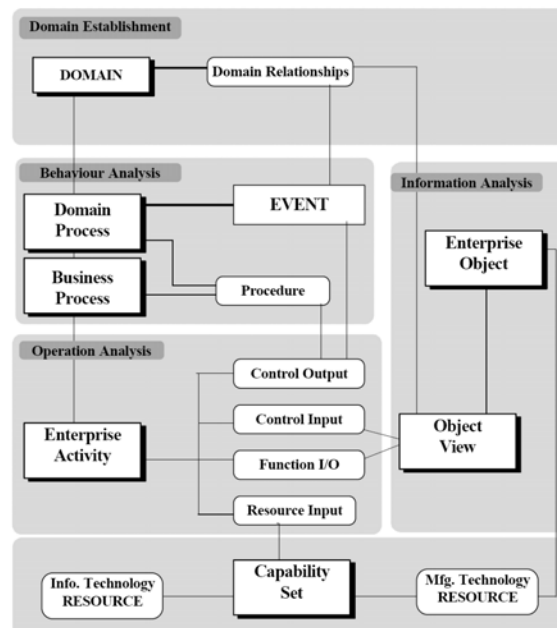


Figura 7-9. Principales conceptos CIMOSA y su relación.(RGCP 1999)

Hasta el momento se ha comentado la arquitectura CIMOSA, los constructores y las relaciones de los mismos en cada vista. La Figura 7-9 muestra la relación de los mismos en el entorno de sus vistas y fuera del mismo.

7.3.2 Análisis de Requerimientos

Una vez los conceptos relacionados con la arquitectura CIMOSA, en general, han sido expuestos, se procederá a identificar cuales son los elementos arquitectónicos necesarios para alcanzar el objetivo: la definición del Modelo de Requerimientos. Los elementos necesarios se pueden identificar de forma intuitiva en base a la observación de las diferentes dimensiones del CUBO CIMOSA. Así de la Dimensión de Modelado, hay que hacer uso de la capa de análisis de Requerimientos. De la Dimensión Genérica se debe hacer uso del nivel particular, ya que se va a realizar un análisis concreto. Y finalmente, de la Dimensión de Vistas se requerirán las Vistas de Funcionalidad, Información y Recursos. *CIMOSA no incluye la Vista Organizacional en el Modelo de Requerimientos.*

El proceso de constructivo se inicia con la recopilación de Requerimientos y su contenido en términos de Requerimientos Funcionales y la generación de todas las Vistas de forma iterativa. En concreto (RGCP 1999) sugiere seguir los siguientes pasos:

- Establecer el Dominio: El usuario debe describir correctamente el subconjunto de la empresa que sea analizar. Para ello debe establecer los Objetivos, las Restricciones, los Límites, los Eventos y los Procesos del Dominio.
- Realizar un Análisis del Comportamiento: Después de identificar los Eventos, el usuario debe definir los Objetivos asociados a los Procesos del Dominio, para descomponerlos en sub-funciones y definir como éstas son secuenciadas para alcanzar el resultado deseado. No se entra en la diferenciación entre Procesos de Negocio y Actividades. En esta fase la Vista Funcional puede quedar finalizada en lo que se refiere al aspecto conductista pero no en el puramente funcional.
- Realizar un Análisis Operacional: En este paso se completa la Vista Funcional estableciendo completamente las entradas y salidas vinculadas a las Actividades Empresariales. Se debe establecer la entrada Funcional, de Recursos y de Control, y la salida de Control y de Recursos. En este punto el usuario debe definir los recursos en términos de Capacidades necesarias y las Clases de Recursos que las pueden soportar.
- Realizar un Análisis de la Información: En este paso el usuario debe establecer vínculos entre la Vista Funcional y de Información. Debe de describir las Vistas de Objetos (OV) que necesitan como entrada las Actividades y consolidarlos con los Objetos de Empresa (EO). Los EO pueden formar niveles de dependencia entre esos, de manera que un EO forme parte de otro u otros EO. La construcción de este nivel se lleva a cabo realizando primero la descripción de los OV y después la definición y estructuración de los mismos. Los elementos de este nivel se pueden generalizar o agregar en un proceso de abstracción.

- Realizar el Análisis de Recursos: Una vez finalizado el análisis operacional es necesario dotar a las Actividades de las Capacidades establecidas. En esta fase se trata de identificar los Conjuntos de Capacidades para reducir su número y consolidar Clases de Actividades que necesitan las mismas Capacidades.
- Verificar la Consistencia del Modelo: Para asegurar la consistencia del modelo CIMOSA proporciona una fuerte relación entre los constructores utilizados en las diferentes Vistas. La Figura 7-10 muestra la relación entre los elementos recomendados para el modelo de Requerimientos.

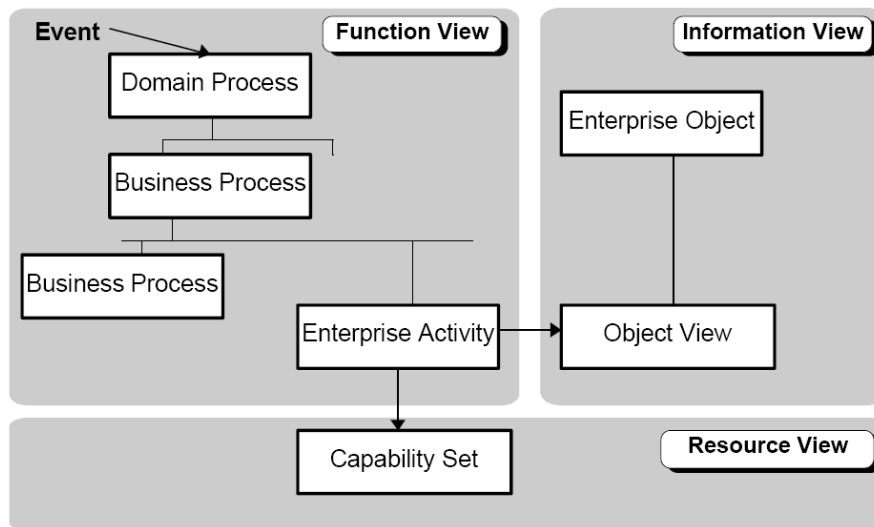


Figura 7-10. Relación entre las vistas del modelo de requerimientos CIMOSA (RGCP 1999).

El objeto del siguiente apartado será el desarrollo del Modelado de Requerimientos para el problema expuesto en el capítulo 5, considerando también los aspectos mencionados en el apartado 7.2, utilizando la propuesta CIMOSA y siguiendo los pasos que se acaban de indicar. En el anexo I se incluye la especificación de todos los elementos considerados en el formato y nivel de detalle propuesto por (Berio, G. & Vernadat, F. B. 1999b).

7.3.2.1 Establecimiento del Dominio

Se ha identificado un solo Dominio: La Programación de la Producción. Este es responsable de generar un Programa Inicial o Predictivo en el cual se secuencian y temporizan los diferentes trabajos incluidos en un periodo determinado, indicando el Recurso sobre el cual van a realizarse las diferentes operaciones, así como la fecha de inicio y fin de la misma. Así mismo

es responsable de reaccionar ante determinados eventos y modificar el Programa Inicial en función de los mismos, en lo que se denomina Programación Reactiva.

El Dominio queda definido de forma por los siguientes campos:

- **Objetivos del Dominio:**
 - Obtener un Programa Inicial o Predictivo de la Producción para un periodo determinado.
 - Obtener un Programa Reactivo de la Producción en función de los eventos percibidos.
- **Restricciones del Dominio:** Son las identificadas en la definición del problema, y más específicamente en el capítulo 5 en el cual se realiza la formulación de los Modelos Predictivo y Reactivo.
- **Procesos del Dominio:** DP-Gestión del Programa Productivo.
- **Eventos:**
 - Recibidos en este Dominio:
 - EV-Nuevo periodo.
 - EV-Reinicio Programa.
 - EV-Desplazamiento.
 - Generados en este Dominio:
 - EV-Inconsistencia de datos.
 - EV-Reducción de Trabajos.

La definición del Dominio, tanto de sus elementos básicos como de sus objetivos y restricciones, se ha realizado en base a los conceptos expuestos en la propia formulación del problema. Por este motivo los objetivos son los relativos a los fines que le son propios a la Programación Predictiva y Reactiva, igual que las restricciones. Aunque pueden existir diversas maneras de enfocar la subdivisión del Dominio en Procesos de Dominio, en este caso se ha creído conveniente establecer un único Proceso de de Dominio. Un único Proceso de Dominio es suficiente para abordar el ámbito que se desea definir, y simplifica el entendimiento del mismo.

En cuanto a los eventos, se establecen tres eventos que pueden ser generados desde fuera de este dominio pero que deben ser identificados e implican la puesta en marcha de alguna

actividad. El evento “Nuevo periodo” que está asociado al inicio de cada periodo de Programación e implica el lanzamiento del Proceso de Negocio de Programación Predictiva. El evento “Desplazamiento” generado desde el Dominio de Control de Operaciones, y que alerta de un desplazamiento a izquierdas o derechas del Programa en curso, lo que implica la puesta en marcha del Proceso de Negocio de Programación Reactiva. El evento “Reinicio Programa” se incluye para dar mayor cobertura funcional al sistema. Permite al programador reiniciar el sistema cuando lo considere oportuno, con ello puede poner en el estado inicial (sin órdenes en curso y máquina disponibles) de un nuevo periodo el sistema.

También existen dos eventos generado desde el propio Dominio. Es primer evento “Inconsistencia de datos”, permite avisar al planificador de que es imposible establecer un Programa con los trabajos que están planificados. De esta forma el planificador puede reaccionar proponiendo otras opciones. En segundo evento, “Reducción de Trabajos” pone en conocimiento del programador que el Programa no resulta posible Programar los trabajos actuales sin incurrir en una no factibilidad, y que se solicita una reducción.

Una vez, dado el primer paso en el proceso, e identificado los elementos de mayor nivel se debe profundizar en primer lugar en la definición de los aspectos relacionados con el comportamiento. En el paso siguiente se va a empezar a de definir con claridad la funcionalidad que debería incluirse en el sistema.

7.3.2.2 Análisis del Comportamiento

En este apartado se empiezan a diferenciar las dos grandes funciones asignadas al Proceso de Dominio: la Programación Predictiva y la Programación Reactiva. Cada una de ellas se identifica como un Proceso de Negocio diferenciado.

Por un lado, la Programación Predictiva es responsable de generar un Programa inicial o preventivo en el cual se secuencian y temporizan los diferentes trabajos incluidos en un periodo determinado, indicando el recurso sobre el cual van a realizarse las diferentes operaciones, así como la fecha de inicio y fin de la misma.

- **Objetivos:** Obtener un Programa inicial o Predictivo, es decir, asignar, secuenciar y temporizar todos los trabajos pendientes para un periodo de tiempo en condiciones deterministas. Debe maximizar el uso de los recursos cumpliendo las fechas más tardías de finalización de los trabajos.

- **Restricciones:** Las expresadas en el modelo formulado en el apartado 5.4.1 del capítulo 5.
- **Comportamiento del Proceso:** Ver diagrama de actividades de la Figura 7-11.
- **Estados de finalización:**
 - Cancelación de Programación por inconsistencia de datos.
 - Programación rechazada.
 - Programación validada.
- **Funciones comprendidas:**
 - Revisión y carga de datos.
 - Cálculo del Programa Predictivo.
 - Ajustes.
 - Validación/Rechazo.

Dentro del caso particular al cual se ciñe la Definición de Requerimientos que se deben incluir ciertas consideraciones que permitan la reutilización de la definición para el mayor número de casos posible. En este sentido se considerado que la Programación Predictiva puede finalizar no sólo con un Programa que sea validado y entre en ejecución, sino con las opciones siguientes:

- Cancelado por que el sistema de información no es capaz de proporcionar un conjunto de datos consistentes para poder iniciar el Proceso de Establecimiento de un Programa. En este caso, se trata de tener en cuenta una circunstancia que puede plantearse en ciertos entornos empresariales donde los datos pueden no ser fácilmente accesibles.
- No se valida el Programa ya que se considera que finalmente no se cree oportuno ponerlo en ejecución.
- El Programa se valida y se pone en funcionamiento en para el periodo correspondiente.

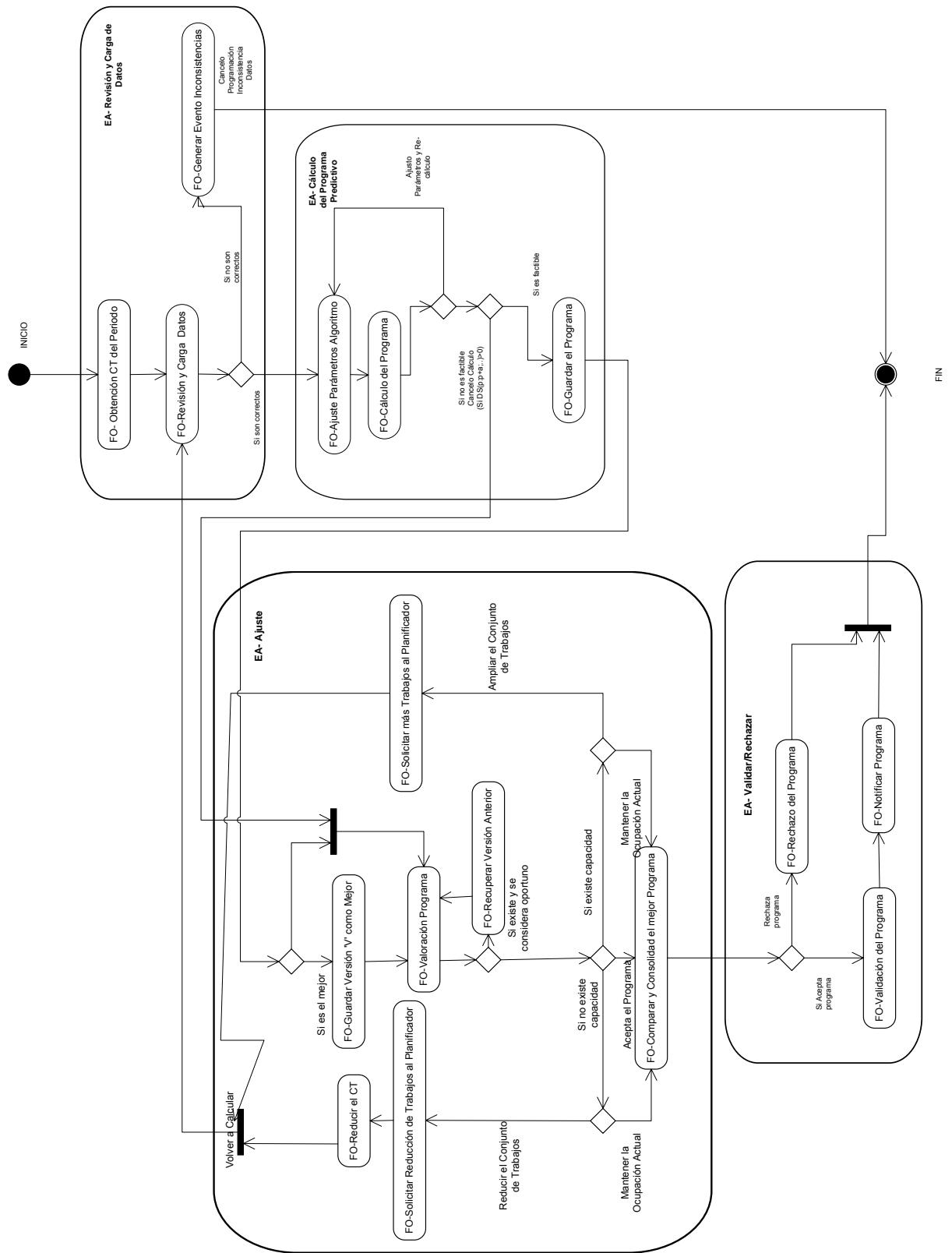


Figura 7-11. Diagrama de Funcional del Proceso de Negocio de "Programación Predictiva" (elaboración propia).

Las funcionalidades básicas del proceso de negocio, denominadas por CIMOSA Actividades Empresariales, se han desgranado en Operaciones Funcionales con el objeto de dar mayor detalle de cada una de ellas.

La primera Actividad que se ejecutaría cuando el Proceso de Negocio se inicia es la de "Revisión y Carga de Datos". Esta actividad se responsabiliza en primer lugar de la recogida de datos, que en ocasiones pueden estar dispersos e incluso en soportes diferentes. En segundo lugar de la conversión de formatos para dejar todos los datos disponibles de forma consistente, posiblemente en un Data-WareHouse. Si no consigue alcanzar la consistencia de datos genera un evento para alerta de ello, y el Proceso de Negocio se detendría en este punto.

La segunda Actividad Empresarial de este proceso es el "Cálculo del Programa Predictivo". Las Actividades comprendidas están destinadas al ajuste de parámetros del algoritmo que se utilice, la ejecución del propio algoritmo y guardar los resultados obtenidos. Se pueden reajustar los parámetros después de conocer los resultados del cálculo en caso de no obtener el resultado deseado, ya sea por no factibilidad o por baja eficiencia. Esta actividad proporciona un Programa factible o no factible que debe ser valorado en la siguiente Actividad Empresarial.

La tercera Actividad del Proceso de Negocio de Programación Predictiva es el "Ajuste". En esta Actividad se trata de valorar el Programa obtenido en el paso anterior y decidir si se acepta, si aún siendo un Programa válido es posible mejorarlo (ampliando el número de trabajos y/o reduciendo la capacidad considerada, y por lo tanto el margen de maniobra), ó si no es un Programa aceptable y debe ser mejorado (reduciendo el número de trabajos y/o aumentando la Capacidad Considerada, "CPC", y reduciendo por lo tanto el margen de maniobra). También se consideran las operaciones funcionales destinadas a mantener el mejor Programa posible a criterio del programador y recuperarlo al final del Proceso de Ajuste, así como el de mantener y recuperar diversas versiones intermedias que permitan continuar el Proceso de Ajuste no necesariamente desde el último resultado obtenido.

En la última Actividad se procede a la Validación del Programa obtenido para su establecimiento como Programa Predictivo en el periodo correspondiente, como al rechazo del mismo por no ajustarse a los criterios de calidad del programador.

Por otro lado, el Proceso de Negocio de Programación Reactiva es responsable de responder ante determinados eventos y modificar el Programa base en función de los mismos, en lo que se denomina Programación Reactiva. Para ello debe identificar la perturbación, actualizar el estado del sistema y volver a generar un Programa a partir del anterior.

- **Objetivos:** Reaccionar eficazmente contra imprevistos que puedan afectar a la Programación de la Producción en el taller. Reprograma la Producción reparando el Programa base teniendo en cuenta las nuevas condiciones. Debe maximizar la robustez del nuevo Programa cumpliendo las fechas más tardías de finalización de los trabajos.
- **Restricciones:** Las expresadas en el modelo formulado en el apartado 5.4.3. del capítulo 5.
- **Comportamiento del Proceso:** Ver diagrama de actividades de la Figura 7-12
- **Estados de finalización:**
 - Cancelación de Programación por inconsistencia de datos.
 - Programación validada.
- **Funcionalidades comprendidas:**
 - Identificación y valoración.
 - Establecimiento de la estrategia de Reprogramación
 - Revisión y carga de datos.
 - Cálculo del Programa Reactivo.
 - Ajustes /Control de la estrategia de Reprogramación.
 - Validar/Rechazar.

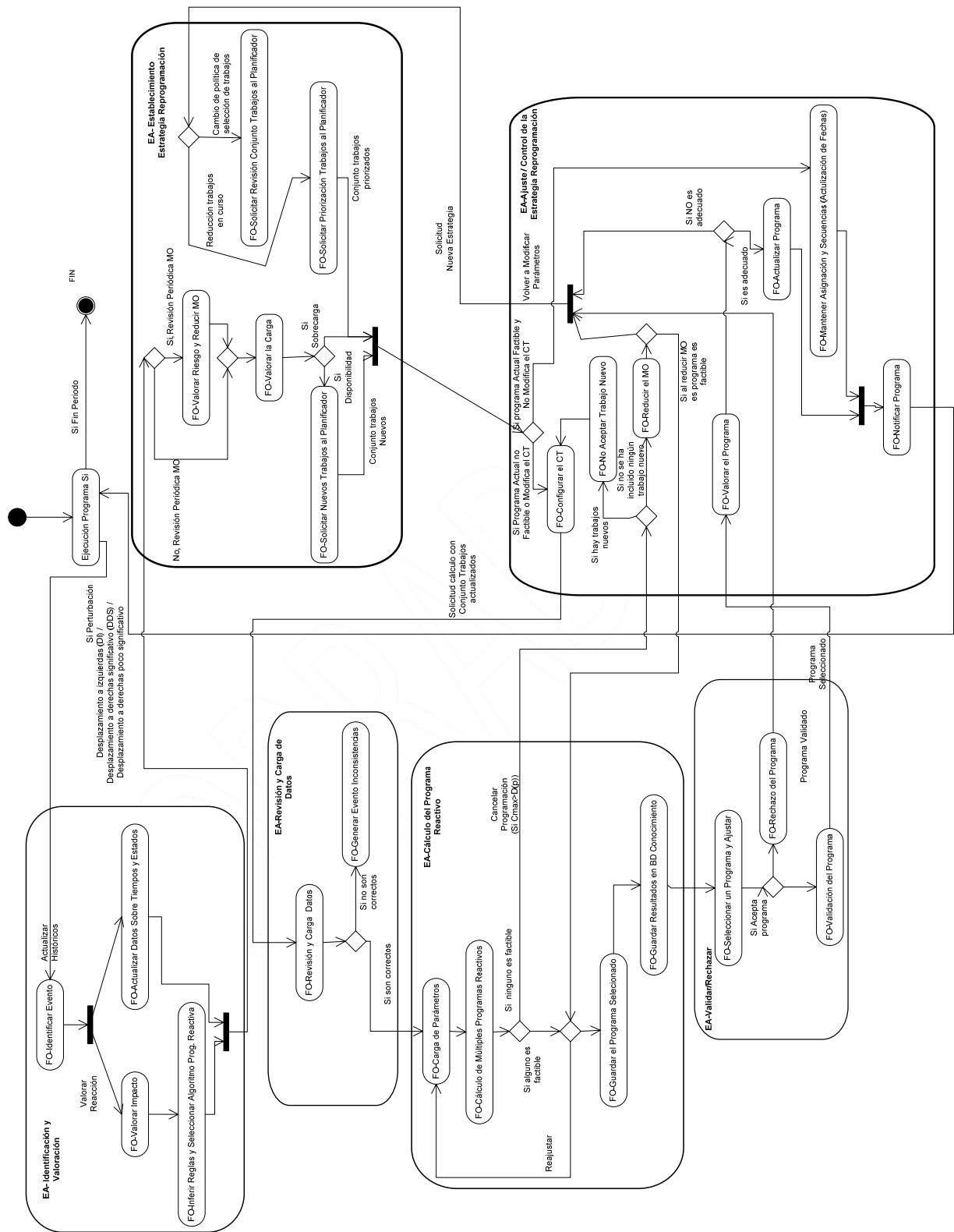


Figura 7-12. Diagrama de Funcional del Proceso de Negocio de "Programación Reactiva" (elaboración propia).

Se ha considerado que mientras la Programación Predictiva suele iniciarse en unas condiciones muy similares en cada caso, en la Programación Reactiva las condiciones de partida pueden ser mucho más amplias. El hecho de depender de la aparición de un evento que puede ser de diferente clase e incluso de diferente intensidad o impacto ha impulsado la incorporación de las bases de datos de conocimiento y motores de inferencia por un lado, y la ejecución de múltiples algoritmos de forma simultánea por otro lado.

Por un lado, la utilización de bases de datos de conocimiento y motores de inferencia es útil para identificar la acción que se debería propiciar en función del Estado del Programa y del evento detectado. La experiencia adquirida a lo largo del tiempo puede ayudar a seleccionar las acciones que han sido más exitosas en situaciones similares anteriores.

Por otro lado, si se considera el uso de técnicas heurísticas como base del Cálculo de Programa Reactivo, se puede plantear la ejecución de varias técnicas de forma simultánea. Las técnicas pueden estar apoyadas en resultados obtenidos anteriormente y seleccionada en base a la base de datos de conocimiento. La ejecución simultánea de varios algoritmos se puede realizar utilizando nuevas tecnologías como los Sistemas Multiagente, y pueden favorecer la obtención de mejores resultados y la realimentación de la base de datos de conocimiento.

El Proceso de Negocio de Programación Reactiva se ha dividido en seis Actividades Empresariales. Debido a las características del propio proceso algunas Operaciones Funcionales que en el caso anterior se encontraban en una misma Actividad Empresarial ahora se han separado.

La primera Actividad que se ha identificado está orientada a la de Identificación y Valoración de las Perturbaciones recibidas. En ella se identifican los tres tipos de eventos que se desean tratar que son: desplazamiento poco significativo de derechas, desplazamiento significativo a derechas, y desplazamiento a izquierdas. Así mismo se identifican directrices, en base a la experiencia almacenada en la base de datos de conocimiento, para los pasos sucesivos como valores del margen operativo, valor de k , o reglas de heurísticas para la Reprogramación.

La segunda actividad establece la Estrategia para la Reprogramación en base a las directivas obtenidas, y a la experiencia del programador. Esta estrategia está directamente vinculada a la *Gestión del Proceso de Programación de la Producción* en su globalidad, y por lo tanto la experiencia del programador se centra en como Ajustar la Calidad en función de lo plantado en la ecuación [5.13] del capítulo 5. Al finalizar esta tarea se debe disponer de la información relativa al número de trabajos deseable y si se debe reducir una lista priorizada de los trabajos. La lista priorizada la proporciona el planificador, pero no es nada más que una orientación.

La tercera Actividad se responsabiliza de Ajustar el conjunto de trabajos (CT) y de analizar la calidad del Programa una vez se han realizado los correspondientes cálculos. Existen dos pasos básicos:

- El primer paso consiste en valorar que trabajos hay que añadir, eliminar o si no hay que modificar el conjunto de trabajos. En caso de tener que eliminar trabajos el programador deberá seguir las recomendaciones realizadas por el planificador en la lista de prioridades, pero en caso de que lo considere oportuno puede relajar dicha recomendación. En todo caso, la propuesta final debe ser aprobada por el planificador si no se sigue la recomendación. En la Figura 7-13 se muestra un detalle de los pasos que debe seguirse en esta Actividad.
- El segundo paso se realiza cuando se termina la Actividad de Cálculo del Programa Reactivo. Si el Programa no se ha podido calcular por una inconsistencia de datos se genera un evento y se finaliza estableciendo un Programa trivial en base a los datos existen en el Programa anterior (por ejemplo, eliminar los trabajos asignados a una máquina averiada y dejar el resto). Si el Programa se rechaza, entonces se intenta volver a establecer una nueva estrategia. Si el Programa se acepta, se valora, y si se considera adecuado se guarda para posteriormente utilizarlos, en caso de no considerarlo adecuado se solicita una nueva estrategia. Cualquier resultado de no factibilidad implica la búsqueda de una nueva estrategia.

Una vez realizado el Ajuste del conjunto de trabajos se ejecuta la tarea de Carga de Datos, cuyo objetivo y actividades son las mismas que en la Programación Predictiva. Y una vez cargados los datos se realiza la tarea de Cálculo del Programa Reactivo. En esta tarea se realizan operaciones de cálculo de múltiples soluciones con diferentes algoritmos heurísticos y ajuste de los parámetros de dichos algoritmos. Así mismo se produce la actualización de la base de datos de conocimiento en base a los resultados obtenidos relacionándolos con la estrategia y el estado del Programa anterior.

Finalmente, se realiza la Actividad de Validación del Programa que ha alcanzado los mejores resultados entre todos los ejecutados. Este Programa aun debe ser valorado en la Actividad de Ajuste y Control de la Estrategia de Reprogramación.

Es conveniente aclarar que CIMOSA no identifica a las Actividades como pertenecientes a un Proceso de Negocio, y que éstas pueden ser compartidas por varios de ellos. No obstante, en este caso no se ha llegado a identificar ninguna Actividad que pueda ser compartida. Aunque las Actividades de Validación o Rechazo y Revisión y Carga de datos son muy parecidas en los Procesos de Negocio de Programación Predictiva y Reactiva, no llegan a ser iguales.

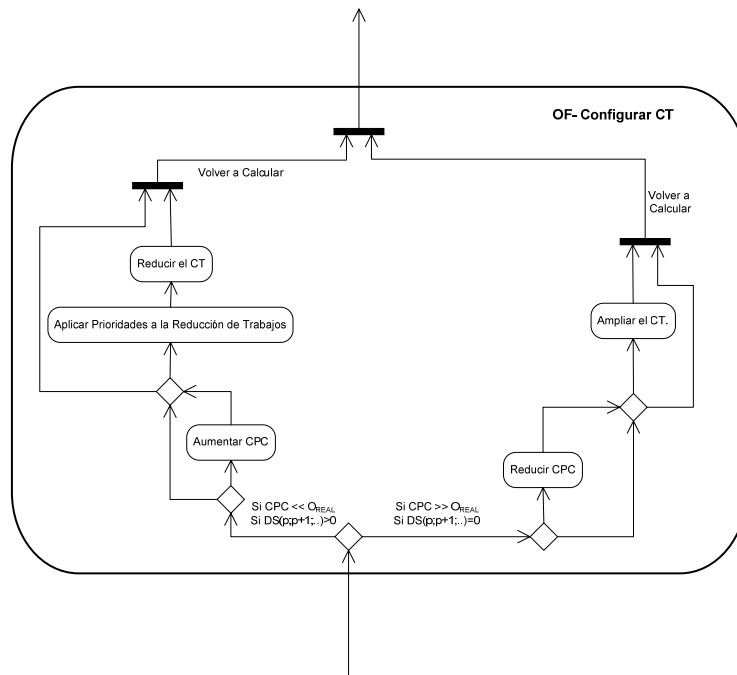


Figura 7-13. Detalle de la función operativa “Configuración del CT” (elaboración propia).

Aunque en este paso no es necesario establecer jerarquías entre los diferentes elementos de la Vista Funcional (Procesos de Negocio, Actividades, etc.) en este punto ya se dispone del detalle suficiente como para establecerlas. En la Figura 7-14 se presenta la Vista Funcional en la cual se incluyen y relacionan los elementos funcionales de mayor nivel de abstracción. En la Figura 7-11 y la Figura 7-12 se detallan las relaciones funcionales de cada uno de los Procesos de Negocio identificados.

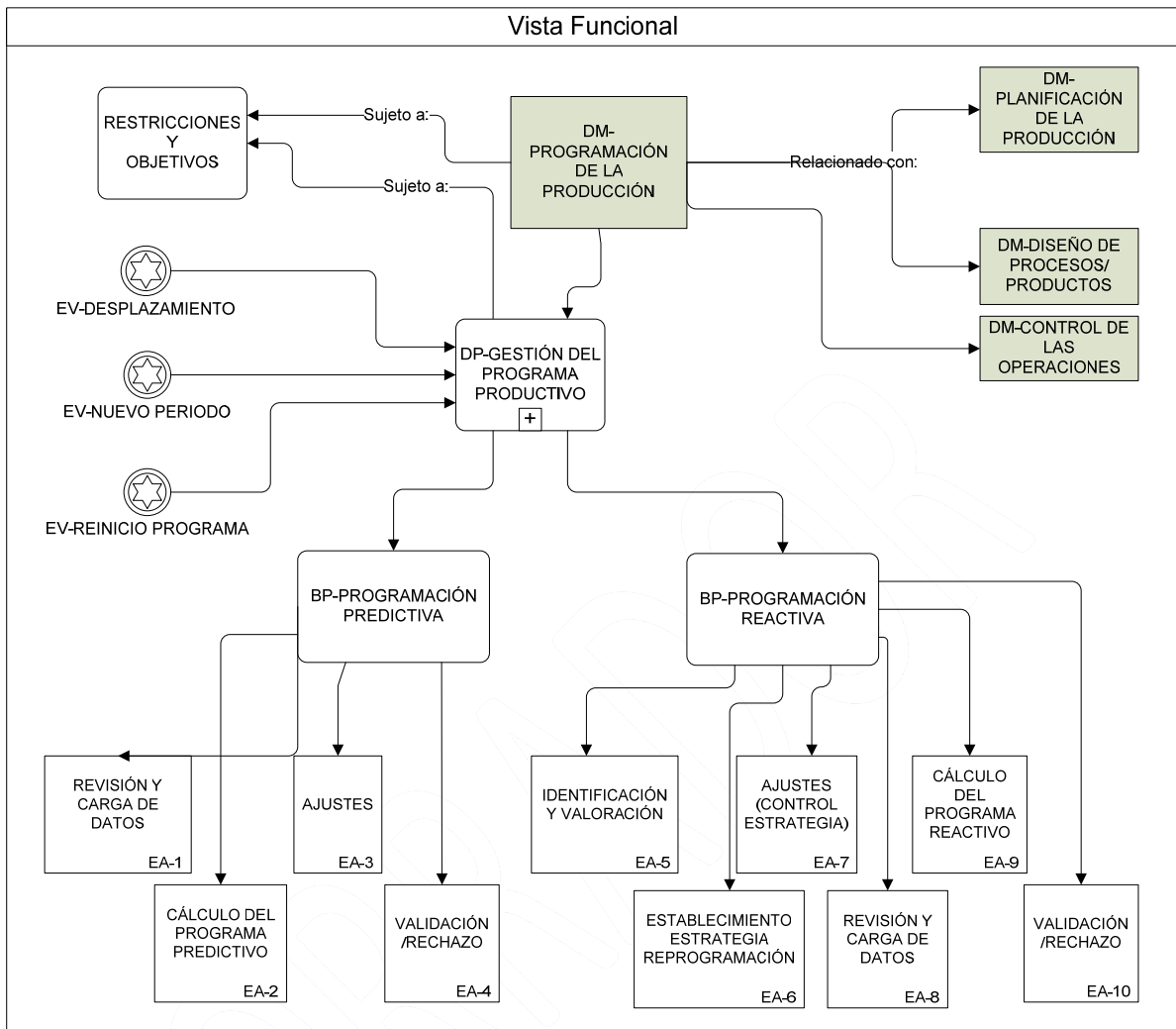


Figura 7-14. Relación entre los elementos de la Vista Funcional del Modelo de Requerimientos de un sistema de Programación de la Producción (elaboración propia).

7.3.2.3 Análisis Operacional

En este punto hay que establecer los elementos de control de las Actividades así como las Capacidades necesarias para su ejecución.

Por lo que tiene que ver con el control de las Actividades, en (RGCP 1999) se establece que el control de entrada está vinculado al conjunto de Vistas de Objetos que restringen la Actividad., y el control de salida por el Estado Final de la Actividad y/o Eventos Generados. Dado que aún

no se han identificado las Vistas de Objetos sólo se utilizarán las restricciones establecidas como elementos de control de entrada en las Actividades. Posteriormente se realizará una vinculación entre Vistas de Objetos y Actividades donde se hará referencia expresa a los elementos de control.

Las Restricciones que actúan como elementos de control en la entrada de la actividad son:

- EA-1 Revisión y carga de datos: Conjunto de Trabajos Seleccionados.
- EA-2 Cálculo del Programa Predictivo: Ausencia de inconsistencias de datos.
- EA-3 Ajuste: Existencia de un Programa de Producción sin Validar o Existencia de una no Factibilidad Continuada.
- EA-4 Validación/Rechazo: Existencia de un Programa de Producción sin Validar.
- EA-5 Identificación y valoración: Existencia de un evento conocido.
- EA-6 Establecimiento de la Estrategia de Reprogramación: Recomendación sobre la estrategia en forma de parámetros del Programa.
- EA-7 Ajuste (Control de la Estrategia de Reprogramación): Definición de la estrategia de reprogramación en forma de parámetros del Programa.
- EA-8 Revisión y carga de datos: Conjunto de Trabajos Seleccionados.
- EA-9 Cálculo del Programa Reactivo: Ausencia de inconsistencias de datos.
- EA-10 Validación/Rechazo: Existencia de un Programa de Producción sin Validar.

El estado y/o eventos que actúan como elementos de control en generados en la salida de la actividad son:

- EA-1 Revisión y carga de datos: “Datos consistentes” o “Datos no consistentes”. En este último caso, genera un evento de Inconsistencia.
- EA-2 Cálculo del Programa Predictivo: Produce un estado de “Programa Calculado” o un evento de no factibilidad.
- EA-3 Ajuste: Produce un estado de “Programa Aceptado” o “Programa no Aceptado”, en el segundo caso implica un re-cálculo.

- EA-4 Validación/Rechazo: Termina con un estado de “Programa Validado” o “Programa Rechazado”.
- EA-5 Identificación y valoración: “Programa Valorado”.
- EA-6 Establecimiento de la Estrategia de Reprogramación: “Estrategia Definida”
- EA-7 Ajuste (Control de la Estrategia de Reprogramación): Produce un estado de “Programa Aceptado”, “Programa no Aceptado”, “Programa Trivial por Inconsistencia de Datos”, o “Trabajos Seleccionados”.
- EA-8 Revisión y carga de datos: “Datos consistentes” o “Datos no consistentes”. En este último caso, genera un evento de Inconsistencia.
- EA-9 Cálculo del Programa Reactivo: Produce un estado de “Programa Calculado” o un evento de no factibilidad.
- EA-10 Validación/Rechazo: Termina con un estado de “Programa Validado” o “Programa Rechazado”.

Por lo que se refiere a las Capacidades, CIMOSA recomienda que en la fase de Modelado de Requerimientos sólo identifiquen los Conjuntos de Capacidades, y en concreto en este paso se trata de identificar habilidades que posteriormente serán englobadas en conjuntos. Algunas de las habilidades necesarias son:

- La habilidad de realizar automáticamente la conexión física con las fuentes de datos y extraer la información que sea necesaria en cada ocasión en función del algoritmo de cálculo que se utilice.
- Ser capaz de realizar la interpretación y transformación de los formatos entre origen y destino. En caso de no poder realizar su cometido debe ser capaz de identificar el problema y generar una alarma.
- La habilidad de crear Programas de Producción eficiente según criterios de uso de los recursos. Se necesita la aplicación de un procedimiento/algoritmo para obtenerlo. Es necesaria en a la Actividad de Cálculo del Programa, ya sea Predictivo o Reactivo.
- La habilidad de identificar posibles mejoras en un Programa de Producción en base a conocimiento y/o experiencias previas. Se trata de poder incorporar modificaciones sobre las restricciones difíciles de indicar de forma cuantitativa e identificar posibles mejoras del Programa propuesto.

- La capacidad de poder identificar a reglas de posibles acciones a tomar en relación con el proceso de Gestión del Programa.

La mayor parte de las Capacidades identificados necesitaran del soporte de un Recurso para su puesta en marcha. En este caso los Recursos serán normalmente software. En todo caso, antes necesitan ser refinadas mediante el establecimiento de Conjuntos de Capacidades.

7.3.2.4 Análisis de la Información

En este paso se trata de identificar los Objetos de Empresa y las Vistas de Objetos. Para realizar este paso se han identificado primero las Vistas de Objetos, que son los conceptos más relacionados con las entradas a las Actividades. A partir de estas se identificaron los elementos de rango superior (objetos de empresas) necesarios para definir los primeros. Las Vistas de Objetos son:

- OV-No disponibilidad por mantenimiento máquina.
- OV-Programa Producción en curso.
- OV-Programa Producción sin validar.
- OV-Programa Producción validado.
- OV-Ocupación programada máquina.
- OV-Capacidad disponible almacén de productos.
- OV-Tiempos de cambio de partida.
- OV-Tiempos de proceso.
- OV-Trabajos pendientes.
- OV-Trabajos Seleccionados.
- OV-Operaciones trabajos.
- OV-Máquinas factibles por trabajo/operación.
- OV-Parámetros del Programa.

- OV-Inconsistencias.
- OV-Reglas de Inferencia.
- OV- Eventos recibidos.

Las Vistas de Objetos contiene toda la información que el sistema necesita para poder realizar su trabajo. La descripción de la Vista, el Objeto de Empresa del que depende, y definición de los campos se detallan en el anexo I. De la misma forma los objetos de empresa que sustentan a las Vistas se detallan en dicho anexo I. En la Figura 7-15 se muestra una representación de la Vista de Objeto y Objetos de Empresa que conforman la Vista Informativa.

Una vez identificadas las Vistas de Objetos se puede mejorar la definición de los controles de entrada de cada Actividad. A continuación se relacionan las Vistas de Objeto que proporciona la información relativa a la restricción de entrada. Estas son:

- EA-1 Revisión y carga de datos: OV-Trabajos Seleccionados.
- EA-2 Cálculo del Programa Predictivo: OV-Inconsistencias.
- EA-3 Ajuste: OV-Programa Producción sin validar.
- EA-4 Validación/Rechazo: OV-Programa Producción sin validar.
- EA-5 Identificación y valoración: OV- Eventos recibidos.
- EA.6 Establecimiento de la Estrategia de Reprogramación: OV-Parámetros del Programa.
- EA-7 Ajuste (Control de la Estrategia de Reprogramación): OV-Parámetros del Programa.
- EA-8 Revisión y carga de datos: OV-Trabajos Seleccionados.
- EA-9 Cálculo del Programa Reactivo: OV-Inconsistencias.
- EA-10 Validación/Rechazo: OV-Programa Producción sin validar.

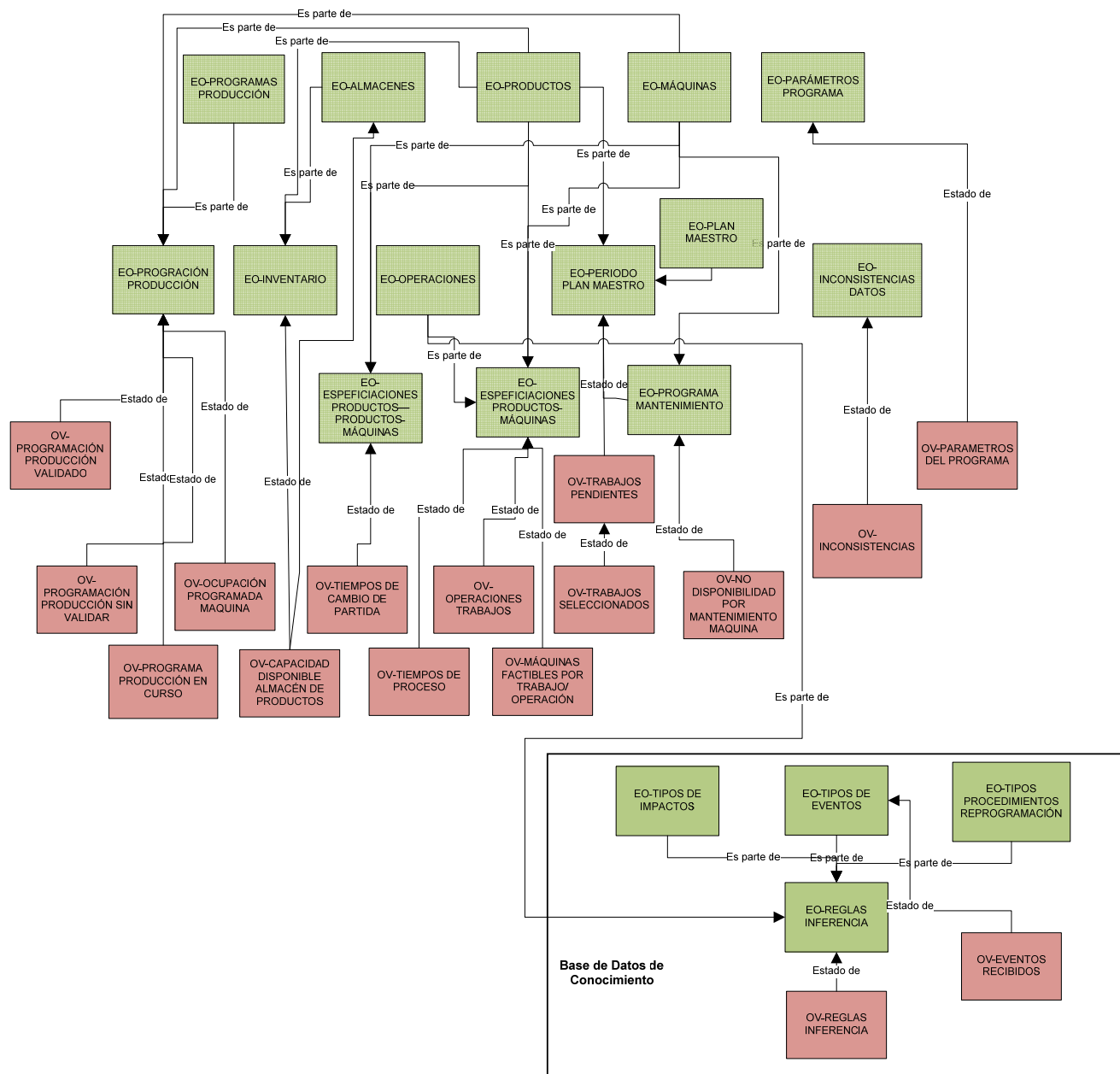


Figura 7-15. Vista Informacional del Modelo de Requerimientos (elaboración propia).

El Modelo propuesto cubre las necesidades del problema que se ha definido previamente pero se le ha dotado de alguna cobertura adicional que permita en un futuro la inclusión de nuevas funcionalidades. Aunque ahora ciertas funciones quedan fuera de los Requerimientos se prevee que para un futuro puedan ser un requerimiento razonable.

7.3.2.5 Análisis de los Recursos

En el caso que se está modelando los recursos necesarios son todos o personales o software. Las Capacidades que se han identificado no están relacionadas con el uso de Recursos Físicos. Esta situación tiene lógica ya que se está modelando una Actividad básicamente intelectual.

Se ha considerado que es suficiente nivel de detalle el establecimiento de los Conjuntos de Capacidades y que no aporta valor añadido la identificación de Recursos concretos. Las Capacidades identificadas son:

- CS-Adquisición, transformación y validación de datos: Se trata de la habilidad de realizar automáticamente la conexión física con las fuentes de datos y extraer la información que sea necesaria en cada ocasión en función del algoritmo de cálculo que se utilice. Al mismo tiempo que obtiene la información debe ser capaz de realizar la interpretación y transformación de los formatos entre origen y destino. En caso de no poder realizar su cometido debe ser capaz de identificar el problema y generar una alarma.
- CS-Procedimientos de Programación de la Producción: Es la habilidad de crear un nuevo Programa de Producción eficiente según criterios de uso de los recursos. Se necesita la aplicación de un procedimiento/algoritmo para obtenerlo.
- CS-Procedimientos de Reprogramación de la Producción: Es la habilidad de modificar el Programa de Producción en curso de forma eficiente balanceando los criterios de uso eficiente de los recursos y minimización de los cambios en el Programa propuesto frente al Programa en curso. Se necesita la aplicación de un procedimiento/algoritmo para obtenerlo.
- CS-Ajuste Fino de Programas: Es la habilidad de identificar posibles mejoras en un Programa de Producción en base a conocimiento y/o experiencias previas. En definitiva se trata de poder incorporar modificaciones sobre las restricciones difíciles de indicar de forma cuantitativa e identificar posibles mejoras del Programa propuesto.
- CS-Capacidad de Inferir Acciones: Consiste en la capacidad de poder identificar a reglas de posibles acciones a tomar en relación con el proceso de gestión del Programa.

Una vez analizados los elementos relacionados con la vista de recursos finaliza el proceso de construcción del modelo de requerimientos. El siguiente paso es verificar la validez el mismo.

7.3.2.6 Verificación de la Consistencia del Modelo

Durante el proceso de construcción del Modelo de Requerimientos se van ido incorporando diferentes elementos arquitectónicos que se ubican en tres vistas diferentes (Funcional, Informacional y de Recursos). En este paso se trata de verificar que cada una de las tres Vistas forman un único Modelo, y que las relaciones entre ellas están perfectamente definidas desde el punto de vista de la arquitectura CIMOSA y tienen la lógica deseada.

Como paso previo se han revisado uno a uno de todas las relaciones existentes en cada Vista con el objeto de alcanzar unos Modelos parciales robustos, y con la ventaja de contar con el Modelo completo ya finalizado. Este paso previo ha tenido como hilo conductor la elaboración de las plantillas descriptivas de cada elemento arquitectónico propuestas por (Berio, G. & Vernadat, F. B. 1999a). En las plantillas no sólo se define con precisión la naturaleza de cada elemento, sino la relación con su entorno. Durante este proceso se ha corregido la configuración de algún elemento, especialmente por lo que se refiere a la depuración de estados iniciales y finales.

Una vez superado el paso previo, y sobre la base de unos Modelos parciales sólidos se ha procedido a la verificación de la relación entre ellos. Para realizar la verificación se ha tenido en cuenta especialmente el análisis e identificación de las Vistas de Objetos que son utilizadas por las Actividades Empresariales como entrada o salida, ya que es una de las relaciones Inter-Vistas más relevante. Durante este proceso se han definido las unidades de Información básica (campos o registros) que cada vista de objeto debería proporcionar o almacenar. Esta definición ha ayudado a transformar conceptos abstractos a concretos. Con la información concreta se ha podido certificar si era posible realizar las funciones asociadas a cada Actividad Empresarial.

La identificación y verificación de la relación entre la Vista Funcional y la Vista de Recursos se basa en el análisis de la relación entre Actividades Empresariales y los Conjuntos de Capacidades. Como se ha resaltado anteriormente, la Programación de la Producción, ya sea Predictiva o Reactiva, es un proceso básicamente carente de elementos o Actividades Físicas, y por este motivo las Capacidades son de tipo aptitudinal. Este chequeo se ha aprovechado para identificar con claridad la aportación esperada en cada uno de los Conjuntos de Actividades.

Como resultado de este proceso de verificación se dispone del anexo I "Definición de los Elementos Arquitectónicos de CIMOSA Implementados para Modelado de Requerimientos del Problema Propuesto", y dentro del cual se encuentra la Figura I-2 donde se puede observar una relación entre los elementos arquitectónicos principales de las vistas Funcional, Informacional y de Recursos. En dicha figura, las actividades EA-1 y EA-8 relacionadas con la

Revisión y Carga de Datos, y las Actividades EA-4 y EA-10 relacionadas con la Validación o Rechazo de Programas se han representado un solo elemento cada una de ellas, ya que a efectos de relación con el resto de elementos son idénticas. Este “ahorro” en la representación permite simplificar relativamente el gráfico.

A partir del Análisis de Requerimientos se realiza la propuesta en la que se establece el conjunto de Actividades que serán asumidas por el sistema para programador de la Producción que se obtendrá como uno de los resultados de esta tesis.

7.3.3 Resumen del Proceso de Modelado de Requerimientos

El modelado de requerimientos ha impulsado el proceso de análisis del problema a partir del cual se empieza a construir mediante un proceso de diseño la solución de la misma. El modelado de requerimientos ha permitido examinar de forma detallada todas aquellas condiciones que deben determinar la Programación de la Producción en el caso presentado en el capítulo 5. Se ha realizado un análisis con múltiples perspectivas o vistas tal y como sugiere el enfoque CIMOSA.

Del trabajo realizado se puede destacar que lo más interesante quizás haya sido la interconexión que se ha realizado entre los elementos que forman parte de cada una de las vistas. Alrededor de los elementos funcionales (Enterprise Activities) se han interconectado los elementos de información (OV) y de recursos (CS). De esta forma se han identificado las necesidades funcionales, que suelen ser las que habitualmente se definen, pero además las necesidades de información y de recursos que son necesarias.

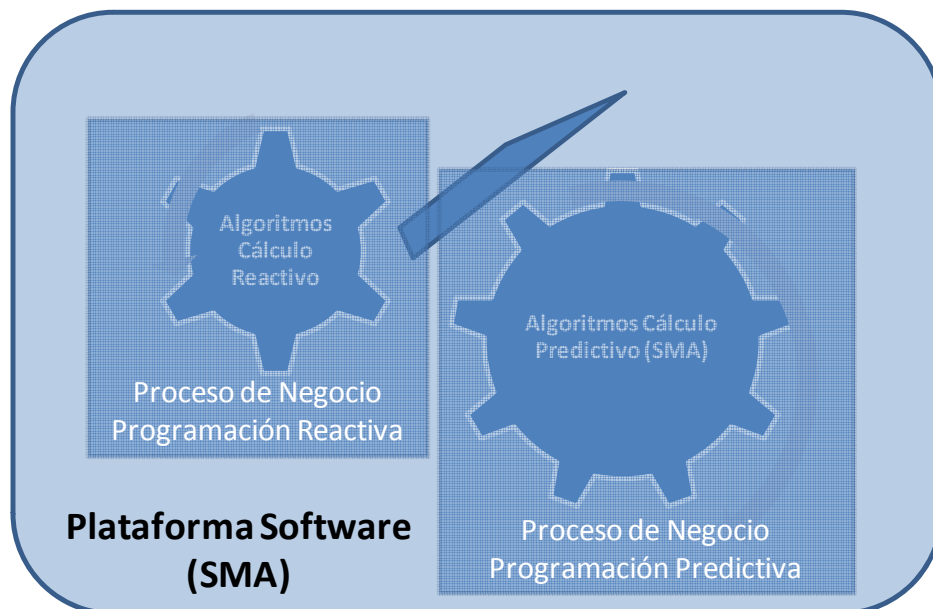
A partir de la los objetivos del Dominio, los eventos implicados, las restricciones y sobre todo los contenidos elaborados en los Procesos de Dominio se procederá a diseñar la solución.

7.4 Diseño de la Plataforma IPSU-MAS

El diseño de la Plataforma software se corresponde con el tercer paso de la metodología aunque no llega a cubrirlo de forma completa. Este tercer paso, es decir el Diseño de la Solución se complementa con el trabajo realizado en los capítulos 8 y 9. En el capítulo 8 se aborda de forma detallada el desarrollo del cálculo del Programa Predictivo mediante el análisis e implementación de diversos algoritmos. De la misma forma, pero focalizados en el cálculo de los Programas Reactivos, se trabajará en el capítulo 9. En ambos capítulos se

ofrecerán datos sobre las prestaciones esperadas mediante un análisis comparativo de los resultados.

Un aspecto característico de la Plataforma software que se propone diseñar es que su desarrollo se basará en el paradigma de los Sistema Multiagente. Como se muestra en la Figura 7-16 **la Plataforma tiene como fin servir de soporte tangible al conjunto de elementos decisionales**, que basados en diversas teorías y propuestas, dan cobertura al Dominio de Proceso para la Gestión del Programa Productivo. Y como se ve en la misma figura se mantiene la estructura definida en el análisis de requerimientos estableciendo dos procesos de negocio cada uno de los cuales está estrechamente relacionado con un problema de cálculo.



Estructura de la Propuesta

Figura 7-16. Relación entre las Vistas del Modelo de Requerimientos (elaboración propia).

Se destaca que ya en la Figura 7-16 se denomina a la Plataforma **IPSU-MAS**, estableciendo de forma inequívoca la adopción del paradigma de Sistema Multiagente en su diseño e implementación.

Hay que subrayar que la Plataforma software que se va a diseñar tiene como fin convertirse en un prototipo y no en un producto final. Por este motivo, sólo se abordan los escenarios de interés para el caso planteado, y además no se afronta ningún aspecto relacionado con el interface de usuario. Esta fase de diseño tiene como objetivo elaborar una proyección

conceptual que será complementada en el capítulo 10 mediante la elaboración de los algoritmos o procedimientos que se necesiten para abordar puntualmente un caso concreto.

Este apartado se organiza de la siguiente manera: En primer lugar se realiza un breve resumen de las metodologías disponibles para el diseño de soluciones basadas en Sistema Multiagente. En segundo lugar se analiza brevemente los elementos más relevantes para el desarrollo de la metodología INGENIAS. En tercer lugar se desarrolla la metodología INGENIAS que es la elegida para el diseño de la Plataforma. En cuarto lugar se realiza una reflexión a modo de resumen de esta fase de diseño.

7.4.1 Metodología para el Diseño de la Plataforma IPSU-MAS

Las metodologías proporcionan medios para construir Sistemas Multiagente de forma disciplinada y repetible. En el capítulo 4 se presentó un resumen de lo que proponen las metodologías actuales. Entre las aproximaciones existentes fueron consideradas interesantes: La metodología para Agentes BDI, esta aproximación, (Kinny, D. & Georgeff, M. 1996), DESIRE, (Brazier, F., Dunin Keplicz, B., Jennings, N., & Treur, J. 1997), GAIA, (Wooldridge, M., Jennings, N. R., & Kinny, D. 1998), MAS-CommonKADS, (Iglesias-Fernández, C. A. 1998), MASSIVE, (Lind, J. 1999), MaSE (Multiagent System Engineering), (Wood, M. F. 2000), AUML, (Odell, J., Parunak, H., & Bauer, B. 2000), TROPOS, (Castro, J., Kolp, M., & Mylopoulos, J. 2002) e MESSAGE, INGENIAS, (Gomez, J. 2002). Entre ellas se ha seleccionado la metodología INGENIAS que está orientada a Agentes y trata de proveer un lenguaje, un método y unas guías de aplicación, centrándose fundamentalmente en las fases de análisis y diseño. Además presenta fundamentalmente como aspecto más interesante la existencia de herramientas que permiten un desarrollo a alto nivel de entidades transformables automáticamente en componentes software.

Después del análisis de las diferentes alternativas consideradas, se valoró de forma muy positiva la posibilidad de emplear INGENIAS, ya que fundamentalmente cubre de forma excelente la fase de diseño, y adicionalmente facilita enormemente la fase de implementación al generar componentes software de forma automática. Así mismo la metodología cuenta con una herramienta que facilita su ejecución. Además, los aspectos formativos relativos al aprendizaje en el manejo de la metodología y la herramienta quedaron ampliamente cubiertos al disponer del material que los propios autores hacen públicos en su página web, así como el apoyo del grupo de investigación GTI (Grupo de Tecnología Informática) de la Universidad Politécnica expertos en el diseño e implementación de este tipo de sistemas con dicha metodología.

7.4.2 Metodología INGENIAS

En este apartado se esboza brevemente la metodología INGENIAS de forma que el lector pueda seguir adecuadamente el proceso de diseño de la Plataforma **IPSU-MAS** basado en dicha metodología.

7.4.2.1 Metodología

El método de desarrollo de Sistemas Multiagentes propuesto en INGENIAS concibe a los mismos como la representación computacional de un conjunto de modelos. Cada uno de estos modelos muestra una visión parcial del sistema: los Agentes que lo componen, las interacciones que existen entre ellos, cómo se organizan para proporcionar la funcionalidad del sistema, qué información es relevante en el dominio y cómo es el entorno en el que se ubica el sistema a desarrollar.

Para especificar cómo tienen que ser estos modelos se definen meta-modelos. Un meta-modelo es una representación de los tipos de entidades que pueden existir en un modelo, sus relaciones y restricciones de aplicación. Los meta-modelos que se describen en INGENIAS son una evolución del trabajo realizado en MESSAGE. En MESSAGE se propusieron meta-modelos para representar Agentes, organizaciones, el dominio, interacciones, tareas y objetivos. INGENIAS introduce ciertas mejoras sobre MESSAGE. El resultado del proceso de mejora es la propuesta basada en **cinco meta-modelos que giran alrededor de dos entidades; la Organización y el Agente.**

		Fases		
		Inicio	Elaboración	Construcción
FLUJOS DE TRABAJO FUNDAMENTALES	Análisis	<p>Generar casos de uso e identificar realizaciones de los casos de uso con modelos de interacciones.</p> <p>Esbozar la arquitectura con un modelo de organización.</p> <p>Generar modelos del entorno para trasladar la captura de los requisitos a los modelos.</p>	<p>Refinar los casos de uso.</p> <p>Generar modelos de Agente para detallar los elementos de la arquitectura.</p> <p>Continuar con los modelos de organización identificando los flujos de trabajo y las tareas.</p> <p>Modelos de tareas y objetivos para generar restricciones de control (objetivos principales, descomposición de objetivos).</p> <p>Refinar modelo de entorno para incluir nuevos elementos.</p>	<p>Estudiar resto de casos de uso.</p>

	Diseño	<p>Generar un prototipo con herramientas de prototipado rápido como ZEUS Agent Tool</p>	<p>Centrar el modelo de organización en el desarrollo de flujos de trabajo.</p> <p>Llevar las restricciones identificadas a modelos de tareas y objetivos para dar detalles acerca de las necesidades y resultados de la tarea en su relación con los objetivos del sistema.</p> <p>Expresar la ejecución de tareas dentro de modelos de interacción.</p> <p>Generar modelos de Agente para detallar patrones de estado mental.</p>	<p>Generar modelos de Agentes, refinar los existentes.</p> <p>Depurar la organización centrando el desarrollo en las relaciones sociales.</p>
--	---------------	---	---	---

Tabla 7-2. Flujos de trabajo en la metodología INGENIAS (Pavon, J. et al. 2003).

El ciclo de vida del desarrollo software de INGENIAS se basa en la propuesta RUP (Rational Unified Process). En el RUP, el esfuerzo del análisis y diseño se encuentra localizado en tres fases: inicio, elaboración y construcción. Dentro de cada fase se desarrollan las iteraciones (ciclos completos de desarrollo incluyendo análisis, diseño, implementación y pruebas) que construyen gradualmente el sistema. Estas asociaciones marcan prioridades a la hora de profundizar en la generación de los distintos modelos (agente, organización, interacción, entorno, objetivo y tareas). En este caso la Tabla 7-2 no hace referencia a las fases de implementación y pruebas por no ser el objetivo del capítulo.

Basándose en el ciclo de vida del desarrollo de software adoptado por INGENIAS en este capítulo de desarrollo la fase de diseño de la Plataforma **IPSU-MAS**.

7.4.2.2 Meta Modelos

Los meta-modelos son una gran ayuda en el proceso de desarrollo de los Sistemas Multiagente, ya que determinan qué entidades tienen que existir y cómo deben conectarse. Desde el punto de vista de ingeniería, lo que interesa de los meta-modelos, además de servir de guía, es cómo pueden ayudar a estructurar el desarrollo del sistema. Para estudiar la aplicación de los meta-modelos a procesos de ingeniería, se han tomado los meta-modelos como lenguaje de especificación del Sistema Multiagente, de forma similar a los diagramas UML (Unified Modeling Language) como especificación de un desarrollo orientado a objetos. Partiendo de este lenguaje se estudian qué actividades y productos son necesarios para generar los modelos que constituyen la especificación del sistema. Estas actividades y productos son relevantes porque se pueden integrarse dentro de un proceso de ingeniería del software. Las actividades se pueden distribuir entre los diferentes componentes de un equipo de trabajo, estableciendo qué productos se esperan de cada actividad y cómo se combinan. Los meta-modelos se presentan de forma separada, son:

- **Meta-modelo de Agente.** El meta-modelo de agente describe Agentes particulares y los estados mentales en que se encontrarán a lo largo de su vida.
- **Meta-modelo de Tareas y Objetivos.** Se usa para asociar el estado mental del agente con las tareas que ejecuta.
- **Meta-modelo de Organización.** Define cómo se agrupan los Agentes, la funcionalidad del sistema y qué restricciones hay que imponer sobre el comportamiento de los Agentes.
- **Meta-modelo de Interacción.** Detallar cómo se coordinan y comunican los Agentes.
- **Meta-modelo de Entorno.** Define qué existe alrededor del nuevo sistema y cómo lo percibe cada agente.

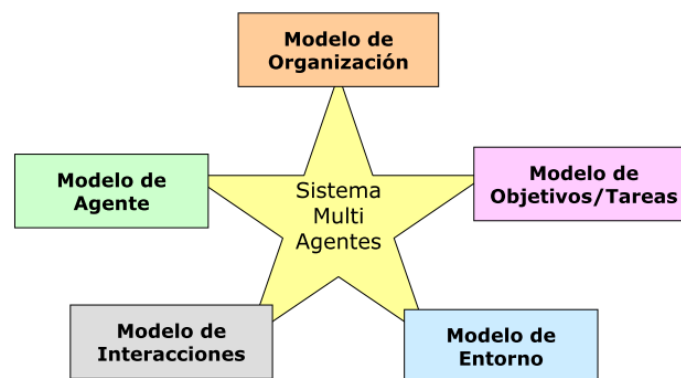


Figura 7-17. Meta-modelos para el Diseño del Sistema Multiagente (Pavon, J. et al. 2003).

La generación de modelos a partir de meta-modelos se guía por el conjunto de actividades que se presentarán a lo largo de este capítulo. Cada modelo estará compuesto de una serie de diagramas que representan parte de la información que se desea transmitir, y que tiene sentido dentro de la totalidad.

7.4.3 Diseño de la Plataforma IPSU-MAS aplicando INGENIAS

En este capítulo se procederá al desarrollo de los diferentes modelos que conforman el núcleo fundamental del diseño de la Plataforma **IPSU-MAS**, en base a los correspondientes meta-modelos. Aunque toda la fase de diseño se ha realizado mediante el ciclo de tres pasos (inicio,

elaboración y construcción), en los que se basa INGENIAS, lo que se muestra en este capítulo es el resultado final.

La Plataforma que se desea diseñar tiene como objetivo dar cobertura a los requisitos establecidos en la fase de análisis de requerimientos. Los dos objetivos básicos establecidos son:

- Ser capaz de realizar el proceso de negocio de Programación Predictiva.
- Ser capaz de realizar el proceso de negocio de Programación Reactiva.

Pero es en este punto donde hay que destacar que ambos procesos se van a desarrollar dentro de un marco de coordinación con otros Procesos de Negocio, en especial con el Proceso de Negocio de Planificación de la Producción. Por este motivo durante el diseño de la Plataforma se ha tenido en cuenta la interacción con ciertos Agentes o roles, según el modelo, que se pueden ubicar en procesos diferentes a los de Programación Predictiva o Reactiva de la Producción.

Aunque en la fase de diseño no se puede llegar a apreciar en su correcta dimensión, es fundamental destacar el peso que adquieren los sistemas y protocolos de comunicación en un Sistema Multiagente. En este sentido se destaca que gracias a la existencia de la Plataforma JADE^{vii} (Java Agent DEvelopment framework) para el desarrollo de Sistemas Multiagente no ha sido necesario diseñar ni programar protocolos ni sistemas de comunicación. JADE es una Plataforma ampliamente utilizada que cumple los estándares de FIPA (Foundation for Intelligent Physical Agents) lo que facilita enormemente el diseño, pero aún más si cabe la implementación de este tipo de sistemas.

A continuación se desarrollo la fase de diseño empezando en primer lugar con una visión general de los diferentes diagramas que conforman cada meta-modelo, así como el papel que cumple. En los siguientes apartados se desarrollan los diagramas correspondientes los modelos de cada meta-modelo, que serán analizados en el nivel de detalle que le corresponde.

^{vii} Tanto la Plataforma JADE como la organización FIPA han sido introducidas y analizadas en el capítulo 4 de esta tesis.

7.4.3.1 Esquema General

Como se ha indicado anteriormente el proceso de diseño se inicia estableciendo el conjunto de casos de uso que representan las situaciones que se desea abordar, que en este caso serán 4 asociados a 4 roles diferentes: Planificador, Coordinador, Operario y Programador.

Una vez explicitados los casos de uso, que servirán de punto de referencia para todo el proceso de diseño, se elaborarán los 5 modelos basados en los meta-modelos propuestos por INGENIAS. En la Tabla 7-3 se resumen los gráficos realizados para cada modelo con el objeto de orientar mejor el proceso de comprensión del trabajo realizado.

Modelo	Gráfico	Comentarios
Modelo de Organización	Modelo de Organización	Gráfico único en el que se resume la visión organizacional que se necesita para el diseño de la Plataforma.
Modelo de Entorno	Modelo de Entorno	Gráfico único en el que se resume el entorno conocido que se necesita para el diseño de la Plataforma.
Modelo de Objetivos y Tareas	Establecer la Producción predictivamente	Este modelo requiere la diferenciación de los 2 grandes objetivos de la Plataforma, que se trabajarán por separado en este punto.
	Establecer la Producción reactivamente	
Modelo de Agentes	Agente Planificador	Para cada uno de los 3 Agentes definidos se realizada una definición de los elementos que lo componen.
	Agente Programador	
	Agente Operario	
Modelo de Interacciones	Modificación de estados mentales	En este modelo además de tratar de definir a los Agentes se hace referencia explícita a ciertos estados que deben ser considerados por los mismos.
	Obtener Planificación periódica	En este modelo se tratan las 5 interacciones fundamentales dentro de la Plataforma IPSU-MAS.
	Notificar Inconsistencias	
	Modificar Planificación	
	Reasignar Producción	
	Notificar problema	
Condiciones iniciadores de las Interacciones	Adicionalmente, se indican las condiciones que provocan el inicio de las mencionadas interacciones.	

Tabla 7-3. Resumen de gráficos correspondientes a cada modelos empleado en el diseño (elaboración propia).

Una vez revisado el esquema general de los diferentes aspectos, particularizaciones, que se tratarán en relación a los diferentes modelos, el lector está en condiciones de seguir con mayor claridad el proceso.

7.4.3.2 Modelo de Casos de Uso

Un modelo de casos de uso captura la funcionalidad deseada por los usuarios en un Sistema de Información. El modelo de casos de uso constituye una herramienta para reflejar los requisitos de una aplicación informática, y de permisos que cada usuario o grupo de usuarios va a disponer en el uso de la aplicación. En el caso de la Plataforma **IPSU-MAS**, se han identificado cuatro perfiles de usuarios (planificador, coordinador, Programador y operario), así como un conjunto de casos de uso que a continuación se detallan.

En la Tabla 7-4 se muestran los 8 casos de uso identificados comentados. Estos casos de uso se emplearán durante la fase de diseño, y serán comprendidos en profundidad a medida que se desvelen los correspondientes modelos.

Entidad	Entidad del meta-modelo	Documentación
Notificar Planificación Periódica	Caso de uso	El planificador indica la Planificación periódica a producir por el conjunto de plantas.
Estimar Planificación por planta	Caso de uso	El coordinador establece la Planificación desagregada por planta.
Notificar Planificación por planta	Caso de uso	El coordinador notifica a las plantas su Planificación para el siguiente periodo.
Obtener Planificación Periódica	Caso de uso	El programador de cada planta consulta la Planificación periódica deseada.
Calcular Programación Predictiva	Caso de uso	El programador calcula la Programación periódica. Puede cambiar los parámetros, establecer ajustes, y validar o rechazar la Programación.
Ajustar Producción	Caso de uso	El programador puede reajustar el Programa Productivo en función del nivel de carga establecido o incidencias acontecidas.
Notificar Problemas	Caso de uso	Los operarios pueden notificar problemas o retrasos a los programadores de su planta.
Consultar Programación	Caso de uso	Los operarios consultan la producción establecida o modificada que les proporciona su programador de planta.

Tabla 7-4. Resumen de los casos de uso planteados en la Plataforma IPSU-MAS (elaboración propia).

Existen roles y funciones que aún no han sido introducidas con detalle pero que están ya incluidas en los casos de uso. Para facilitar la visión general de los mismos se debe comentar que se ha planteado un sistema capaz e afrontar un sistema productivo multi-planta en el que existe la figura de un planificador de la Producción, un coordinador de la Producción entre las plantas, un programador de la Producción por planta y algunos operarios por cada planta. Desde la Figura 7-18 a la Figura 7-21 se relacionan los roles con los casos de uso En todos los casos se ha seguido la notación descrita en la Tabla 7-5.

Notación:	
	representa a un role
	representa a un caso de uso.
	representa a un elemento de la organización.
	representa a un flujo de trabajo
	representa un objetivo.
	representa a un recurso software
	representa a un recurso físico.
	representa una tarea.
	representa una interacción.
	representa una creencia.
	representa un estado mental.
<u>WFPlays</u> →	Indica que un Role (elemento sucesor) es jugado por un Agente.
<u>AHasMS</u> →	Indica que un Estado Mental (elemento sucesor) se puede ser elaborado por un Agente.
<u>AConstainsME</u> →	Indica que un Estado Mental (elemento predecesor) está contenido en un Conjunto de Estados Mentales.
<u>«ParticipatesInUseCase»</u> →	Indica que elemento del origen participa en un caso de uso.
<u>«Includes»</u> →	Indica que el caso de uso situado en el origen participa en un caso de uso del destino.
<u>GTPursues</u> →	Indica que un objetivo (elemento sucesor) es perseguido por un elemento predecesor.
<u>OHasWF</u> →	Establece una relación de pertenencia de un Flujo de Trabajo (elemento sucesor) con respecto al predecesor.
<u>OHasGroup</u> →	Establece una relación de pertenencia de un elemento organizativo (elemento sucesor) con respecto al predecesor.
<u>WFContainsTask</u> →	Establece una relación de pertenencia de un Flujo de Trabajo (elemento sucesor) con respecto a otro Flujo de Trabajo (predecesor).
<u>OHasMember</u> →	Establece una relación de pertenencia de un role (elemento sucesor) con respecto al predecesor.
<u>ApplicationBelongsTo</u> →	Indica que una aplicación software (elemento sucesor) le pertenece a un elemento predecesor.
<u>EResourceBelongsTo</u> →	Indica que un recurso físico (elemento sucesor) le pertenece a un elemento predecesor.
<u>Satisfaction</u> →	Indica que un Flujo de Trabajo (elemento sucesor) debe satisfacer un determinado objetivo.
<u>WFDescomposes</u> →	Indica que un Flujo de Trabajo (elemento sucesor) se descompone en una tarea.
<u>WFResponsible</u> →	Indica que un tarea (elemento sucesor) es responsabilidad e un role.
<u>GTCreates</u> →	Indica que un tarea (elemento sucesor) necesita iniciar una interacción.

Tabla 7-5. Notación empleada en las figuras que representan los Casos de Uso y los Modelos.

Como ejemplo se comenta el caso de uso de la Figura 7-21. La funcionalidad del sistema se divide en un conjunto de casos de uso, que a su vez pueden subdividirse en funcionalidad más específica (ver relaciones entre casos de uso etiquetadas como <<include>>). Es decir, que parte de la funcionalidad del Sistema Multiagente es *calcular la Programación Predictiva*, y parte de la funcionalidad que compone tal cálculo (<<include>>) es *establecer parámetros*. Relacionado con un caso de uso existe un perfil de usuario que hace uso de esa funcionalidad del Sistema de Información. Así en el caso de la Figura 7-21 un usuario con permisos (role) de programador puede invocar la funcionalidad de calcular Programación predictiva, pero un operador no puede hacer uso de tal funcionalidad, hecho que se indica al no relacionar el rol de operario con el caso de uso calcular Programación Predictiva.



Figura 7-18. Modelo de casos de uso del Planificador (elaboración propia).

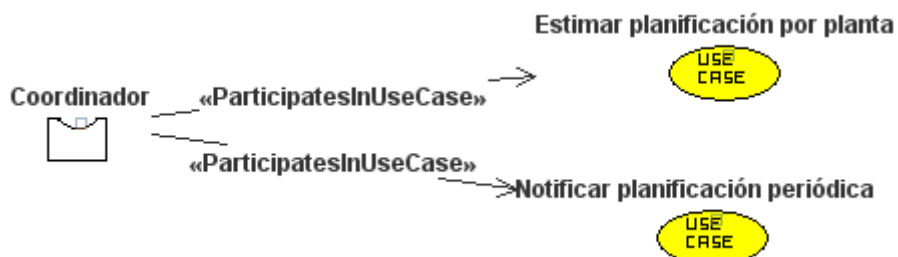


Figura 7-19. Modelo de casos de uso del Coordinador (elaboración propia).



Figura 7-20. Modelo de casos de uso del Operario (elaboración propia).

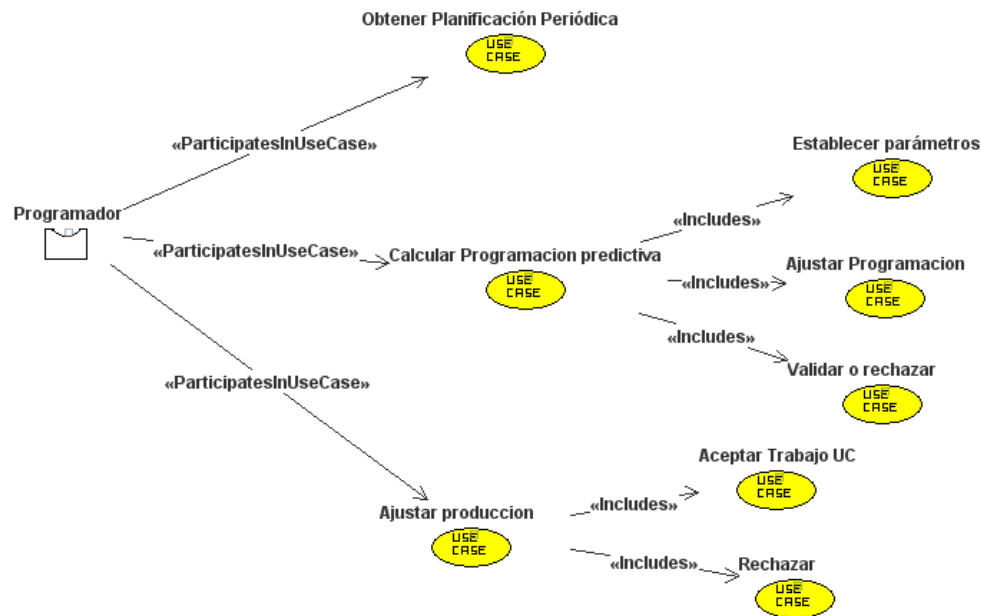


Figura 7-21. Modelo de casos de uso del Programador (elaboración propia).

7.4.3.3 Modelo de Organización

El modelo de organización muestra la estructura organizativa de la empresa y su relación con la estructura del Sistema de Información del Sistema Multiagente. Así pues, el modelo de organización aproxima la arquitectura del sistema en un Sistema Multiagente.

En el modelo se establece la división de la empresa en grupos (en nuestro caso una *sede central* y un conjunto de *plantas*, ver Figura 7-22). Asociada a cada distribución existe un conjunto de roles que serán asumidos por Agentes software, aunque podrían ser asumidos por personas. Los roles identificados, y que posteriormente se explicarán, como ya se ha avanzado, son los de *Planificador*, *Coordinador*, *Programador* y *Operario*. Los Agentes que asuman los roles realizarán tareas colaborando entre sí en lo que se denominan flujos de trabajo, en los que interactuarán e intercambiarán información. En nuestro caso, los principales flujos de trabajo son *Establecer Producción Predictiva* y *Reajustar Producción reactivamente* (ver Figura 7-22). El estudio de estos flujos de trabajo se realiza mediante el modelado de objetivos y tareas, y el análisis de las interacciones en el modelo de interacciones (ver secciones 7.4.3.5 y 7.4.3.7).

En la Tabla 7-6 se resumen las entidades que forman parte del modelo de organización que se desea representar.

Entidad	Entidad del meta-modelo	Documentación
Planificador	Rol	Representa a la persona responsable de calcular la Planificación de la Producción a partir de la previsión de la demanda, y de la información de capacidades proporcionados por el agente coordinador.
Coordinador	Rol	Se puede considerar como un planificador de segundo nivel. Se encarga de desagregar la Planificación de la Producción que le proporciona el agente planificador, y de emitir planificaciones de la Producción por planta en función del stock, y la capacidad de cada planta. A su vez, se encarga de modificar y balancear las cargas productivas en cada una de las plantas. Su trabajo se realiza en equipo con el agente planificador.
Programador	Rol	Se encarga de establecer una Programación de la Producción a partir de la Planificación de la Producción que se convierte en el trabajo diario que envía a cada agente operario. Modifica la Programación diaria de acuerdo a información recibida por los Agentes operarios, y en caso de implicar un rechazo o aceptación de Producción se comunica con el agente coordinador.
Operario	Rol	Se encargan de recibir las órdenes productivas y notificarlas al operario. Monitorizan el volumen de trabajo realizado y prevén retrasos o adelantos que notifican al agente programador. En caso de rotura de máquinas también notifican al agente programador.
Empresa	Grupo	Nivel organizativo principal.
Sede central	Grupo	La empresa se compone de una sede central, que es una división que agrupa el planificador y coordinador y que es común a todas las plantas.
Planta	Grupo	La empresa puede estar compuesta de una o más plantas. El modelo está preparado para ser escalable a varias plantas.
Establecer Producción predictiva	Flujo de trabajo	Conjunto de actividades que tienen como objetivo obtener una Programación Predictiva.
Reajustar Producción reactivamente	Flujo de trabajo	Conjunto de actividades que tienen como objetivo obtener una Programación Reactiva.

Tabla 7-6. Resumen de las entidades del modelo de organización planteados en la Plataforma IPSU-MAS (elaboración propia).

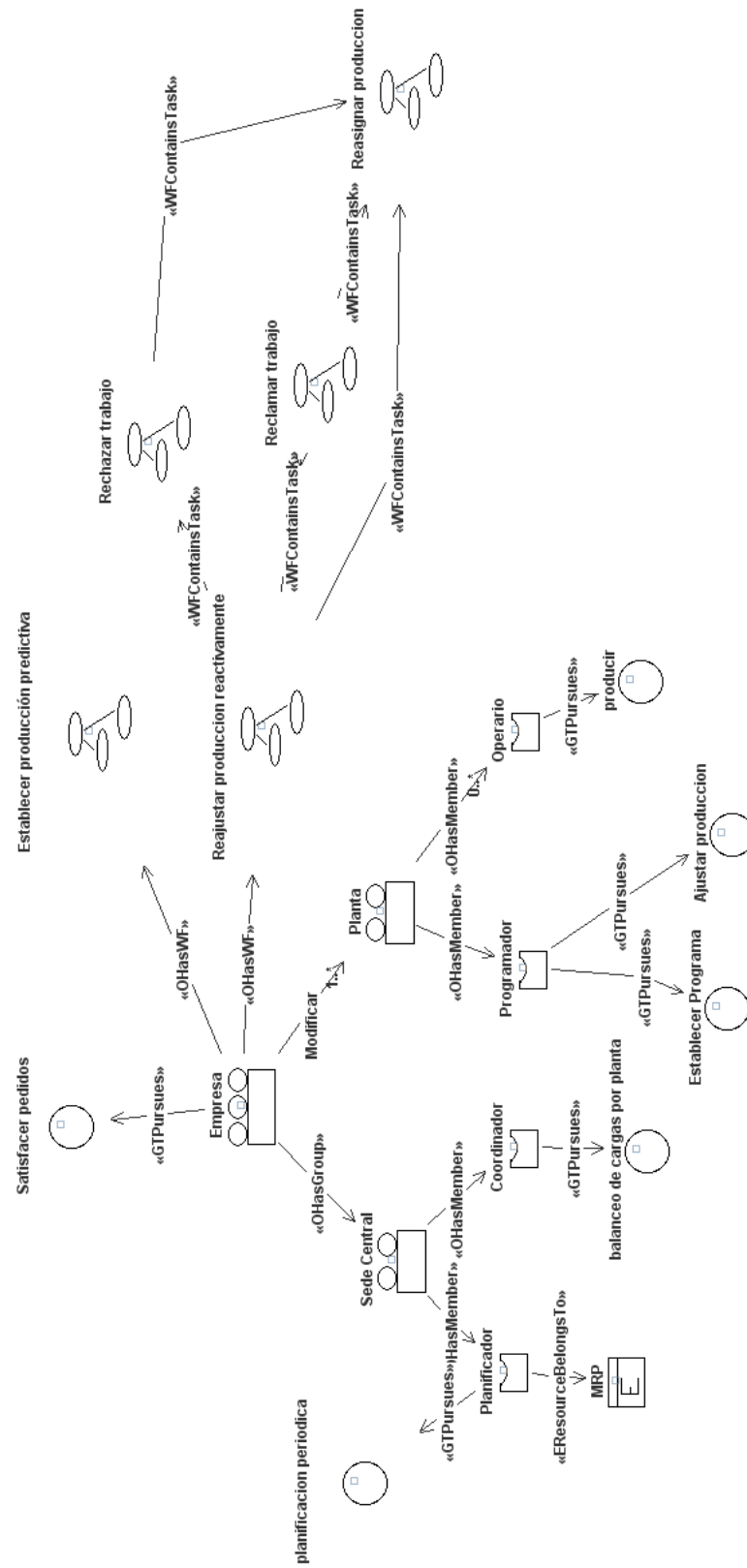


Figura 7-22. Modelo de Organización (elaboración propia).

Una vez conocida la organización en general e identificados los actores fundamentales es necesario revisar el conjunto de recursos que están disponibles en el entorno.

7.4.3.4 Modelo de Entorno

En este modelo el propósito no es generar representaciones del mundo en el que se ubica el Sistema Multiagente, sino una discretización donde categorizar el tipo de entidades relevantes en el entorno y restringir la interacción con las mismas. Así, el entorno sólo contendrá recursos, aplicaciones y Agentes, y se limitará la percepción y actuación de los Agentes.

La Tabla 7-7 muestra las entidades que se desea considerar en el modelo de entorno. Se trata tanto de elementos físicos, como las máquinas, como elementos software pre-existentes.

Entidad	Entidad del meta-modelo	Documentación
MRP	Aplicación externa	El MRP es una aplicación usada por el planificador que a su vez cae fuera del Sistema Multiagente para la Programación de la Producción, aunque suponen fuentes de interacción e información para nuestro problema.
Aplicación Programador Producción	Aplicación interna	Aplicación que implementará el cálculo de Programación de la Producción de acuerdo con la propuesta de esta tesis. Esta aplicación será usada por el programador (ver Figura 7-22).
Horno	Recurso	Representa el conjunto de máquinas de cada planta que se encarga de cocer el bizcocho.
Embaladora / Clasificadora	Recurso	Representa el conjunto de máquinas de cada planta que se encarga de embalar y clasificar las baldosas.
Prensadora y Esmaltadora	Recurso	Representa el conjunto de máquinas de cada planta que se encarga de prensar la tierra atomizada y esmaltar el bizcocho.

Tabla 7-7. Resumen de las entidades del modelo de entorno planteados en la Plataforma IPSU-MAS (elaboración propia).

En la Figura 7-23 se representa como la planta cuenta con un conjunto de recursos a su disposición formado por los sistemas de información MRP y Aplicación de producción, así como las máquinas de cada una de las 3 líneas productivas.

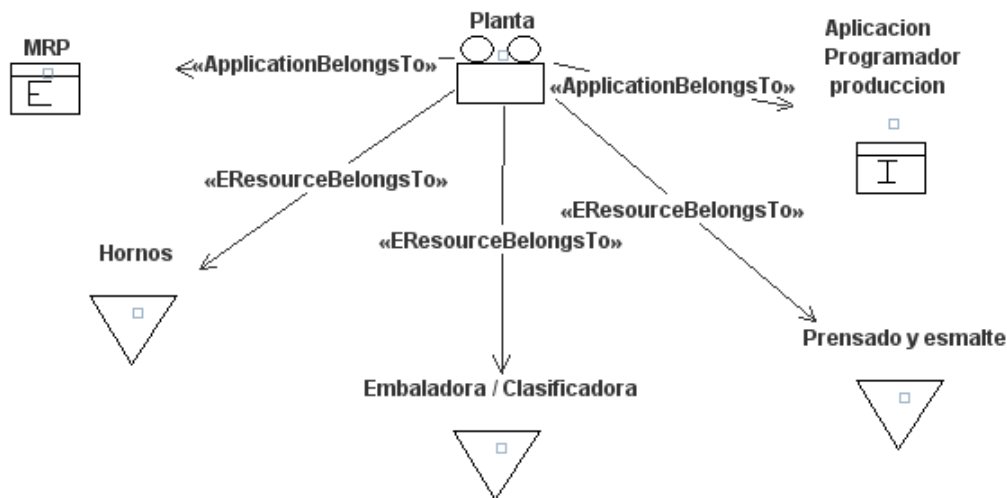


Figura 7-23. Modelo de entorno (elaboración propia).

Una vez se han fijado las entidades de la organización y del entorno la metodología indica que se deben fijar los objetivos y tareas de cada elemento activo.

7.4.3.5 Modelo de Objetivo y Tareas

El modelo de objetivos y tareas tiene como propósito recoger las motivaciones del Sistema Multiagente, definir los flujos de trabajo identificados en el modelo de organización, mediante la descomposición en tareas, en las que se identifican interacciones llevadas a cabo y responsabilidades (roles o Agentes). Muchas de las tareas establecidas en esta fase de diseño han sido elaboradas en la fase de análisis correspondiente al capítulo anterior. Esta información constituye parte de la especificación de cómo se quiere que sea el control del Agente a alto nivel. En el modelo de Agentes se especificarán los Procesadores y Gestores de Estado Mental que establecen el proceso deliberativo y de mantenimiento del conocimiento del Agente y en el modelo de interacción los estados mentales requeridos por parte de un conjunto de Agentes que interactúen. Aquí se trata de poder expresar cuáles son las consecuencias de ejecutar las tareas y de porqué se deberían llegar a ejecutar.

Se han especificado dos modelos de objetivos y tareas, uno por cada uno de los macro flujos de trabajo identificados en el modelo de organización (ver Figura 7-22): *establecer Producción Predictiva* y *Reajustar Producción Reactivamente*.

Establecer Producción Predictiva

Este modelo de objetivos y tareas especifica el conjunto de tareas a realizar para establecer la Producción Predictiva. En la Figura 7-24 se muestra como esta tarea se basa en 3 objetivos que pretende conseguir; Satisfacer la Planificación, Uso Eficiente de los Recursos, Establecer Programa Productivo. Para satisfacerlos debe poner en marcha una serie de tareas, identificadas en el análisis del Proceso de Negocio de Programación de la Producción Predictiva, como son; *Revisar y Cargar los Datos, Calcular el Programa Predictivo, Ajustar, Validar o Rechazar la Programación*. Pero también se han tenido que incluir otras tareas no identificadas en dicho análisis previo, como son; *Notificar la Planificación Periódica o Notificar Operarios* de la Programación. Estas dos tareas se pueden considerar como destinada a establecer un proceso de comunicación que inicialmente no se identificó.

Algunas tareas son iniciadas por el Agente Planificador como la de Notificar Planificación Periódica, y otras son iniciadas por el Agente Programador como la de Revisar y Cargar datos. En todo caso, la ejecución de muchas de las tareas implica el establecimiento de una conversación entre Agentes para intercambiar información.

En el modelo también se especifica responsabilidad de roles en la ejecución de las tareas, hecho que se complementa con los diagramas de Agentes y de interacción. En la Tabla 7-8 se comentan las 4 interacciones que forman parte de este modelo de Objetivos y Tareas.

Entidad	Entidad del meta-modelo	Documentación
INT: Obtener Planificación Periódica	Interacción	Corresponde con la interacción en la que el agente planificador proporciona la Planificación periódica a producir al programador de cada planta, teniendo en cuenta las capacidades de la planta para ese periodo
INT: Notificar Inconsistencias	Interacción	Corresponde con la interacción entre el agente programador y el planificador, en el que el programador notifica que la Planificación es inconsistente con las capacidades productivas de la planta
INT: Modificar Planificación	Interacción	Corresponde con la interacción entre el agente programador y el planificador, en el que el programador solicita o rechaza trabajos de o a la Planificación para ajustar su Programación de la planta a sus capacidades productivas
INT: Notificar Programación Producción Modificada	Interacción	Corresponde con la interacción entre el agente programador de una planta y los Agentes operarios de una planta, en la que el programador tras calcular o recalculer la Programación de la Producción notifica la Programación a los operarios

Tabla 7-8. Resumen de las interacciones del modelo de objetivos y tareas (Predictivos) planteados en la Plataforma IPSU-MAS (elaboración propia).

Los modelos de objetivos y tareas muestran una descomposición de flujos de trabajos en tareas, pero no especifican el orden de ejecución o condicionalidad en la ejecución de las tareas, hecho que si muestran el diagrama de actividades. Para el caso del cálculo de la Programación predictiva, el diagrama de actividades que muestra el flujo de ejecución de actividades es el mostrado en la Figura 7-11 y para la Programación Reactiva se muestra en la Figura 7-12.

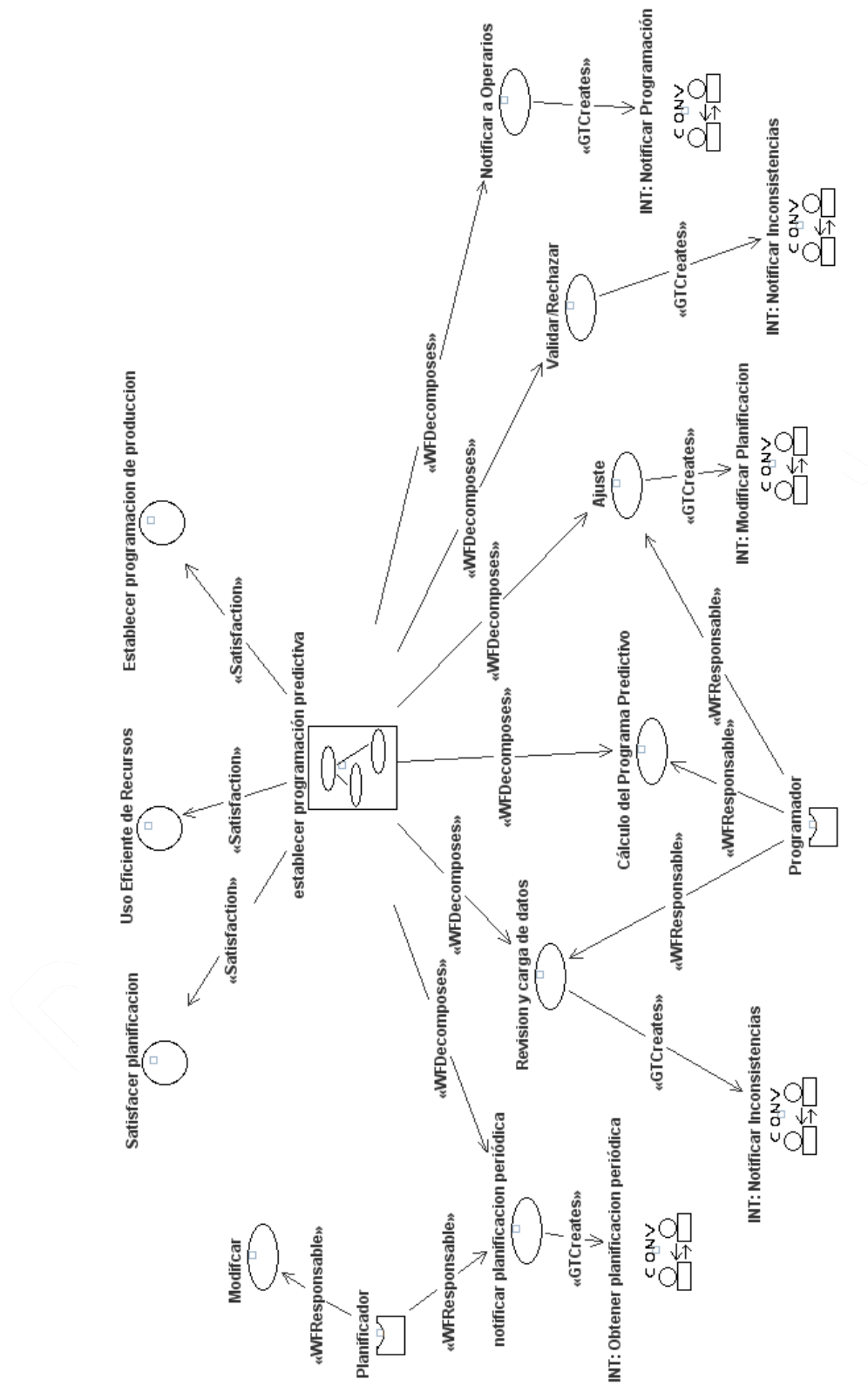


Figura 7-24. Modelo de tareas y objetivos de la Producción Predictiva (elaboración propia).

Ajustar Producción Reactivamente

Este modelo de objetivos y tareas especifica el conjunto de tareas a realizar para establecer la Producción Reactiva. En la Figura 7-25 se muestra como esta tarea se basa en 4 objetivos que pretende conseguir; Satisfacer la Planificación, Uso Eficiente de los Recursos, Mantener el Programa Estable, Establecer Programa Productivo. Para satisfacerlos debe poner en marcha una serie de tareas, identificadas en el análisis del Proceso de Negocio de Programación de la Producción Reactiva, como son; *Establecimiento Estrategia de Reprogramación, Revisión y Carga de Datos, Cálculo del Programa Reactivo, Validad o Rechazar, y Ajuste / Control de la Estrategia de Reprogramación*. A diferencia de lo que ocurrió en el modelo anterior aquí no ha sido necesario incluir tareas adicionales a las establecidas inicialmente. Todas las tareas son iniciadas por el Agente Programador.

En el modelo también se especifica responsabilidad de roles en la ejecución de las tareas, hecho que se complementa con los diagramas de Agentes y de interacción. En este modelo se hacen uso de algunas interacciones entre Agentes que ya se habían identificado en el modelo de Establecimiento de Programación Predictiva, ya que la naturaleza de las comunicaciones entre Agentes es la misma y las semejanzas de algunas tareas en ambos flujos de trabajo son evidentes. En la Tabla 7-9 se describen las interacciones que se llevan a cabo como resultado o como inicio de algunas de las tareas.

Entidad	Entidad del meta-modelo	Documentación
INT: Notificar problema	Interacción	Se corresponde con la interacción en la que el Agente Operario notifica al Agente Programador de un retraso o adelanto con respecto a la producción programada.
INT: Notificar inconsistencias	Interacción	Se corresponde con la interacción entre el Agente Programador y el Planificador, en el que el Programador notifica que la Planificación es inconsistente con las capacidades productivas de la planta.
INT: Modificar Planificación	Interacción	Se corresponde con la interacción entre el Agente Programador y el Planificador, en el que el Programador rechaza trabajos de la Planificación o solicita más para ajustar su Programación de planta a las capacidades productivas.
INT: Notificar Programación Producción Modificada	Interacción	Corresponde con la interacción entre el Agente Programador de una planta y los Agentes Operarios de una planta, en la que el Programador tras calcular o recalcular la Programación de la Producción notifica la Programación a los Operarios.

Tabla 7-9. Resumen de las entidades del modelo de objetivos y tareas (Reactivo) planteados en la Plataforma IPSU-MAS (elaboración propia).

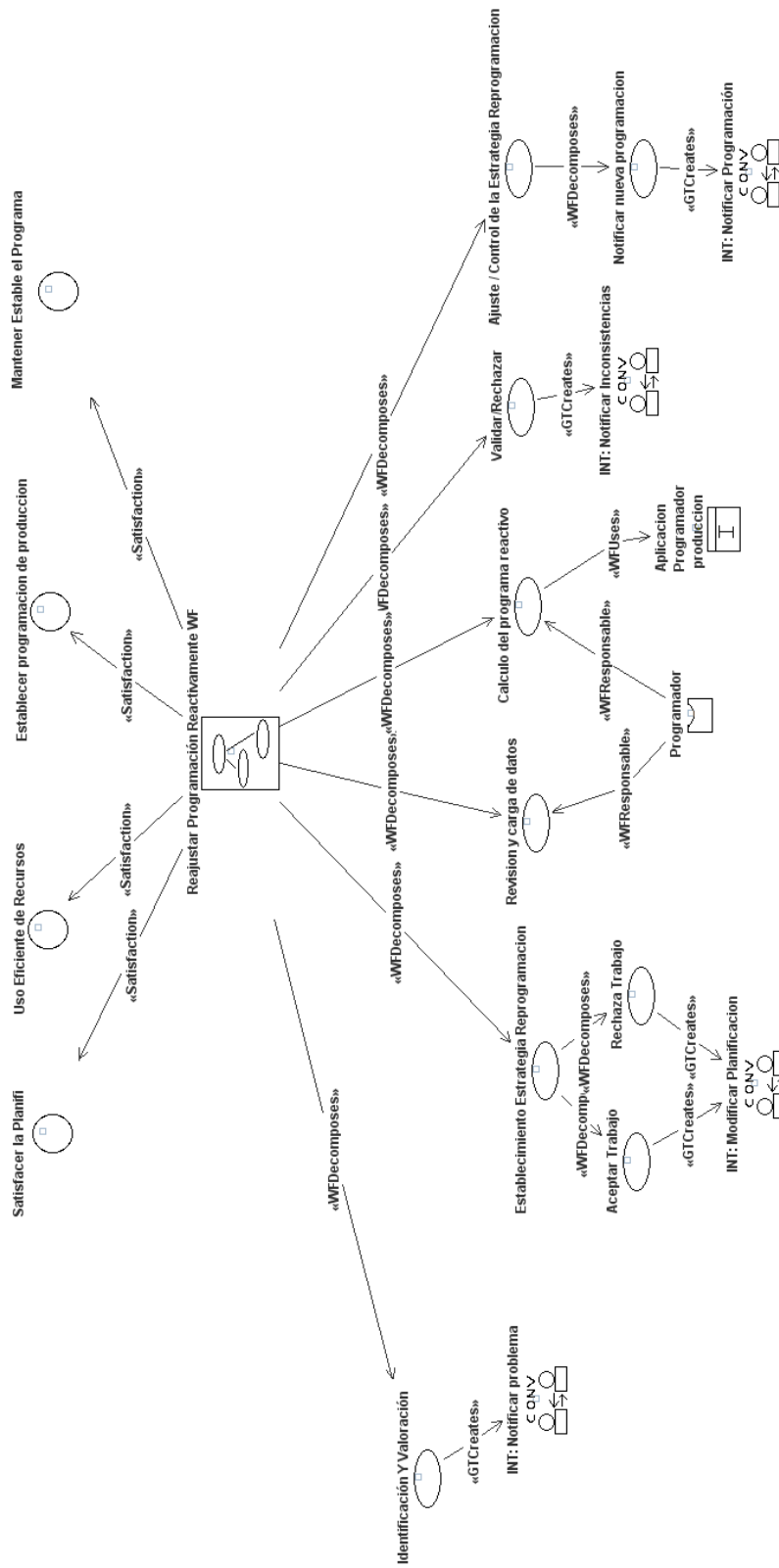


Figura 7-25. Modelo de tareas y objetivos de la Producción Reactiva (elaboración propia).

Para el reajuste de Programación reactivamente el diagrama de actividades que se observa el flujo de ejecución de actividades es el mostrado en la Figura 7-42.

7.4.3.6 Modelo de Agentes

El modelo de agente se usa para describir Agentes particulares excluyendo las interacciones con otros Agentes. Es decir, que en este modelo se describe la funcionalidad del Agente (tareas de las que es responsable), sus objetivos, los roles que juega, aplicaciones de las que hace uso, y su control. Para que un Agente pueda cumplir con las tareas propias del role que se le asigna necesita un soporte sobre el cual mantener sus Creencias sobre el entorno, en el enfoque DBI. Por este motivo, cada Agente tiene asociado un conjunto de Estados Mentales, que actúa como memoria de hechos en la que se identifican los atributos en los que se retienen la información necesaria para llevar a cabo sus objetivos (identificadas mediante *FrameFacts* en los diagramas) a partir de los cuales se establecen las creencias de los Agentes, mostradas mediante nubes.

Se han identificado 3 agentes: Agente Planificador, Agente Programador y Agente Operario. Todos ellos serán comentados a continuación.

El **Agente Planificador**, es un Agente de contexto, ya que su implementación no forma parte del sistema de Programación que se plantea en la tesis, pero sí que es necesario identificarlo en el Sistema Multiagente ya que es una fuente de información de las Planificaciones de Producción periódica por planta que el programador de la Producción necesita. En la Figura 7-26 se observa como este Agente desempeña los roles de planificador y de coordinador, interactúa con una aplicación MRP para obtener la Planificación de la Producción para el conjunto de plantas, se identifican los objetivos del Agente, y las tareas de las que es responsable (*Notificar Planificación Periódica, Modificar Planificación Periódica*). El Agente dispone de un estado mental en el que registra información acerca de la *Planificación Periódica agrupada* y de la *Planificación Periódica desagrupada por plantas* (ver caja *FrameFact* en la Figura 7-26). El Agente en función de sus creencias (Estado Mental en un instante) y de sus Objetivos tomará la iniciativa de calcular lo que modificará su estado mental al de **Elaborando Planificación**, cuando esta tarea acabe notificará la Planificación Periódica al Programador. Cuando el Programador confirme la recepción del Plan el estado Mental debería cambiar su al de **Plan en Curso**, lo que implica la creencia de que el Plan se puede alcanzar sin dificultades a priori. Previo a la ejecución, o durante la ejecución, puede surgir un evento que suponga cambiar a estado mental que indica que existen problemas en el desarrollo del Programa por falta o exceso de capacidad. Los posibles problemas deben ser solucionados mediante la tarea Modificar el Plan Maestro. En todo caso, si eso ocurre el Agente debería

situarse en un estado de **Problemas en el Programa**. Además antes de iniciarse la ejecución del Programa es posible que el Programador le notifique al Planificado de la existencia de una inconsistencia en los datos lo que implicaría que este Agente se situaría en un estado de **Datos Inconsistentes** si finalizaría su existencia.

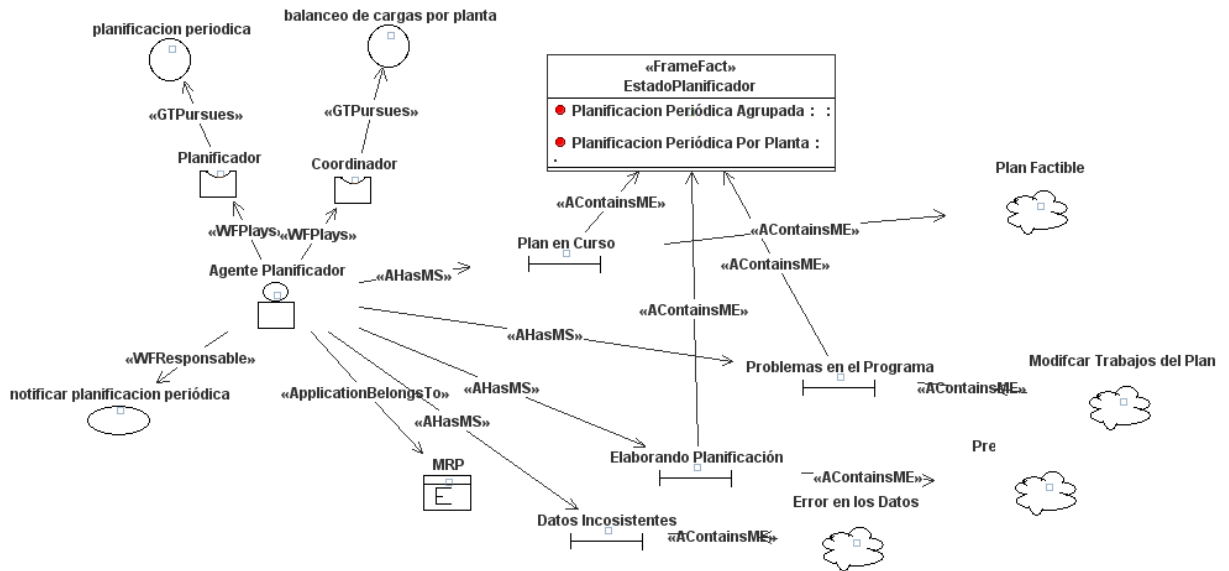


Figura 7-26. Agente planificador (elaboración propia).

En la Tabla 7-10 se muestra un resumen de los roles y tareas asignados, así como las conversaciones contempladas.

Componentes del Agente Planificador	
Conversaciones Previstas	<ul style="list-style-type: none"> • Iniciar notificación Planificación periódica • Recibir Modificar Planificación • Notificar Inconsistencias
Tareas asignadas	<ul style="list-style-type: none"> • Notificar Planificación periódica • Modificar Planificación periódica
Roles	<ul style="list-style-type: none"> • Planificador • Coordinador

Tabla 7-10. Resumen de los componentes del agente planificador (elaboración propia).

El **Agente Programador** es el Agente clave de la propuesta, ya que además de encargarse del cálculo de la Programación Predictiva y los ajustes Reactivos, gestiona los operarios, identifica retrasos y se comunica con el planificador para la modificación del Plan Productivo según las capacidades productivas en cada momento. En la Figura 7-27 se observa como este Agente desempeña el rol de programador, interactúa con una aplicación de Programación de la Producción que implementa los algoritmos que se propondrán en esta tesis, y es responsable de las principales tareas de la Programación y Ajuste de la Producción. El Agente dispone de un conjunto de estados mentales en el que registra información acerca de la planta que Programa, como son: la Planificación inicial que recibe del planificador, la Programación calculada inicialmente y sus modificaciones, el Margen Operativo inicial y el actualizado, la Estrategia Empleada, así como otros parámetros (ver caja *FrameFact* en la Figura 7-27).

Inicialmente el Agente Programador puede estar siempre en una situación de **Espera**. Se asume que esta situación se dará siempre entre periodo y periodo. Bajo esta estado el Agente cree que se recibirá un mensaje del Planificador Notificando la Planificación Periódica. Cuando se recibe el Plan se inicia el proceso de elaboración del Programa que puede culminar sin dificultades llegando ponerlo en ejecución, en cuyo caso el Agente se situaría en un estado de **Programa en Curso**. Sin embargo, si durante la elaboración del Programa se produce alguna dificultad de capacidad que implique solicitar una modificación de los trabajos al Planificador entonces se llegaría a un estado de **Solicitud Modificación**. Si se produjera la necesidad de revisar el programa cuando ya está en curso el Agente se situaría en un estado mental de **Reprogramación**, pudiéndose volver a dar la circunstancia de que se necesite solicitar una modificación del número de trabajos en cuyo caso el estado mental sería **Solicitud de Modificación Reprogramando**. Una situación en la que se puede encontrar el Agente en el momento de empezar a realizar el proceso de programación predictiva es que los datos con los que debe trabajar sean erróneos con lo que este Agente pasará a un estado de **Datos Inconsistentes**.

En la Tabla 7-11 se muestra un resumen de los roles y tareas asignados, así como las conversaciones contempladas.

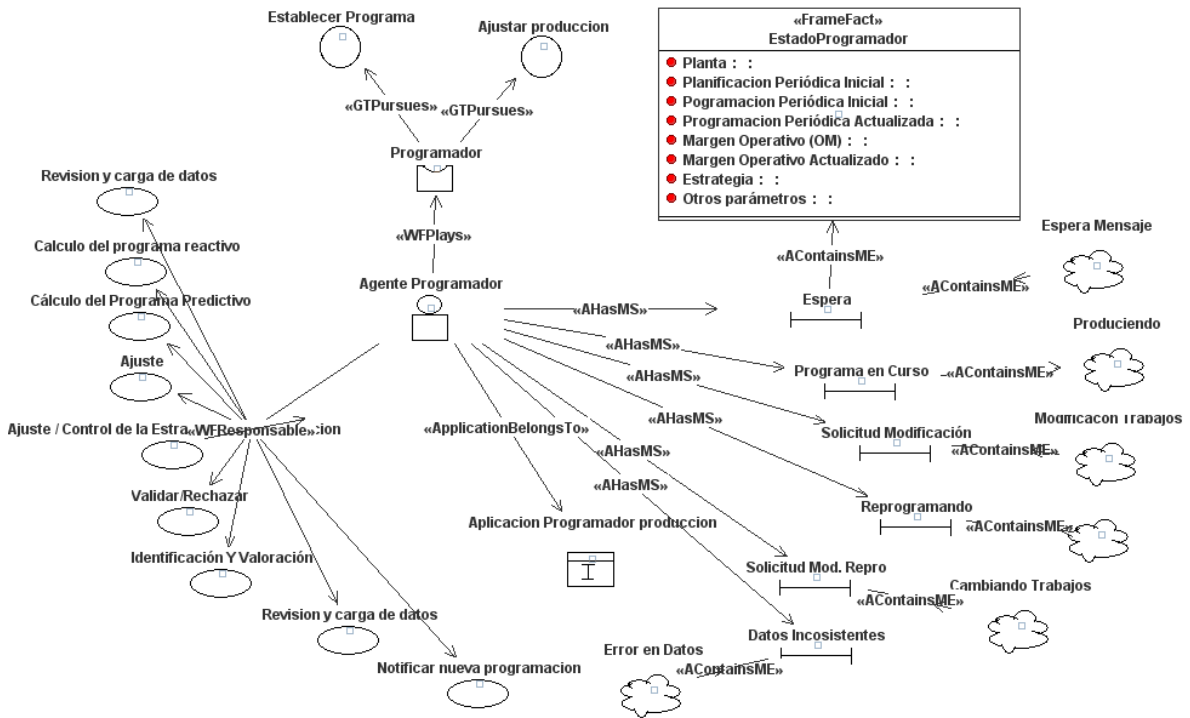


Figura 7-27. Agente Programador (elaboración propia).

Componentes del Agente Programador	
Conversaciones Previstas	<ul style="list-style-type: none"> • Recibir notificación Planificación periódica • Iniciar Notificar inconsistencias • Iniciar Modificar Planificación • Iniciar Reasignar Producción • Recibir Notificar Problema
Tareas asignadas	<ul style="list-style-type: none"> • Revisión y carga de datos • Cálculo del Programa Reactivo • Cálculo del Programa Predictivo • Ajuste • Ajuste / Control de la Estrategia Reprogramación • Validar/Rechazar • Recalcular Margen Operativo • Revisión y carga de datos • Notificar nueva Programación
Roles	<ul style="list-style-type: none"> • Programador

Tabla 7-11. Resumen de los componentes del agente programador (elaboración propia).

El **Agente Operario** es un Agente asignado de una etapa en cada una de las plantas. Así, para cada etapa y planta existirá un Agente Operario. El Agente recibe la Programación de la Producción y sus ajustes, y se encarga de producir de acuerdo a la misma. A su vez, si existe una incidencia o problema que ponga en peligro la producción de acuerdo con la Programación en su etapa correspondiente, el Agente operario lo notifica al Agente programador para que tome las acciones oportunas. En la Figura 7-28 se indica como este Agente desempeña el rol de operario y es responsable de notificar problemas (retrasos o adelantos). El Agente dispone de un estado mental en el que registra información acerca de: La planta y etapa a la que pertenece, la Programación Periódica inicial establecida por el programador que ha de notificar para producir, la Programación Periódica actualizada y niveles de alerta en función de los problemas que acontezcan (ver caja *FrameFact* en la Figura 7-28). En concreto este Agente puede pasar por 3 estado mentales. Su estado inicial sería **En Espera** de que se le notifiquen las órdenes de producción, una vez recibidas éste pasaría a un estado de **Trabajo en Curso**, y si se detecta algún tipo de adelanto o retraso sobre el programa en curso se alcanzaría un estado de **Problemas en Taller**.

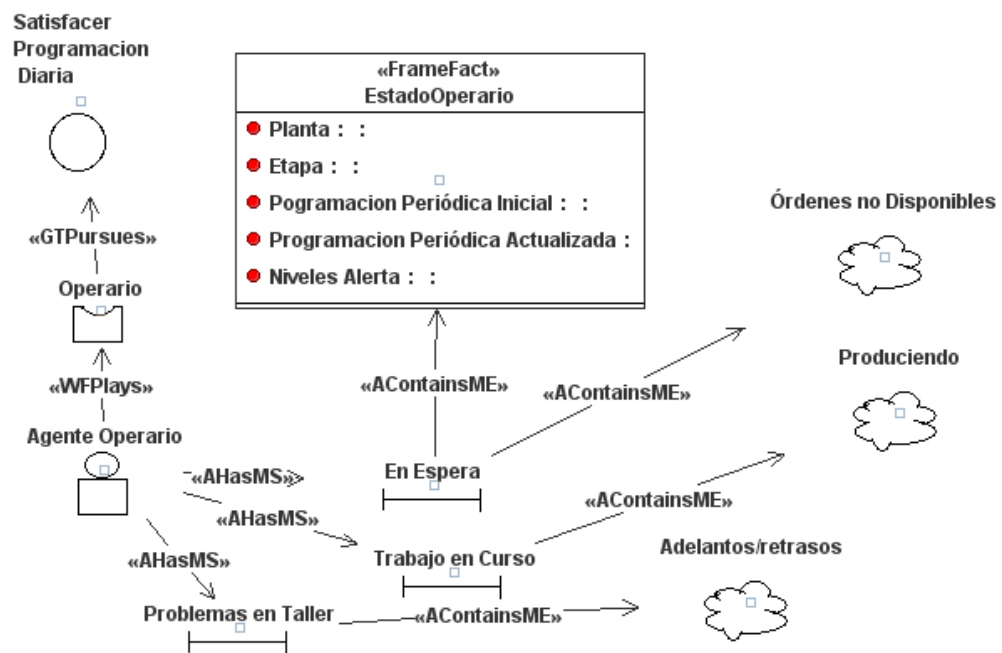


Figura 7-28. Agente Operario (elaboración propia).

En la Tabla 7-12 se muestra un resumen de los roles y tareas asignados, así como las conversaciones contempladas.

Componentes del Agente Operario	
Conversaciones Previstas	<ul style="list-style-type: none"> • Recibir Reasignar Producción • Iniciar Notificar Problema / Retraso
Tareas asignadas	<ul style="list-style-type: none"> • Notificar Problemas / retraso
Roles	<ul style="list-style-type: none"> • Operario

Tabla 7-12. Resumen de los componentes del agente operario (elaboración propia).

El estado mental de los Agentes es modificado cuando acontecen tareas en el Sistema Multiagente. Existen algunos cambios que deben ser identificados con claridad. Así, cuando se recibe la Planificación periódica el Agente Programador modifica su estado mental, hecho que se refleja en la Figura 7-29. En este caso el Agente actualiza la información básica sobre la que trabaja, que no es otra que el conjunto de trabajos que deben ser programados. Esto implica la puesta en marcha de algunas actividades.

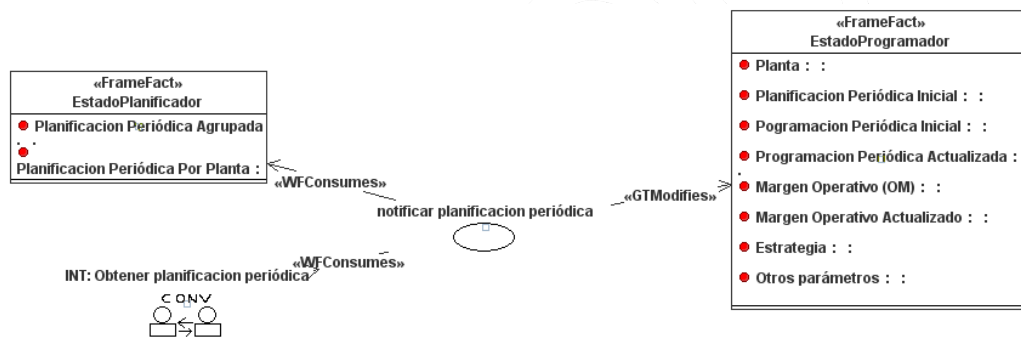


Figura 7-29. Modificación de estados mentales por Notificar Planificación Periódica (elaboración propia).

De igual manera, la modificación de los parámetros que aplica el Agente Programador en el cálculo de la Producción es el resultado de la introducción mediante interfaces de un software de tales parámetros por un usuario experto (ver Figura 7-30).

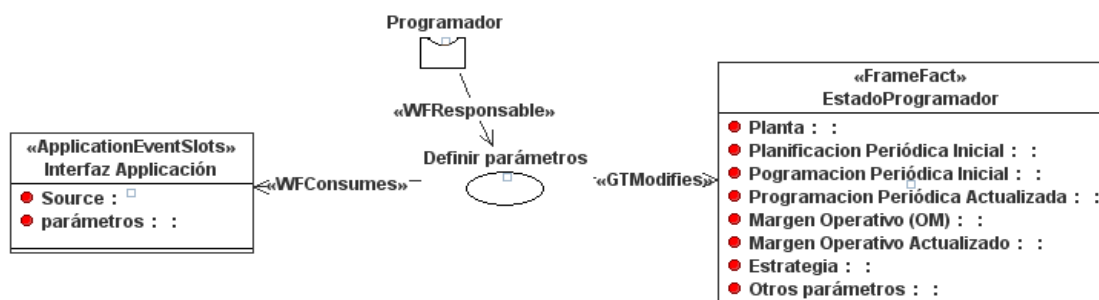


Figura 7-30. Modificación de estados mentales por definir parámetros (elaboración propia).

Las modificaciones del estado mental del Agente Programador con la Programación Periódica actualizada es debido al reajuste del Programa Productivo (ver Figura 7-31) o bien al cálculo de la Programación Predictiva (ver Figura 7-32).

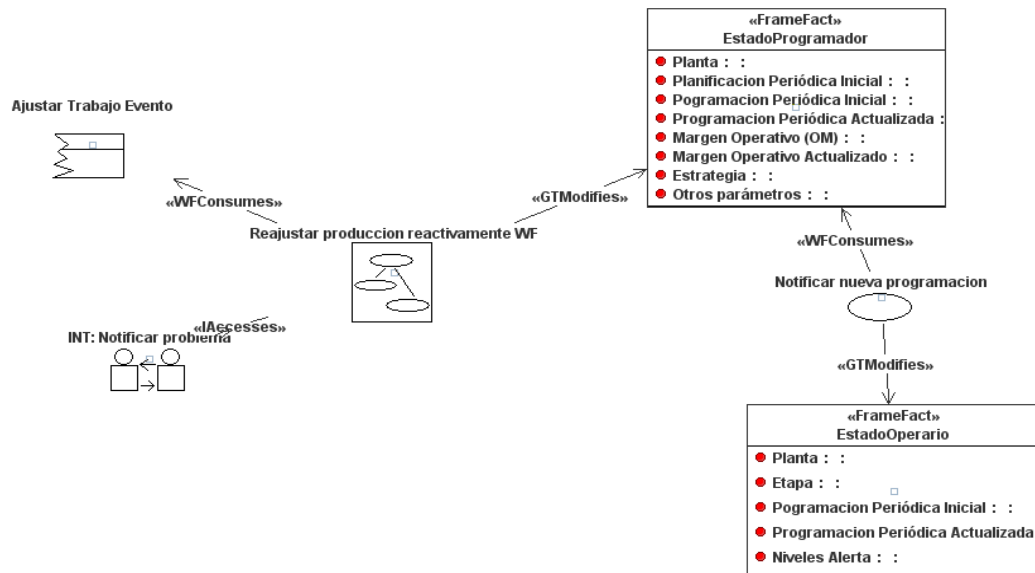


Figura 7-31. Modificación de estados mentales por reajustes de la Producción (elaboración propia).

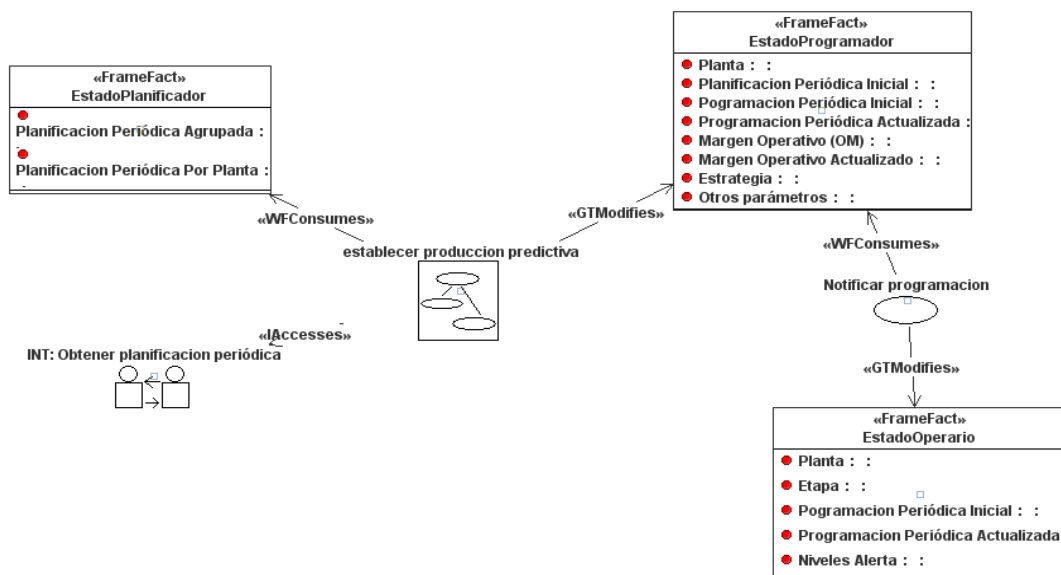


Figura 7-32. Modificación de estados mentales por establecimiento de Producción Predictiva (elaboración propia).

En la mayoría de las ocasiones los cambios en los estados mentales son producidos o generan interacción entre los Agentes, o las dos cosas a la vez. En el siguiente apartado se destacan las interacciones más relevantes.

7.4.3.7 Modelo de Interacciones

Como parte de la ejecución de tareas existe la necesidad de que diversos Agentes interactúen entre sí para el intercambio de información (estados mentales, objetivos, hechos) entre ellos. La interacción entre Agentes se especifica mediante modelos de interacción. **En un modelo de interacción se especifican los Agentes iniciadores de una interacción y los Agentes que colaboran en la misma.** A su vez, se hace referencia a una especificación del protocolo de interacción entre los Agentes, como se muestra por ejemplo en la Figura 7-33. La especificación del protocolo de interacción empleada en este trabajo ha sido la notación del grupo GRASIA⁸, aunque existen alternativas como son UML o AUML. La especificación del protocolo de una interacción especifica el conjunto de pasos, su orden, y participantes para llevar a cabo la interacción entre Agentes, como se muestra por ejemplo en la Figura 7-34.

Las interacciones entre Agentes son parte de la ejecución de una tarea dentro de un flujo de trabajo o workflow. Las tareas que inician las interacciones entre Agentes pueden ser lanzadas por eventos de todo tipo, por interacciones entre Agentes, por una persona real mediante una interfaz, o por la satisfacción de una condición. Las tareas a su vez muestra el cambio o transferencia de información entre Agentes, y como afecta tal ejecución a los hechos (framefacts) y al estado mental del agente.

En este apartado se abordarán las siguientes interacciones: *Obtener Planificación Periódica*, *Notificar Inconsistencias*, *Notificar Inconsistencias*, *Notificar Programación de Producción Revisada*, y *Notificar Problema*.

En el modelo de interacción “*Obtener Planificación Periódica*”, se especifica el objetivo de tal interacción, y se indica que el iniciador de la interacción es el Agente Planificador (relación “Initiates”) como se puede ver en la Figura 7-33, y se especifica en la Tabla 7-13. El Agente que participa en la interacción es el Agente Programador. A su vez, se indica un diagrama en el que

⁸ El grupo GRASIA (GRupo de Agentes Software: Ingeniería y Aplicaciones) pertenece a la Universidad Complutense de Madrid y está coordinado por el Dr. Juan Pavón Mestras. Su página web es [\\www.grasia.fdi.ucm.es](http://www.grasia.fdi.ucm.es).

establecer el protocolo que especifica el conjunto de pasos y mensajes intercambiados para llevar a cabo la interacción, que se muestra en la Figura 7-34. El protocolo indica que la interacción de obtener la Planificación periódica consiste en un primer paso (cada paso de un protocolo se denomina “interaction unit” en la terminología de INGENIAS) en la que el Agente Planificador solicita al Agente Programador de cada planta la capacidad productiva de su planta. La naturaleza de esa solicitud se indica mediante el atributo “SpeechAct” de la unidad de interacción (“Interaction Unit”), que en este paso es de tipo “query”, ya que es una solicitud de información. El mensaje de solicitud se indica mediante el atributo “info” de la “interaction unit” que en este caso es “PP_request_capacidades” donde PP indica las iniciales de los Agentes entre los que se produce la interacción, en nuestro caso Planificador y Programador, y “request capacidades” hace referencia al mensaje de solicitud de capacidades (ver Figura 7-34).

Después de solicitar las capacidades, el Agente Programador notifica al planificador su capacidad. El Planificador tras saber esa información hace un cálculo balanceado de las Planificaciones Periódicas de cada planta y las notifica (ver Figura 7-34).

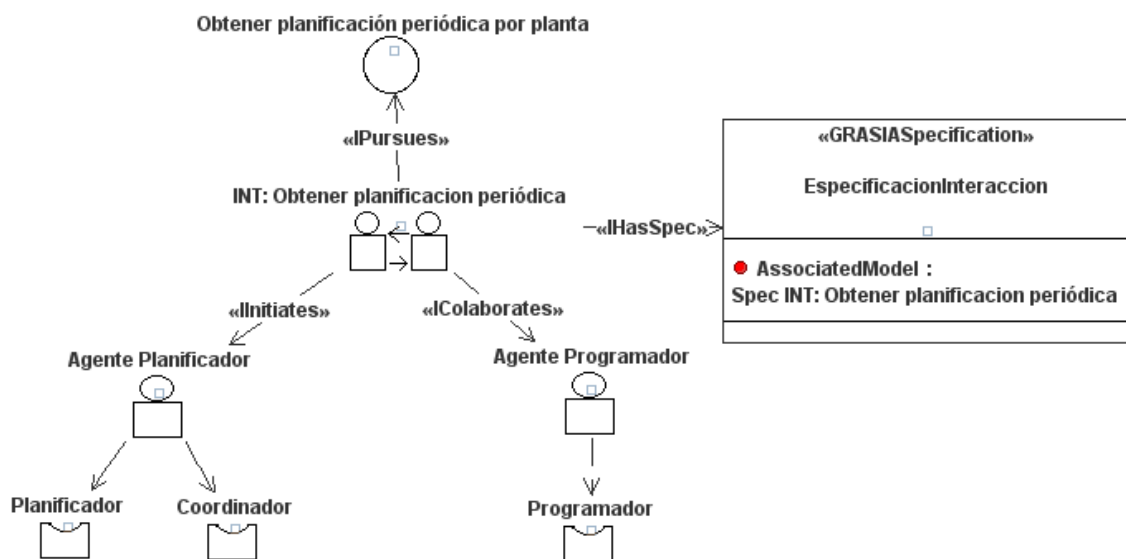


Figura 7-33. Modelo INT: Obtener Planificación Periódica (elaboración propia).

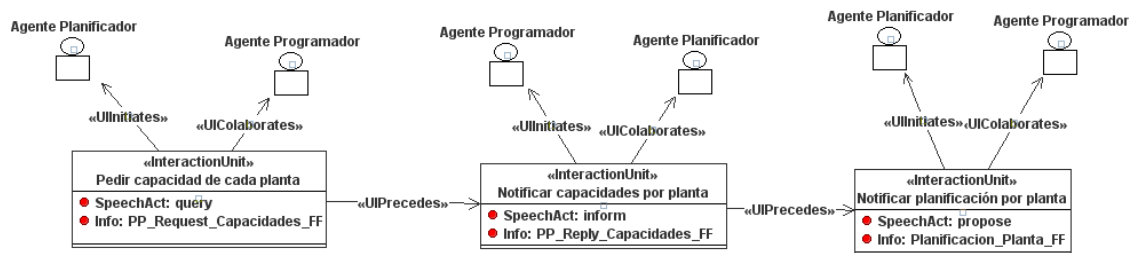


Figura 7-34. Protocolo Obtener Planificación Periódica (elaboración propia).

Entidad	Entidad del meta-modelo	Documentación
Pedir capacidad de cada planta	Interaction Unit	Paso en el que el Planificador solicita a cada Agente Programador de cada planta la capacidad productiva para el periodo que el Planificador desea planificar. El Planificador necesita saber si existen problemas de producción o si ya disponen de stock cada una de las plantas para determinar un balanceo adecuado de la Planificación en el conjunto de plantas de la empresa.
Notificar capacidades por planta	Interaction Unit	Cada Programador notifica las capacidades productivas del periodo solicitado por el Planificador.
Notificar Planificación por planta	Interaction Unit	El Planificador obtiene la Planificación para el conjunto de plantas y notifica la Planificación Periódica a cada uno de los Agentes Programadores.

Tabla 7-13. Resumen de las entidades del protocolo “Obtener Planificación Periódica” planteado en la Plataforma IPSU-MAS (elaboración propia).

La interacción “**Notificar Inconsistencias**” consiste en la notificación de una Planificación inviable de satisfacer por el Agente Programador por alguna inconsistencia en los datos necesarios para el proceso tal y como se puede ver en la Figura 7-35. Tal interacción es iniciada por el Agente Programador y en ella colabora el Agente Planificador. El protocolo que especifica la interacción se muestra en la Figura 7-36. En ella se emite un mensaje de tipo “rechazo” al Agente Planificador en el que indica la Planificación o parte de la misma rechazada y la razón del mismo. En la Tabla 7-14 se comenta el elemento principal del protocolo.

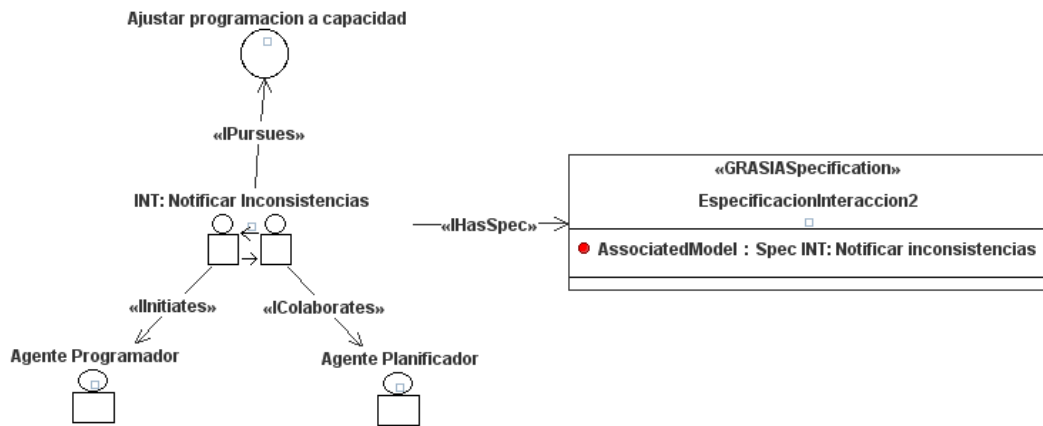


Figura 7-35. Modelo INT: Notificar Inconsistencias (elaboración propia).

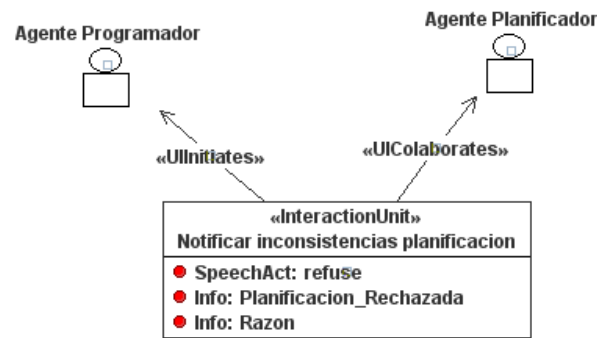


Figura 7-36. Protocolo Notificar Inconsistencias (elaboración propia).

Entidad	Entidad del meta-modelo	Documentación
Notificar inconsistencias de Planificación	Interaction Unit	En caso de que la Planificación sea imposible de satisfacer por la existencia de errores en los datos.

Tabla 7-14. Resumen de las entidades del protocolo “notificar inconsistencias” planteado en la Plataforma IPSU-MAS (elaboración propia).

La interacción “**Modificar Planificación**” resultante de la tarea de “Ajuste en la Programación Predictiva”, incluida en el flujo de trabajo de la Figura 7-24, así como por la tarea “Establecimiento de Estrategia de Reprogramación” en el flujo de trabajo “Reajuste de Programación reactivamente”, como se puede ver en la Figura 7-25, es iniciada por el Agente

Programador y en ella colabora el Agente Planificador (ver Figura 7-37). El conjunto de pasos (las ya comentadas “Interaction Units”) que componen el protocolo se divide en dos: la “modificación de la Planificación” debida al rechazo de trabajo por falta de capacidad de producción de la planta o la “solicitud de más trabajo” por capacidad desaprovechada en la planta. Estas dos alternativas se puede ver en la Figura 7-38 y es especifican en la Tabla 7-15. En el caso de rechazar trabajo el conjunto de pasos del protocolo es el siguiente: el Agente Programador notifica el rechazo de trabajo al Agente Planificador indicando la planta que esta rechazando trabajo y el conjunto de trabajos que no han recibido ninguna operación aún. El Agente Planificador consulta a los Agente Programadores del resto de plantas la disponibilidad de capacidad (ver “Interaction Unit” Consultar Disponibilidad de Reubicación). Cada Programador notifica su capacidad. Y finalmente el Planificador notifica la nueva propuesta y se inicia el Reajuste de la Programación.

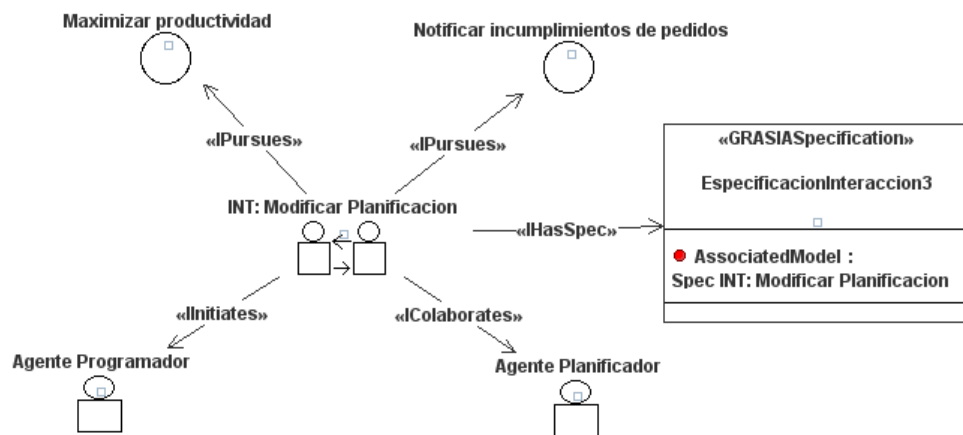


Figura 7-37. Modelo INT: Modificar Planificación (elaboración propia).

En el caso de solicitar más trabajos, el programador notifica la solicitud al planificador, y este se lo proporciona, lo que conlleva el inicio del ajuste de la Programación.

Entidad	Entidad del meta-modelo	Documentación
Rechazar Trabajo	Interaction Unit	El agente programador notifica rechazo de trabajo al agente planificador.
Consultar Disponibilidad de Reubicación	Interaction Unit	El agente planificador consultar las posibilidades de reubicación o alternación temporal de los planes..
Notificar Disponibilidad de Capacidad	Interaction Unit	Cada agente programador consultado notifica la disponibilidad de capacidad en el periodo indicado.
Reasignar Planificación	Interaction Unit	El agente planificador notifica a los Agentes implicados la nueva propuesta de Planificación.
Solicitar Trabajo	Interaction Unit	El agente programador solicita la asignación de más trabajo al agente planificador.
Asignar Trabajo Nuevo	InteractionUnit	El agente planificador asigna trabajo adicional al agente programador.

Tabla 7-15. Resumen de las entidades del protocolo "Modificar Planificación" planteado en la Plataforma IPSU-MAS (elaboración propia).

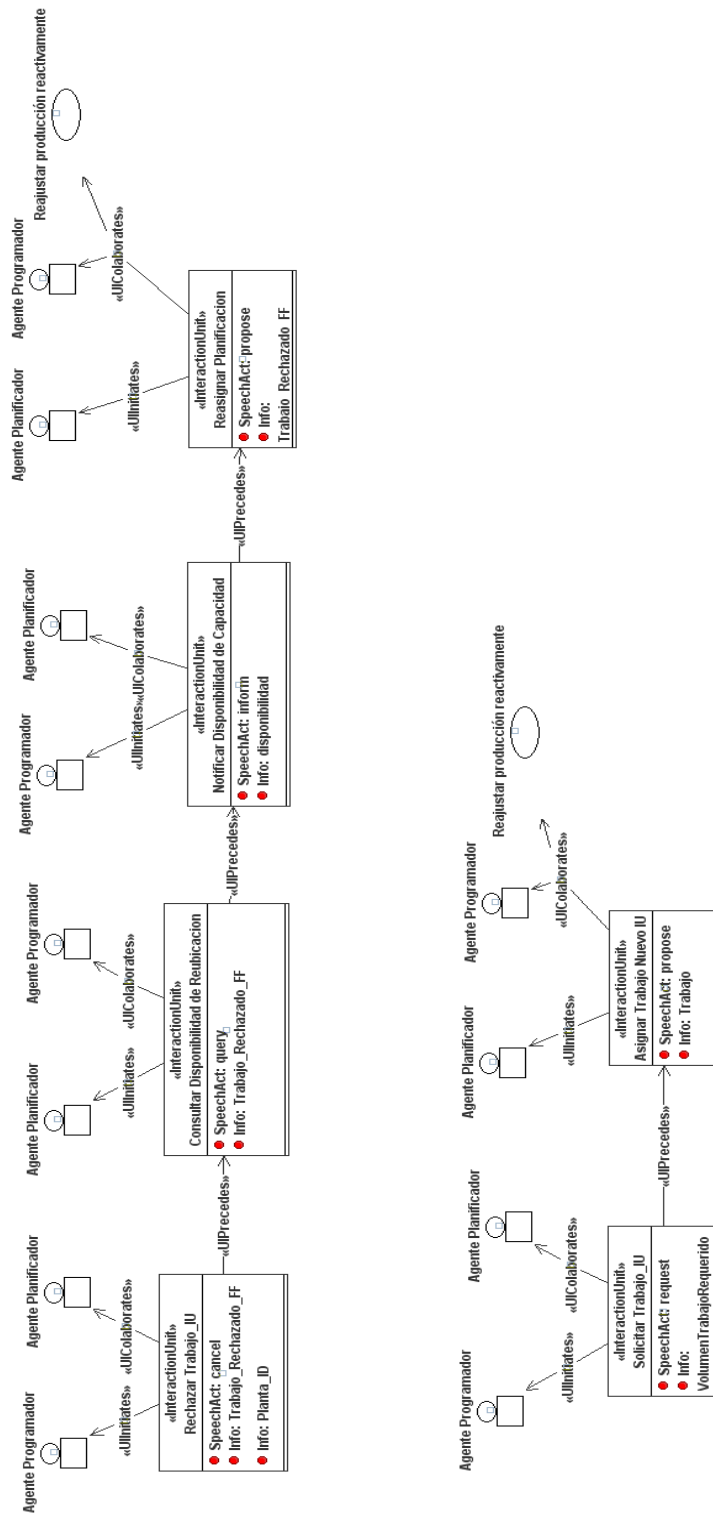


Figura 7-38. Protocolo Modificar Planificación (elaboración propia).

La interacción **“Notificar Programación de Producción Revisada”** es la consecuencia de la obtención de una Programación ya sea Predictiva, como un ajuste Reactivo, ya que consiste en la notificación al Agente Operario de las Órdenes de Fabricación que ha de producir en un periodo, como se observa en la Figura 7-39. El protocolo que describe tal interacción es sencillo, ya que consiste en un mensaje de notificación informativo de la Programación a cada uno de los Operarios. La representación del protocolo se puede ver en la Figura 7-40, y la especificación se indica en la Tabla 7-16.

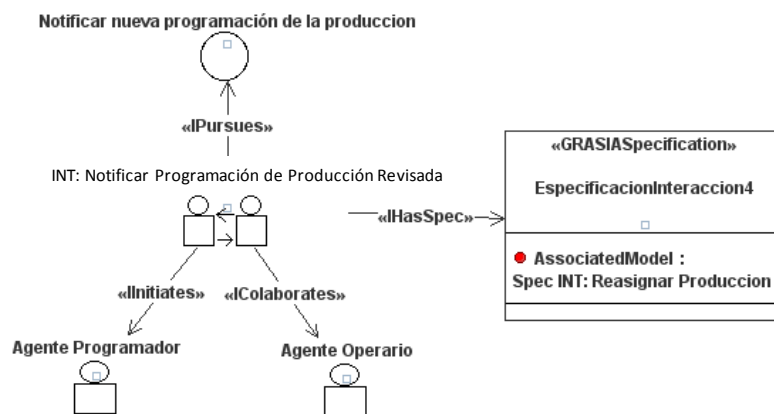


Figura 7-39. Modelo INT: Notificar Programación de Producción Revisada (elaboración propia).

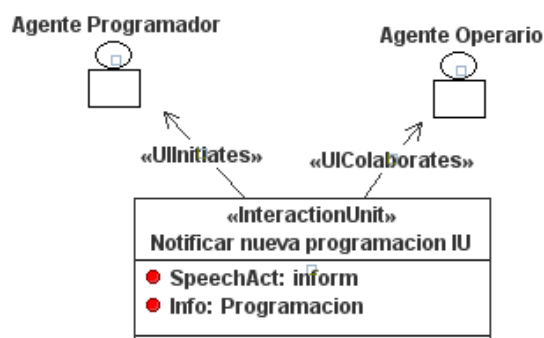


Figura 7-40. Protocolo Notificar Programación de Producción Revisada (elaboración propia).

Entidad	Entidad del meta-modelo	Documentación
Notificar nueva Programación	Interaction Unit	El Agente Programador notifica al Agente Operario la existencia de una nueva Programación de la Producción.

Tabla 7-16. Resumen de las entidades del protocolo "Notificar Programación de Producción Revisada" planteado en la Plataforma IPSU-MAS (elaboración propia).

La interacción "Notificar Problema" es iniciada como consecuencia de la detección de una persona o aplicación del retraso o adelanto de una operación que implica la modificación de la capacidad productiva de una etapa, y por tanto, la incapacidad o riesgo de no cumplir la Programación establecida por el Programador en la planta. Esta interacción iniciada por el Agente Operario y en la que colabora el Agente Programador, consiste de un protocolo muy sencillo en el que el Agente Operario notifica del problema, y nivel de alerta del mismo, como se especifica en la Tabla 7-17 y se puede ver representado en la Figura 7-42.

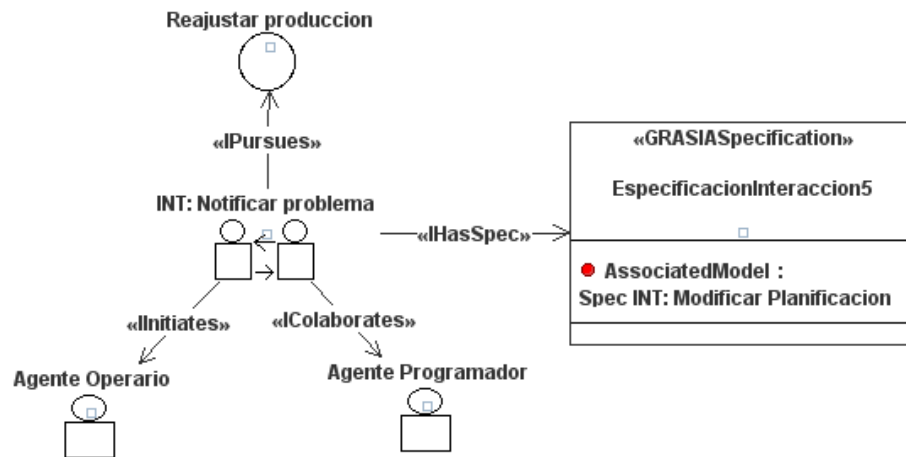


Figura 7-41. Modelo INT: Notificar Problema (elaboración propia).

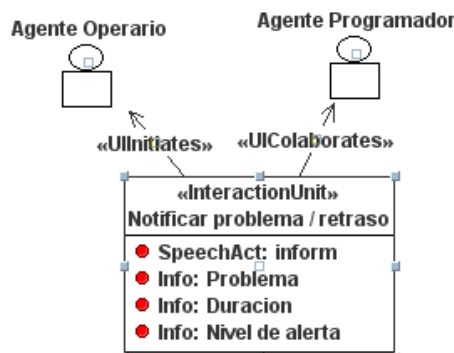


Figura 7-42. Protocolo Notificar Problema (elaboración propia).

Entidad	Entidad del meta-modelo	Documentación
Notificar Problema	Interaction Unit	El Agente Operario notifica al Agente Programador de un problema o un retraso que afecta a la consecución de la Programación establecida por el Agente Programador

Tabla 7-17. Resumen de las entidades del protocolo “Notificar Problema” planteado en la Plataforma IPSU-MAS (elaboración propia).

Una vez descritas las interacciones y protocolos de las interacciones, resta por describir como la intervención de usuarios reales o el acontecimiento de algunas condiciones como es el recalcular temporal del Margen Operativo y su evaluación, pueden determinar el inicio de algunas de esas interacciones. En la Figura 7-43 se realiza una representación gráfica de las mismas.

El re-cálculo del Margen Operativo es realizado periódicamente o en cada modificación del Programa, y su evaluación puede llevar asociada la necesidad de recalcular el Programa Productivo en caso de que los trabajos hayan consumido el Margen Operativo, lo que implicaría rechazar algún trabajo. También se puede dar el caso contrario, y al evaluar el Margen Operativo, se puede detectar la disponibilidad de capacidad productiva suficiente como para solicitar algún trabajo adicional al Agente Planificador. A su vez, en la Figura 7-43, se muestra como la notificación de problemas es iniciada o bien por un operario real (ente humano) mediante el uso de una aplicación informática (denominada “aplicación de usuario”).

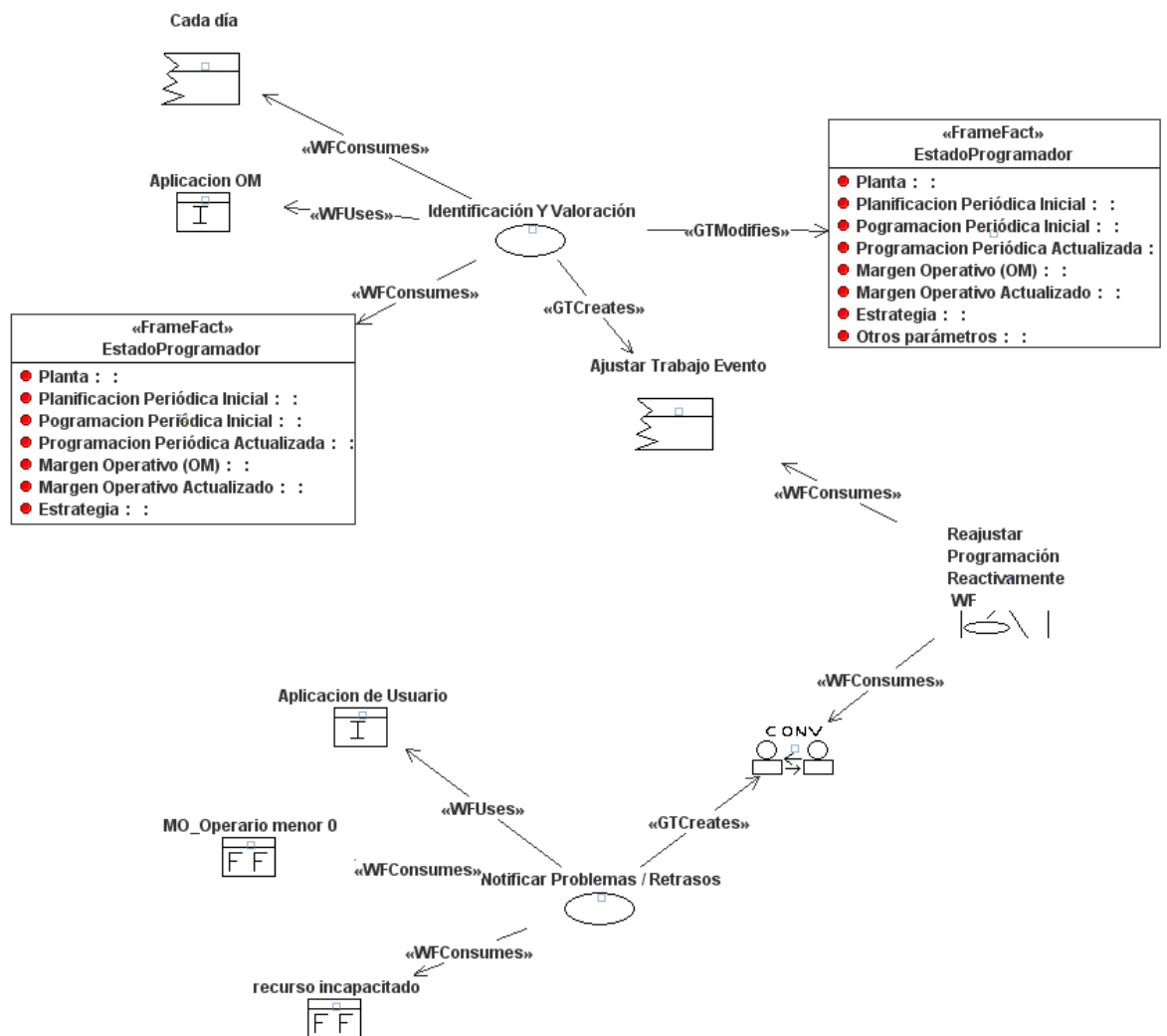


Figura 7-43. Condiciones iniciadoras del Ajuste de la Producción (elaboración propia).

Con el conjunto de modelo (Organización, Entorno, Objetivos y Tareas, Agentes e Interacciones) se dispone de una base suficiente para comenzar la implementación de la Plataforma **IPSU-MAS**.

7.4.4 Relación entre la Metodología de Análisis y de Diseño

Este apartado se destina a analizar brevemente la relación existente entre CIMOSA e INGENIAS. Después de un proceso en el que se combinan de forma novedosa ambos enfoques,

y en un punto del desarrollo de la propuesta en el que ambos ya son conocidos con cierto detalle es conveniente analizar cual ha sido el grado de aportación de cada uno.

En la Tabla 7-18 se muestra una relación entre los modelos INGENIAS y CIMOSA. La información se estructura en la tabla de forma que existe una columna en la que se ubican los 5 modelos empleados con INGENIAS para diseñar la Plataforma **IPSU-MAS**, y dos columnas adicionales en las que se muestran las aportaciones que se han necesitado tener disponibles para elaborar cada modelo INGENIAS. Estas aportaciones son por un lado las realizadas con el análisis de recursos obtenido con CIMOSA, y por otro lado, análisis y reflexiones ad-hoc para construir el modelo correspondiente alcanzando el objetivo asignado por INGENIAS a dicho modelo.

Modelos INGENIAS	Aportación Vistas CIMOSA	Aportaciones Adicionales
ORGANIZACIÓN	Flujos de trabajo → FUNCIONAL (Nivel Proceso Negocio)	Roles: A partir de información de la Vista Funcional: Dominios relacionados y eventos. Grupos: Análisis de la realidad.
ENTORNO	Aplicación (Programación Producción): RECURSOS	Aplicación (MRP): Complementada mediante Análisis de la realidad. Recurso: Complementado mediante Análisis de la realidad.
OBJETIVOS y TAREAS	<i>Tantos modelos como Procesos de Negocio CIMOSA</i> Objetivos: FUNCIONAL (Asociados al Proceso de Negocio) Tareas: FUNCIONAL (Enterprise Activity) Interacciones: FUNCIONAL (Detalle de las Enterprise Activity)	Roles: A partir del Modelo de Organización INGENIAS. Interacciones: Complementado con el reparto de Roles.
AGENTES	Estados Mentales: FUNCIONAL (Detalle de las Enterprise Activity) y INFORMACIÓN.	Se ha identificado un agente por Role, excepto en el caso del Agente Planificador. Tareas: A partir del Modelo de Objetivos y Tareas INGENIAS. Roles: A partir del Modelo de Objetivos y

		<p>Tareas INGENIAS.</p> <p>Interacciones: A partir del Modelo de Objetivos y Tareas INGENIAS.</p> <p>Recursos: A partir del Modelo de Entorno INGENIAS.</p>
INTERACCIONES	<p>Objetivos: FUNCIONAL (Detalle de las Enterprise Activity).</p> <p>Protocolos: INFORMACIÓN.</p>	<p>Objetivos: Complementados con un análisis detallado de la Tarea.</p> <p>Tareas: A partir del Modelo de Objetivos y Tareas o Agentes INGENIAS.</p> <p>Agentes: A partir del Modelo de Objetivos y Tareas o Agentes INGENIAS.</p> <p>Protocolos: Complementados con un análisis detallado de la Tarea.</p>

Tabla 7-18. Aportaciones básicas para la construcción de los modelos INGENIAS (elaboración propia).

En el modelo de Organización se manejan tres elementos conceptuales: Flujos de Trabajo, Grupos, Roles. Por un lado, los flujos de trabajo hacen referencia a procesos con un alto nivel de abstracción que en este caso están fuertemente ligados a los conceptos de Proceso de Negocio identificados con CIMOSA en la fase de análisis conductual. Por otro lado, ni los Grupos, que hacen referencia a niveles jerárquicos en los que se desarrolla el proceso, ni los roles, que se emplean para agrupar responsabilidades sobre diversas tareas, que están relacionados con la vista de Organización se han desarrollado en el análisis CIMOSA. Estos dos últimos elementos han necesitado de una reflexión complementaria a CIMOSA aunque, en la parte relativa a los roles, se ha basado en gran medida en la información recogida sobre los Dominios relacionados al de Programación de la Producción y eventos generados y recibidos.

En el modelo de Entorno se distinguen dos elementos: Las aplicaciones y los recursos. De las aplicaciones sólo las internas (Programador Producción) se concretan a partir del contenido de la vista de Recursos elaborada en CIMOSA, ya que esta vista se centra sólo en el Dominio de Programación de la Producción y no analiza los recursos del entorno. No obstante, cabe recordar que la fase desarrollada con CIMOSA en relación a la vista de Recursos sólo se detallan las capacidades y no los recursos físicos, por este motivo es necesario realizar una aportación adicional basada en la experiencia para estos y para las aplicaciones externas (MRP).

En el modelo de Objetivos y Tareas sólo se *identifican* y se *relacionan* los objetivos, las tareas o procesos, los roles que intervienen y las conversaciones o interacciones que se puedan dar.

Para el desarrollo de este modelo se ha seguido la división propuesta por CIMOSA en Procesos de Negocio, generando en este caso un modelo de Objetivos y Tareas para el Proceso de Negocio de Programación Predictiva y otra para el de Programación Reactiva. Los objetivos y las tareas se obtienen directamente de lo establecido en el Proceso de Negocio correspondiente definido en CIMOSA. Las tareas se especifican al mismo nivel que las Enterprise Activities de CIMOSA. En este modelo se emplean aquellos Roles que se considera oportuno de aquellos que fueron previamente definidos en modelo INGENIAS. Las Interacciones se definen a partir del estudio conductual realizado con CIMOSA y que concluyó con el desarrollo de secuencias de tareas^{ix} que constituirían cada una de las Actividades Empresariales. Mediante el análisis de esa secuencia se identifica que se debe hacer y teniendo en cuenta los roles identificados se identifican cuantas interacciones sean necesarias.

En el modelo de Agentes sólo se *identifican y se relacionan* las tareas, los roles que asume un agente e interacciones que se puedan dar. También se puede complementar con la identificación de los Recursos y los Estados Mentales^x. Se ha considerado oportuno identificar un agente por Role, excepto en el caso del Agente Planificador en el que se agrupan los roles de coordinador y planificador. Todos los elementos han sido previamente identificados y relacionados en el modelo de Objetivos y Tareas, por lo que en esta fase lo que hay que hacer es segmentar y agrupar los Roles, las tareas y las posibles interacciones vinculadas a los mismos a un agente concreto. En este caso el proceso de división y asignación de dichos elementos no se basa de forma directa en ninguna vista de CIMOSA, está vinculado con la teoría de Agentes y la proyección que se realiza en base a esta de las futuras posibilidades de coordinación y colaboración entre Agentes. Por lo que se refiere a los recursos vinculados con un agente éstos se obtiene del modelo de Entorno. Los Estados Mentales son situaciones estables en las que se encuentra un agente en función de la información disponible y que se modifica por ejecución de una tarea. Dicho Estados Mentales se definen en base al conocimiento obtenido en la vista funcional y de información realizado en CIMOSA.

Finalmente en el modelo de Interacciones se definen tantas interacciones como se hayan identificado en el modelo de Tareas y Objetivos o en el modelo de Agentes. Para cada interacción se define su objetivo y con que protocolo se desarrolla la conversación y se identifica la tarea en la que se realiza y los que Agentes intervienen. Con la información

^{ix} Las secuencias de tareas no fueron hechas públicas en el desarrollo del modelo CIMOSA, y se mantiene como material de trabajo.

^x Se puede entender como Estado Mental de un Agente al conjunto de cláusulas lógicas mediante las cuales el Agente representa su conocimiento.

obtenida en la vista funciona, en la que se describe detalladamente los procesos, y en la vista de información en la que se detallan los datos disponibles para la toma de decisiones, se desarrollan los objetivos y los protocolos. Las tareas y los Agentes ya han sido identificados en el modelo de Tareas y Objetivos o en el modelo de Agentes.

7.5 Implementación de la Plataforma IPSU-MAS

El objetivo de la fase de implementación es dar forma mediante un código ejecutable al diseño obtenido en la fase anterior. En este sentido cabe volver a recordar que en la implementación se incluye el desarrollo de los algoritmos de cálculo de la Programación Predictiva y Reactiva que se analizarán en los capítulos 8 y 9 respectivamente. Así mismo, existen algunos aspectos que no han sido desarrollados en detalle en la fase de diseño ya que el objetivo de esta fase era realizar un diseño no detallado, pero que éste será complementado por la elaboración de los algoritmos/procedimientos que se necesiten para abordar puntualmente un caso real. En el capítulo 10 en el que se abordará la aplicación de la Plataforma a un caso particular se comentarán y propondrán estos procedimientos detallados y puntuales.

En este apartado sólo se pretende comentar algunos aspectos relativos a la implementación, pero por ser un trabajo puramente de codificación se ha considerado que su aportación a este trabajo es mínima.

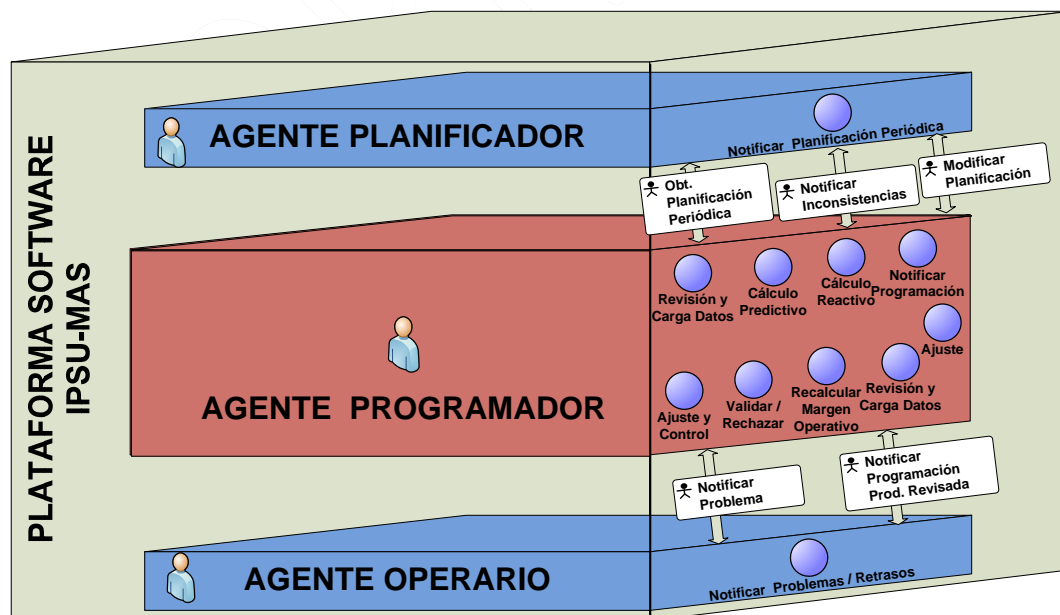


Figura 7-44. Visión de los componentes fundamentales de la Plataforma IPSU-MAS (elaboración propia).

La Figura 7-44 muestra de forma resumidas elementos más relevantes a la hora de codificar la Plataforma **IPSU-MAS**. Por un lado, se muestran los 3 Agentes identificados: El agente Planificador, el agente Programador y el agente Operario. Por otro lado, se muestran las tareas que debe realizar y las interacciones en las que debe participar cada agente en base al enfoque que se le ha dado a la Plataforma. Como se ve los Agentes Planificador y Operario, que se han definido como de contexto, deben ser implementados porque aunque disponen de una funcionalidad muy reducida son básicos en el proceso. Dado que este trabajo se centra en el proceso de Programación de la Producción, las tareas de los 2 Agentes de contexto se implementaran en base a procesos decisionales muy simples que serán definidos en su mayoría en el capítulo 10.

La Plataforma **IPSU-MAS** ha sido implementada utilizando el lenguaje de programación JAVA para su desarrollo, en concreto se ha utilizado el entorno de desarrollo de libre distribución ECLIPSE 3.4, que se puede descargar de <http://www.eclipse.org/platform>. Se ha seleccionado el lenguaje de programación JAVA por ser en el cual está desarrollada la propuesta JADE que ha servido de soporte.

Como se ha dicho en el párrafo anterior para el desarrollo de la Plataforma **IPSU-MAS** se han empleado un conjunto de facilidades agrupada en lo que se conoce como Java Agent DEvelopment framework (JADE), en concreto se ha empleado la versión 3.5. JADE es una solución de uso libre que compila una de las soluciones de base más completas en cuanto a funcionalidad al mismo tiempo que cumple los estándares internacionales que regulan el desarrollo de aplicaciones con Agentes.

7.6 Conclusiones

En este capítulo se han abarcado las cuatro primeras fases de la metodología propuesta en el capítulo 6. Sin embargo, no todas han quedado cubiertas en la misma medida. Por un lado, las **fases 1 y 2, correspondientes a la Definición Elemental del Problema y al Análisis de requerimientos se han desarrollado por completo**, pudiendo dar por definitivos los resultados obtenidos. Por otro lado, la **fase 3 en la que se realiza el Diseño de la Solución queda pendiente de las aportaciones de los capítulos 8 y 9** en relación a los algoritmos de cálculo, y en menor medida del diseño y desarrollo de funciones puntuales que no se han afrontado en este capítulo. Por último, la fase 4 no ofrece ningún resultado inteligible de forma razonable que pueda ser mostrado en formato papel al ser aquella en la que se genera la codificación del software. En todo caso, se puede considerar que la revisión de forma conjunta del trabajo y los resultados alcanzados en el conjunto de estas cuatro fases, tal y como se ha planteado, aporta

una visión bastante ilustrativa y útil para analizar una de las aportaciones más relevantes que se ofrece en la Tesis: La Plataforma **IPSU-MAS**.

Las dos primeras fases relacionadas con la visión general y detallada de los requerimientos han permitido realizar una aproximación en primer lugar y un sólido modelo en segundo lugar. Hay que resaltar que **la aplicación de la metodología CIMOSA es escasa en la literatura y que no han sido pocos los problemas de interpretación de la propuesta**, finalmente resueltos con bastante reflexión y colaboración de otros miembros del grupo de investigación CIGIP. Una de las ventajas de la aplicación de CIMOSA, y uno de los grandes logros alcanzados, ha sido el análisis del problema con un enfoque múltiple (vistas) que además ha quedado sustentado en el anexo I y que podrá ser empleado en otras propuestas.

La fase de diseño de la Plataforma IPSU-MAS ha servido, además de para alcanzar esquema de la solución que se propone, para **poner de manifiesto la forma de enfocar la solución a un problema basado en un Sistema Multiagente**. Se ha podido observar como en el proceso de diseño han cobrado una enorme relevancia los aspectos “sociales” de los elementos principales, los Agentes. Los diseños basados en propuestas de software tradicional suelen preocuparse exclusivamente de objetivo a alcanzar y actividades a realizar para cumplir los primeros. Sin embargo, cuando se aborda el diseño de un Sistema Multiagente el foco de atención se aleja permitiendo considerar un campo de actuación más amplio. De esta forma, se empieza con considerar el **sistema organizativo y los recursos del entorno**, que posteriormente influirán de forma decisiva en la consecución de los objetivos. Posteriormente, y una vez fijados los tradicionales objetivos y tareas, se analizan con detenimiento las entidades, Agentes, a los cuales se vinculan con su entorno y sus tareas, y de forma absolutamente relevante, se desarrollan los escenarios vinculados a las interacciones entre Agentes. Sin duda, un lector no familiarizado con este proceso puede haber quedado sorprendido por los aspectos considerados.

El objetivo final nunca ha sido detallar todos y cada uno de los aspecto relacionados con el diseño de la Plataforma, sólo aquellos más destacables, sobre todo vinculados a la metodología INGENIAS. En todo caso, los elementos generados son una base sólida en la que apoyar la siguiente fase, es decir, la implementación. Aunque en el proceso de diseño se ha realizado un ajuste de los resultados, sin duda, en la fase de implementación será necesario detallar algunas cuestiones que no han sido abordadas aún.

En los capítulos 8 y 9 se ampliará el diseño incluyendo algoritmos para el cálculo de Programas de Producción Predictivo y Reactivo respectivamente. Una vez completado el desarrollo de todos los elementos que deben formar parte de la Plataforma en el capítulo 10 se realizará una prueba del funcionamiento conjunto de toda la Plataforma. Será en dicho capítulo cuando

adicionalmente se desarrollen aspectos muy concretos en relación con algunos procesos decisionales que se hay dejado planteados pero no resueltos en éste por no ser objeto del trabajo que se plantea aunque sí necesarios para un funcionamiento de conjunto.

7.7 Bibliografía

Berio, G. & Vernadat, F. B. 1999a. *New developments in enterprise modelling using CIMOSA. Computers in Industry, 40(2-3): 99-114.*

Berio, G. & Vernadat, F. B. 1999b. *New developments in enterprise modelling using CIMOSA. Computers in Industry, 40(2-3): 99-114.*

Brazier, F., Dunin Keplicz, B, Jennings, N., and Treur, J. *Desire:Modelling multi-agent systems in a compositional formal framework. 1997.*

Castro, J., Kolp, M., & Mylopoulos, J. 2002. *Towards requirements-driven information systems engineering: The Tropos project. Information Systems.*

ESPRIT Consortium AMICE 1993. *CIMOSA - Open System Architecture for CIM.*

Gomez, J. 2002. *Modelado de Sistemas Multiagente. , Departamento de Sistemas Informáticos y Programación, Universidad Complutense de Madrid.*

Iglesias-Fernández, C. A. 1998. *Definición de una metodología para el desarrollo de sistemas multiagente. Departamento de Ingeniería de Sistemas Telemáticos. Universidad Politécnica de Madrid.*

K.Kosanke 1995. *CIMOSA - Overview and Status. Computers in Industry, 27(2).*

Kinny, D. and Georgeff, M. *Modelling and design of multi-agent systems. Technical Report 59. 1996. Melbourne, Australia, Australian Artificial Intelligence Institute.*

Kosanke, K. 1995. *Cimosa - Overview and Status. Computers in Industry, 27(2): 101-109.*

Lind, J. 1999. *MASSIVE: Software Engineering for Multi-agent Systems. DFKI.*

M.Zelm, F.Vernadat, & K.Kosanke 1995. *The CIMOSA business modelling process. Computers in Industry, 21(2).*

Odell, J., Parunak, H., and Bauer, B. *Extending UML for agents. Proceedings of the Agent-Oriented Information Systems Workshop , 3-17. 2000.*

Pavon, J. & Gomez-Sanz, J. 2003. *Agent oriented software engineering with INGENIAS. Multi-Agent Systems and Applications II, Proceedings, 2691: 394-403.*

RGCP. *From CimOsa Architecture to CimOSa/rg Methodology. 1999. www.rgpc.com.*

Wood, M. F. 2000. *Multiagent Systems Engineering: A Methodology for Analysis and Design of Multiagent Systems*. . Air Force Institute of Technology.

Wooldridge, M., Jennings, N. R., & Kinny, D. 1998. *A Methodology for Agent-Oriented Analysis and Design*. Seattle, WA.

BORRADOR

**Algoritmos para la Programación
Predictiva de la Producción en un
Taller de Flujo Híbrido**

Capítulo

8

Tabla de contenido

8.	Capítulo 8	471
8.1	Introducción	471
8.2	Ámbito de Aplicación y Metodología	472
8.2.1	Identificación del Problema	472
8.2.2	Identificación de la Propuesta	475
8.2.3	Metodología para el Desarrollo y Evaluación de la Propuesta	476
8.2.3.1	El Juego de Datos	480
8.2.3.2	La Medida para la Evaluación de los Algoritmos.....	494
8.3	Diseño de un Nuevo Algoritmo Genético basado en Sistemas Multiagente Aplicado al Taller de Flujo Híbrido con Tiempos de Cambio de Partida Dependientes de la Secuencia. 496	
8.3.1	Estructura Tradicional del Algoritmo Genético	496
8.3.2	Algoritmo SMAGA	503
8.3.2.1	Papel del Sistema Multiagente en el Algoritmo.....	504
8.3.2.2	Funcionamiento General del Algoritmo.....	507
8.3.2.2.1.	Representación de los Individuos	509
8.3.2.2.2.	Identificación de los Agentes y el Sistema Multiagente	512
8.3.2.2.3.	Generación de la Población	514
8.3.2.2.4.	Procedimiento de Selección.....	520
8.3.2.2.5.	Evaluación de la Soluciones	524
8.3.2.2.6.	Operadores de Cruce	525
8.3.2.2.7.	Selección de los Operadores de Cruce.....	531
8.3.2.2.8.	Aplicación de los Operadores de Cruce	534
8.3.2.2.9.	El Operador de Mutación.....	541

8.3.2.2.10.	Aprendizaje Genético.....	545
8.3.2.2.11.	Operador de Reinicio de la Población.....	551
8.3.2.3	Implementación del Algoritmo	553
8.4	Desarrollo del Diseño Experimental.....	553
8.4.1	Parametrización	554
8.4.2	Plan de Experimentación y Condición de Finalización	557
8.4.3	Procedimiento de Ejecución los Experimentos.....	559
8.4.4	Análisis de los resultados	561
8.5	Evaluación Comparativa.....	571
8.5.1	Análisis de los resultados para el Caso P13.....	574
8.5.2	Análisis de los resultados para el Caso P3.....	578
8.6	Conclusiones	582
8.7	Bibliografía	584

Índice de figuras

Figura 8-1. Diagrama de Funcional del Proceso de Negocio de “Programación Predictiva” (elaboración propia).	474
Figura 8-2. Esquema de una Algoritmo Genético (elaboración propia).....	497
Figura 8-3. Primer paso del operador GPX (elaboración propia).	498
Figura 8-4. Segundo paso del operador GPX (elaboración propia).....	499
Figura 8-5. Ejemplo de SWAP Mutation.....	500
Figura 8-6. Ejemplo de SHIF Mutattion (elaboración propia).	500
Figura 8-7. Esquema general de los procesos correspondientes al algoritmo SMAGA (basado en (Zhong, W. C. et al. 2004)).	508
Figura 8-8. Ampliación de las actividades de los operadores correspondientes al algoritmo SMAGA (elaboración propia).....	509
Figura 8-9. Representación genética ordinal para un caso de 20 trabajos (elaboración propia).	510
Figura 8-10. Diagrama Gantt para una la representación genética del individuo de la Figura 8-9 (elaboración propia).	511
Figura 8-11. Representación del Sistema Multiagente (elaboración propia).	513
Figura 8-12. Modelo de Agentes-Solución en rejilla o matriz base (elaboración propia).....	514
Figura 8-13. Relación entre Matriz Base y Matriz de Aprendizaje (elaboración propia).	515
Figura 8-14. Primer paso del operador SB2OX (elaboración propia).	527
Figura 8-15. Segundo paso del operador SB2OX (elaboración propia).....	527
Figura 8-16. Tercer paso del operador SB2OX (elaboración propia).	528
Figura 8-17. Competidores locales en la matriz base para el Agente-Solución (2,2) (elaboración propia).	529
Figura 8-18. Primer paso del operador NCO (elaboración propia).	530
Figura 8-19. Segundo paso del operador NCO (elaboración propia).	531
Figura 8-20. Diagrama de flujo para sustitución de la secuencia actual por una nueva después de un cruce.	537
Figura 8-21. Ejemplo de población con matriz base de dimensión 4x4 (elaboración propia)..	539
Figura 8-22. Despliegue de la lista de posiciones ocupadas para la aplicación del operador de cruce (elaboración propia).	540
Figura 8-23. Mutación por intercambio en proximidad (elaboración propia).....	542
Figura 8-24. Esquema general de los procesos correspondientes al procedimiento de aprendizaje genético (basado en (Zhong, W. C. et al. 2004)).....	545

Figura 8-25. Desplazamiento a derechas de la secuencia original para la obtención del elemento de la fila 2 (elaboración propia).	547
Figura 8-26. Desplazamiento a izquierdas de la secuencia original para la obtención del elemento de la fila 3 (elaboración propia).	548
Figura 8-27. Intercambio de posiciones pares e impares de la secuencia original para la obtención del elemento de la fila 4 (elaboración propia).	548
Figura 8-28. Intercambio de dos posiciones de la secuencia original para la obtención del elemento de la fila 1, en las columnas 2, 3 y 4 (elaboración propia).	548
Figura 8-29. Gráfico de probabilidad normal (Y_i , F_i) para la variable IPSOVEPT del conjunto SSD10_P13_20 en relación a las frecuencias relativas acumuladas.	561
Figura 8-30. Residuos estudentizados frente a la variable IPSOVEPT del conjunto experimental SSD10_P13_20.....	562
Figura 8-31. Residuos estudentizados frente a los niveles del factor GsM.	563
Figura 8-32. Residuos estudentizados frente a los niveles del factor FdM.....	563
Figura 8-33. Residuos estudentizados frente a los niveles del factor FmG.	564
Figura 8-34. Residuos estudentizados frente al orden de ejecución.	565
Figura 8-35. Gráfico de efectos principales de los factores del conjunto experimental SSD10_P13_20.....	566
Figura 8-36. Gráfico de interacción entre los factores del conjunto experimental SSD10_P13_20.....	566
Figura 8-37. Gráfico del efecto principal del factor RM de la configuración P3.	570
Figura 8-38. Diagrama Gantt de un Programa Productivo para un taller de flujo híbrido del tipo P13 (elaboración propia).	575
Figura 8-39. Representación gráfica de los valores IPSOVEPT para los conjuntos experimentales P13.....	576
Figura 8-40. Representación gráfica de los valores IPSOVEPT de los algoritmos GAH, SMAGA-1 y SMAGA-2 para los conjuntos experimentales P13.....	578
Figura 8-41. Diagrama Gantt de un Programa Productivo para un taller de flujo híbrido del tipo P3 (elaboración propia).	579
Figura 8-42. Representación gráfica de los valores IPSOVEPT para los conjuntos experimentales P3.....	580
Figura 8-43. Representación gráfica de los valores IPSOVEPT de los algoritmos GAH, SMAGA-1 y SMAGA-2 para los conjuntos experimentales P3.....	581

Índice de ilustraciones

Ilustración 8-1. Pseudocódigo correspondiente al algoritmo voraz para el Taller de Flujo (elaboración propia). 516

Ilustración 8-2. Pseudocódigo correspondiente a la aplicación del algoritmo original voraz en un Agente-Solución (elaboración propia)..... 519

Ilustración 8-3. Pseudocódigo correspondiente a la aplicación del algoritmo voraz en el Agente-Gestor (elaboración propia). 520

Ilustración 8-4. Pseudocódigo correspondiente al procedimiento básico de selección y cruce (elaboración propia). 521

Ilustración 8-5. Pseudocódigo correspondiente al procedimiento modificado de selección y cruce (elaboración propia). 523

Ilustración 8-6. Pseudocódigo del procedimiento de selección y cruce para el operador NCO (elaboración propia). 530

Ilustración 8-7. Pseudocódigo correspondiente a la selección del operador de cruce (elaboración propia). 533

Ilustración 8-8. Pseudocódigo correspondiente al método de relevo generacional de los operadores de cruce (elaboración propia). 538

Ilustración 8-9. Pseudocódigo correspondiente al procedimiento para la aplicación del operador de mutación (elaboración propia). 543

Ilustración 8-10. Pseudocódigo correspondiente al método de relevo generacional del operador de mutación (elaboración propia). 545

Ilustración 8-11. Pseudocódigo para la aplicación del operador de reinicio de la población (elaboración propia). 552

Índice de tablas

Tabla 8-1. Relación entre diferentes tipos de análisis de residuos y su aplicación para verificar características (elaboración propia). 479

Tabla 8-2. Resumen de la estructura de ejemplares de (Taillard, E. 1993). 482

Tabla 8-3. Ejemplo de ejemplar Taillard 019, generada por (Taillard, E. 1993). 483

Tabla 8-4. Ejemplo de ejemplar del grupo SDST10 para Taillard 003, generada por (Vallada, E. et al. 2003). 487

Tabla 8-5. Resumen de la estructura de ejemplares de (Vallada, E. et al. 2003). 488

Tabla 8-6. Ejemplo de ejemplar del grupo SSD10_P13 para Taillard 009, generada por (Vallada, E. et al. 2003). 492

Tabla 8-7. Resumen de los conjunto experimentales empleadas en el diseño experimental (elaboración propia). 493

Tabla 8-8. Distribución de Ejemplares en grupos para dos conjuntos experimentales (elaboración propia). 494

Tabla 8-9. Relación entre las ventajas generales y las características deseadas (elaboración propia). 506

Tabla 8-10. Conjunto de parámetros fijos (elaboración propia). 554

Tabla 8-11. Conjunto de parámetros ajustables (elaboración propia). 555

Tabla 8-12. Estructura de los factores ajustables para los algoritmo SMAGA-1 y SMAGA-2 (elaboración propia). 556

Tabla 8-13. Niveles asignados a los parámetros ajustables para los algoritmo SMAGA-1 y SMAGA-2 (elaboración propia). 558

Tabla 8-14. Matriz de correlaciones para el conjunto experimental SSD10_P13_20. 564

Tabla 8-15. ANOVA correspondiente al conjunto experimental SSD10_P13_20. 565

Tabla 8-16. Valores seleccionados para los parámetros de los diferentes conjuntos experimentales del algoritmo SMAGA-1 (elaboración propia). 567

Tabla 8-17. Resumen de los valores asignados a los parámetros del algoritmo SMAGA-2 para cada conjunto experimental (elaboración propia). 569

Tabla 8-18. ANOVA correspondiente IPSPPVEPT de la configuración P13. 569

Tabla 8-19. ANOVA correspondiente IPSPPVEPT de la configuración P3. 570

Tabla 8-20. Valor de IPSOVEPT para cada uno de los conjuntos experimentales de tipo P13 para los diez algoritmos seleccionados (elaboración propia). 573

Tabla 8-21. Valor de IPSOVEPT para cada uno de los conjuntos experimentales de tipo P3 para los diez algoritmos seleccionados (elaboración propia). 573

Tabla 8-22. Diferencias entre el mejor IPSOVEPT y el de cada algoritmo en problemas P13 (elaboración propia). 577

Tabla 8-23. Diferencias entre el mejor IPSOVEPT y el de cada algoritmo en problemas P3
(elaboración propia). 581

8.1 Introducción

Uno de los objetivos de esta tesis es el análisis de un conjunto de algoritmos para la **Programación Predictiva de la Producción**, que permita la selección de uno o varios de ellos, en base a su eficiencia en el ámbito del problema planteado en el marco de la Industria Cerámica. En este capítulo se va a avanzar en esta dirección, ya que se realizará el diseño e implementación de nuevas propuestas para Programas Predictivos de la Producción aplicados al caso de un Taller de Flujo Híbrido con Tiempos de Cambio de Partida Dependiente de la Secuencia.

Las nuevas propuestas que se presentan en este capítulo son variaciones que se han desarrollado a partir de un **Algoritmo Genético** que, diseñado e implementado basándose en el paradigma de los **Sistemas Multiagente**, se espera que proporcione soluciones alternativas a las actuales para las situaciones reales que se pueden encontrar en la Industria Cerámica. Dicho algoritmo se ha denominado **SMAGA** (Scheduling Multi-Agent Genetic Algorithm). La adecuación de los Algoritmos Genéticos como procedimiento de resolución, la aplicación de Sistemas Multiagente y el análisis de los resultados de los algoritmos que serán integrados en la Plataforma **IPSU-MAS** (Integrated Platform for Scheduling under Uncertainty based on **Multi-Agent Systems**), diseñada en el capítulo anterior, son en líneas generales los aspectos que serán abordados en este bloque.

Este capítulo forma parte de la fase 3 de la metodología propuesta en el capítulo 6, es decir, del Diseño de la Solución. Y en el se pueden destacar 3 etapas que resumen los objetivos del capítulo. En primer lugar, se dedica un gran esfuerzo a **diseñar el algoritmo SMAGA** a partir de la estructura modificada de un Algoritmo Genético, sobre la cual se describe función a función, como se combinan los elementos típicos de este tipo de metaheurística con los conceptos del paradigma de Agentes. Así por un lado, conceptos como los de Alelo o Gen se trabajan al mismo nivel que el de Agente para alcanzar la definición de los individuos de la población. Y por otro lado, funciones como la de los operadores de cruce o mutación se hacen compatibles con el requisito de coordinar a los individuos del Sistema Multiagente. Esta primera tarea finaliza con la implementación de dos versiones del algoritmo que son: **SMAGA-1 y SMAGA-2**. En segundo lugar, se realiza un importante trabajo de **ajuste de parámetros** para intentar buscar el máximo rendimiento del mismo. Esta tarea se fundamenta en el **Diseño de Experimentos** con un importante soporte en conceptos y herramientas estadísticas. Y por último, en tercer lugar **se realiza un análisis comparativo del comportamiento de los algoritmos propuestos** con otros métodos, para poder sí valorar la calidad de los mismos. Esta

tarea se basa en el Diseño de Experimentos, y la selección de un juego de datos y de unos algoritmos de referencia adecuados. El empleo de un **juego de datos público y estándar** permite un mayor grado de contraste de la propuesta, al mismo tiempo que evita la implementación y testeo de los algoritmos de referencia, facilitando el uso de datos ya publicados.

El trabajo desarrollado en el presente capítulo se organiza de la siguiente forma: En primer lugar se expondrá el ámbito de aplicación del algoritmo SMAGA y la metodología con la que se desarrollará la propuesta. En segundo lugar se realizará el diseño del algoritmo. En tercer lugar, una vez implementada la propuesta, se procederá al juste paramétrico del algoritmo en cuya fase se diferenciará entre SMAGA-1 y SMAGA-2. En cuarto lugar, se realizará un análisis comparativo entre los algoritmos SMAGA-1 y SMAGA-2 parametrizados con otras propuestas relevantes existentes. Y finalmente, se comentarán brevemente las conclusiones del capítulo.

8.2 Ámbito de Aplicación y Metodología

En este apartado se quiere dar respuesta al “que” y al “como”, es decir, cual es el problema concreto que se desea abordar, y como se va actuar para garantizar la validez de los resultados alcanzados.

8.2.1 Identificación del Problema

Según se comentó en el capítulo 5, “Descripción y Modelado Matemático del Problema”, en esta tesis se propone realizar una Plataforma Software, que permita afrontar el problema de la Programación de la producción en un entorno de incertidumbre en coordinación con el proceso de Planificación de la Producción. Así mismo, se identificaron los requerimientos de la misma mediante la aplicación de la propuesta CIMOSA, en base a los cuales se realizó el diseño de la misma en el capítulo 7. Dentro del amplio conjunto de responsabilidades que se la atribuyeron a dicha Plataforma en el capítulo 7, se destacan dos grandes “Procesos de Negocio (BP)”ⁱ: El “BP-Programación Predictiva”, y el “BP-Programación Reactiva”.

ⁱ El autor sigue la nomenclatura propuesta por CIMOSA. En el capítulo 5 de esta tesis se dispone del análisis de requerimientos de la Plataforma Software en relación con el trabajo de Programación de la Producción.

Uno de los objetivos de este capítulo es el desarrollo de un algoritmo, con diferentes variantes, que pueda formar parte del núcleo del Proceso de Negocio para la Programación Predictiva. El algoritmo propuesto no constituye por sí sólo dicho proceso, ya que el mismo está formado por un conjunto, de lo que CIMOSA denomina, Actividades Empresariales. Estas son:

- EA-1: Revisión y carga de datos.
- EA-2: Cálculo del Programa Predictivo.
- EA-3: Ajustes.
- EA-4: Validación / Rechazo.

En la Figura 8-1 se muestra el diagrama funcional del proceso completo, dentro del cual se describen las responsabilidades de cada actividad empresarial. El algoritmo que se propone debe asumir las funciones atribuidas a la actividad empresarial EA-2, en la que se sitúa la función asociada al cálculo del Programa Predictivo. Las acciones comprendidas en la actividad empresarial AE-2 están destinadas al ajuste de parámetros del algoritmo que se utilice, la ejecución del propio algoritmo y el registro y transferencia de los resultados obtenidos a la actividad empresarial AE-3.

La funcionalidad asignada a la actividad empresarial AE-2 se debe implementar para dar lugar a una aplicación software. Una de las características que se ha considerado necesaria es que el software resultante permita el reajuste de los parámetros después de conocer los resultados del cálculo en caso de no obtener el resultado deseado. Además se incorporará una pequeña utilidad que permita visualizar los resultados obtenidos mediante un diagrama Gantt.

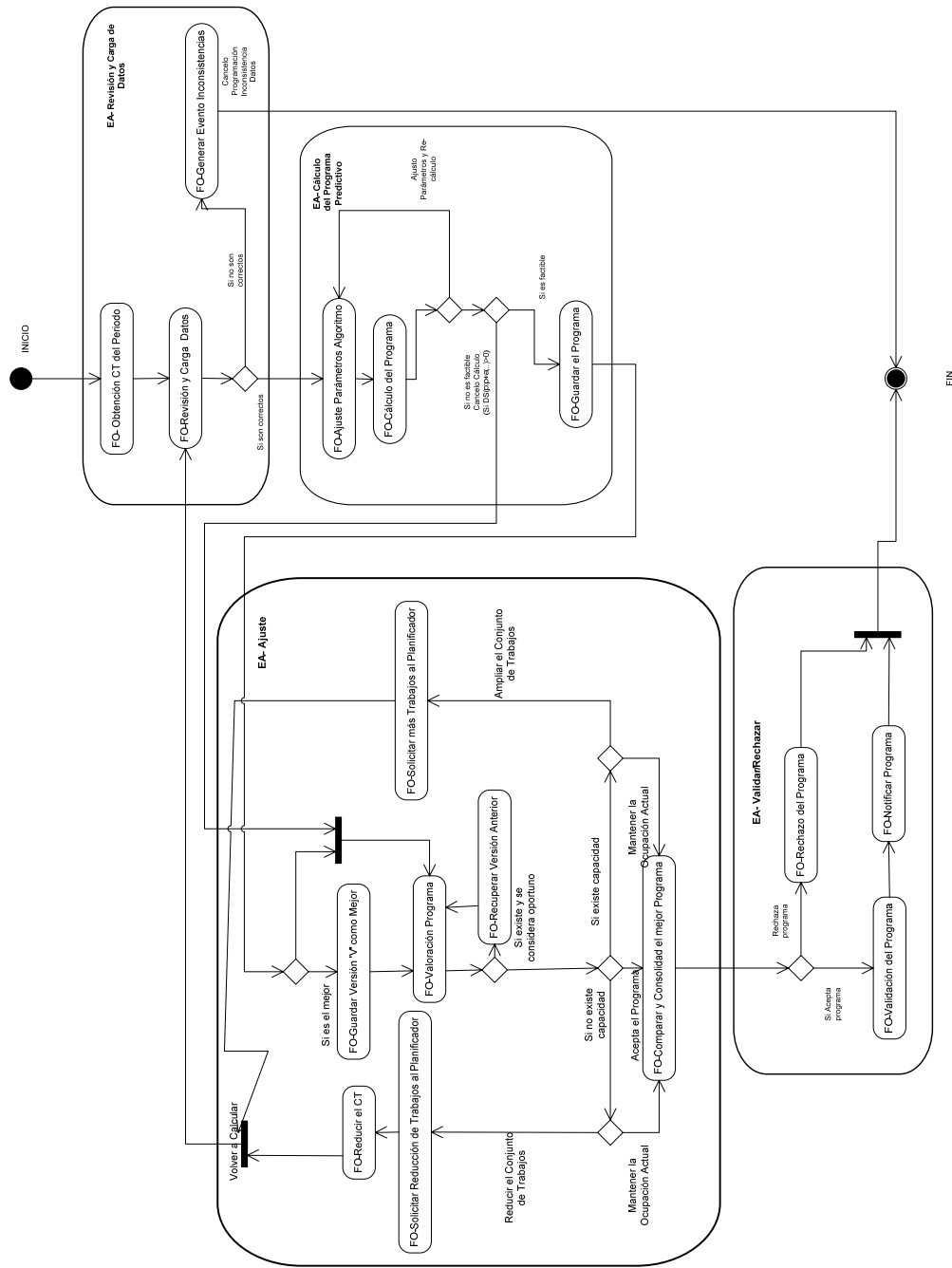


Figura 8-1. Diagrama de Funcional del Proceso de Negocio de "Programación Predictiva" (elaboración propia).

Se desea resaltar que lo que en capítulos anteriores se denominó, "Gestión del Programa Productivo", es la función que asumirán el conjunto de procesos que forman la Plataforma, y no específicamente la actividad EA-2, que contribuye a una función muy concreta y puntual. La Gestión del Programa Productivo es un concepto que va más allá del cálculo de un Programa Productivo, y que surge como consecuencia de la decisión de incluir dentro del problema a

tratar los efectos de la Incertidumbre. En base a la definición de requerimientos empleada debe quedar claro que la actividad EA-2 resolverá un problema de Programación Predictiva, es decir estática y sin Incertidumbre. Cuando debido a la Incertidumbre se modifiquen las condiciones del Programa Productivo en curso, y sea necesario adaptarlo a las nuevas circunstancias, se activarán otras actividades empresariales incluidas en el “BP-Programación Reactiva” que abordarán y resolverán esa parte del problema.

El Problema Predictivo que se aborda en la actividad EA-2, y por lo tanto, el algoritmo que se propone en este capítulo, fue definido en un lenguaje natural y matemático en el capítulo 5. En concreto en el apartado 5.2.2.2 se comentan las hipótesis que se plantean para el modelo Predictivo determinista, y en el apartado 5.4.1 y 5.4.2 se desarrolla el modelo matemático que define sin ambigüedades el problema se pretende resolver en este capítulo.

Como recordatorio cabe indicar que basándose en la notación establecida por (Conway, R. W., Maxwell, W. L., & Miller, L. W. 1967) para identificar los problemas de secuenciación, el problema Predictivo determinista que se pretende resolver podría ser representado como $n/r/F/k_1, \dots, k_r$ -parallel/setup/Cmax. Es decir se trata de un Taller de Flujo Híbrido con r etapas, y k_1 máquina en paralelo en la primera etapa, k_2 en la segunda y k_r en la última, con Tiempos de Setup Dependientes de la Secuencia, cuya función objetivo es minimizar el Cmax o makespan.

8.2.2 Identificación de la Propuesta

Para poder obtener un Programa Productivo Predictivo se propone el diseño e implementación de un nuevo Algoritmo Genético (AG). La diferencia fundamental entre la propuesta que realiza en esta tesis y los AG tradicionales, **es que en este caso, se realiza una hibridación del algoritmo combinando la estructura básica de un AG con las características de los Sistemas Multiagente (SMA).**

La propuesta presente busca disponer de un algoritmo que base su funcionamiento, y en gran medida su eficiencia, en la explotación de ciertas características propias de los SMA. **El objetivo es que la hibridación se produzca de forma que los Agentes tomen el rol de “seres vivos” y por lo tanto, de individuos de la población del AG.** Este tipo de hibridación (AG+MAS) es casi inexistente en la literatura, y se ha considerado que puede ofrecer muy buenos resultados.

Este algoritmo será una pieza más dentro de la Plataforma **IPSU-MAS**, y por lo tanto se desea que los resultados que con él se obtengan sean competitivos. Por este motivo, una vez

definido el algoritmo, se va a realizar un calibrado de los parámetros del mismo, y posteriormente un análisis comparativo de los resultados que permitan contrastar su calidad.

8.2.3 Metodología para el Desarrollo y Evaluación de la Propuesta

Debido a las características de la propuesta que se realiza, los aspectos metodológicos han sido tenidos muy en cuenta en el desarrollo de la misma.

En primer lugar se desea indicar que dado que se va a desarrollar un algoritmo basado en un Sistema Multiagente se ha considerado la posibilidad de utilizar la metodología INGENIAS desarrollada por (Pavon, J. & Gomez-Sanz, J. 2003) para obtener un software basado en Agentes que permita abordar el problema de Programación Predictiva de la Producción en una Empresa Cerámica en los términos planteados. Sin embargo, cuando se terminó de definir las líneas maestras del algoritmo se percibió una gran distancia entre la metodología INGENIAS y el trabajo que se debería realizar para plasmar las ideas en un software. Efectivamente, INGENIAS es una propuesta muy acertada para el diseño y posterior desarrollo aplicaciones de cierta envergadura. Sin embargo, aunque la aportación de los Agentes en el algoritmo que se propone es muy relevante, la complejidad del diseño e implementación es escasa. En el algoritmo que se propone el número Agentes y roles de los mismos es sencillo en cuanto a su estructura. Por este motivo se ha optado por no emplear una metodología específica para el desarrollo de Agentes y seguir los pasos habituales en el diseño de un Algoritmo Genético.

En segundo lugar habría que señalar que el proceso de ajuste de los parámetros del algoritmo SMAGA será complejo. Además del método de experimental seguido para el ajuste, se detallará en apartados diferenciados la descripción del juego de datos y la definición de la variable de respuesta.

La mayoría de los procedimientos metaheurísticos que se conocen incorporan un factor de aleatoriedad en su ejecución, lo que hace difícil poder establecer una relación directa entre el valor de los parámetros utilizados y los resultados obtenidos sin realizar un estudio estadístico adecuado. El Diseño de Experimentos es una metodología basada en herramientas matemáticas y estadísticas cuyo objetivo es ayudar al experimentador a seleccionar la estrategia experimental óptima que permita alcanzar, y evaluar los resultados obtenidos, con el mínimo coste y la máxima fiabilidad en las conclusiones.

La aplicación del diseño de experimentos requiere considerar las etapas que se comentarán a continuación:

1. Comprender el problema y definir claramente el objetivo.
2. Identificar los factores que potencialmente podrían influir en la función objetivo, y los valores que éstos pueden tomar. Entre estos valores se buscará la información necesaria.
3. Establecer una estrategia experimental, llamada Plan de Experimentación.
4. Efectuar los experimentos con los valores de los factores decididos en el punto tres para obtener los valores de las respuestas estudiadas.
5. Responder las preguntas planteadas, sea directamente, sea utilizando un modelo matemático. Si es necesario, volver a la etapa uno.

En relación con el primer paso se puede decir que el problema que se desea abordar en la experimentación es el de identificar el conjunto de parámetros con los que el algoritmo SMAGA pueda obtener los mejores resultados promedio posible para el problema especificado en el apartado 8.2.1. Dado que los resultados no sólo están condicionados por el tipo de algoritmo empleado para la Programación de la Producción, sino también por el tipo concreto de configuración de taller sobre el que se trabaje (básicamente del número de etapas y del número de máquinas por etapa), el número de trabajos que se procese, y la eficiencia de los recursos (básicamente de los tiempos de proceso y de cambio de partida), es necesario tener en cuenta la combinación del tipo de algoritmo junto con el tipo de problema para el cual se realizan los experimentos. En este aspecto, el objetivo es establecer tantos conjuntos de parámetros con combinaciones existan. Dicho conjunto serán especificados en el apartado donde se explicita el juego de datos empleado.

Una vez se discriminen los parámetros con los que el algoritmo SMAGA alcanza los mejores resultados para cada configuración de problema, se procederá a realizar una comparación entre dichos resultados y los resultados publicados que se han obtenidos con otros algoritmos, para los mismos casos. Este análisis comparativo es un paso que se realizará una vez concluido el ajuste paramétrico para el cual se realiza el diseño experimental.

Por lo que se refiere a los pasos desde el 2 al 5 se cree más conveniente desarrollar su contenido dentro del apartado 8.4, ya que la descripción de los mismos implica un amplio conocimiento de la propuesta que aún no ha sido desarrollada.

En cualquier caso, será necesario disponer de una herramienta que permita establecer conclusiones claras y fiables. En este caso se ha considerado que existe una variable explicada (dependiente o endógena) cuyo valor es función de diversas variables explicativas

(independientes o exógenas). Donde además la variable explicada es métrica y las variables explicativas son no métricas. En este contexto, habitual por otra parte en este tipo de ajustes algorítmicos, el uso de un test paramétrico es fundamental. Uno de los test paramétricos más potentes, y ajustados a las características de este caso, es el del análisis de la varianza o ANOVA.

Un punto importante, que se debe tenerse en cuenta, cuando se trabaja con un test ANOVA es que este debe cumplir varias condiciones, cuyo incumplimiento invalida cualquier conclusión que se pueda obtener. Estas condiciones son:

- **Normalidad.** Todas las variables que intervienen en un método de análisis multivariable deben ser normales.
- **Homocedasticidad.** Es una propiedad fundamental del modelo de regresión lineal general. Se dice que existe homocedasticidad cuando la varianza de los errores estocásticos de la regresión son los mismos para cada observación. Lo que significaría que habría una distribución de probabilidad de idéntica amplitud para cada variable aleatoria. Esta cualidad es necesaria, según el Teorema de Gauss-Markov, para que en un modelo los coeficientes estimados sean los mejores o eficientes, lineales e insesgados.
- **Linealidad.** Suele ser un supuesto implícito en todas las técnicas multivariante basadas en medidas de correlación. Como los efectos no lineales nunca están representados en el valor de la correlación, su presencia tendría efectos nocivos en el modelo multivariante.
- **Ausencia de autocorrelación.** La autocorrelación de una serie temporal discreta de un proceso es simplemente la correlación de dicho proceso con una versión desplazada en el tiempo de la propia serie temporal.
- **Ausencia de multicolinealidad.** La multicolinealidad es una situación en la que se presenta una fuerte correlación entre variables explicativas del modelo.

Es por lo tanto requisito imprescindible comprobar el cumplimiento de las distintas hipótesis, para lo cual en esta tesis se aplicarán fundamentalmente métodos gráficos.

Para el análisis de la normalidad se utilizará el “gráfico de probabilidad normal (Y_i, F_i)”, que presenta en el eje de abscisas los valores de la variable explicada o endógena (Y_i), y en el ordenadas las frecuencias relativas acumuladas de dichos valores (F_i). La normalidad de los

datos será perfecta cuando el gráfico de puntos resulte ser una línea recta situada sobre la diagonal del primer cuadrante.

Para el análisis de la homocedasticidad se utilizarán los medios gráficos. El medio gráfico empleado por excelencia es el “gráfico de residuos (X_t, \bar{u}_t) y (\hat{Y}_t, \bar{u}_t)”, que permite disponer los residuos, que pueden estar estudentizadosⁱⁱ (\bar{u}_t) o no, en el eje de ordenadas, frente a las variables exógenas (X_t), en el eje de abscisas, que deben presentar estructura aleatoria libre de tendencia. El gráfico de los residuos contra cada variable exógena permite detectar como la variable culpable del no cumplimiento de la condición es aquellas cuyo gráfico se separa más de la aleatoriedad. De idéntica forma se empleará el mismo gráfico con la variable endógena, que nuevamente puede estar estudentizada (\hat{Y}_t) o no.

El análisis de la linealidad se detectará con el “gráfico de los residuos (X_t, \bar{u}_t) y (\hat{Y}_t, \bar{u}_t)”, ya utilizado para la verificación de la condición de homocedasticidad.

La multicolinealidad se detectará con el análisis de la matriz de correlaciones. Valores altos en esta matriz son síntoma de posible dependencia entre variables. El análisis de la autocorrelación se realizará con el estudio del “gráfico de residuos (t, \bar{u}_t)”, donde los residuos, que pueden estar estudentizados (\bar{u}_t) o no, se sitúan en el eje de ordenadas contra el tiempo (t) o el orden de ejecución de la pruebas en el eje de abscisas. Si se detecta una tendencia creciente o decreciente puede existir autocorrelación.

Como se ha visto el análisis de los residuos es fundamental para comprobar el cumplimiento de las condiciones de homocedasticidad, linealidad, y autocorrelación. A continuación se ofrece una tabla resumen de los diferentes análisis residuales en función de la propiedad que se desea verificar.

Propiedad	(t, \bar{u}_t)	(X_t, \bar{u}_t)	(\hat{Y}_t, \bar{u}_t)
Homocedasticidad		✓	✓
Linealidad		✓	✓
Autocorrelación	✓		

Tabla 8-1. Relación entre diferentes tipos de análisis de residuos y su aplicación para verificar características (elaboración propia).

ⁱⁱ Consiste en el cálculo de la siguiente expresión para cada residuo: (residuo-media de los residuos)/desviación estándar de los residuos.

Una vez definido el esquema básico por el cual se guiará el desarrollo experimental que se va a desarrollar en este capítulo se explicarán con detalle los aspectos relacionados con el juego de datos empleado y la medida (variable explicada o endógena) que se empleará para la evaluación del algoritmo.

8.2.3.1 El Juego de Datos

En este trabajo se ha considerado fundamental la utilización de un juego de datos estándar. Su función es la ayudar a contrastar la calidad de los resultados que produce el algoritmo **SMAGA** en un conjunto de situaciones representativas. Estos datos deben considerar diversas configuraciones del sistema (nº etapas, nº máquinas por etapa), del número de trabajos, y de la eficiencia de los recursos (tiempos de proceso, y de cambio de partida), y deben ser conocidos y utilizados por la comunidad científica. Sobre dicho datos el algoritmo **SMAGA** debe calcular un Programa Productivo Predictivo intentando minimizar el makespan. La existencia de un juego estándar presenta dos ventajas fundamentales en el proceso de experimentación científica:

- Los resultados obtenidos pueden ser contrastados por la comunidad científica con mayores garantías, ya que se pueden comparar los resultados obtenidos para múltiples escenarios conocidos, y sobre los cuales hay un gran número de investigadores trabajando. Esto a su vez, facilita el proceso de publicación de resultados.
- Simplifica el proceso de experimentación. Ya que una forma de contrastar los resultados obtenidos por un algoritmo, especialmente metaheurístico, es la comparación con otros procedimientos. En este caso, no es necesario ejecutar otros algoritmos, ya que los resultados son conocidos, para el mismo juego de datos con el que se ejecuta la propuesta que se desea testear. Al no tener que ejecutar dichos algoritmos tampoco es necesario programarlos, con el consiguiente ahorro de tiempo, y reducción de fuentes de error.

En múltiples ocasiones los experimentadores no han podido acceder a un juego de datos conocido, ya que las características del problema que abordan eran lo suficientemente especiales como para que no existan propuestas que hayan favorecido la generación de un banco de pruebas estable y contrastado. En este caso, dicha circunstancia se ha podido solventar. Existen varios bancos de pruebas con datos disponibles para la experimentación, donde los diferentes avances alcanzados son dados a conocer por sus autores. Sin embargo, en el caso del Taller de Flujo se deben de diferenciar dos escenarios, el problema del Taller de Flujo de Permutación, y el resto de configuraciones Talleres de Flujo. Para el Taller de Flujo de

Permutación, un problema que ha concentrado más atención que el resto de los mencionados, existen diversas opciones para poder elegir un juego de datos estándar ((Watson, J. P., Barbulescu, L., Whitley, L. D., & Howe, A. E. 2002),(Taillard, E. 1993) , (Brucker, P., Heitmann, S., & Hurink, J. 2003)). Fuera de esta configuración se no se han encontrado bancos de prueba de uso “reconocido”. Por este motivo, se ha optado por seguir los pasos de la propuesta empleada por (Ruiz, R. 2003). Se trata de utilizar un banco de pruebas específico para el caso que se desea analizar en este trabajo; El Taller de Flujo Híbrido con Tiempos de Cambio de Partida Dependientes de la Secuencia, generados a partir de un banco de pruebas contrastado por la comunidad científica.

En (Ruiz, R. 2003) se encontró con la misma dificultad, encontrar un juego de datos para configuraciones específicas del Taller de Flujo, por lo que optó por emplear una serie de benchmarks que previamente había publicado junto otras compañeras en (Vallada, E., Ruiz, R., & Maroto, C. 2003). El banco de datos fue publicado originalmente en (Vallada, E. et al. 2003), y se trata de una adaptación del juego de datos establecido en (Taillard, E. 1993). Dicho juego de datos, adaptado para el Taller de Flujo y de Flujo Híbrido ambos con Tiempos de Cambio de Partida Dependiente de la Secuencia, y ha sido posteriormente puesto a disposición de la comunidad científica para que ésta trabaje con ellos en sus respectivas propuestas y faciliten sus nuevos resultados que mejoren a los actualmente establecidos. La selección de un conjunto de problemas adaptados desde el conjunto de ejemplares de Taillard no sólo se realiza en (Ruiz, R. 2003), sino también en otros casos, como en (Brucker, P. et al. 2003). Por estos motivos se han considerado que los ejemplares de (Vallada, E. et al. 2003) constituyen un banco de pruebas ajustado a las necesidades de este trabajo, y con incipiente uso científico. A continuación se detalla el método de generación, la estructura y disponibilidad de los mismos.

Dado que los datos que se emplean para el contraste del algoritmo **SMAGA** toman como base el conjunto de ejemplares de Taillard se va a comentar brevemente dicha propuesta. En (Taillard, E. 1993) se estableció un banco de pruebas para el Taller de Flujo de Permutación compuesto de un total de 120 problemas de diversos tamaños. Cada problema está compuesto por un conjunto de trabajos a los cuales se le asignó tiempos de proceso extraídos a partir de una distribución aleatoria uniforme $U[0..99]$. Taillard propuso una clasificación formada por doce tipos de configuración. Cada grupo se caracteriza por considerar el mismo número de trabajos (n) y de máquinas (m). A su vez, dentro de cada grupo se generaron diez ejemplares para la misma configuración de máquinas y trabajos pero con diferente asignación de tiempos de proceso a cada uno. Un resumen de la estructura del banco de pruebas se puede ver en la Tabla 8-2. Los ejemplares son accesibles desde la propia página web del Taillard en <http://mistic.heig-vd.ch/taillard/problemes.dir/problemes.html>.

Grupos de Ejemplares	Nº de Máquinas	Nº de Trabajos	Nº de Instancias
Grupo_20x5	5	20	10
Grupo_20x10	10	20	10
Grupo_20x20	20	20	10
Grupo_50x5	5	50	10
Grupo_50x10	10	50	10
Grupo_50x20	20	50	10
Grupo_100x5	5	100	10
Grupo_100x10	10	100	10
Grupo_100x20	20	100	10
Grupo_200x10	10	200	10
Grupo_200x20	20	200	10
Grupo_500x20	20	500	10

Tabla 8-2. Resumen de la estructura de ejemplares de (Taillard, E. 1993).

Un ejemplo para un ejemplar con veinte trabajos y diez máquinas es el que se refleja en la Tabla 8-3. En la representación que se ofrece en dicha tabla existen diez grupos de dos columnas. En cada grupo la primera columna hace referencia al identificador de la máquina o etapa (0..9) y el segundo al tiempo de proceso. A su vez existen veinte filas. Cada fila representa la información de un trabajo de los veinte existentes.

Son muchos los autores que a lo largo de casi dos décadas han dedicado parte de su esfuerzo a resolver estos problemas. Existen algunos ejemplares resueltos de manera óptima, y otros problemas que aún están abiertos. En el anexo II, la tabla II.1 presenta un resumen de los ejemplares propuestos con los resultados obtenidos en cada una. Los problemas resueltos o cerrados presentan el mismo valor para la cota inferior (CI) y para la cota superior (CS), mientras que los no resueltos o abiertos disponen de ambas cotas diferenciadas. En dicha tabla se puede ver como aun existen 27 problemas abiertos, aunque en todos los casos la diferencia entre la cota superior y la inferior es muy pequeña.

Los problemas generados por Taillard presentan una ventaja adicional y es la existencia de una amplia gama de datos en función del número de trabajos y de máquinas. Esto permite testear cualquier algoritmo con problemas próximos a muchas realidades industriales.

0	39	1	9	2	72	3	93	4	8	5	75	6	16	7	98	8	65	9	21
0	16	1	19	2	13	3	77	4	83	5	70	6	98	7	49	8	93	9	98
0	84	1	71	2	49	3	20	4	29	5	16	6	85	7	46	8	14	9	61
0	51	1	7	2	37	3	1	4	87	5	72	6	63	7	66	8	73	9	78
0	19	1	23	2	89	3	60	4	77	5	85	6	10	7	63	8	97	9	6

0 57 1 7 2 37 3 36 4 15 5 94 6 14 7 88 8 17 9 34
0 33 1 65 2 58 3 85 4 26 5 49 6 82 7 41 8 91 9 53
0 1 1 56 2 4 3 18 4 97 5 40 6 39 7 13 8 71 9 63
0 23 1 37 2 13 3 97 4 44 5 64 6 23 7 43 8 59 9 3
0 39 1 87 2 48 3 24 4 85 5 98 6 29 7 54 8 45 9 50
0 27 1 43 2 4 3 60 4 22 5 46 6 30 7 69 8 45 9 29
0 37 1 4 2 43 3 28 4 17 5 18 6 99 7 97 8 21 9 29
0 37 1 92 2 18 3 94 4 47 5 47 6 34 7 10 8 98 9 20
0 1 1 75 2 9 3 34 4 30 5 2 6 16 7 51 8 64 9 68
0 82 1 86 2 81 3 61 4 10 5 77 6 34 7 95 8 50 9 84
0 24 1 26 2 66 3 10 4 84 5 74 6 28 7 51 8 74 9 29
0 28 1 17 2 82 3 4 4 13 5 77 6 69 7 92 8 77 9 75
0 98 1 7 2 21 3 83 4 26 5 32 6 85 7 71 8 32 9 85
0 74 1 80 2 60 3 91 4 16 5 65 6 50 7 98 8 70 9 98
0 36 1 24 2 26 3 38 4 48 5 91 6 58 7 33 8 95 9 68

Tabla 8-3. Ejemplo de ejemplar Taillard 019, generada por (Taillard, E. 1993).

El conjunto de ejemplares de (Taillard, E. 1993) son conocidas y reconocidas. Sin embargo, como ya se ha comentado, no ofrece un juego de datos para el Taller de Flujo Híbrido con Tiempos de Cambio de Partida Dependientes de la Secuencia. Es este punto donde se ha considerado oportuno apoyarse en los avances publicados por el Dr. Ruben Ruiz en diversas contribuciones donde si se aborda a esta necesidad.

Es originalmente en (Vallada, E. et al. 2003) donde se propone un método para generar unos ejemplares para el caso del Taller de Flujo con Tiempos de Cambio de Partida Dependientes de la Secuencia a partir de la propuesta de mencionada de Taillard. Para la configuración indicada es necesario incorporar una matriz por cada máquina existente en la cual se informe sobre el tiempo necesario para realizar el cambio entre un “trabajo i” que se acaba de ejecutar y un “trabajo j” que se pueda ejecutar posteriormente en dicha máquina. Los autores proponen la construcción de cuatro grandes grupos de 120 problemas cada uno de ellos. Las características de cada problema, es decir el número de trabajos, número de máquinas y tiempos de proceso, son exactamente los mismos que los propuestos por (Taillard, E. 1993), y comentados anteriormente, pero los autores añaden tantas matrices como máquinas existen en cada problema con los Tiempos de Cambio de Partida.

En el primer grupo de problemas establece un ratio entre los tiempos de proceso y los tiempos de cambio de partida de diez, es decir, los tiempos de cambio son el 10% de los tiempos de proceso. Los tiempos de cambio de partida en este primer grupo de problemas se distribuyen uniformemente entre uno y nueve, $U[1, 9]$. A este primer grupo los autores le denominan SSD10. Los otros tres grupos van aumentando el ratio entre los tiempos de proceso y de

cambio de partida. El segundo grupo, denominado SSD50, tiene un ratio de dos, es decir los tiempos de cambio de partida son el 50% los de tiempos de proceso, con lo que se caracterizan por una distribución uniforme U[1, 49]. El tercer grupo, denominado SSD100, tiene un ratio de uno, es decir, los tiempos de cambio de partida y los de proceso son del mismo orden y ambos se caracterizan por una distribución uniforme U[1, 99]. El último grupo, llamado SSD125, tiene un ratio de 0.8. En este caso los tiempos de cambio de partida son un 25% superiores a los de proceso, luego se distribuirán como una uniforme U[1, 124]. Los autores justifican esta distribución después de examinar la literatura donde encontraron que múltiples autores comprobaron que el ratio entre los tiempos de proceso y de cambio de partida afecta a las prestaciones de los métodos propuestos. Estos problemas se pueden descargar de la página web <http://soa.iti.es/instancias-de-problemas/>.

Un ejemplo para un ejemplar con veinte trabajos y cinco máquinas, empleada en (Ruiz, R. 2003) y (Ruiz, R., Maroto, C., & Alcaraz, J. 2005) es la que se refleja en la Tabla 8-4. La estructura inicial correspondiente a los cinco grupos de dos columnas y veinte filas es la relacionada con los tiempos de proceso, igual que se comentó para el caso del Taller de Flujo de Permutación. Después de dicho bloque se ubican cinco bloques adicionales identificados por M0..M4 que se corresponden con las matrices de tamaño veinte por veinte que proporcionan la información sobre los Tiempos de Cambio de Partida de todos los trabajos en cada una de las cinco máquinas.

20 5
0 77 1 39 2 14 3 11 4 83
0 94 1 31 2 21 3 2 4 13
0 9 1 46 2 15 3 36 4 84
0 57 1 18 2 10 3 30 4 46
0 29 1 93 2 85 3 89 4 20
0 79 1 58 2 46 3 10 4 33
0 55 1 85 2 42 3 88 4 74
0 73 1 58 2 18 3 22 4 42
0 65 1 97 2 36 3 31 4 33
0 86 1 10 2 2 3 9 4 71
0 25 1 79 2 44 3 43 4 32
0 39 1 93 2 89 3 91 4 48
0 76 1 2 2 6 3 26 4 42
0 24 1 87 2 3 3 3 4 99
0 38 1 17 2 1 3 75 4 7
0 5 1 18 2 43 3 99 4 54
0 91 1 10 2 81 3 63 4 8
0 29 1 50 2 57 3 83 4 73
0 22 1 8 2 76 3 70 4 30
0 27 1 26 2 59 3 84 4 75
SSD

M0																			
0	7	9	1	6	3	8	5	7	6	8	3	4	7	3	4	1	9	3	2
3	0	4	3	5	2	9	6	8	6	9	1	8	9	1	8	2	2	1	5
1	3	0	2	2	2	1	8	5	4	2	4	1	4	9	1	1	1	4	8
6	7	6	0	1	1	4	3	9	1	8	2	3	1	4	9	3	1	7	9
6	8	7	8	0	8	2	8	5	2	3	7	4	3	7	3	5	4	9	1
5	1	7	3	7	0	8	6	3	8	7	9	8	2	3	2	9	2	6	5
8	3	7	8	9	1	0	9	6	5	7	2	5	9	2	2	6	5	3	3
9	3	8	7	3	7	2	0	3	6	5	7	2	5	2	9	8	6	1	9
5	7	6	3	9	3	5	6	0	2	2	8	9	2	7	6	3	3	7	5
6	3	6	8	8	3	1	6	9	0	2	1	3	2	7	3	9	1	7	8
7	2	3	3	2	5	3	9	3	9	0	8	5	9	8	2	4	6	5	5
3	9	3	2	5	8	9	2	8	4	2	0	6	9	4	5	9	2	8	8
8	7	8	3	4	5	1	3	4	9	5	4	0	2	2	5	6	9	7	1
8	4	2	6	9	5	6	1	9	5	8	6	5	0	3	6	4	2	1	2
3	2	2	1	8	7	3	2	4	6	9	2	7	5	0	7	7	2	1	9
4	3	4	9	1	9	5	7	8	5	5	4	3	7	8	0	9	4	2	3
5	9	2	7	8	2	7	7	8	2	8	2	9	7	8	7	0	3	3	7
2	3	3	8	7	3	3	5	6	2	4	6	9	1	2	7	2	0	1	1
7	2	9	9	6	5	5	3	8	9	1	7	8	1	8	9	2	1	0	3
7	7	8	7	6	6	3	6	2	6	7	7	1	5	6	9	2	7	7	0
M1																			
0	3	7	1	7	4	7	9	1	9	1	8	2	2	8	4	2	5	3	2
2	0	3	8	1	1	9	6	9	3	9	6	7	2	5	3	3	7	9	7
5	1	0	9	6	1	2	8	2	4	7	9	6	7	6	6	9	9	8	3
7	8	7	0	2	3	1	4	5	1	6	4	5	5	7	5	5	7	5	3
6	6	6	6	0	8	6	7	7	1	7	2	9	8	7	3	6	4	3	9
7	8	5	9	2	0	8	2	8	5	8	8	9	3	3	9	7	7	5	3
2	4	7	1	8	8	0	8	6	9	7	5	3	2	3	4	3	3	4	9
5	6	9	6	1	1	3	0	3	3	5	2	3	7	2	6	6	9	5	5
8	4	3	3	6	1	4	5	0	1	1	5	5	3	4	2	2	3	9	9
6	5	5	6	6	4	2	8	7	0	6	6	4	9	8	6	8	7	8	3
8	7	8	5	3	2	7	7	5	5	0	2	9	2	3	5	5	5	8	8
8	8	6	5	7	7	9	9	9	8	7	0	9	4	3	4	3	7	9	8
1	4	5	4	3	4	4	5	2	2	4	4	0	7	3	9	1	7	2	2
7	1	8	8	9	6	1	5	8	3	8	9	3	0	7	8	3	4	5	7
6	7	7	5	9	5	4	4	6	7	1	4	4	9	0	8	5	1	5	2
4	1	4	1	6	6	6	9	6	7	8	3	3	6	9	0	3	5	7	7
4	7	2	2	3	1	2	8	7	4	9	5	6	2	2	4	0	5	5	3
3	4	1	3	5	3	5	8	6	2	5	2	8	1	5	1	9	0	1	3
8	9	1	8	6	8	5	2	3	9	7	1	3	7	4	9	5	5	0	1
1	3	5	7	1	6	4	2	5	7	4	1	5	8	3	7	1	8	5	0
M2																			
0	1	8	5	1	9	3	7	2	7	6	8	2	9	6	3	8	4	2	1
8	0	4	8	9	6	9	1	5	5	7	2	9	3	9	9	6	9	8	7
6	1	0	7	3	9	1	2	5	1	6	2	8	9	2	6	1	2	2	8
6	4	8	0	6	7	4	6	1	3	5	8	5	1	1	9	1	9	4	7
5	1	3	5	0	9	5	2	4	1	3	3	6	3	6	2	2	2	7	2

6	9	1	9	3	0	4	4	4	9	3	3	3	2	2	5	3	8	9	3
6	8	8	5	2	2	0	5	9	3	5	6	5	4	5	4	8	8	5	1
9	9	5	3	8	3	6	0	7	2	2	1	5	6	2	5	1	4	3	3
6	2	8	5	6	2	3	2	0	7	9	8	7	2	2	4	4	2	8	3
2	8	2	5	3	3	2	3	7	0	1	1	1	4	7	5	1	5	9	8
1	4	3	2	2	8	2	6	4	8	0	6	5	1	9	2	5	7	6	1
6	2	5	3	2	3	8	8	9	6	1	0	1	7	9	8	4	3	9	9
6	8	8	8	7	4	4	8	9	5	7	4	0	1	3	6	3	5	2	1
8	4	1	7	9	3	9	1	1	2	7	8	5	0	2	8	2	7	5	2
2	6	1	3	1	2	9	6	1	5	6	8	5	3	0	9	1	2	1	4
1	6	5	2	7	5	4	4	2	1	8	5	8	1	3	0	6	7	9	8
3	7	2	2	8	6	5	8	8	4	6	2	2	9	6	7	0	3	2	2
1	5	1	1	5	8	2	7	1	3	4	2	4	3	4	2	3	0	2	8
4	7	8	6	5	2	7	8	3	3	1	6	2	7	2	5	6	3	0	7
5	5	2	5	8	9	4	2	8	6	2	1	4	6	8	3	9	1	1	0
M3																			
0	8	1	6	3	9	1	6	6	2	4	1	5	7	8	2	5	8	9	7
2	0	7	2	2	8	3	7	1	7	2	9	1	4	5	7	9	4	8	7
4	2	0	1	9	5	5	8	7	1	5	8	9	1	9	1	1	8	7	4
5	6	9	0	9	7	3	9	3	3	3	3	2	5	5	7	8	1	4	6
9	2	2	9	0	7	4	3	5	8	4	2	7	7	1	5	9	2	7	9
6	5	2	9	2	0	7	6	1	3	5	2	4	9	8	7	3	2	3	2
4	3	8	3	6	7	0	7	7	9	4	7	5	8	8	7	6	9	4	7
4	2	6	8	5	5	8	0	5	4	2	5	9	1	7	2	8	2	1	3
8	8	6	2	5	1	5	2	0	1	4	9	4	2	2	1	4	7	6	2
7	5	6	3	6	6	4	6	1	0	3	9	1	6	2	7	5	4	7	2
6	1	6	7	8	1	9	7	8	1	0	4	6	5	3	5	6	1	1	9
1	9	1	5	2	6	7	4	8	5	6	0	6	6	6	1	8	9	6	5
4	9	1	2	8	4	9	7	6	9	2	3	0	6	2	1	6	2	2	1
7	4	3	8	5	1	3	8	5	3	8	6	8	0	8	7	1	6	7	1
6	7	3	8	5	7	6	8	8	3	9	4	8	6	0	9	8	9	9	9
1	5	2	5	3	2	5	9	5	5	9	8	9	7	9	0	8	3	2	2
6	8	4	7	9	5	1	2	8	3	8	4	6	1	5	3	0	3	3	9
9	4	6	6	7	9	8	3	5	5	5	1	7	3	9	1	7	0	6	6
5	5	4	6	3	1	8	6	6	8	9	6	6	9	8	1	8	4	0	9
3	7	1	6	2	7	5	7	6	6	5	2	9	1	2	9	2	1	6	0
M4																			
0	1	5	4	3	1	5	9	2	9	6	3	4	8	3	1	6	8	5	9
4	0	1	3	3	6	2	3	5	5	5	8	9	4	6	2	2	3	6	3
5	5	0	7	8	3	7	5	3	5	8	7	6	3	3	8	1	2	7	4
5	6	1	0	5	6	2	2	3	5	9	3	4	5	4	3	4	4	9	7
3	4	3	9	0	1	9	5	2	3	2	5	5	3	5	9	4	9	4	6
8	7	9	9	4	0	5	9	1	4	7	2	7	3	1	6	8	8	6	8
2	3	4	5	9	4	0	5	2	9	4	3	8	3	1	5	9	1	3	9
2	8	4	3	9	4	8	0	9	9	9	3	5	2	6	1	3	3	3	8
1	6	3	5	9	9	6	5	0	8	8	1	8	7	9	6	3	8	2	6
5	4	9	2	7	7	9	2	5	0	9	4	6	1	7	7	4	7	5	1
2	8	8	9	2	2	5	8	1	6	0	9	6	5	9	5	4	5	7	6

4	1	3	6	7	7	4	8	8	5	8	0	8	5	5	2	4	6	2	9
5	5	1	5	5	2	6	6	9	1	8	3	0	3	5	3	2	3	7	2
7	9	4	1	3	9	6	5	5	3	4	7	8	0	3	2	4	2	5	9
3	4	5	2	6	7	7	7	6	8	9	1	6	7	0	5	9	4	7	4
6	9	5	8	6	9	2	9	9	1	9	1	9	7	2	0	3	5	9	1
8	8	7	1	4	9	2	6	4	5	1	4	3	8	3	2	0	8	7	1
6	4	5	5	9	4	4	6	9	7	7	5	2	1	3	8	2	0	1	2
2	7	9	9	2	5	5	1	8	6	4	7	9	9	3	7	7	3	0	6
8	3	9	1	9	3	5	5	9	5	6	2	5	8	1	2	9	7	4	0

Tabla 8-4. Ejemplo de ejemplar del grupo SDST10 para Taillard 003, generada por (Vallada, E. et al. 2003).

Igual que ocurre con el banco de pruebas de Taillard para la propuesta de Vallada y otros existen resultados publicados. En el anexo II, la tabla II.2, muestra los mejores resultados obtenidos para el problema de minimización del makespan en un Taller de Flujo con Tiempos de Cambio de Partida Dependientes de la Secuencia utilizando el banco de datos proporcionado por (Vallada, E. et al. 2003).

Finalmente, para el caso del Taller de Flujo Híbrido con Tiempos de Cambio de Partida Dependientes de la Secuencia (Vallada, E. et al. 2003) proponen doce conjuntos de 110 problemas cada uno de ellos que son accesibles desde <http://soa.iti.es/instancias-de-problemas>. Los problemas propuestos se basan a su vez en el método de generación y el banco de datos empleado en el caso del Taller de Flujo con Tiempos de Cambio de Partida Dependientes de la Secuencia. En este caso los autores proporcionan en cada etapa tantos tiempos de proceso como recursos (máquinas) existen, pero a su vez, deben decidir cuantas máquinas no relacionadas existen por etapa. El método utilizado por los autores fue el de establecer tres grandes grupos. En el primer grupo se utiliza una distribución aleatoria uniforme entre uno y tres, $U [1..3]$, para decidir el número de máquinas por etapa, en el segundo grupo el número de máquinas por etapa fue siempre 3, y en el tercer, y último grupo, fue en todos los casos de 3 máquinas por etapa. Dentro de cada gran grupo se ubican cuatro subgrupos que se corresponden con las variaciones de tiempo de cambio de partida propuestas para el problema anterior es decir, SDST10, SDST50, SDST100 y SDST125. Esto conforma los doce grupos de problemas. Cada grupo está identificado por un nombre compuesto por dos partículas, una que identifica al subconjunto de cambio de partida, y otra al gran grupo para la distribución del número de máquinas por etapa. Un ejemplo es el del grupo SSD10_P13, que identifica al grupo formado por 110 problemas con tiempos de cambio de partida generados con una relación del 10% con respecto al tiempo de proceso y con una distribución entre una y tres máquinas por etapa.

Uniendo los doce grupos de problemas tenemos un total de 1.320 problemas diferentes, cuya estructura se puede consultar en Tabla 8-5. Los autores resaltan que el número de etapas y el número de trabajos dentro de cada uno de los doce grupos es idéntico a los problemas de

Taillard, es decir, hay once tipos de problemas de entre veinte trabajos y cinco etapas hasta doscientos trabajos y veinte etapas. En este banco de datos los autores han eliminado los problemas de quinientos trabajos y veinte etapas por ser problemas de un tamaño demasiado grande.

Ejemplares	Entre 1..3 Máquinas por	2 Máquinas por	3 Máquinas por
	Etapa.	Etapa	Etapa
	P13	P2	P3
Tiempos cambio 10%	SSD10_P13	SSD10_P2	SSD10_P3
Tiempos cambio 50%	SSD50_P13	SSD50_P2	SSD50_P3
Tiempos cambio 100%	SSD100_P13	SSD100_P2	SSD100_P3
Tiempos cambio 125%	SSD125_P13	SSD125_P2	SSD125_P3

Tabla 8-5. Resumen de la estructura de ejemplares de (Vallada, E. et al. 2003).

El formato de los ficheros de entrada cambia ligeramente, con respecto a los anteriores. En (Ruiz, R. 2003) se emplea una estructura en la cual, para poder considerar todas las restricciones, en la primera línea aparecen tres números, el primero representa el número de trabajos (n), el segundo el número de máquinas totales (m) y el tercero el número de etapas (e). En la segunda línea, los autores han dispuesto, un total de “e” datos, que representan el número de máquinas no relacionadas que existen por etapa. Tras estas dos líneas aparece una matriz con los tiempos de proceso que tiene tamaño nxm, es decir, se especifican todos los tiempos de proceso para todas las máquinas, no por etapas. Al terminar esta matriz aparecen m matrices de tamaño nxn, que son las matrices de los tiempos de cambio de partida para cada una de las m máquinas. Un ejemplo puede ser el problema ta009_SSD10_P13, de veinte trabajos y cinco etapas, que se muestra en la Tabla 8-6.

20 8 5
2 2 1 1 2
0 74 1 74 2 34 3 83 4 79 5 10 6 73 7 76
0 32 1 22 2 84 3 26 4 54 5 44 6 19 7 85
0 67 1 30 2 3 3 65 4 78 5 24 6 69 7 61
0 73 1 89 2 60 3 47 4 29 5 85 6 12 7 32
0 7 1 61 2 34 3 94 4 8 5 75 6 73 7 90
0 78 1 39 2 70 3 75 4 38 5 66 6 85 7 61
0 64 1 15 2 57 3 30 4 97 5 49 6 23 7 87
0 98 1 69 2 8 3 1 4 61 5 95 6 53 7 51
0 60 1 57 2 74 3 71 4 10 5 61 6 16 7 25
0 89 1 9 2 13 3 46 4 37 5 19 6 88 7 73
0 49 1 13 2 37 3 87 4 16 5 47 6 8 7 93
0 2 1 71 2 87 3 78 4 78 5 84 6 26 7 28
0 79 1 2 2 71 3 76 4 96 5 13 6 42 7 90

0 79 1 34 2 89 3 75 4 9 5 11 6 58 7 94
0 53 1 49 2 57 3 55 4 91 5 19 6 63 7 59
0 59 1 65 2 70 3 94 4 46 5 98 6 7 7 64
0 16 1 94 2 3 3 98 4 52 5 2 6 2 7 2
0 90 1 96 2 43 3 63 4 79 5 85 6 44 7 16
0 3 1 47 2 14 3 83 4 45 5 44 6 38 7 35
0 76 1 35 2 26 3 19 4 97 5 7 6 24 7 53
SSD
M0
0 4 4 1 1 6 7 3 7 7 1 8 6 9 6 9 5 1 8 8
5 0 6 2 9 1 7 7 2 3 9 6 4 2 7 6 1 2 7 1
4 5 0 6 9 9 5 4 4 8 1 6 4 7 6 1 7 2 4 8
7 9 6 0 7 1 4 2 3 8 3 6 5 9 7 3 1 7 5 8
8 7 7 5 0 9 9 6 8 2 8 5 8 3 1 4 9 6 1 4
2 8 9 1 9 0 5 5 8 5 9 1 4 3 8 7 6 5 9 6
2 5 8 2 1 2 0 9 1 8 4 1 7 2 7 2 7 8 3 5
2 8 1 3 4 6 6 0 1 1 4 4 3 7 8 6 3 9 6 8
5 3 7 9 3 9 9 6 0 6 1 2 4 5 4 8 3 8 1 1
1 6 9 5 7 6 9 1 2 0 1 9 4 5 6 7 2 6 7 6
9 7 1 4 4 3 9 4 3 8 0 1 1 6 8 3 7 6 6 6
4 5 7 9 4 8 5 6 1 6 4 0 5 8 8 1 3 4 3 3
9 1 1 5 1 7 2 3 4 1 8 7 0 7 3 8 3 9 7 1
3 4 9 5 3 2 6 7 5 5 5 2 2 0 6 9 1 1 3 6
4 4 6 5 6 6 6 1 1 2 6 4 5 5 0 8 3 1 2 7
4 7 7 2 8 7 4 2 4 9 7 3 8 9 3 0 4 2 4 5
9 2 9 4 7 4 8 5 4 7 9 9 1 9 4 7 0 7 9 4
3 8 1 6 5 8 5 1 5 4 9 4 8 3 6 8 3 0 4 1
6 1 9 1 4 6 3 9 8 9 7 7 4 4 4 4 7 6 0 7
3 8 4 9 8 9 3 1 8 6 4 4 7 9 6 1 4 5 1 0
M1
0 4 6 3 3 3 1 7 3 2 9 3 1 9 1 6 9 4 6 2
1 0 8 8 4 4 8 7 3 2 9 8 6 7 1 3 7 1 7 7
8 3 0 5 1 9 7 9 5 1 4 8 6 6 1 9 1 3 9 5
9 5 5 0 1 1 1 5 7 6 8 2 8 6 2 3 9 3 9 5
8 8 3 7 0 5 7 3 6 9 5 1 4 7 2 1 9 9 4 7
7 3 3 5 9 0 8 3 9 9 6 5 1 4 4 5 9 3 4 8
5 6 7 4 6 1 0 3 9 7 9 2 4 4 6 6 2 1 4 9
8 3 2 8 2 4 6 0 8 3 6 1 2 5 1 8 2 4 6 3
2 8 8 4 4 9 1 9 0 2 9 6 5 3 4 3 3 5 7 3
5 4 6 4 5 3 3 7 5 0 9 2 6 8 9 9 5 8 1 9
3 3 3 2 2 3 4 8 6 1 0 4 8 6 1 1 9 6 8 1
4 6 5 9 4 1 8 5 2 4 3 0 2 7 2 8 4 7 2 8
4 3 7 2 9 9 6 8 6 1 6 6 0 4 1 2 2 6 7 5
6 8 7 7 7 2 1 1 9 6 2 9 6 0 8 3 4 5 7 3
3 7 2 2 2 6 1 6 7 7 6 8 9 7 0 1 2 7 2 3
9 2 7 5 1 8 6 9 6 9 6 1 2 6 5 0 1 7 7 2
1 4 4 6 2 3 7 4 9 1 7 3 2 6 3 6 0 8 5 7
8 6 4 1 6 1 5 4 3 4 4 9 1 9 5 7 8 0 7 6
1 6 1 2 6 4 9 8 6 7 5 9 4 5 2 7 3 4 0 1
1 7 7 5 3 4 1 1 1 2 7 6 3 7 4 9 8 2 3 0

M2
0 1 3 4 2 4 9 3 7 6 2 9 3 7 4 2 7 9 1 1
5 0 3 1 5 8 6 6 7 3 8 8 8 4 2 4 7 6 6 1
7 1 0 9 3 1 5 3 2 3 5 9 4 5 5 1 3 3 2 7
3 4 8 0 7 3 2 1 2 2 6 1 9 9 6 3 1 9 1 9
1 3 2 9 0 8 3 7 2 2 5 7 4 5 6 3 6 6 5 8
3 6 2 7 1 0 3 8 5 3 8 9 6 9 7 7 6 5 2 5
8 3 9 5 1 2 0 1 6 3 5 4 3 8 4 8 4 1 3 3
5 6 9 6 4 6 8 0 5 7 5 6 1 7 5 7 2 7 9 6
6 2 2 8 4 2 4 5 0 3 7 4 5 4 3 6 6 8 2 2
7 3 4 9 5 8 3 8 4 0 9 3 9 7 1 1 8 4 2 3
7 2 1 4 5 8 5 6 3 1 0 1 7 3 7 7 1 1 5 6
8 2 1 1 5 9 9 1 2 3 7 0 9 2 7 6 9 8 6 4
8 4 6 8 8 8 7 3 4 4 4 4 0 1 6 6 6 7 6 2
9 3 3 5 9 5 4 6 8 5 8 8 5 0 7 2 8 2 6 1
8 8 1 2 7 5 2 1 4 8 7 8 9 5 0 1 5 1 3 2
2 2 1 2 3 1 1 3 7 1 7 2 6 8 9 0 6 9 5 7
2 6 4 1 8 8 9 2 7 2 2 8 1 8 9 5 0 6 8 9
9 1 3 5 5 5 3 6 3 1 2 5 3 3 1 9 9 0 2 3
5 6 5 8 7 9 4 6 6 8 8 2 9 6 2 1 7 8 0 6
3 9 8 8 1 8 8 2 3 5 7 5 9 2 1 8 2 6 6 0
M3
0 3 1 3 8 5 5 6 8 5 7 2 6 7 7 3 6 6 7 9
9 0 4 9 1 6 7 3 3 7 8 5 8 4 3 6 7 1 2 3
2 6 0 4 6 3 8 6 1 5 8 9 6 9 3 7 8 2 6 6
7 9 2 0 7 5 1 4 9 7 9 7 8 5 2 2 2 2 9 2
1 3 2 9 0 4 4 1 2 1 1 1 7 3 1 5 4 9 5 8
2 4 4 3 7 0 3 5 5 4 4 3 8 6 6 6 2 3 5 5
2 2 2 6 9 9 0 4 3 2 7 6 8 1 2 6 1 2 9 7
3 5 3 7 8 5 7 0 7 2 5 8 3 8 7 9 6 6 6 5
7 3 9 4 7 6 6 3 0 4 6 8 8 2 2 6 9 8 6 1
5 1 8 7 8 4 5 1 5 0 8 3 1 7 3 5 5 3 5 5
9 2 8 5 3 1 8 1 8 4 0 4 1 2 2 5 4 1 6 4
2 6 2 6 9 5 9 3 2 6 7 0 3 1 3 9 1 6 6 3
9 2 1 1 9 1 3 2 3 7 4 3 0 1 5 6 8 2 4 5
8 3 8 9 8 8 1 6 5 6 8 4 3 0 3 1 2 1 1 5
4 7 7 2 2 9 2 3 3 3 3 5 2 9 0 2 8 6 7 3
6 1 4 4 1 3 1 9 5 8 8 8 5 1 1 0 9 1 2 6
4 5 3 6 7 4 2 6 7 6 4 5 1 7 2 7 0 5 5 5
4 8 6 5 4 6 3 8 6 1 2 5 1 1 2 6 3 0 9 8
7 6 6 6 3 8 9 4 5 8 2 3 5 5 1 6 8 6 0 8
1 2 4 8 7 6 7 6 1 2 2 9 9 5 4 9 8 3 6 0
M4
0 4 5 7 5 4 4 9 1 1 6 8 2 9 7 5 2 9 5 9
2 0 3 2 6 2 9 2 6 4 7 4 3 2 3 3 3 8 5 3
6 2 0 4 8 9 3 4 9 4 8 8 4 5 7 5 7 1 4 8
8 2 6 0 5 4 7 9 3 5 7 9 2 2 8 6 7 4 3 6
9 6 1 1 0 5 1 7 2 5 5 4 6 7 3 8 5 1 2 1
9 9 8 1 5 0 7 9 5 7 7 7 6 3 3 9 8 5 5 5
1 4 5 7 3 5 0 7 5 1 8 2 6 1 4 6 7 3 6 4

5	3	2	2	1	9	1	0	9	8	4	6	2	7	1	2	5	8	2	6
1	8	6	2	9	2	3	5	0	3	9	5	2	2	4	2	3	8	3	4
3	6	8	8	9	3	3	1	4	0	7	2	5	1	4	6	6	3	5	7
2	8	2	1	5	7	8	2	8	8	0	6	6	1	6	3	6	8	5	8
7	2	7	5	2	8	1	9	6	5	1	0	9	8	3	3	6	4	8	5
5	2	2	6	4	1	5	6	9	1	4	9	0	5	4	1	9	7	3	6
3	6	1	5	6	9	9	5	2	9	6	3	6	0	8	6	9	4	6	4
8	9	9	6	6	4	4	3	9	4	1	8	8	3	0	3	7	5	3	3
3	4	6	8	9	3	3	1	4	9	5	7	8	6	9	0	2	3	4	3
9	6	7	2	4	9	6	1	1	7	3	3	6	6	4	8	0	9	4	9
5	2	8	3	6	8	8	8	1	4	4	8	5	3	4	6	4	0	3	7
3	1	6	5	9	5	5	8	2	1	4	3	8	8	5	6	9	3	0	5
1	7	8	4	4	2	6	9	6	1	6	2	2	3	2	1	5	8	7	0
M5																			
0	6	9	2	1	2	2	5	1	5	8	9	3	2	4	5	1	2	4	6
1	0	5	1	4	2	2	8	8	3	8	1	5	2	1	7	3	2	7	2
9	9	0	3	9	5	5	3	6	3	1	3	3	4	8	5	2	4	2	1
6	8	1	0	8	3	8	2	8	2	5	3	8	8	4	9	4	2	6	2
2	3	3	2	0	1	1	6	6	9	6	9	6	4	4	3	2	9	1	8
2	3	6	9	9	0	9	4	1	8	9	3	9	5	1	3	7	9	2	7
9	2	5	1	2	7	0	7	4	4	6	3	8	9	8	4	4	4	6	6
6	6	5	3	4	2	2	0	7	6	2	7	6	7	2	6	5	7	1	9
6	3	9	9	1	9	4	8	0	3	9	1	4	1	4	6	9	1	2	1
4	8	6	4	2	5	4	8	9	0	8	5	5	3	3	8	3	7	5	3
4	4	5	6	5	3	4	8	7	3	0	5	7	6	2	7	2	3	1	8
9	4	9	9	6	4	4	4	3	3	1	0	8	5	4	8	8	7	2	7
2	1	4	3	1	4	7	4	1	2	1	3	0	5	8	4	6	2	7	7
1	3	6	5	2	2	8	4	3	7	1	9	8	0	6	2	5	5	7	2
4	1	2	5	6	5	2	8	7	2	9	4	6	3	0	6	2	2	6	7
3	4	4	7	3	4	6	1	1	3	8	5	1	1	7	0	7	9	6	2
4	8	6	6	3	5	5	8	7	3	3	3	9	6	6	2	0	4	7	6
9	6	4	9	7	7	4	2	1	6	9	5	4	6	1	9	1	0	3	1
6	8	1	8	6	5	2	2	3	2	9	5	4	6	1	2	7	2	0	1
4	7	1	9	7	5	6	9	2	6	2	9	3	4	1	5	3	5	1	0
M6																			
0	1	2	4	5	2	5	3	6	3	8	3	3	1	9	1	5	4	8	7
9	0	5	6	4	8	6	6	2	6	4	6	2	2	4	8	1	5	6	6
7	5	0	1	9	1	1	6	7	8	2	2	7	8	1	1	3	8	9	4
6	9	7	0	3	6	6	9	6	9	2	5	4	3	6	2	8	9	7	1
6	9	3	8	0	3	7	7	6	2	7	4	6	8	4	7	6	6	2	9
5	5	6	5	6	0	3	3	1	2	1	4	6	5	1	1	5	3	6	7
8	1	4	2	2	6	0	7	7	7	1	8	8	4	3	1	6	7	9	5
5	5	9	1	8	1	4	0	5	7	7	9	4	8	8	9	4	7	4	4
9	2	7	7	5	7	1	3	0	8	3	1	3	8	1	4	8	5	4	3
9	3	4	2	4	8	5	5	9	0	8	9	2	3	9	8	9	1	2	7
4	8	4	8	1	6	4	9	7	9	0	5	5	6	1	6	9	9	6	4
8	7	3	4	9	9	1	6	6	3	4	0	2	7	1	9	7	5	4	2
6	4	1	1	4	9	6	1	4	5	4	2	0	7	6	1	3	2	6	3
3	1	1	3	2	8	1	6	1	8	8	8	3	0	3	4	4	4	1	8
2	8	6	7	5	1	5	1	4	3	6	9	9	6	0	9	6	5	4	7

2	3	8	4	6	9	6	6	3	2	4	8	6	9	4	0	4	1	6	1
5	4	7	2	4	4	6	5	9	8	2	3	6	9	1	8	0	7	8	7
8	4	8	6	5	3	7	3	2	7	4	9	9	2	8	2	6	0	4	1
5	6	3	7	8	7	7	2	1	9	5	8	3	2	8	7	2	1	0	2
9	1	5	9	5	7	3	8	2	6	1	4	8	5	6	2	1	9	2	0
M7																			
0	7	9	4	2	8	9	1	4	8	4	6	3	2	8	1	4	2	8	7
7	0	2	7	2	8	6	1	4	3	4	4	8	3	3	5	3	6	5	2
2	6	0	6	9	2	2	6	8	6	1	8	2	5	1	7	4	8	3	7
8	2	4	0	4	6	6	8	8	6	2	6	6	6	3	4	2	1	3	5
9	3	1	6	0	1	2	4	5	5	1	2	2	3	6	5	7	1	4	3
2	8	4	7	2	0	9	9	5	2	6	7	1	4	6	6	6	3	7	7
8	5	3	4	5	8	0	1	3	5	3	3	6	8	7	3	6	7	9	3
2	9	1	2	9	2	5	0	9	7	8	9	8	1	2	2	2	4	8	5
3	5	6	1	9	8	4	3	0	5	7	6	3	3	9	5	9	5	3	6
6	2	6	3	1	4	1	6	2	0	5	2	4	6	4	4	8	7	5	4
3	9	6	3	7	4	1	7	6	4	0	1	5	2	9	4	2	1	1	1
4	6	2	5	2	8	4	9	5	6	3	0	8	1	7	6	9	2	7	1
2	5	2	5	7	4	7	4	7	9	1	3	0	7	5	5	4	5	8	7
3	4	3	4	8	9	4	5	4	9	7	5	6	0	7	5	7	9	7	3
6	9	6	1	7	1	7	5	6	9	8	3	7	5	0	8	5	9	5	6
9	2	4	3	5	8	9	4	2	2	6	3	1	1	7	0	9	7	9	9
7	6	3	4	1	3	8	5	8	5	2	6	4	1	9	2	0	1	8	2
7	7	6	7	8	9	9	1	5	8	7	7	4	3	4	6	6	0	9	3
8	7	5	1	3	1	1	9	4	8	7	3	9	3	5	2	4	2	0	4
3	4	1	1	8	3	8	6	9	6	3	2	3	3	7	7	3	2	1	0

Tabla 8-6. Ejemplo de ejemplar del grupo SSD10_P13 para Taillard 009, generada por (Vallada, E. et al. 2003).

Igual que ocurre con el juego de datos anterior existen unos resultados publicados. En el anexo II, las tabla II.3 a tabla II.14, muestran los mejores resultados obtenidos para el problema de minimización del Cmax en un Taller de Flujo Híbrido con Tiempos de Cambio de Partida Dependientes de la Secuencia utilizando el banco de datos proporcionado por (Vallada, E. et al. 2003).

Se ha considerado que si bien, el juego de datos propuesto por (Vallada, E. et al. 2003) es válido en su composición, ya que permite analizar un amplio conjunto de escenarios, posiblemente la utilización completa del mismo pueda obligar a emplear un tiempo excesivo para la ejecución de los experimentos. Como se comentará posteriormente, la ejecución de algoritmos que incluyen tecnología basada en Agentes, puede suponer un incremento en los tiempos de ejecución de los mismos respecto a la tecnología tradicional. Por este motivo, sólo se han empleado para el proceso experimental los datos correspondientes a la columna uno y tres de la Tabla 8-5 (marcadas en gris), que se consideran suficientemente representativas para el objetivo del estudio. Se trata del conjunto de ejemplares correspondientes a los casos relacionados con entre 1 y 3 máquinas por etapa, y los que presentan 3 máquinas por etapa.

Cada uno de ellos con los 4 juegos de datos de los distintos tiempos de cambio de partida dependientes de la secuencia (SSD10, SSD50, SSD100 y SSD125). Queda fuera del análisis los ejemplares con 2 máquinas por etapa.

Dado que el objetivo del diseño experimental es establecer el conjunto de parámetros del algoritmo SMAGA más adecuado para diversas configuraciones, es importante dividir el juego de datos en bloques de configuraciones representativas. En este sentido hay que recordar que es necesario que la parametrización cubra, al menos, aquellos problemas que se puedan dar en el caso Industrial Cerámico, y si es posible, algún caso adicional que permita un análisis más general sin comprometer el proceso de análisis. Por este motivo se ha dividido el juego de datos en bloques que permitan, no sólo discernir entre diversas situaciones en relación al tiempo de cambio de partida y máquinas asignadas por etapa, sino también en cuanto al número de trabajos que se desea programar. Con este objetivo se estableció la Tabla 8-7 que consta de 16 entradas, cada una de las cuales será referida como "Conjunto Experimental".

El ajuste de parámetros se realizará de forma individualizada para cada uno de los conjuntos experimentales indicados en la Tabla 8-7.

Ejemplares	Entre 1..3	Entre 1..3	3 Máquinas por	3 Máquinas por
	Máquinas por	Máquinas por	Etapa	Etapa
	Etapa	Etapa	y 20 Trabajos.	y 20 Trabajos.
	y 20 Trabajos.	y 50 Trabajos.		
	P13_20	P13_50	P3_20	P3_50
Tiempos cambio 10%	SSD10_P13_20	SSD10_P13_50	SSD10_P3_20	SSD10_P13_50
Tiempos cambio 50%	SSD50_P13_20	SSD50_P13_50	SSD50_P3_20	SSD50_P13_50
Tiempos cambio 100%	SSD100_P13_20	SSD100_P13_50	SSD100_P3_20	SSD100_P13_50
Tiempos cambio 125%	SSD125_P13_20	SSD125_P13_50	SSD125_P3_20	SSD125_P13_50

Tabla 8-7. Resumen de los conjunto experimentales empleadas en el diseño experimental (elaboración propia).

Con el objeto de realiza un mejor ajuste de los parámetros cada tipo de configuración relacionada con las máquinas disponibles en cada etapa (P13 y P3) se ha desdoblado con el objeto de diferenciar los casos en los que se consideran veinte trabajos, y los casos en los que se consideran cincuenta trabajos. De esta manera las filas de la tabla marcan las diferentes opciones en cuanto a tiempos de cambio de partida (SSD10, SSD50, SSD100 y SSD125), y cada columna diferencia entre diversas configuraciones de taller y número de trabajos (P13_20, P13_50, P3_20 y P3_50).

Para refinar el análisis y poder considerar cual es el comportamiento del algoritmo en función del número de etapas del taller se han seleccionado y agrupado los ejemplares de cada conjunto experimental. Cada conjunto experimental dispone de hasta 110 ejemplares, como se indica en la Tabla 8-2 (exceptuando la entrada 500x20 que no se incluye en este caso), pero éstas sólo son las sesenta primeras las que consideran los casos de veinte y cincuenta trabajos. Entre éstas se ha decidido utilizar sólo los treinta primeros ejemplares pares (ejemplares ta002, ta004..ta060), para realizar el experimento más ágilmente sin perder representatividad. Esto supone disponer de quince ejemplares para una configuración de 20 trabajos, distribuidos un grupo de cinco ejemplares para un taller con cinco etapas denominado G20_5, un grupo de cinco ejemplares para un taller con diez etapas denominado G20_10, y otro grupo de cinco ejemplares para un taller con 20 etapas denominado G20_20, y de otros quince ejemplares para una configuración de 50 trabajos con la misma distribución que en el caso anterior, es decir compuesta por los grupos G50_5, G50_10, y G50_20. En la Tabla 8-8 se muestra, como ejemplo, una distribución por grupos para los conjuntos experimental SSD10_P13_20 y SSD10_P13_50. Dado que cada conjunto experimental está compuesto por 3 grupos y cada grupo por 5 ejemplares se emplearán un total de 240 ejemplares en cada tratamiento.

SSD10_P13_20			SSD10_P13_50		
G20_5	G20_10	G20_20	G50_5	G50_10	G50_20
ta_002	ta_012	ta_022	ta_032	ta_042	ta_052
ta_004	ta_014	ta_024	ta_034	ta_044	ta_054
ta_006	ta_016	ta_026	ta_036	ta_046	ta_056
ta_008	ta_018	ta_028	ta_038	ta_048	ta_058
ta_010	ta_020	ta_030	ta_040	ta_050	ta_060

Tabla 8-8. Distribución de Ejemplares en grupos para dos conjuntos experimentales (elaboración propia).

El objetivo es encontrar los parámetros más adecuados del algoritmo SMAGA para cada uno de los 16 conjuntos experimentales. Y posteriormente emplear los resultados obtenidos con los parámetros elegidos para ser comparados, por grupos, con otras propuestas sobre los mismos ejemplares para determinar la eficiencia del algoritmo SMAGA. Se ha tenido en cuenta, y así se refleja en las tablas correspondientes del anexo II, que los resultados obtenidos sobre dicho juego de datos se ha ido mejorando progresivamente desde al año 2003.

8.2.3.2 La Medida para la Evaluación de los Algoritmos

En todo proceso de configuración y análisis de un algoritmo es necesario establecer al menos una variable que permita su evaluación. En este apartado dicha variable, que en la terminología experimental se denomina variable explicada o endógena del modelo, podría ser el makespan de cada Programa Productivo obtenido con el algoritmo SMAGA. Sin embargo en

esta tesis se utilizará como medida comparativa el Incremento Porcentual Sobre el Óptimo, o la cota mínima conocida, del Valor Examinado Promedio (IPSOVEP) de un problema o ejemplar dado. Esta medida se puede expresar como:

$$IPSOVEP = \frac{Actual_sol - Mejor_sol}{Mejor_sol} * 100$$

La variable “Actual_sol” representa al valor del makespan obtenido con el algoritmo que se desea evaluar con un ejemplar determinado. La variable “Mejor_sol” representa al valor del mejor makespan conocido para dicho ejemplar. Así por ejemplo, para el ejemplar “ta002” del conjunto de problemas SDST10_20_P13 empleado en el Taller de Flujo Híbrido con Tiempos de Cambio de Partida Dependientes de la Secuencia el mejor resultado conocido es 1.029, y por lo tanto un ejecución que obtuviera 1.050 alcanzaría un IPSOVEP de 2,04, es decir es un 2,04% peor que la mejor solución conocida. Valores positivos de IPSOVEP implican que el algoritmo valorado presenta un makespan peor que la referencia utilizada, y valores de IPSOVEP negativos implican que se ha mejorado la referencia.

Atendiendo a una subdivisión realiza en grupos de ejemplares y conjunto experimentales se han introducido otras dos variables. La variable **IPSOVEPG** tendrán en cuenta el IPSOVEP promedio para un todos los ejemplares de un grupo, y la variable **IPSOVEPT** tendrá en cuenta el IPSOVEP promedio para un conjunto experimental completo.

El objetivo que se persigue al establecer IPSOVEP como referencia en la comparación es doble:

- Por un lado, se considera que esta forma de expresar el resultado facilita mucho la comprensión del hecho fundamental que se persigue, que es saber como de bueno o malo es el resultado alcanzado.
- Por otro lado, se dispone de un amplio y pormenorizado análisis presentado por (Ruiz, R. 2003) en el cual se aportan los resultados obtenidos con varios de los métodos más relevantes empleados en el cálculo de Programas Predictivos para el tipo de taller que se van analizar y con el mismo juego de datos. Estos resultados se han publicado empleando una variable denominada IPSO que es equivalente a la que se propone en esta tesis, IPSOVEP.

En el proceso de Parametrización se trabajará con la variable IPSOVEPT, mientras que el análisis comparativo se trabaja con la variable IPSOVEPTG que proporciona mayor detalle.

8.3 Diseño de un Nuevo Algoritmo Genético basado en Sistemas Multiagente Aplicado al Taller de Flujo Híbrido con Tiempos de Cambio de Partida Dependientes de la Secuencia.

Cabe recordar que en este capítulo se desea abordar solamente el problema de obtención o cálculo de un Programa Predictivo a partir de una condiciones definidas y conocidas de ante mano. Y como se ha desarrollado en los capítulos dedicados al estado del arte existen un conjunto de técnicas que han sido empleadas en repetidas ocasiones para la Programación de la producción de Talleres de Flujo en sus diversas configuraciones, y que han demostrado ser muy eficaces. Entre dichas soluciones se encuentran los Algoritmos Genéticos (AG).

A continuación se comentará en primer lugar cual es la estructura y funcionamiento de un Algoritmo Genético general, para posteriormente introducir un nuevo Algoritmo Genético basado en Sistemas Multiagente. El algoritmo propuesto se denomina **SMAGA** (Scheduling MultiAgent Genetic Algorithm). El algoritmo se desarrolla en base al concepto de Sistemas Multiagente con la intención de proporcionar un valor añadido.

8.3.1 Estructura Tradicional del Algoritmo Genético

Un Algoritmo Genético es un procedimiento basado en la analogía con la evolución de los seres vivos. La premisa que subyace en este tipo de enfoques es, que se puede encontrar soluciones aproximadas a problemas de gran complejidad computacional mediante un procedimiento de evolución simulada matemáticamente en un ordenador (Holland, J. H. 1975), (Goldberg, D. E. 1989).

Un Algoritmo Genético básico consta de cuatro fases bien definidas: Generación de la Población, Selección, Operación Genética, y Actualización de la Población. Se puede ver un esquema funcional del mismo en la Figura 8-2.

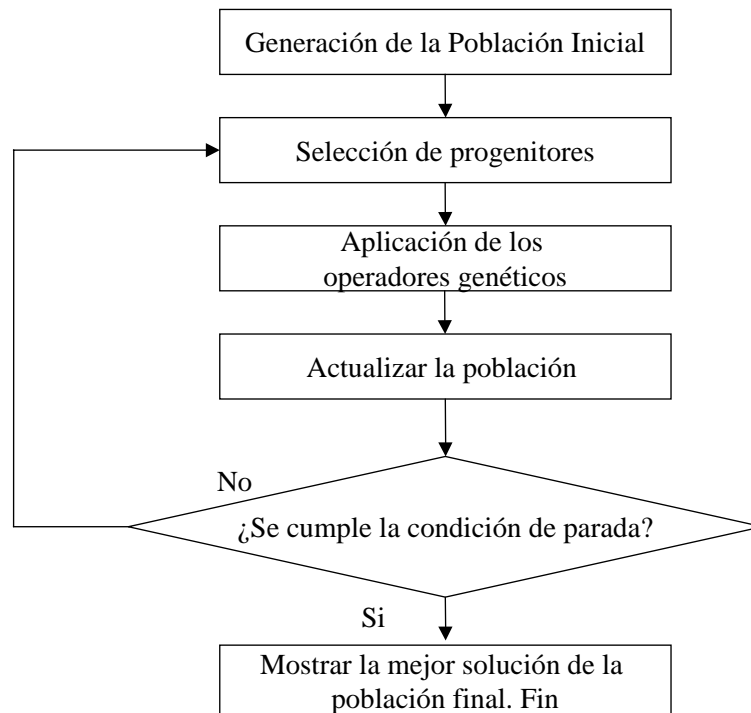


Figura 8-2. Esquema de una Algoritmo Genético (elaboración propia).

En primer lugar se genera una “población de individuos” formada por un conjunto de soluciones al problema que pueden ser creadas de manera aleatoria o mediante alguna heurística conocida. Aunque en general, existen tantos procedimientos para generar soluciones iniciales como procedimientos existen para tratar un problema, se suele descartar todos aquellos que impliquen un elevado coste computacional. La información relevante de cada individuo es almacenada.

A continuación se realiza un proceso de selección de individuos “padres” donde una o más soluciones son escogidas mediante algún procedimiento basado en mayor o menor medida en el azar, pero teniendo en cuenta el valor de la función de bondad (fitness), que mide el grado de adaptación del individuo a su entorno y que usualmente depende de la función objetivo. Esto significa que, en la mayoría de las propuestas, antes de la selección se realiza una evaluación de los individuos, y también que el proceso de selección utiliza el valor de bondad para favorecer la probabilidad de elegir individuos “buenos” en términos de la función objetivo.

Después se aplica a los individuos elegidos algún operador genético. Los operadores genéticos por excelencia son el “Cruce” y la “Mutación”. Ambos operadores se aplican con un cierto grado de aleatoriedad.

El operador de cruce tiene como objetivo explorar en el espacio de soluciones mediante la generación de nuevos individuos mediante combinación de individuos ya existentes en la población. Existen múltiples operadores de cruce desarrollados, aunque para el tipo de representación que se realizará en este problema los mejores candidatos son aquellos que se basan en el intercambio de posiciones en una secuencia de genes dada. Entre estos podemos encontrar desde los más simples como el Order Crossover (OX) de (Davis, L. 1985), o el Partially Matched Crossover (PMX) de (Goldberg, D. E. & Lingle, Jr. R. 1985), y algunas propuestas más elaboradas y recientes como los operadores Similar Job Order Crossover (SJOX), Similar Block Order Crossover (SBOX), Similar Job 2-point Order Crossover (SJ2OX) o Similar Block 2-Point Order Crossover (SB2OX) de (Ruiz, R. 2003).

Dos operadores ampliamente utilizados en el desarrollo de Algoritmos Genéticos son el Uniform Order based Crossover (UOBX) propuesto por (Syswerda, G. 1996) y Generalized Position Crossover (GPX) propuesto (Mattfeld, D. C. 1996). A continuación se va a ilustrar el funcionamiento de un operador de cruce describiendo el operador GPX.

El operador GPX propuesto por (Mattfeld, D. C. 1996) se puede emplear para combinar dos individuos cuya representación genética esté basada en una secuencia de trabajos, y genera a su vez otros individuos. Éste consta de dos pasos:

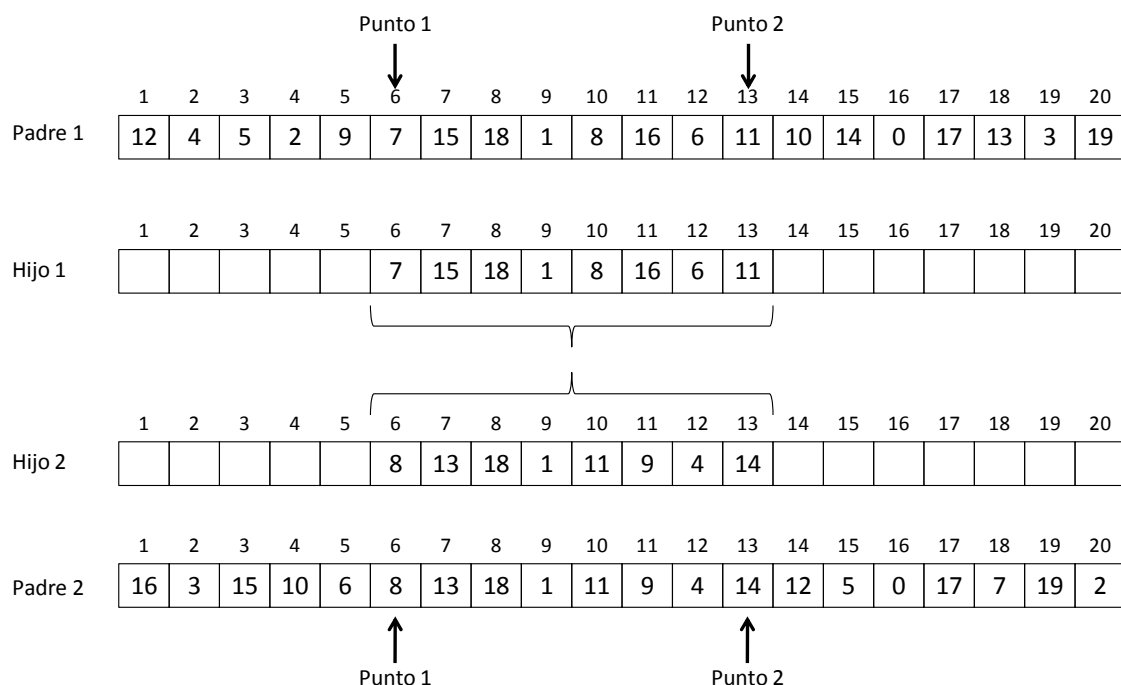


Figura 8-3. Primer paso del operador GPX (elaboración propia).

1. El primer paso consiste en determinar dos puntos de forma aleatoria, de tal forma que, todos los genes del progenitor o padre 1, situados entre ambos puntos, son transferidos al descendiente o hijo 1. De la misma forma el padre 2 transfiere el mismo bloque al hijo 2. Esta operación se puede ver en la Figura 8-3ⁱⁱⁱ.

2. El segundo paso consiste en que el padre 1 transfiere todos los genes no pertenecientes al bloque determinado por los puntos 1 y 2, al hijo 2. De la misma forma actuará el padre 2, en este caso transfiriendo su carga genética la hijo 1. En la Figura 8-4 se puede ver una ilustración de este paso.

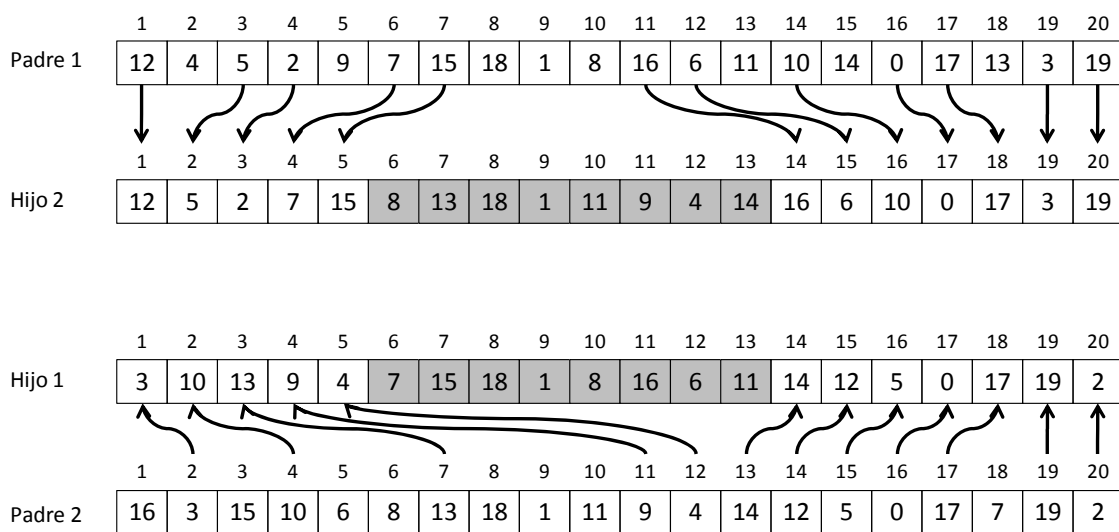


Figura 8-4. Segundo paso del operador GPX (elaboración propia).

Existen otras propuestas como el operador Neighborhood Orthogonal Crossover (NOX) de (Zhong, W. C., Liu, J., Xue, M. Z., & Jiao, L. C. 2004), que aunque no se pueden aplicar sobre el tipo de representación genética escogida para este problema sugiere una forma de orientar los cruces hacia competiciones locales como se verá posteriormente.

La aplicación del operador de mutación implica la modificación un individuo ya existente en la población mediante una operación que normalmente suele ser sencilla, por ejemplo modificar el valor de gen. El objetivo que se persigue con la aplicación de este tipo de operadores es evitar una rápida convergencia de la población provocando alteraciones genéticas que impidan

ⁱⁱⁱ La representación de los individuos se comenta en el apartado 8.3.2.2.1.

que los individuos se asemejen entre sí rápidamente. Igual que ocurre con el operador de cruce, la forma de aplicar el operador depende mucho del tipo de representación genética que se emplee. Para el caso basado en una representación, como se indicó en la situación anterior, se puede decir que existen dos tendencias claras en cuanto al tipo de operador de mutación: las basadas en un intercambio de posiciones (SWAP Mutation) y las basadas en un desplazamiento de la secuencia (SHIFT Mutation). En la Figura 8-5 se presenta un ejemplo de mutación por intercambio de posiciones en la que las posiciones 8 y 14 intercambian sus valores. Y en Figura 8-6 se muestra un ejemplo de mutación mediante desplazamiento en la que la posición 14 ocupa el lugar de la posición 8 y desplaza los valores 18, 1, 8, 9, 6 y 11 una posición a la derecha.

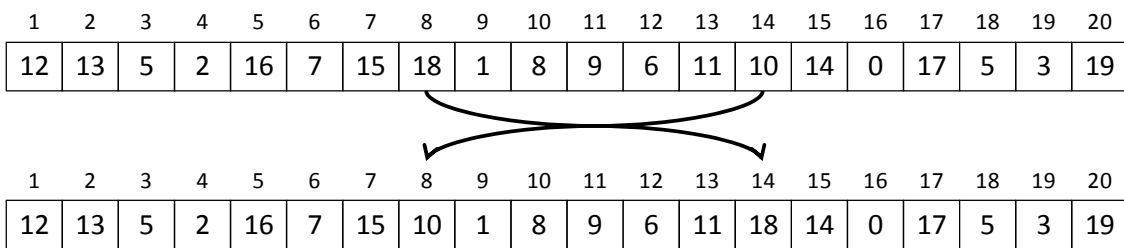


Figura 8-5. Ejemplo de SWAP Mutation.

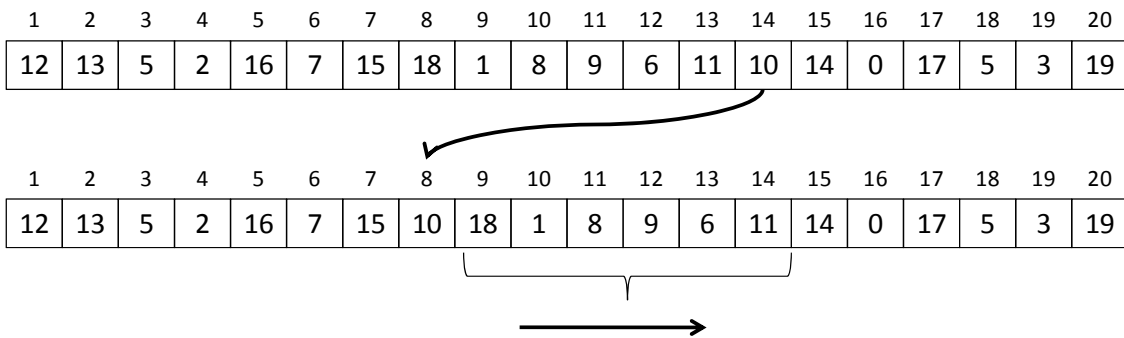


Figura 8-6. Ejemplo de SHIF Mutattion (elaboración propia).

El último paso de un Algoritmo Genético es la evaluación de la bondad de las nuevas soluciones generadas (individuos hijos), y la modificación de la población en función de algún criterio previamente establecido.

El procedimiento descrito se repetirá en múltiples ocasiones hasta que se alcance una condición de finalización.

En general los AG tienen como objetivo buscar una solución para lo cual mantienen un conjunto de soluciones que, como se ha comentado, se van mezclando de manera que se intenta favorecer la generación de nuevas y mejores soluciones. De esta forma se pretende conseguir una mejora en la bondad de los individuos que componen la población a lo largo de las generaciones. En el diseño de un buen Algoritmo Genético hay que tener en cuenta dos cuestiones clave, que son: **La diversidad de la población y la presión del proceso selectivo.**

La diversidad de la población es una característica que está relacionada con la similitud o diferenciación de los individuos. Una población diversificada implica individuos que muestran diferencias significativas, y por el contrario una población poco diversificada presenta individuos muy similares. La importancia de este concepto estriba en que si se utilizan individuos muy parecidos en el operador de cruce lo normal es que se generen nuevos individuos semejantes a sus progenitores. Si esto sucede a lo largo de muchas generaciones lo que ocurre es que el algoritmo estará analizando soluciones muy próximas unas a otras, y que probablemente pertenezcan a un óptimo local. La diversidad de la población es importante para evitar caer en óptimos locales. Aunque inicialmente la composición de la población es independiente del proceso de selección, a lo largo de la evolución del algoritmo ésta dependerá mucho de dicho proceso.

La presión del proceso selectivo puede ser entendida como la capacidad, que el proceso de selección de individuos que se realiza antes de aplicar los operadores genéticos, tiene para favorecer la elección de unos o de otros individuos. Normalmente, los AG están enfocados a favorecer la selección de los “buenos” individuos. Sin embargo, una fuerte presión implicaría que la selección se realizase sobre un pequeño grupo de individuos, una elite de la población. Una fuerte presión implicaría una rápida falta de diversidad, y una convergencia prematura en un óptimo local. No obstante, la falta de presión es decir, que la bondad de los individuos no sea muy relevante en el proceso de selección, impide que se realice un proceso de refinamiento y mejora de algunas de las soluciones más prometedoras, esta situación se denomina convergencia lenta.

Un Algoritmo Genético debe mantener el equilibrio de manera que no caiga en una convergencia prematura ni en una convergencia lenta.

El empleo de Algoritmos Genéticos implica el uso de una serie de términos específicos que es necesario comentar. Algunos de los más relevantes son:

- **Cromosoma:** Representa la solución a un problema dado. Está compuesto por un conjunto de “genes” dispuestos en serie. También es conocido por “individuo” o “genotipo”.

- **Gen:** Representa la unidad de información indivisible de un cromosoma o individuo.
- **Alelo:** Es el nivel de información que representa el valor que puede tomar un determinado gen.
- **Población:** Es como se denomina a un conjunto de cromosomas o individuos.
- **Representación Genética:** Es la forma de codificar las soluciones, es decir, de representar los cromosomas y elementos dependientes de estos (genes, alelos, etc.).
- **Fenotipo:** Es un concepto relacionado con el significado de los individuos. En el caso de este trabajo, sería equivalente al valor del makespan que se obtiene con cada solución.
- **Valor de Adecuación o Fitness:** Es una medida de lo bueno o malo que resulta un individuo en relación con el resto de individuos. El fitness está muy relacionado con la función objetivo pero no tiene que coincidir necesariamente con ésta.
- **Generación:** Este término se refiere a la composición de una población en un instante dado.

Una característica importante de los Algoritmos Genéticos es que mientras que la mayoría de las metaheurísticas generan una única solución en cada iteración, los AG trabajan sobre un conjunto de soluciones (población), alterando la población en cada iteración. Por lo general, son algoritmos neutrales al contexto, y no consideran ni explotan la información del problema dentro del proceso de búsqueda. La generación de nuevas soluciones es aleatoria por naturaleza. Los AG utilizan una elección al azar para guiar la búsqueda, repitiendo el proceso durante un número relativamente largo de iteraciones para dar lugar a un número muy grande de soluciones.

Como ya se comentó en el capítulo 3, este tipo de algoritmo ha sido ampliamente utilizado para resolver problemas de Programación de la producción en múltiples tipos de taller y circunstancias. No todas las tentativas se han basado en la estructura tradicional del Algoritmo Genético, sino que se ha realizado diversas propuestas innovadoras como son:

- Algoritmo macroevolucionarios (Marin, J. & Sole, R. V. 1999).
- Algoritmos Genéticos innumes (Zhang, R. & Wu, C. 2009; Zuo, X. Q., Mo, H. W., & Wu, J. P. 2009).

- Algoritmos ortogonales Genéticos (Naderi, B., Zandieh, M., & Roshanaei, V. 2009; Zhong, W. C., Liu, J., & Jiao, L. C. 2005).
- Algoritmos microgenéticos (Coello, C. A. C. & Pulido, G. T. 2005; Zhou, Z. F. & Harris, K. D. M. 2008).

Sobre esta base conceptual se presenta una nueva propuesta que también realiza una serie de modificaciones sobre la base del Algoritmo Genético tradicional que permite la incorporación de los Sistemas Multiagente, y que a su vez, favorece mejoras en las prestaciones del AG aplicado a la Programación de la Producción en un Taller de Flujo Híbrido con Tiempo de Cambio de Partida Dependiente de la Secuencia.

8.3.2 Algoritmo SMAGA

La propuesta que a continuación se presenta tiene como antecedentes los trabajos de los profesores Weicai Zhong y Jing Liu de la Universidad de Xidian (R.P. China) y sus colaboradores. Dicho equipo tiene entre sus publicaciones un conjunto de comunicaciones y artículos enfocados a problemas de optimización en los que emplean técnicas basadas en la evolución de los sistemas naturales. Algunos ejemplos relevantes son (Liu, J., Zhong, W. C., & Hao, L. C. 2007; Liu, J., Zhong, W. C., & Jiao, L. C. 2006; Liu, J., Zhong, W. C., Jiao, L. C., & Liu, F. 2004; Zhong, W. C. et al. 2005; Zhong, W. C. et al. 2004). Entre éstos, fue su propuesta para afrontar problemas generales de optimización numérica (Zhong, W. C. et al. 2004), en la que los autores tienen como objetivo plantear un método que posibilite el establecimiento de los valores a los parámetros de múltiples funciones matemáticas conocidas, no necesariamente lineales, la que el autor de este trabajo ha considerado interesante como punto de partida de un nuevo algoritmo.

En (Zhong, W. C. et al. 2004) se presenta una arquitectura en la que se hibridan los Sistemas Multiagente y los Algoritmos Genéticos. En ella se utilizan los Sistemas Multiagente como potenciadores de ciertas características de los Algoritmos Genéticos. Por este motivo el enfoque general, es decir, la estructura del algoritmo, de la propuesta de Zhong, ha sido respetada casi en su totalidad. No obstante, **sobre la estructura general de la propuesta de Zhong se han incorporado una serie de novedades respecto del algoritmo original**. Se debe remarcar que el tipo de problema que en esta tesis se pretende abordar es muy diferente al abordado por Zhong, por lo que los procedimientos particulares que se implementan en (Zhong, W. C. et al. 2004) para llevar a cabo funciones como la mutación, el cruce, etc. han sido sustituidos por otros completamente diferentes.

En los apartados siguientes se realizará, en primer lugar, la identificación de las ventajas competitivas que los Sistemas Multiagente deben aportar, a continuación la descripción del algoritmo SMAGA. Durante la exposición de la propuesta se remarcarán cuales son los elementos originales, y cuales son las aportaciones realizadas. En apartados posteriores se realizará la parametrización de dicho algoritmo, y una vez ajustados los parámetros, se procederá a su comparación con otros métodos empleados para la resolución del mismo problema.

8.3.2.1 Papel del Sistema Multiagente en el Algoritmo

En este apartado se pretende identificar el papel que el SMA debe jugar en el algoritmo SMAGA de forma específica, y relacionar éste con las ventajas de carácter general que se subrayaron en el capítulo 4.

Como punto de partida, y manteniendo la idea general de aprovechar las características de los Sistemas Multiagente para dotar a los Algoritmos Genéticos de mayor proximidad a los sistemas naturales, se propone que SMAGA potencie:

- La generación de nuevos individuos basándose en una **competencia local** como ocurre en la naturaleza, y no global como se propone en la mayoría de los Algoritmos Genéticos.
- El **ajuste dinámico de algunos parámetros** en base a la capacidad de aprendizaje de los individuos.
- **Aplicación diferenciada de los operadores genéticos** mediante la diferenciación de los individuos.

En la naturaleza, la evolución de los individuos mediante las nuevas generaciones tiene una característica destacable, como es que las nuevas generaciones contienen múltiples individuos, y que estos se obtienen fruto de una competencia directa en el entorno de cada individuo. Esta forma de actuar no se suele ver reflejada en el procedimiento seguido por los Algoritmos Genéticos tradicionales donde se suele realizar una selección global de candidatos a partir de los cuales obtener uno o varios individuos nuevos. Por este motivo **se propone disponer de un sistema que obtenga en cada generación múltiples individuos, procedentes de diversos progenitores, cuya selección se realice en un ámbito reducido, su ámbito local.**

El alcance del material genético a partir del cual se crearán nuevos individuos será global, es decir, toda la población podrá disponer de aquella información genética que se considere

relevante. **La diferencia fundamental de la propuesta que se plantea debe residir en la forma de hacer llegar dicho material genético.** Para que la información genética de un individuo ubicado en un entorno llegue a otro ubicado en otro entorno diferente, ésta debe ir pasando poco a poco entre origen y destino a través de ámbitos locales contiguos. Mientras que con un sistema basado en una competencia global podría producirse la transferencia genética de forma más directa. La propuesta basada en competencia local puede provocar, a su vez, que se examinen de forma más exhaustiva soluciones próximas a una dada. Obviamente, como desventaja fundamental de este enfoque se encuentra la posibilidad de que se produzca una homogenización de la población, y una convergencia rápida a un óptimo local, para lo que hay que poner medios que eviten dicha situación.

Por lo que se refiere a la posibilidad de ajustar los valores de algunos parámetros, hay que indicar que esta característica se ha vinculado a la capacidad que tienen ciertos sistemas de reconocer que durante su funcionamiento las condiciones de trabajo pueden cambiar. Y que en algunos casos, es mejor modificar la configuración inicial con el objeto de reconducir al sistema, y obtener mejores resultados finales. En este sentido **se ha incorporado un sencillo sistema de observación que permite aprender sobre los resultados obtenidos en ciertas funciones y modificar los valores de los parámetros que las regulan si es necesario.** En concreto, se trata de parámetros que tienen que ver con el tipo de algoritmo aplicado en la generación de una nueva solución, el tipo de operador de cruce que se aplica, y el valor de la probabilidad de cruce que se aplica. El conjunto de parámetros ajustable de forma dinámica mediante el aprendizaje podría extenderse a otras variables pero se ha considerado, mediante experimentos previos, que los parámetros indicados son los más indicados.

En lo referente a la gestión diferenciada de los individuos, **se pretende que no exista un comportamiento único para todos los individuos del sistema.** Esto se produce habitualmente en entornos cuyos elementos son de por sí diferentes. Por ejemplo dentro de los Algoritmos Genéticos se puede asumir que no toda la información genética evoluciona con el mismo patrón de comportamiento, como ocurre con los individuos reales. En este caso, la propuesta se refiere al comportamiento, es decir al hecho de que materiales genéticos idénticos pueden ser tratados de forma diferenciada en circunstancias iguales o diferentes. La incorporación de los Sistemas Multiagente facilita enormemente el diseño e implementación de roles que facilitan esta forma de trabajar. No obstante, en este caso la diferenciación se extiende también, y fundamentalmente, a elementos cuya funcionalidad es la misma. Como anticipo de la arquitectura del sistema propuesto, se puede avanzar que un individuo genético se corresponde con una solución, y que ésta se corresponde con un Agente. En este contexto se ha aplicado la diferenciación posibilitando que cada Agente contemple una parte de su aprendizaje de forma individualizada, y establezca de forma independiente a la del resto de Agentes (soluciones) los valores que correspondan a la aplicación del operador, y la

probabilidad de cruce. Se puede decir que la capacidad de aprendizaje se aprovecha para diferenciar a los individuos y alcanzar la cierta independencia en los Agentes.

Ventajas Generales	Características Deseadas	Comentarios
Permite de forma natural la convivencia de diferentes roles (gestiones).	Diferenciación	Permite establecer individuos con diferentes funciones y facilita su ejecución en un entorno donde cada Agente actúa en base a su rol.
Permiten la obtención de valores adecuados, en su caso óptimos, de diversas medidas de forma simultánea y adaptándose al entorno cambiante	Aprendizaje y Diferenciación	Aunque no se busca la optimización de múltiples medidas, el proceso de aprendizaje permite que cada individuo se adapte a su nuevo estado (representado por una solución), explore sus posibilidades y se adapte a lo que es mejor en cada momento.
Representan de forma natural la evolución temporal de los acontecimientos.	Competencia Local y Diferenciación	La evolución de produce en un entorno reducido, tal y como ocurre en la naturaleza. Esto supone que en una generación se han producido múltiples situaciones de competencia local. A su vez cada individuo actúa en función de sus circunstancias (entorno).
Son Sistemas idóneos para entornos con Incertidumbre	Aprendizaje	Son capaces de identificar diversos estados y actuar en consecuencia.
Facilitan el Diseño y la Implementación de una Solución	Diferenciación.	Los SMA permiten implementar mediante un único código individuos idénticos que posteriormente evolucionan de forma independiente.

Tabla 8-9. Relación entre las ventajas generales y las características deseadas (elaboración propia).

En la Tabla 8-9 se establece una relación entre las características que se desean incorporar (evolución, aprendizaje y diferenciación) al algoritmo mediante los Sistema Multiagente con las ventajas funcionales generales de los mismos destacadas en el capítulo 4.

Son por lo tanto, ***la diferenciación, el aprendizaje y la evolución local*** la esencia básica que los **Sistemas Multiagente deben aportar al algoritmo SMAGA**. No será la única aportación que realicen. Debido a la forma de enfocar el desarrollo de la propuesta se podrán dar a conocer algunas características secundarias del proceso de hibridación. Estas consecuencias, positivas y negativas, se resumirán una vez determinada la estructura del algoritmo implementado, y analizada su efectividad.

8.3.2.2 Funcionamiento General del Algoritmo

El objeto de este apartado es dar una visión general del funcionamiento del algoritmo SMAGA. De esta forma, y basándose en un lenguaje natural, el lector podrá tener una idea de conjunto de la propuesta que se plantea.

En la Figura 8-7 se muestra un esquema general del algoritmo SMAGA. En el se muestran los pasos más relevantes en la ejecución del mismo. Si se compara con el esquema general de un Algoritmo Genético, ver Figura 8-2, con esquema propuesto para SMAGA se puede observar que existe un elemento funcional innovador destacado en SMAGA, como es el aprendizaje genético^{iv}. Otro aspecto, diferenciador aunque menos novedoso, es el establecimiento de una condición para la regeneración de la población si se cumple cierta restricción. No obstante, existe multitud de apreciaciones en lo referente a la aplicación del resto de elementos funcionales que se irán desvelando posteriormente.

Se puede decir, al hilo de la figura anterior, que existen cuatro grandes actividades funcionales destacables y un tipo de toma de decisión fundamental en la ejecución del algoritmo SMAGA. Las actividades fundamentales son: La generación de la población inicial, la selección de los progenitores para la aplicación del operador de cruce, y su propia aplicación, la aplicación del operador de mutación y la aplicación de un proceso de aprendizaje genético. El tipo de decisión más importante que se debe tomar durante la ejecución del algoritmo es la de establecer el relevo generacional, es decir, el cambio de soluciones existentes en la población por otras nuevas. Esta decisión se debe tomar después de la aplicación del operador de cruce, en el que se puede llegar a obtener un nuevo individuo, la aplicación del operador de mutación, en el que se puede producir la modificación de un individuo existente, y la

^{iv} En el algoritmo SMAGA existen varios procesos de aprendizaje diferentes al aprendizaje genético propuesto por (Zhong, W. C. et al. 2004).

aplicación del proceso de aprendizaje genético, en el que de nuevo, se puede llegar a obtener un nuevo individuo.

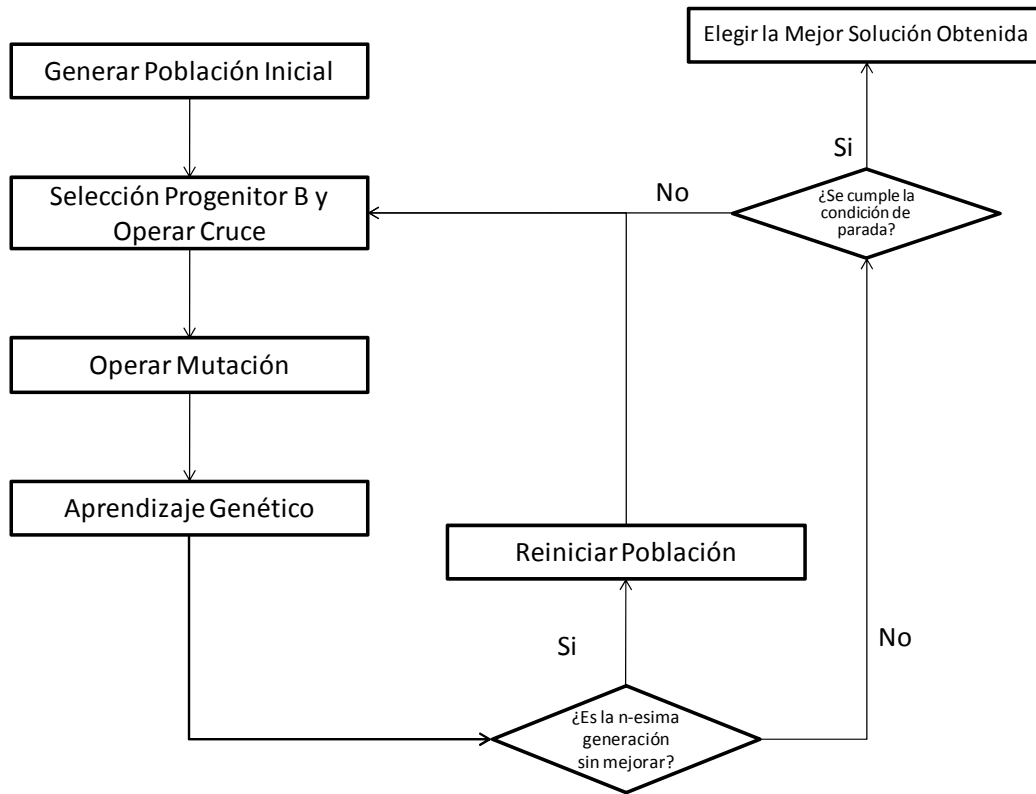


Figura 8-7. Esquema general de los procesos correspondientes al algoritmo SMAGA (basado en (Zhong, W. C. et al. 2004)).

Como se puede deducir de lo comentado en el párrafo anterior, en el algoritmo SMAGA se ha decidido que en aquellos casos en los que se aplique un operador de genético directamente (cruce o mutación) o indirectamente (aprendizaje genético), sólo se obtendrá un individuo nuevo o modificado. En general, en los Algoritmos Genéticos es normal que los operadores de cruce ofrezcan la posibilidad de generar más de un nuevo individuo, y lo mismo puede ocurrir con los operadores de mutación. No obstante, en este algoritmo se reduce a la generación de un solo individuo. En caso, de que el operador, por su naturaleza, ofrezca más de una solución el conjunto se reducirá a una, eliminando los individuos que peor “fitness” posean.

Los diversos procesos (operadores de cruce, mutación o aprendizaje) se aplican siguiendo el esquema propuesto en la Figura 8-8. Esto implica que cuando se aplica un operador se hace de forma secuencial a todos los elementos de la población, y hasta que no se ha finalizado su aplicación en todos los individuos no se ejecuta la siguiente operación.

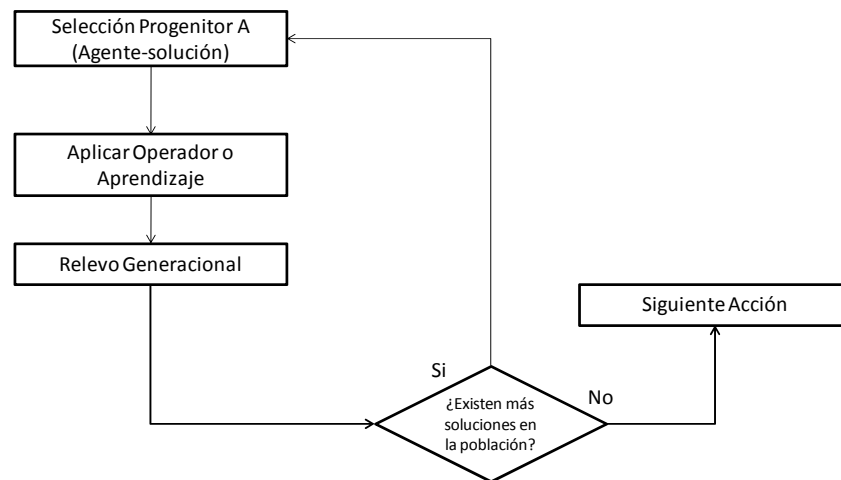


Figura 8-8. Ampliación de las actividades de los operadores correspondientes al algoritmo SMAGA (elaboración propia).

A lo largo de la descripción del algoritmo se irán proponiendo diversas alternativas de funcionamiento a partir de la configuración que se realice en algunos parámetros. Dichas alternativas serán analizadas como factores en el análisis que se realizará para el ajuste paramétrico del algoritmo, cada uno de los cuales podrá contar con varios niveles. Hay que resaltar que en la configuración de un Algoritmo Genético pueden influir un conjunto de factores que puede llegar a ser muy elevado. Con la premisa de conseguir un algoritmo competitivo, pero con la necesidad de realizar un análisis abordable, se ha decidido reducir los factores sujetos a estudio fijando inicialmente aquellas opciones que han alcanzado buenos resultados en experimentos preliminares. En este sentido se ha procedido a profundizar aquellas opciones fundamentalmente asociadas a las características peculiares de la propuesta que se realiza.

8.3.2.2.1. Representación de los Individuos

Un Algoritmo Genético trabaja sobre los individuos o cromosomas, que son una representación o codificación de las soluciones del problema, no necesariamente las soluciones en sí. Una solución para un Programa de un Taller de Flujo Híbrido, con o sin Tiempos de Cambio de Partida, viene dada por la Programación de todas las tareas, es decir, necesitaríamos saber los instantes de comienzo y finalización de cada tarea, así como la identificación de la máquina sobre la que se realiza. En definitiva, una solución debe ofrecer información tanto sobre la secuencia, como la asignación, y los tiempos de inicio y fin de las operaciones.

Por lo que se refiere a los problemas de secuenciación y asignación, en la literatura se pueden encontrar aproximaciones en donde en la fase de resolución se han abordado ambos problemas, y aproximaciones en donde en la fase de resolución se ha afrontado el problema de secuenciación siendo en la fase de valoración de los resultados donde se ha tenido en cuenta el problema de asignación. Según las referencias analizadas en el capítulo 3 se ha podido constatar como la aproximación de ambos problemas de forma conjunta suele ser compleja, y en general, no ofrece buenos resultados. Por este motivo, se ha preferido desde el inicio considerar la segunda aproximación, abordando los dos problemas de forma diferenciada.

Al plantear de forma diferenciada el problema de secuenciación del problema de asignación, se ha considerado adecuado que la representación de la solución esté fuertemente vinculada a la secuencia. Tomando la secuencia como elemento básico de información, y mediante la aplicación de un simple procedimiento, se realiza la asignación a las máquinas, y la temporización de los instantes de inicio y fin de cada trabajo en las diferentes etapas y máquinas requeridas. Con el planteamiento propuesto, la información necesaria para representar una solución para el caso concreto del Taller de Flujo Híbrido es sencilla. Es suficiente con que la representación genética de un individuo se realice mediante una secuencia de "n" genes, uno por cada trabajo. En dicha secuencia el orden que ocupa cada trabajo es relevante. Este tipo de representación se denomina ordinal. Se puede ver un ejemplo para veinte trabajos en Figura 8-9. En esta figura se observa como existen 20 genes, que se corresponden con cada posición de la secuencia, y en cada gen se almacena el identificador de cada uno de los 20 trabajos disponibles. Según esta representación el trabajo 14 sería el primero de la secuencia, el trabajo 3 el segundo, y así sucesivamente hasta el trabajo 12 que sería el vigésimo y último.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
14	3	9	7	8	19	11	6	13	16	15	10	5	18	17	2	1	4	0	12

Figura 8-9. Representación genética ordinal para un caso de 20 trabajos (elaboración propia).

Es destacable como la configuración del taller, es decir el número de etapas y máquinas, no influye en la codificación de la solución. Será el proceso de asignación el que tenga en cuenta la configuración del taller. El procedimiento que se seguirá para la asignación consistirá en elegir los trabajos de la secuencia configurada, de uno en uno y en el orden determinado previamente, y establecer la máquina con la que se realizará cada operación por orden de precedencia. La máquina seleccionada en cada etapa debe ser aquella que permita finalizar lo antes posible la operación correspondiente en dicha etapa. Una vez completadas todas las operaciones de un trabajo se elige el siguiente de la secuencia hasta finalizar.

Como muestra práctica de la transformación de un individuo, y por lo tanto, una solución expresada como secuencia en un Programa Productivo, se ofrece el diagrama Gantt de la Figura 8-10. En dicha figura se puede observar como la secuencia correspondiente al individuo de la Figura 8-9 se repite en cada una de las etapas de la que consta el taller que sirve de ejemplo. Existen tiempos de proceso indicados en color y tiempos de cambio de partida, entre cada trabajo, indicados en gris claro. Se puede decir que el cálculo del diagrama Gantt es el proceso de decodificación o cálculo del fenotipo, en este caso el valor del makespan obtenido. En el ejemplo de la Figura 8-10 el valor del fenotipo o makespan es de 1.436 uds.

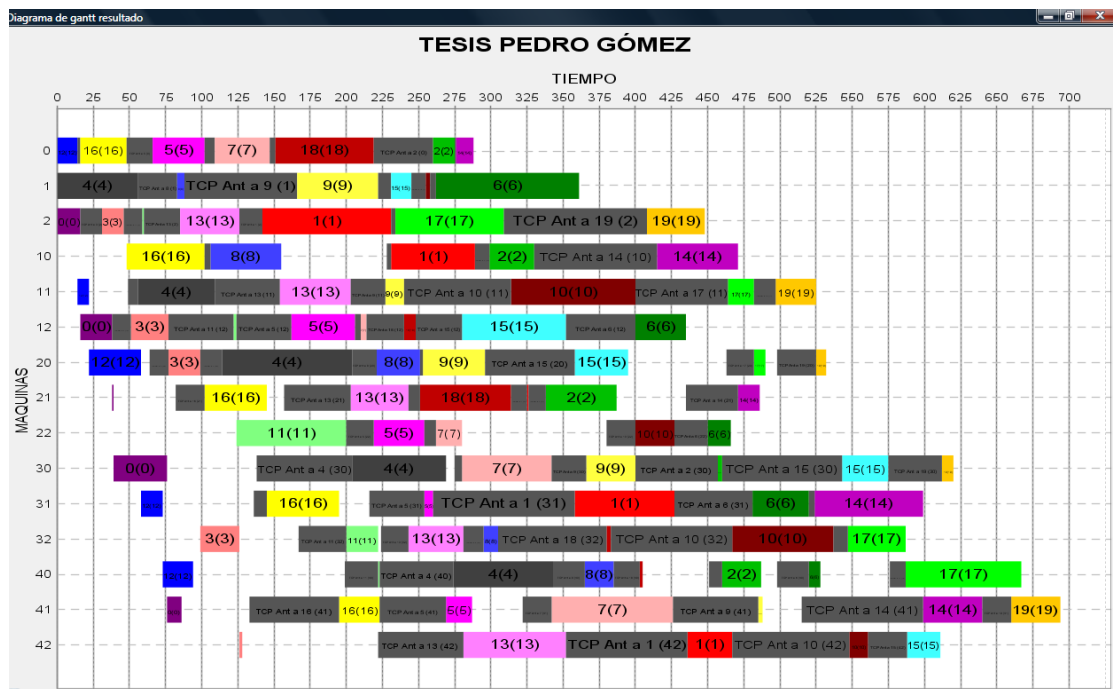


Figura 8-10. Diagrama Gantt^v para una la representación genética del individuo de la Figura 8-9 (elaboración propia).

En el ejemplo anterior existen cuatro etapas, cada una de ellas con tres máquinas (etapa 0: máquinas 0, 1 y 2; etapa 1: máquinas 10, 11 y 12; etapa 2: máquinas 20, 21 y 22; etapa 3: máquinas 30, 31 y 32).

^v Utilidad para la representación de Diagramas Gantt obtenida gracias a la colaboración desinteresada de José Ángel y Francisco Gómez Gómez programadores pertenecientes al Grupo de Investigación ROGLE.

Por último, se debe remarca que la representación genética propuesta no se basa en la propuesta de (Zhong, W. C. et al. 2004), debido a que el tipo de problema que dichos autores plantean solucionar es sustancialmente diferente del que se plantea en este trabajo. La elección de la representación ordinaria se ha fundamentado en las experiencias manifestadas por diversos autores en la literatura revisada.

8.3.2.2.2. Identificación de los Agentes y el Sistema Multiagente

En el caso del algoritmo SMAGA se ha seguido el criterio establecido en (Zhong, W. C. et al. 2004) en cuanto a la identificación de una solución con un Agente. Sin embargo, es relevante indicar que en la propuesta SMAGA el Agente contiene a la solución. Este matiz implica que el Agente se generará en la fase de inicialización del algoritmo, y que éste será caracterizado por un conjunto de parámetros que determinarán su forma de actuar durante la ejecución del algoritmo, dejando de existir cuando el algoritmo finalice. Sin embargo, la vida de las soluciones será más corta, siendo normal que un Agente contenga a lo largo de su vida a varias soluciones diferentes. En todo caso, en un instante dado un Agente contendrá sólo a una solución, y una solución sólo existirá contenida en un Agente. La idea subyacente es que el Agente sea quien gestione a la solución que contiene aplicando sobre ella los diferentes operadores genéticos, y dando lugar, si procede, a una nueva solución. En este sentido, el Agente es una entidad activa mientras que la solución es pasiva, simplemente una información gestionada por el Agente.

Desde el punto de vista, del diseño del Sistema Multiagente, se ha considerado que deben existir dos tipos de Agentes, tal y como se muestra en la Figura 8-11. Estos son:

- **El Agente-Solución:** Es el Agente que contiene a una solución, y la gestiona.
- **El Agente-Gestor:** Es el Agente responsable de coordinar a la población de Agentes-Solución.

En la Figura 8-11 se puede ver una representación del Sistema Multiagente, donde se destaca que la comunicación se produce entre los Agentes-Solución, y entre Agentes-solución y Agente-Gestor. Si bien, la comunicación entre Agentes-Solución es limitada como posteriormente se comentará.

Sin entrar en el diseño software del Sistema Multiagente se ha considerado que es necesario avanzar una visión simplificada de la configuración básica de la arquitectura del Sistema Multiagente, con el fin de dar una idea de conjunto del mismo, y facilitar así la comprensión del propio algoritmo.

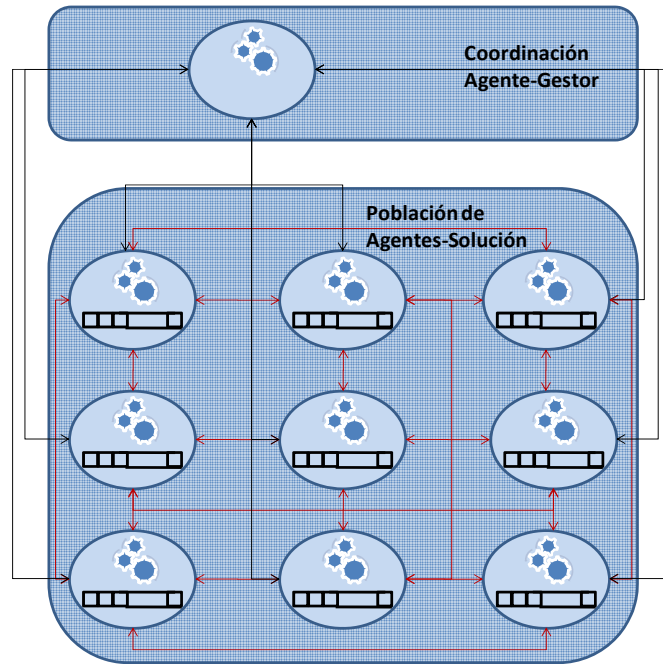


Figura 8-11. Representación del Sistema Multiagente (elaboración propia).

Por lo que se refiere al número y configuración de los agentes-solución, tal y como se establece en (Zhong, W. C. et al. 2004), se ha definido una población en la cual los agentes están dispuestos en forma de rejilla o matriz base de tamaño $L \times L$, como se intuye en la Figura 8-11. En esta matriz base los agentes solución se comunican entre sí, pero únicamente con sus vecinos. Se considera que un agente-solución situado en la posición (i,j) de la matriz base, y representado por $L_{i,j}$, tiene por vecinos al conjunto formado por $\{(i', j), (i'', j), (i, j'), (i, j'')\}$, tal que:

$$i' = \begin{cases} i-1 & \text{si } i \neq 1 \\ L & \text{si } i = 1 \end{cases} \quad i'' = \begin{cases} i+1 & \text{si } i \neq L \\ 1 & \text{si } i = L \end{cases} \quad j' = \begin{cases} j-1 & \text{si } j \neq 1 \\ L & \text{si } j = 1 \end{cases} \quad j'' = \begin{cases} j+1 & \text{si } j \neq L \\ 1 & \text{si } j = L \end{cases}$$

En la Figura 8-12 se puede ver una representación de la matriz de Agentes-Solución y la conectividad de los mismos.

Por conectividad entre los Agentes-Solución se debe entender la posibilidad de establecer algún tipo de relación entre ellos. Por lo tanto, según el esquema propuesto por (Zhong, W. C. et al. 2004) cada Agente-Solución se puede relacionar con cuatro vecinos (superior, inferior, derecha e izquierda). En el algoritmo SMAGA la aplicación de la conectividad se reduce al ámbito del operador de cruce.

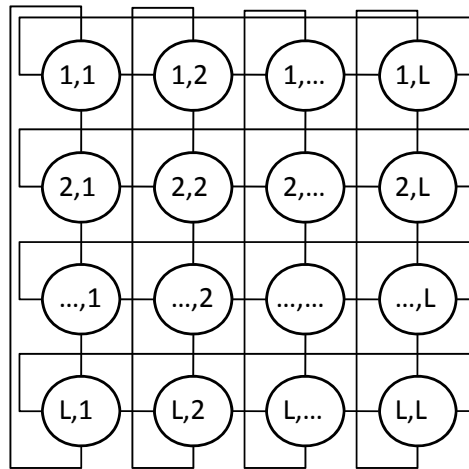


Figura 8-12. Modelo de Agentes-Solución en rejilla o matriz base (elaboración propia).

Esta disposición en rejilla, junto con el tipo de conexión permitido, posibilita que cada Agente-Solución trabaje en un entorno reducido, y que sus relaciones puedan ser consideradas locales. Es decir, esta arquitectura es una forma de implementar la competencia local que se ha identificado como una de las características básicas del sistema propuesto.

8.3.2.2.3. Generación de la Población

Dos son los aspectos que deben definirse en relación a la población: Su tamaño y el método de generación de la misma.

En primer lugar se abordará cual debe ser el tamaño de la población. Aunque en general, el tamaño de una población no tiene porque ser constante a lo largo de la ejecución de un Algoritmo Genético en este caso se ha decidido que el número de individuos se mantenga invariable. Dicho tamaño, en este caso invariable durante la ejecución del algoritmo, condiciona la intensidad de las interacciones, y por lo tanto de la exploración de soluciones. Suele ser habitual que se identifiquen las poblaciones con numerosos individuos con entornos de fuerte competencia y gran presión genética. Así mismo, es frecuente que se identifiquen las poblaciones reducidas con entornos de competencia reducida. En el caso del algoritmo SMAGA, se ha seguido la propuesta de (Zhong, W. C. et al. 2004), de forma que existen tantas soluciones como elementos existen en la matriz, en este caso Agentes-Solución. Y en este sentido, la definición de la matriz implica el establecimiento de un determinado tamaño de población. En (Zhong, W. C. et al. 2004) se recomienda un tamaño de matriz entre 5x5 y 10x10. En este caso, y después de un estudio preliminar, se ha decidido establecer el tamaño de matriz en 6x6. Este tamaño ha demostrado proporcionar un buen equilibrio entre resultados

obtenidos y el coste computacional. El estudio preliminar se ha realizado mediante experimentos “no formales”, con un conjunto reducido de problemas, y no se ha podido observar ningún tipo de mejora significativa con el incremento del tamaño de la matriz. En concreto se ha analizado la opción del tamaño 8x8 y 9x9, además de la opción elegida.

En relación con el tamaño de la matriz base se debe considerar que cada uno de los Agentes-Solución definidos en la matriz tendrá la oportunidad de acceder a un proceso de aprendizaje, que como posteriormente se explicará, implicará la ejecución de una versión reducida del propio algoritmo SMAGA. Como se puede ver en la Figura 8-13 por cada individuo de la matriz base existirá otra nueva matriz cuadrada, que en este caso se denominará de aprendizaje. La dimensión de la matriz de aprendizaje es $sL \times sL$, y suele ser menor al de la matriz base. Por lo tanto, el incremento unitario en la matriz base supone un incremento computacional sustancial del esfuerzo destinado a tratar los nuevos individuos.

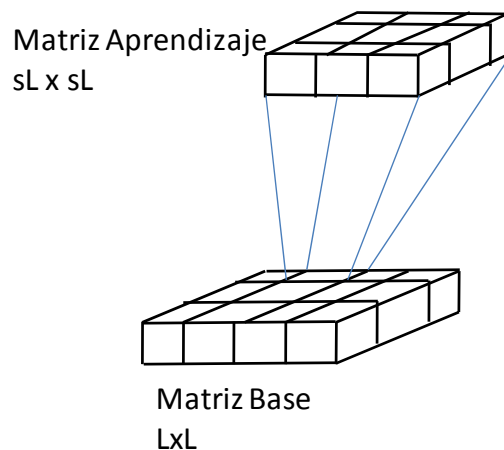


Figura 8-13. Relación entre Matriz Base y Matriz de Aprendizaje (elaboración propia).

En segundo lugar, y por lo que se refiere a la generación de las soluciones, se ha intentado mantener la filosofía que tiene como objetivo el equilibrio entre el coste computacional y la calidad de las soluciones obtenidas. Hay que tener en cuenta que los procedimientos de generación de nuevas soluciones no sólo serán empleados en la fase de inicialización del algoritmo, sino también en un procedimiento adicional para la regeneración de la población a gran escala. Este procedimiento de regeneración se utilizará en caso de que durante un número determinado de generaciones no se mejore el valor de la mejor solución. Este último procedimiento se comentará con detalle posteriormente. En todo caso, desde el punto de vista del impacto computacional, lo relevante es que el procedimiento se empleará en múltiples ocasiones, ya sea para la generación inicial o regeneración posterior de la población.

Para la obtención de soluciones se ha escogido un procedimiento de generación de soluciones basado en un algoritmo voraz^{vi}. Se han valorado otros tres métodos alternativos como son: generación aleatoria de soluciones, el algoritmo NEH propuesto por (Nawaz, M., Enscore, E. E., & Ham, I. 1983) y el algoritmo NEH modificado propuesto por (Ruiz, R. 2003). Sin embargo, se ha considerado que en el balance entre calidad de las soluciones y coste computacional, la propuesta escogida se sitúa entre las más competitivas.

```
// Algoritmo voraz Taller de Flujo
Conjunto TNS={1...N}; //Todos los trabajos
Secuencias[N,N] //Contendrá N secuencias con N trabajos

Para i=1 hasta N {
    Secuencias[i,1]=i;
    Extraer_de_TNS(i); //Elimina al trabajo i del conjunto TNS
    orden=2;
    Mientras (orden<=N) {
        Mejor_TCP= Infinito //Contiene el mejor tiempo de cambio de partida obtenido.
        Siguiente_trabajo=nulo; //Contiene al trabajo candidato a ser el siguiente de una secuencia
        Para j=1 hasta N {
            Si (Pertenece_a_TNS(j)) { //Es cierto si el trabajo j pertenece al conjunto TNS
                Si (TCP(Secuencias[i,orden-1],j)<Mejor_TCP) {
                    Mejor_TCP= TCP(Secuencias[i,orden-1],j)
                    Siguiente_trabajo=j;
                }
            }
        }
        Secuencias[i,orden]=Siguiente_trabajo
        orden=orden+1
        Extraer_de_TNS(Siguiente_trabajo)
    }
}
Mejor_secuencia=Calcula_mejor_makespan(Secuencias) //A partir de las secuencias obtenidas calcula
//Makespan de cada una e identifica a la mejor
```

Ilustración 8-1. Pseudocódigo correspondiente al algoritmo voraz para el Taller de Flujo (elaboración propia).

^{vi} Un algoritmo voraz se caracteriza por intentar solucionar un problema empleando un procedimiento que consiste en construir una solución paso a paso, donde cada paso supone añadir un nuevo elemento, con la esperanza de que cada decisión parcial conduzca a una solución óptima al final del proceso constructivo.

Algoritmo voraz Empleado

Inicialmente se define el algoritmo voraz aplicado a un taller de flujo. El método consiste en el establecimiento de dos conjuntos, trabajos no secuenciados (TNS) y trabajos secuenciados (TS), y en la aplicación de un criterio de selección para extraer los trabajos del conjunto TNS y transferirlos al conjunto TS. El orden en que se realiza el transvase del conjunto TNS al conjunto TS coincide con el orden asignado al trabajo en la secuencia. El funcionamiento se muestra de forma más detallada en la Ilustración 8-1.

El procedimiento se inicia incluyendo un trabajo cualquiera en el conjunto TS, y dejando el resto en el conjunto TNS. Dicho trabajo será considerado como el primero de la secuencia. A partir de esta estado el procedimiento de selección tiene como objetivo explorar todas las combinaciones entre el último trabajo seleccionado (t_i) y el conjunto de trabajos no secuenciados aún. Calculando para cada combinación el tiempo de cambio de partida correspondiente. Resultará elegido el trabajo con menor tiempo de cambio de partida. El trabajo seleccionado se eliminará del conjunto TNS y se incluirá en el conjunto de TS, considerándose el trabajo t_{i+1} , y ocupando el puesto $i+1$ en la secuencia. El proceso de selección se realizará hasta que el conjunto TNS esté vacío. En dicho instante el conjunto TS incluirá a todos los trabajos, que se desea secuenciar, y el orden en el que han sido elegidos definirá la secuencia.

Dado que la secuencia obtenida depende exclusivamente del trabajo seleccionado en primer lugar el algoritmo prescribe que se realice el procedimiento preliminar iniciando la secuencia con todos y cada uno de los trabajos existentes. La secuencia elegida finalmente será aquella que presente menor makespan entre todas las secuencias generadas.

A partir del algoritmo voraz planteado se realiza una modificación con el fin de posibilitar su aplicación a un Taller de Flujo Híbrido. Dicha modificación consiste en aplicar el algoritmo a una máquina ubicada en una etapa elegida aleatoriamente. En concreto se ha decidido que por simplicidad la máquina elegida sea siempre la primera máquina que se identificó en el momento de recoger la información de cada etapa. Existen otras posibles modificaciones del algoritmo, pero todas las opciones barajadas implicaban un incremento importante del esfuerzo computacional con respecto a la propuesta original. La modificación que se indica mantiene el coste computacional original.

Una vez conocido el procedimiento propuesto para la obtención de soluciones, tanto en la población inicial como en sucesivas regeneraciones de ésta, se desea llamar la atención sobre lo siguiente:

- El proceso de generación de soluciones se realiza de forma independiente en cada Agente-Solución. A su vez, cada Agente-Solución aplicará el algoritmo voraz tomando como referencia una etapa, probablemente diferente, en cada ocasión. Por lo tanto, durante la ejecución del algoritmo SMAGA, se podría disponer de una muestra de cuales han sido los resultados de la ejecución del algoritmo voraz en las diferentes etapas, y por los diferentes Agentes-Solución. Dicha muestra puede ser empleada para tomar decisiones más acertadas que una simple elección aleatoria.
- El resultado que se obtiene de la ejecución del algoritmo voraz es una solución, que en ningún caso es dependiente de la herencia que pudiera portar el Agente-Solución que lo ejecuta, como por ejemplo la secuencia anterior, o sus vecinos. El resultado depende básicamente de los tiempos de cambio de partida que se establecieron en la máquina elegida, y por lo tanto, son dependientes de la etapa seleccionada e independientes del Agente-Solución que lo ejecuta.
- El algoritmo voraz basa la secuencia generada en los tiempos de cambio de partida, buscando establecer la secuencia que acumule menor cantidad de los mismos. En este sentido, su aplicación puede ser muy interesante realizarla sobre una etapa que suponga un claro cuello de botella en cuanto a los tiempos de cambio de partida.

Analizando las reflexiones que se han destacado anteriormente se ha llegado a la conclusión de que puede ser útil establecer un proceso de aprendizaje que permita orientar al Agente-Solución en la aplicación del algoritmo de generación de soluciones. Se trata fundamentalmente de un proceso de aprendizaje colaborativo en el que participan todos los Agentes-Solución, y cuyos resultados sean compartidos. El objetivo del proceso de aprendizaje es detectar si existe una etapa con tiempos de cambio de partida que puedan implicar una situación de cuello de botella para que esta sea elegida como referencia en la ejecución del algoritmo voraz.

Se permitirá que el aprendizaje sea una opción parametrizable por el usuario. De acuerdo con esta posibilidad se ha configurado un parámetro denominado “**Generador Dinámico**” que puede tomar dos valores: activado o desactivado. Si está activado se posibilita el aprendizaje, y si está desactivado se trabaja sin aprendizaje. La configuración de los diferentes parámetros que se identifiquen durante la descripción del algoritmo SMAGA se realizará durante el proceso de ajuste paramétrico que se describirá en un apartado posterior.

En caso de permitir el aprendizaje se habilitará a todos los Agentes-Solución para que estos recojan información sobre el comportamiento del algoritmo voraz en las distintas etapas, y

para que esta información sea tratada por el Agente-Gestor, y puesta a disposición de todos los Agentes-Solución de manera que se comparta la información obtenida.

El diseño que se realizó en relación con el proceso de aprendizaje se muestra en forma de pseudocódigo en la Ilustración 8-2 y la Ilustración 8-3. Este consiste en establecer una etapa del taller de Flujo Híbrido como dominante. Inicialmente esta etapa dominante se designará de forma aleatoria. Cada vez que un Agente-Solución aplique el algoritmo voraz realizará un sorteo basado en una distribución uniforme entre 0 y 1. Si el resultado del sorteo es menor o igual al parámetro “**Cuota de Dominio**” establecido en 0.80, es decir en el 80% de las ocasiones, se aplica el algoritmo sobre la etapa dominante. En caso de que el sorteo proporcione un número superior a 0.8, el 20% de las opciones restantes, las posibilidades de aplicación se dividen a partes iguales entre las etapas no dominantes. Por ejemplo, si el taller tiene 6 etapas, y la etapa 3 es la dominante, se le asigna un 80% de opciones a la etapa 3 un 4% de posibilidades a las etapas 1, 2, 4, 5 y 6. Cada vez que se ejecuta el algoritmo en una etapa, cuando se obtiene la solución, se guarda el makespan vinculado a dicha solución, que es el que servirá para medir la calidad del resultado.

```
// Aprendizaje sobre la aplicación del Algoritmo Voraz

// En cada Agente
Si (Sorteo(0,1)<=0.8) etapa_seleccionada = Obten_Etapa_Dominante()
Sino { Etapa_seleccionada=Sorteo_entero(1, numero_etapas)
      } Mientras (etapa_seleccionada<> Etapa_Dominante)
Secuencia=AlgoritmoVoraz(etapa_seleccionada)
makespan=Calcula_makespan(Secuencia)
Guarda(makespan, etapa_seleccionada)
```

Ilustración 8-2. Pseudocódigo correspondiente a la aplicación del algoritmo original voraz en un Agente-Solución (elaboración propia).

El algoritmo voraz se ejecutará al inicio del algoritmo SMAGA, cuando se generen los Agentes-Solución, y se defina una solución vinculada a cada uno de ellos. También se ejecutará cuando se realice el procedimiento de regeneración de la población ordenado por el Agente-Gestor. La aplicación del algoritmo voraz es masiva, en el sentido que afecta a toda o gran parte de la población, y sincronizado, en tanto que es el Agente-Gestor de SMAGA quien decide el instante. Justo antes de la ejecución generalizada, que se produce en un procedimiento de regeneración de la población, es cuando el Agente-Gestor realiza un análisis de los datos obtenidos por la aplicación del algoritmo voraz en las diferentes etapas, y por los diferentes Agentes-Solución, y establece la etapa dominante para la siguiente tanda de ejecuciones (la de menor makespan promedio). En la fase inicialización de los Agentes-Solución, dado que aún no

existen datos, se aplicará el algoritmo voraz en una etapa seleccionada aleatoriamente por parte de cada uno de los Agentes-Solución.

```
// Aprendizaje sobre la aplicación del Algoritmo Voraz

// En el Gestor

Calcula_Etapa_Dominante()
Aplica_Regeneracion_Poblacion()
```

Ilustración 8-3. Pseudocódigo correspondiente a la aplicación del algoritmo voraz en el Agente- Gestor (elaboración propia).

Se puede considerar, por lo tanto, que existe una variable dinámica, que se ha denominado “Etapa Dominante”, que condiciona fuertemente la aplicación del algoritmo de generación de soluciones. Otra posibilidad, en relación con la aplicación del algoritmo voraz es la de no posibilitar que los Agentes aprendan, y que por lo tanto, cada Agente-Solución fije aleatoriamente la “Etapa Dominante”. Esta segunda forma de trabajar, “Sin Aprendizaje” consiste actuar en los casos de regeneración de la misma forma que actúan que los Agentes-Solución en su fase de inicialización.

8.3.2.2.4. Procedimiento de Selección

El proceso de selección de individuos es aquel que tiene como objetivo discriminar entre la población a los individuos que participarán en la producción de una nueva generación, es decir en la aplicación del operador de cruce. En este caso, la esencia del procedimiento se ha heredado de (Zhong, W. C. et al. 2004). El proceso se basa en que cada Agente-Solución pertenece a un entorno con el que puede interactuar, y que sólo en dicho entorno puede establecer la competencia para conseguir ser progenitor de un nuevo individuo. El proceso de selección está basado en el cumplimiento de las siguientes premisas:

- **Ser universal**, es decir, todos los individuos podrán ser candidatos a progenitor.
- **Con implementación local**. Cada individuo sólo competirá con los individuos de su entorno para alcanzar el derecho a ser progenitor.
- **Con selección múltiple**. En cada generación se escogen múltiples parejas de individuos para generar a un nuevo individuo. Por lo tanto, en una sola iteración es

posible incorporar a varias nuevas soluciones a la matriz. Se trata de nuevas soluciones que sustituirán a otras que hasta ese momento formaban parte de la población.

El procedimiento considerará que en cada iteración del algoritmo todos los Agentes-Solución son candidatos a convertirse en progenitores. El procedimiento básico se posiciona en la celda (1,1) de la matriz básica, y recorre todos los Agentes-Solución de izquierda a derecha, y de arriba a abajo. Para cada Agente-Solución situado en (i,j) se calcula el makespan (mk_{ij}), así como el makespan de todos sus vecinos. Se identifica el vecino con menor makespan ($mk_{im,jm}$). Si el mk_{ij} resulta ser menor que $mk_{im,jm}$ se consideran que no existen candidatos seleccionables. En otro caso, se selecciona la pareja ((i,j),(im,jm)) como progenitores de un nuevo individuo que podrá formar parte de la nueva generación. En la Ilustración 8-4 se expresa en pseudocódigo el procedimiento comentado.

```
// Procedimiento básico de selección de progenitores y Cruce
progenitorA=nulo;
progenitorB=nulo;

Para i=1 hasta L
  Para j=1 hasta L {
    Mkij=Calcular makespan (i,j)
    (Mkimjm, pos_i, pos_j)=Calcular menor_makespan (Vecinos(i,j))
    Si (Mkij>Mkimjm) {
      progenitorA=agente(i,j)
      progenitorB=agente(pos_i, pos_j)
    }
    Si ((progenitorA<>nulo) y (progenitorB<>nulo) Cruce (progenitorA, progenitorB)
  }
}
```

Ilustración 8-4. Pseudocódigo correspondiente al procedimiento básico de selección y cruce (elaboración propia).

Como se ha podido observar la selección se realiza mediante interrogación directa a todos los individuos, uno a uno, dando así la opción a que cada uno de ellos pueda ser uno de los dos progenitores necesarios para realizar el cruce. En el pseudocódigo de la Ilustración 8-4, se ha denominado “progenitor A” al Agente-Solución actual, es decir (i,j). El otro progenitor, el denominado “progenitor B”, se elije mediante competencia como ya se ha dicho. Para la elección del “progenitor B” se debe de tener en cuenta la calidad de los individuos. La competencia en base a la calidad de los individuos se realiza exclusivamente entre los cuatro vecinos del “progenitor A”, es decir por competencia local. Esta competencia local es mucho más “relajada” que la competencia global dando así la oportunidad de ser progenitores a individuos de calidad media.

Este procedimiento de selección permite que las buenas soluciones locales puedan ser empleadas como material genético en todos sus vecinos, y mediante el paso de las generaciones, que dichas soluciones se distribuyan a toda la población en alguna medida. Una de las diferencias que se espera en el comportamiento de esta propuesta es el relacionado con la composición de la población. Se espera que la composición de la población, por lo que a su calidad se refiere, sea mucho más homogénea que con procedimientos de selección basados en competencia global. Obviamente, este comportamiento de la población sugiere la puesta en marcha de una serie de medidas que eviten la convergencia prematura, y garanticen la existencia de un cierto grado de heterogeneidad en la población. Estas medidas se han introducido en el propio operador de cruce, y se explicarán posteriormente.

Un aspecto adicional a tener en cuenta es que si el procedimiento proporciona a todos los individuos la oportunidad de participar en la generación de una nueva solución esto pueda provocar dos situaciones no deseadas:

- Que el ciclo de vida de una solución sea muy corto. Es decir, que se genere una nueva solución, y sin haber participado en ningún tipo de intercambio genético sea sustituida por otra nueva solución. Dicha solución habría supuesto un coste computacional que no es transformado en una aportación genética de ningún tipo.
- Si participan todos y cada uno de los individuos en un proceso de transformación genética el coste computacional puede ser muy elevado, y hacer poco operativo el algoritmo debido a su lentitud.

Por eso, a este procedimiento básico, se le ha introducido una modificación de elaboración propia que se puede considerar una aportación para el algoritmo SMAGA. Esta consiste en emplear una versión del tradicional factor de cruce, pero en este caso vinculado al ciclo de vida de las soluciones, que se ha denominado “**factor de riesgo**”. Antes de explicar en que consiste, y como se aplica el “factor de riesgo”, es importante recordar que un Agente-Solución puede modificar la solución que contiene dando lugar a una nueva. El “factor de riesgo” es una variable asignada al Agente, y no a la solución, aunque el valor del factor pueda evolucionar en función de la solución.

El “factor de riesgo” consiste en asignar una probabilidad a cada Agente-Solución para poder participar en el proceso de selección. Dicha probabilidad se establece inicialmente en cada Agente-Solución de forma aleatoria entre cuatro valores posibles (0.2, 0.4, 0.6 ó 0.8). Cada Agente-Solución tiene por lo tanto un valor en el factor que no tiene porque ser igual al de sus vecinos. La asignación de probabilidad se hace siguiendo una distribución uniforme entre los valores 1 y 4, ya que son cuatro los posibles valores iniciales. El “factor de riesgo” se emplea cada vez que se desea aplicar el operador genético de cruce, de forma que antes de proceder a

su aplicación de realiza un sorteo. Dicho sorteo consiste en obtener de forma aleatoria, mediante una distribución uniforme entre 0 y 1, un valor que si resulta ser menor que el “factor de riesgo” posibilita la aplicación del operador genético, y sino impide su aplicación.

El valor asignado al “factor de riesgo” puede evolucionar durante la ejecución del algoritmo SMAGA. El valor del “factor de riesgo” asignado a un Agente se reduce en 0.05 unidades en cada generación. Esto implica que en cada generación se reducen las probabilidades de ser seleccionado. No obstante, cuando un Agente modifica la solución el “factor de riesgo” se vuelve a situar en el valor original. Esto supone que cuando una solución es nueva la probabilidad de participar en operaciones es mayor que cuando lleva cierto tiempo en el sistema. En resumen, **el “factor de riesgo” es un parámetro que inicialmente se fija de forma aleatoria, para favorecer comportamientos heterogéneos en la población, y que posteriormente favorece a las soluciones que llevan menos tiempo en la población.** En todo caso, el usuario puede fijar el valor mínimo que pueda llegar a alcanzar el “factor de riesgo”, ya que un factor muy bajo supondría que un Agente determinado tiene muy pocas opciones de ser seleccionado. Dicho valor mínimo será establecido como un parámetro, denominado **“Riesgo Mínimo”**. Aunque el valor de “Riesgo Mínimo”, o cota mínima, es común para todos los Agentes-Solución, la evolución del valor del “factor de riesgo” sigue una evolución independiente en cada Agente-Solución.

En la Ilustración 8-5 se muestra el procedimiento modificado de selección de progenitores para la aplicación del operador de cruce, en el que se incluye la aplicación del “factor de riesgo”.

```
// Procedimiento modificado de Selección de progenitores y Cruce
progenitorA=nulo;
progenitorB=nulo;

Para i=1 hasta L
  Para j=1 hasta L {
    Si (aleatorio(0,1)<factor_Riesgo) {
      Mki=Calcular makespan (i,j)
      (Mkimjm, pos_i, pos_j)=Calcular menor_makespan (Vecinos(i,j))
      Si (Mki>Mkimjm) {
        progenitorA=agente(i,j)
        progenitorB=agente(pos_i, pos_j)
      }
      Si ((progenitorA<>nulo) y (progenitorB<>nulo) Cruce (progenitorA, progenitorB)
    }
  }
}
```

Ilustración 8-5. Pseudocódigo correspondiente al procedimiento modificado de selección y cruce (elaboración propia).

Será el Agente-Gestor el responsable de aplicar la reducción en el valor de la variable “factor de riesgo” en cada Agente-Solución, ya que se trata de una acción vinculada al concepto de evolución de la población, en concreto, es necesario conocer cuando se ha finalizado una generación. Se considera que se ha alcanzado el fin de una generación cuando todos los Agentes-Solución han explorado la posible aplicación de los operadores de cruce, mutación, y el aprendizaje genético.

Igual que ocurría con el proceso de aprendizaje empleado para la generación de soluciones se ha considerado que la reducción del valor del “factor de riesgo” puede estar vinculado a un parámetro denominado, en esta ocasión, “**Evolución del Factor de Riesgo**”, cuyos valores pueden ser activado o desactivado. En caso de estar activado se permite la evolución, en caso contrario no existe dicha evolución, y por lo tanto el parámetro “Riesgo Mínimo” no se utiliza.

8.3.2.2.5. Evaluación de la Soluciones

La actividad de evaluación consiste en valorar de forma cuantitativa cada solución de la población. El objetivo es sencillo, se trata de poder disponer de un criterio objetivo que permita elegir entre dos o más Agentes-Solución o soluciones generadas en competencia. En concreto la competencia se produce cuando:

- Un Agente-Solución debe elegir a un vecino para que este se convierta en progenitor.
- Un Agente-Solución obtiene más de un descendiente y debe quedarse sólo con uno de ellos.
- Un Agente-Solución debe decidir si sustituye la solución que contiene por otra después de un proceso de cruce, mutación o aprendizaje genético.

Se ha decidido simplificar esta actividad empleando de forma directa el valor de makespan de las soluciones. Por lo tanto, el valor de adecuación o “fitness” de una solución será el makespan calculado a partir de la secuencia que representa a la solución. El cálculo más costoso de este proceso es, sin duda, el cálculo del Programa Productivo a partir del cual se establece el makespan. Todas las propuestas analizadas, que cuentan con buenos resultados, proponían el empleo del makespan. Las más sencillas con una aplicación directa y otras empleando proceso de escalado como en (Ruiz, R. 2003). En el caso del algoritmo SMAGA se no se emplea escalado de ningún tipo.

8.3.2.2.6. Operadores de Cruce

La aplicación del operador de cruce es una de las funciones que más se repite durante la ejecución de un Algoritmo Genético. Por lo tanto, es una de las funciones que más impacto tiene en el resultado final. Sin embargo, la elección de un tipo de operador de cruce adecuado, en términos de resultados obtenidos, es enormemente difícil.

Normalmente, en función de la representación genética de la solución se suele aplicar un tipo de operador genético u otro. En el caso de la representación ordinal es habitual emplear operadores genéticos basados en el intercambio de posiciones de los genes. Dentro de este tipo de operadores existe una gran variedad, y sin embargo no se puede decir a priori cual es el operador que mejor se adapta a un determinado problema. Suele ser necesario un estudio experimental, lo que sin duda es costoso de plantear fuera del entorno investigador. Una vez seleccionado un subconjunto aquellos operadores que se consideran más adecuados, ¿por qué no dejar que sea el propio algoritmo quien decida cual es mejor? Suele ser difícil determinar a priori que operador va a proporcionar los mejores resultados. Normalmente, y más en un entorno que depende en gran medida del azar, las decisiones, en este caso el operador de cruce que se aplica, pueden decantarse hacia un tipo de operador u otro en función de las circunstancias temporales que se den en cada momento. Ello implicaría que no sería malo plantearse una aplicación mixta de las decisiones, es decir, aplicar algunas veces un operador y otras veces otro operador.

En base al razonamiento anterior se ha considerado la hipótesis consistente en que el Agente-Solución tenga la capacidad de elegir por si mismo cual es el mejor operador de cruce de los que tiene disponibles, y aplicar en cada caso aquel que considere mejor. La experimentación que se propone para el operador de cruce del algoritmo SMAGA va dirigida a determinar si es adecuada la hipótesis considerada o no lo es.

Se ha optado por la implementación de dos tipos de operadores de cruce diferentes cuya aplicación gestionará cada uno de los Agentes-Solución. Al dotar a los Agentes-Solución de la posibilidad de elegir entre dos algoritmos, se ha pretendido seguir apostando por la línea de aprendizaje y capacidad de autogestión de los Agentes. Propiedades que son habituales en el ámbito de los Sistemas Multiagente. Se ha reducido el campo de decisión a sólo dos operadores de cruce, ya que se considera un número adecuado, para la observación experimental. Con dos operadores se podrá observar fácilmente si efectivamente los Agentes son capaces de aprender, y decantarse por aquel algoritmo que mejor valor de adecuación les proporcione. Un número más elevado de operadores de cruce puede ayudar a obtener mejores resultados, pero este no es el objetivo principal.

A los Agentes-Solución se les proporcionará, de forma deliberada, un operador de cruce que se considera muy bueno, y otro cuyos resultados no han sido contrastados, y que por lo tanto, no garantiza buen comportamiento. Con ello se desea ver si efectivamente los Agentes saben detectar el operador más adecuado, y si realmente existe gran diferencia entre ellos. Uno de los operadores de cruce es el denominado “Similar Block 2-point Order Crossover (SB2OX)”, que fue empleado por (Ruiz, R. 2003) con gran éxito. El otro operador es una adaptación del denominado “Neighborhood Competition Operator (NCO)” empleado por (Zhong, W. C. et al. 2004), cuya aplicación no está contrastada para el problema del Taller de Flujo Híbrido con Tiempos de Cambio de Partida Dependientes de la Secuencia.

Similar Block 2-point Order Crossover (SB2OX)

El operador de cruce SB2OX fue propuesto y empleado por (Ruiz, R. 2003) en un Algoritmo Genético que tenía como objetivo resolver diversos problemas de Programación de la Producción en un Taller de Flujo Híbrido con Tiempos de Cambio de Partida Dependientes de la Secuencia. En concreto, el juego de datos (problemas) para los que se empleó coinciden con los que se propone resolver con el algoritmo SMAGA. Es decir, el operador SB2OX ya fue utilizado en un Algoritmo Genético diferente al propuesto en este trabajo, pero para resolver el mismo tipo de problema, y con el mismo juego de datos para el que se plantea emplear SMAGA. Los resultados obtenidos por (Ruiz, R. 2003) fueron excelentes al situar el algoritmo que propuso como el más eficaz entre los que se contrastó.

Para la aplicación del operador de cruce SB2OX se necesita disponer de dos progenitores, progenitor A (padre 1) y Progenitor B (padre 2). La selección de los mismos se realiza, tal y como, se describió en el apartado 8.3.2.2.4. El operador SB2OX genera dos nuevas soluciones (hijo 1 e hijo 2). Los pasos a seguir para su aplicación son:

1. Los bloques con al menos dos valores idénticos en ambos padres se copian directamente a los hijos. En la Figura 8-14 se puede observar como las posiciones 8 y 9, y 16 y 17 son idénticas en ambos padres, y por lo tanto se transfieren directamente a los hijos correspondientes.

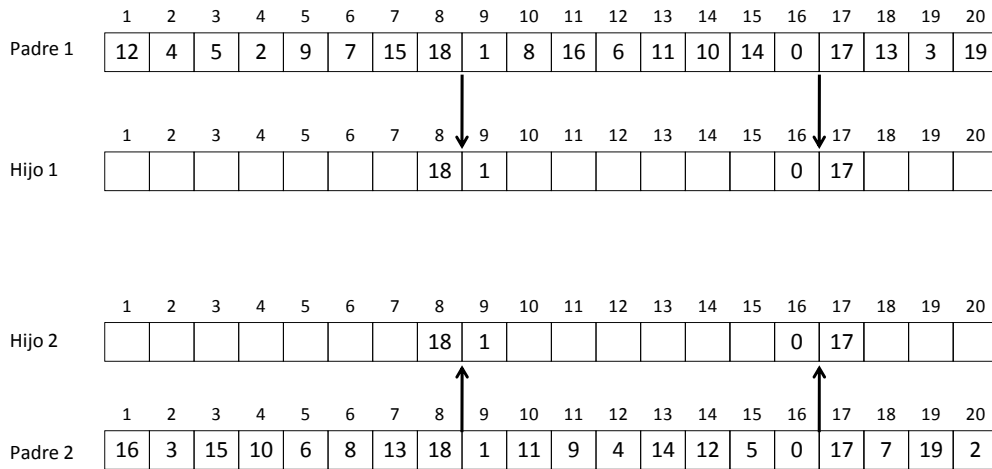


Figura 8-14. Primer paso del operador SB20X (elaboración propia).

- Se generan dos puntos de cruce de forma aleatoria (punto 1 y punto 2). Ambos valores se obtienen mediante una distribución uniforme en el rango entre 1 y el tamaño de la secuencia. El hijo 1 hereda todos los valores del padre 1 situados entre ambos puntos de cruce, y lo mismo ocurre con el hijo 2 con respecto al padre 2. En el ejemplo de la Figura 8-15, el punto de corte 1 se corresponde con la posición 6, y el punto de corte 2 se corresponde con la posición 13. Los valores 7, 15, 18, 1, 8, 16, 6 y 11 del padre 1 se transfieren al hijo 1, y los valores 8, 13, 18, 1, 11, 9, 4, y 14 se transfieren del padre 2 al hijo 2.

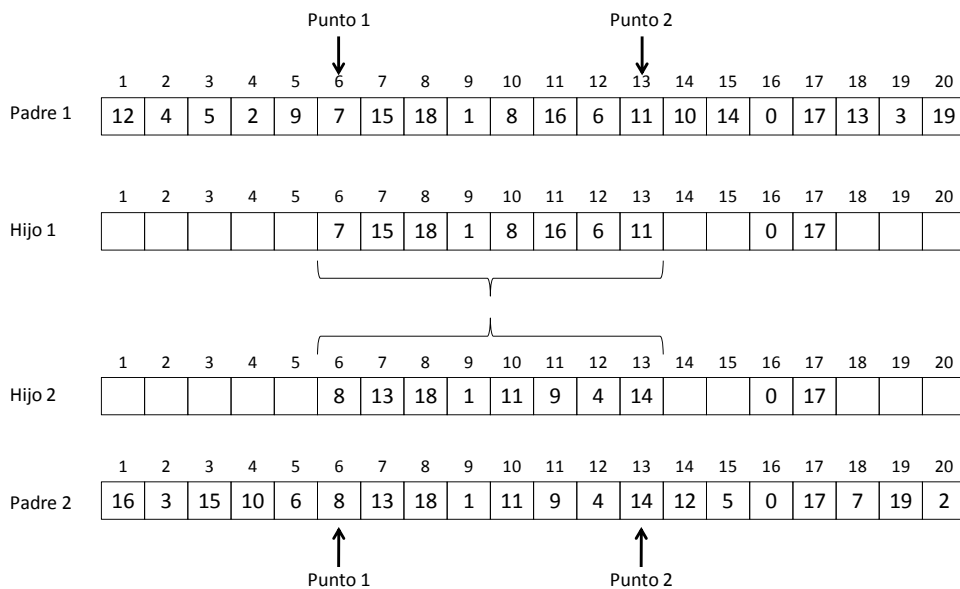


Figura 8-15. Segundo paso del operador SB20X (elaboración propia).

3. El hijo 1 hereda todas las posiciones faltantes del padre 2 y el hijo 2 hereda todas las posiciones faltantes del padre 1. En la Figura 8-16 se puede observar como el padre 1 inicia la transferencia de genes al hijo 2 tomando como punto de partida la primera posición de la secuencia. Si el valor a transferir ya se encuentra en el destino, por ejemplo la posición 2 del padre 1, entonces se salta a la siguiente posición.

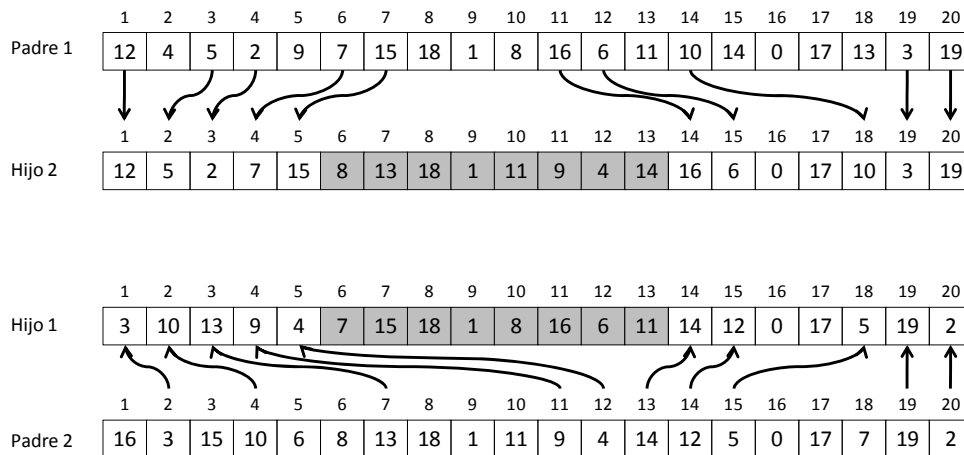


Figura 8-16. Tercer paso del operador SB2OX (elaboración propia).

Se debe considerar que la solución está formada por una secuencia cíclica por lo que se refiere al empleo de los puntos de corte. Se debe remarcar que la aportación del procedimiento de selección y cruce para el operador SB2OX debe suponer un impulso positivo para el Algoritmo Genético SMAGA.

Neighborhood Competition Operator (NCO)

El algoritmo “Neighborhood Competition Operator (NCO) es presentado en (Zhong, W. C. et al. 2004) como un operador genético exploratorio que permite generar y analizar nuevas soluciones, que en el caso referenciado se empleaba para establecer el valor de una serie de parámetros de una función objetivo. El valor de los parámetros podía situarse dentro de un rango continuo de números reales, y con este operador se pretendía establecer, y analizar el efecto de valores intermedios para algunos parámetros entre los valores próximos a los fijados por una solución progenitora. La solución progenitora que se elige para explorar los nuevos valores es la solución con mejor fitness dentro de un entorno local.

En el caso de una representación ordinal de la solución el objetivo perseguido originalmente no puede ser asumido, ya que en el caso anterior el valor de cada posición de la solución era independiente del resto. Sin embargo, cuando se trata de una secuencia al variar el valor de

una posición hay que variar obligatoriamente el valor de otra posición del resto de la solución, es decir, hay dependencia. La dependencia complicaría mucho la aplicación del NCO, y además el efecto de un cambio o no equivaldría a la modificación de un parámetro, es decir, no existe la misma relación causa-efecto en los dos problemas.

Lo que se ha considerado interesante de la propuesta de (Zhong, W. C. et al. 2004) es que no propone un operador de cruce típico en el que se generan nuevas soluciones a partir de dos progenitores, sino que se genera una sola solución a partir de un único progenitor, y que este es el mejor de su entorno. Se trata, por lo tanto, de sustituir una solución por una evolución de la mejor solución del entorno, esta es la novedad. Por este motivo, los autores no lo identifican directamente como un operador de cruce sino como un operador genético competitivo. En todo caso, supone una evolución de una determinada solución. Otro aspecto interesante es que la búsqueda la mejor solución se realiza mediante una competencia local.

Para la propuesta actual la selección se realiza, tal y como, propone (Zhong, W. C. et al. 2004), y como se ha explicado en el apartado 8.3.2.2.4. Una vez se ha seleccionado a un Agente, que inicialmente se identifica como progenitor A, se procede a seleccionar al progenitor B. Según el algoritmo descrito en el apartado 8.3.2.2.4 el progenitor B es el ganador en el proceso de competencia local. En el ejemplo de la Figura 8-17 se muestra como si el algoritmo está evaluando la posibilidad de aplicar el operador de cruce en el Agente-Solución de la posición (2,2) de la matriz base éste establecerá una competencia local con los Agentes solución (1,2), (2,1), (2,3) y (3,2). La solución del Agente-Solución con menor makespan será considerado el progenitor B. En el caso del operador de cruce NCO el único progenitor que se utiliza es el B, por lo que el progenitor A será ignorado, a no ser que sea el que menor makespan presente en cuyo caso será considerado a todos los efectos como progenitor B.

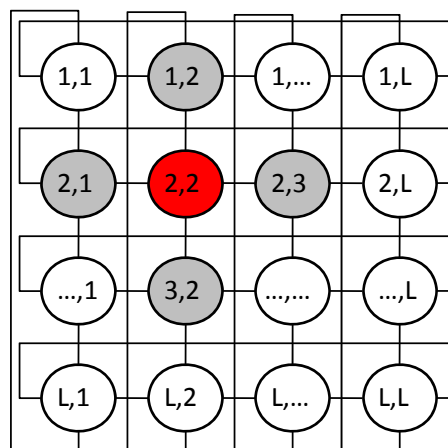


Figura 8-17. Competidores locales en la matriz base para el Agente-Solución (2,2) (elaboración propia).

Como consecuencia de lo descrito en el párrafo anterior se ha adaptado el procedimiento establecido en el apartado 8.3.2.2.4 para clarificar que el procedimiento de cruce NCO sólo requiere del progenitor con mejor makespan. La Ilustración 8-6 proporciona el pseudocódigo correspondiente al procedimiento de selección y cruce para el operador NCO.

```
// Procedimiento modificado de Selección de progenitores para el operador de cruce NCO
progenitorB=nulo;

Para i=1 hasta L
  Para j=1 hasta L {
    Si (aleatorio(0,1)<factor_Riesgo) {
      Mkij=Calcular makespan (i,j)
      (Mka, pos_i, pos_j)=Calcular menor_makespan (Vecinos(i,j))
      Si (Mkij>Mka) progenitorB=agente(pos_i, pos_j)
      Sino progenitorB=agente(i,j)
      Si (progenitorB<>nulo) Cruce_NCO (progenitorB)
    }
  }
}
```

Ilustración 8-6. Pseudocódigo del procedimiento de selección y cruce para el operador NCO (elaboración propia).

La implementación del operador varia sensiblemente respecto de la propuesta de (Zhong, W. C. et al. 2004), ya que los autores realizan primero una modificación de los valores de los genes y posteriormente una alteración en el orden de los mismos. De la propuesta original sólo se respeta la forma de alterar el orden de los genes. A continuación, y una vez establecido el procedimiento de selección de progenitores, se define el funcionamiento del operador de cruce NCO. Este consiste en:

1. Se elige de forma aleatoria dos puntos de cruce. Mediante una distribución uniforme entre 1 y el tamaño de la secuencia. Se obtienen dos valores.

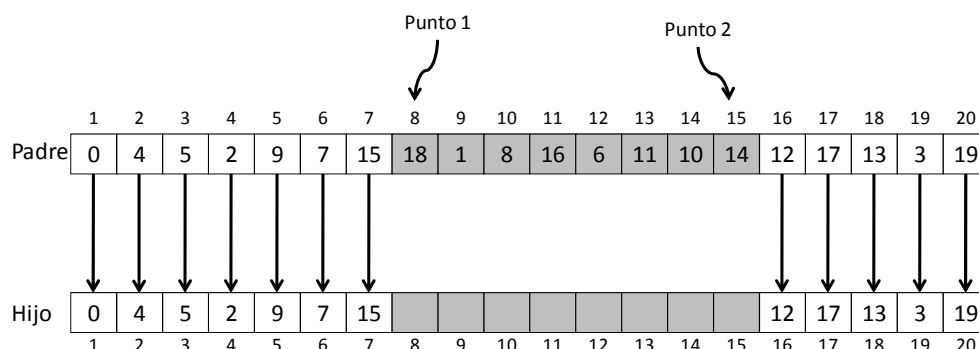


Figura 8-18. Primer paso del operador NCO (elaboración propia).

2. Se transfiere toda la información genética que no se encuentra entre el punto de corte 1 y el punto de corte 2, desde el padre hasta el hijo.
3. La información genética que se encuentra entre ambos puntos de corte, configura una secuencia parcial que, se transfiere desde el padre hasta el hijo, de tal forma que la posición de los genes en el hijo se corresponde con el orden inverso de la posición de los genes del padre en la secuencia parcial.

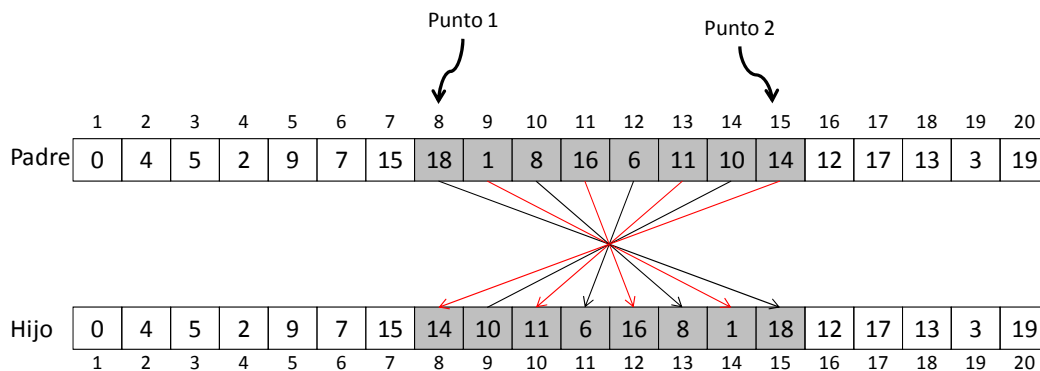


Figura 8-19. Segundo paso del operador NCO (elaboración propia).

Igual que en el caso del operador de cruce SB2OX, se debe considerar que la solución está formada por una secuencia cíclica por lo que se refiere al empleo de los puntos de corte.

Una vez finalizada la descripción del el procedimiento de selección y cruce para el caso del operador NCO, se debe remarcar que éste se plantea con un desconocimiento previo de cual puede ser su comportamiento, y por lo tanto su aportación al Algoritmo Genético.

La forma en la que se combinan ambos operadores de cruce, SB2OX y NCO, para realizar la función de cruce empleando cierta capacidad de aprendizaje se describe con detalle en el apartado siguiente.

8.3.2.2.7. Selección de los Operadores de Cruce

En este apartado se detalla como se decide que operador de cruce se aplica en cada ocasión, cuestión que es relevante al disponer de dos opciones.

Se aborda el método para la selección de los operadores de cruce dentro de la lógica establecida en el Algoritmo Genético, esquematizada en la Figura 8-7. El método que se propone se basa en dejar que el Agente-Solución tenga disponible siempre la opción de aplicar el operador de cruce SB2OX y NCO, pero que cada vez que tenga que realizar un cruce evalúe cual es la opción más adecuada. Dicha evaluación es realizada de forma individual por cada Agente-Solución, y depende de lo que éste ha aprendido en función de su experiencia aplicando operadores de cruce. Con este enfoque se pretende alinear la funcionalidad de los Agentes con sus características básicas, y en este caso, con su autonomía y capacidad de aprendizaje.

La aplicación del método se inicia cuando el Agente-Solución realiza un sorteo empleando una distribución uniforme entre 0 y 1, para obtener un valor. El rango $[0, 1]$ se divide en dos sub-rangos continuos, el primero es el definido por $[0, \text{factor límite}]$ y vinculado al operador NCO, y el segundo es el definido por $]\text{factor límite}, 1]$ y vinculado al operador SB2OX. Si el valor obtenido en el sorteo pertenece al primer sub-rango se aplica el operador NCO, y sino el operador SB2OX. La variable “**factor límite**” es definida por el usuario, e inicialmente tiene un valor de 0.5 para el algoritmo SMAGA. Por lo tanto, inicialmente ambos operadores tienen las mismas oportunidades de ser aplicados.

Una vez se ha finalizado de aplicar el operador de cruce correspondiente, en función del sorteo, se guarda el valor del makespan obtenido con la nueva solución^{vii} generada vinculada al operador con el que se ha obtenido. Así mismo, al finalizar la aplicación del operador de cruce, se verifica si se ha aplicado diez veces el operador NCO o SB2OX, si alguno de los dos ha alcanzado dicho valor se procede a modificar el valor actual de la variable “factor límite”. La modificación de la variable “factor límite” consiste en incrementar en 0.01 unidades el valor si se observa que el operador de cruce NCO ha obtenido un makespan medio mejor que el operador SB2OX, promediando los valores obtenidos de makespan de las diversas soluciones generadas desde la última vez que se modificó el valor de dicha variable. Si por el contrario el operador NCO ha obtenido un makespan medio peor que el operador SB2OX entonces el valor de “factor límite” se reduce en 0.01 unidades. Una vez modificado el valor de “factor límite” los valores de makespan obtenido en el último periodo (desde la última modificación del valor de la variable “factor límite”), se borran para iniciar otro periodo con una valoración independiente de la anterior.

^{vii} Como se comentará posteriormente los operadores de cruce sólo proporcionarán una solución.

Como se ha indicado en el párrafo anterior la variable “factor límite” va modificando su valor de forma que el operador con mejor comportamiento, en cuanto al valor de makespan de las soluciones generadas, va progresivamente ampliando sus posibilidades de ser elegido. Para garantizar que los dos operadores mantienen opciones de ser seleccionados en cualquier caso, y así disponer de información sobre su comportamiento, se ha establecido una cota inferior de 0.1 y una cota superior de 0.9 para la variable “factor límite”. Hay que destacar que al realizarse una valoración del comportamiento de los operadores, y modificación de la variable “factor límite”, cada vez que un operador se aplica diez veces, es decir cuando el operador de cruce se ha aplicado entre 10 y 20 veces, se garantiza que esta modificación se realiza un número muy elevado de veces. La frecuencia con la que se realiza el cambio en el valor de la variable “factor límite” es importante, ya que una frecuencia baja impediría que el Agente-Solución se posicione en el valor más adecuado de forma rápida, impidiendo de esa forma, que el operador que mejor se comporta obtuviera la cuota que le corresponde durante un periodo significativo de la ejecución del algoritmo, viéndose de esa forma desvirtuado el enfoque propuesto.

Para poder gestionar el proceso de aprendizaje, se ha creado un parámetro denominado “**Evolución del Factor Límite**” al cual el usuario puede darle el valor de “activado” o “desactivado”. En caso de estar activado el valor del “factor límite” se modifica, tal y como se ha explicado, en función de la evolución de los resultados obtenidos en la aplicación de cada operador de cruce. En caso, de estar desactivado el valor del “factor límite” no se modifica nunca, y por lo tanto, se conserva el valor inicial de 0.5. En este último caso el Agente-Solución se considera que no aprende. El empleo de este parámetro permite al usuario utilizar de la forma que considere más oportuno el algoritmo, y en el caso de la experimentación, permite analizar el comportamiento del algoritmo SMAGA en ambas situaciones.

```
// Procedimiento de Cruce (NCO/SB2OX)
Descendiente=nulo

Si (Aleatorio (0,1)< factor_limite) {
    descendiente=Aplicar_logica_cruce_NCO (progenitorA, progenitor B)
    Mk=Calcular makespan (descendiente)
    Si (Activacion_factor_limite) Actualiza_evolucion_NCO (Mk)
}Sino{
    descendiente=Aplicar_logica_cruce_SB2OX (progenitorA, progenitor B)
    Mk=Calcular makespan (descendiente)
    Si (Activacion_factor_limite) Actualiza_evolucion_SB2OX (Mk)
}
Si ((Activacion_factor_limite) y (fin_ciclo_cruce())) Calcula_factor_limite()
```

Ilustración 8-7. Pseudocódigo correspondiente a la selección del operador de cruce (elaboración propia).

El pseudocódigo correspondiente a la aplicación del operador de cruce, que se puede ver en la Ilustración 8-7, se es el que sería invocado por la función denominada “Cruce” que se empleó en el procedimiento modificado de selección y cruce de progenitores de la Ilustración 8-5.

8.3.2.2.8. Aplicación de los Operadores de Cruce

En este apartado se describe el método mediante el cual se realiza la incorporación de la nueva o las nuevas soluciones a la población, es decir método para el relevo generacional.

Se ha decidido que en el algoritmo SMAGA los operadores de cruce generen únicamente una solución nueva o hijo. Como ya se explicó en el apartado anterior el operador SB2OX proporciona dos soluciones nuevas, y el operador NCO sólo una. Por lo tanto, es necesario establecer un procedimiento adicional de selección para el operador de cruce SB2OX, que permita escoger una de las dos soluciones generadas. En el caso del operador NCO no se necesita realizar ninguna modificación por lo que se refiere al número de soluciones generadas. El método propuesto para reducir la descendencia del operador SB2OX se ha diseñado para que este en línea con la filosofía que se propone para realizar el relevo generacional. Esta filosofía consiste en:

- **Intentar no repetir soluciones en la población** con el objeto de emplear el espacio de soluciones con el mayor número de soluciones diferentes.
- **Intentar realizar una sustitución de soluciones que permita que el nivel promedio de bondad de las mismas sea cada vez mejor.** Se propone sustituir las soluciones existentes por otras mejores si es posible o por otras peores siempre que no se alejen mucho del nivel de bondad existentes. Para dar un margen en la exploración de soluciones se permite que una solución dada se cambie por otra peor, siempre que no exista mucha diferencia entre el fitness de ambas. Esto permite que pueda producirse un relevo de soluciones incluso cuando estas están muy próximas al óptimo, lo que permite seguir explorando el espacio de soluciones incluso en situaciones donde es difícil mejorar.

Cuando se plantea como objetivo no repetir soluciones se hace de una forma flexible. No repetir soluciones implica verificar si una secuencia nueva existe ya en la población. Este proceso es computacionalmente costoso, por lo que se ha decidido realizar dicha comprobación empleando sólo el valor del makespan que se obtiene con cada solución. Dicho valor se calcula siempre que se genera una nueva solución por lo que no supone un coste adicional excesivo. Se sabe que dos soluciones pueden tener el mismo valor de makespan y poseer secuencias

diferentes, pero esta situación es en términos de probabilidades baja. Asumiendo el riesgo de considerar a dos soluciones iguales siendo diferentes se ha optado por emplear el makespan como medio de discriminar entre soluciones.

La modificación propuesta para el operador SB2OX consiste en seleccionar al hijo cuyo valor de makespan sea menor siempre que dicho valor no se encuentre repetido en los vecinos^{viii} con los que se comunica (izquierda, derecha, arriba y abajo). Si el valor el valor de makespan de ambos hijos se encuentra ya entre sus vecinos se elige como descendiente al de menor makespan.

Tanto en el caso del operador de cruce SB2OX, después de la selección de un descendiente, como en el caso del operador de cruce NCO, que sólo proporciona un descendiente, es necesario plantearse cual es el destino del mismo. En el algoritmo SMAGA se ha determinado que el relevo generacional debe consistir en la sustitución de una solución existente por una nueva solución siempre que se cumplan una serie de requisitos. Los requisitos exigidos para realizar la modificación de la secuencia que representa a la solución en una gente-solución son los que deben garantizar la aplicación de la filosofía propuesta para el relevo generacional. Estos son, por el orden indicado, los siguientes:

1. Se verifica que el makespan de obtenido con la secuencia de la nueva solución generada con el operador de cruce del Agente-Solución en curso es diferente al de todos sus vecinos. Se entiende por vecinos de un Agente-Solución de la matriz básica a los Agentes-Solución de la izquierda, de la derecha, de arriba y de abajo. Si el valor del makespan se repite en algún caso se descarta al descendiente y no se realiza el relevo generacional.
2. Si se ha pasado el filtro del punto anterior, se verifica que el valor del makespan de la nueva solución generada es menor o igual al *valor corregido* del makespan de la solución actual. Se entiende por valor corregido a multiplicar el valor del makespan por un factor denominado "**Factor Margen**". El valor del margen siempre es mayor que uno, por lo que lo que se hace mediante esta comprobación es verificar que la nueva solución presenta un fitness mejor que la anterior o, en el caso de ser peor, se encuentra próximo a la solución actual. La variable "factor margen" es un parámetro que fija el usuario, y que será establecido en la fase de Parametrización del algoritmo. En todo caso, es necesario resaltar que valores altos del "factor margen" implican

^{viii} El concepto de agente vecino se aborda en el apartado 8.3.2.2.2.

facilidad en el proceso de relevo generacional, y análisis de un amplio espectro de soluciones, que incluye las soluciones con baja o muy baja bondad. Las soluciones cuyo valor de makespan supera el valor corregido del makespan de la solución actual son rechazadas y no se realiza el relevo generacional.

3. Si se ha pasado la restricción del punto anterior, se verifica que el valor del makespan de la nueva solución no se encuentre repetido en el conjunto de la población, y no sólo en los vecinos. Esta restricción no se aplica de forma estricta. Dado que se sabe que las soluciones que llegan hasta este punto son soluciones relativamente buenas se ha decidido realizar esta verificación sólo para las soluciones con un valor que se diferencia con el valor del mejor makespan de la población actual en un máximo de K unidades. Se genera una lista de K valores cuyo valor más bajo es el del mejor makespan conocido en el conjunto de la población. En dicha lista, inicialmente todos los valores están libres. Siempre que una solución cumple esta restricción la secuencia que representa a la nueva solución sustituye a la secuencia con la que Agente-Solución en curso representaba a la solución actual. Además se anota que el valor correspondiente al makespan de dicha solución se ha ocupado para que ninguna otra nueva solución con el mismo valor pueda introducirse en la población. Si el valor del makespan alcanzado con la nueva solución es mejor (menor) que el mejor valor obtenido en el conjunto de la población la lista de valores ocupados se borra. También se borra la lista cuando se realiza un proceso de reinicio de la población. La información contenida en la lista es pública y accesible para todos los Agentes-Solución de la matriz base.

En la Figura 8-20 se muestra un esquema de la propuesta empleada para la sustitución de la actual por una nueva después de una operación de cruce.

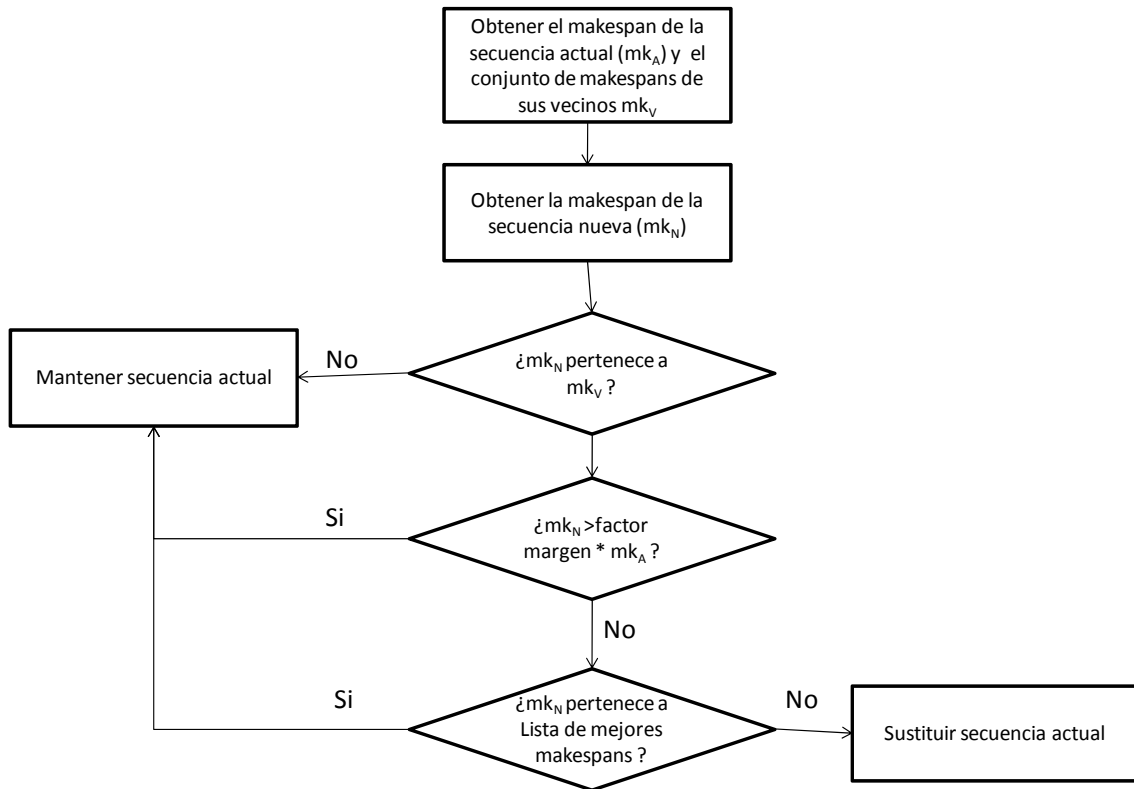


Figura 8-20. Diagrama de flujo para sustitución de la secuencia actual por una nueva después de un cruce.

En la Ilustración 8-8 se muestra el pseudocódigo asociado al método de relevo generacional utilizado en los operadores de cruce. Dicho código se corresponde con la invocación de la función “logica_de_cruce_NCO” o “lógica_de_cruce_SB2OX” empleado en la Ilustración 8-7.

```

// Aplicación de la lógica de cruce NCO/SB2OX
descendiente=nulo
descendiente1=nulo
descendiente2=nulo

Si (NCO) {
    descendiente=operar(progenitorB)
    Mk=Calcular makespan (descendiente)
    Si (existe_en_vecindario(Mk)==cierto) retorna
}

Si (SB2OX) {
    (descendiente1, descendiente2)=operar(progenitorA, progenitorB)
    //Elige a uno de los dos descendientes.
    Mk1=Calcular makespan (descendiente1)
    Mk2=Calcular makespan (descendiente2)
    Si (Mk1<=Mk2)
        descendiente=descendiente1
        Mk=Mk1
        Si ((existe_en_vecindario(Mk1)==cierto) y (existe_en_vecindario(Mk2)=falso)){
            descendiente=descendiente2
            Mk=Mk2
        }
    Si (Mk1<=Mk2)
        descendiente=descendiente1
        Mk=Mk1
        Si ((existe_en_vecindario(Mk1)==cierto) y (existe_en_vecindario(Mk2)=falso)){
            descendiente=descendiente2
            Mk=Mk2
        }
}

//En este punto ya se dispone de un solo descendiente que no se encuentra en el vecindario (en el caso
de SB2OX es posible que si se encuentre).

Si (Mk>=makespan_actual*Factor_Margen) retorna
Si (Esta_en_lista(Mk)==cierto) retorna
//Si se ha llegado hasta aquí se cambia la secuencia actual por la nueva secuencia.
Sustituye_secuencia (progenitorA, descendiente)
    
```

Ilustración 8-8. Pseudocódigo correspondiente al método de relevo generacional de los operadores de cruce (elaboración propia).

A continuación se ilustrará el procedimiento descrito para el relevo generacional mediante un pequeño ejemplo. Supóngase que disponemos de una población formada por una matriz básica de dimensión 4x4, y que en este instante se está ejecutando el operador de cruce sobre

el Agente-Solución (2,2). La Figura 8-21 se emplea como representación gráfica de dicha matriz base. Como se puede ver en la figura cada Agente-Solución está identificado por una dupla, que indica la fila y la columna que identifica su posición en la matriz, y dispone de un valor asociado que representa al makespan o fitness de cada Agente-Solución. En el ejemplo, el Agente-Solución (2,2) tiene asociado un makespan de 1.800 unidades. Considérese un valor del parámetro “factor margen” de 1.05. Independientemente de cual sea el operador de cruce empleado. Supóngase que ya se dispone de un descendiente, una nueva solución, cuya evolución analizaremos en función de los diferentes valores que puede alcanzar su makespan.

Supóngase un valor de makespan de la nueva solución de 2.015. El procedimiento de aplicación del operador de cruce haría lo siguiente:

1. ¿Es 2.015 diferente del valor de makespan de los Agentes-Solución (1,2), (3,2), (2,1) y (2,3)? La respuesta es sí, ya que: Agente-Solución (1,2)=1.580, Agente-Solución (3,2)=1.980, Agente-Solución (2,1)=2.050 y Agente-Solución (2,3)=2.030. Por lo tanto, se continúa analizando las restricciones.
2. Dado que el “factor margen” es 1.05, y que el valor actual de makespan es de 1.800, ¿El nuevo valor de makespan es inferior a $1.800 * 1.05 = 1.890$? La respuesta es No. Y por lo tanto, la nueva solución se rechaza y no se realiza relevo generacional.

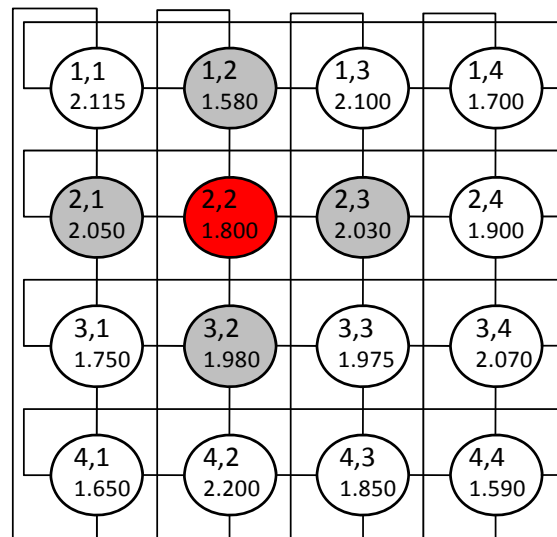


Figura 8-21. Ejemplo de población con matriz base de dimensión 4x4 (elaboración propia).

Supóngase ahora que el valor de la nueva solución es 1.845. Si se vuelve a aplicar el procedimiento obtenemos:

1. ¿Es 1.845 diferente del valor de makespan de los Agentes-Solución (1,2), (3,2), (2,1) y (2,3)? La respuesta es sí, ya que: Agente-Solución (1,2)=1.580, Agente-Solución (3,2)=1.980, Agente-Solución (2,1)=2.050 y Agente-Solución (2,3)=2.030. Por lo tanto, se continúa analizando las restricciones.
2. Dado que el “factor margen” es 1.05, y que el valor actual de makespan es de 1.800, ¿El nuevo valor de makespan es inferior a $1.800 \cdot 1.05 = 1.890$? La respuesta es Sí. Por lo tanto, se continúa analizando las restricciones.
3. Para un valor de $K=500$, ¿El valor de makespan de la nueva solución existe ya en la población cuyo makespan está comprendido entre $[1.580+1, 1.580+500]$?. Para dicho caso se analiza la lista de valores ocupados. En la Figura 8-22 se ha representado una lista de valores ocupados que podría corresponder al ejemplo en curso. En la lista cada individuo dispone de un índice que va desde 1 hasta 500 para distinguir los 500 valores que se analizan, así mismo se indica el valor del makespan con el que se corresponde cada elemento, y por último se indica mediante un color verde que el valor no ha sido ocupado y un color rojo que ya existe.

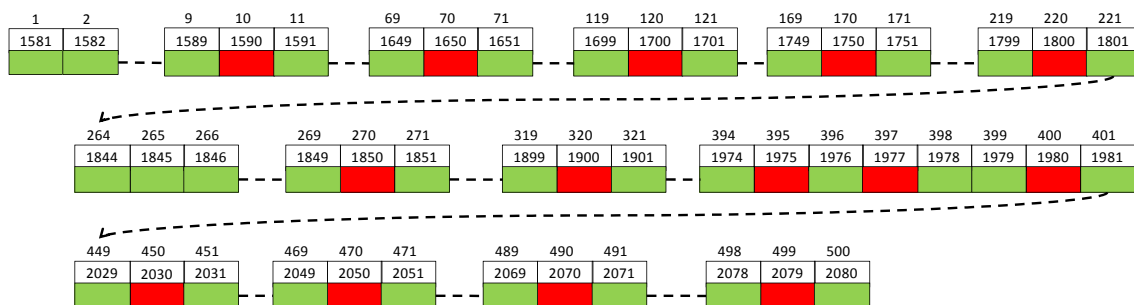


Figura 8-22. Despliegue de la lista de posiciones ocupadas para la aplicación del operador de cruce (elaboración propia).

Como se puede ver la posición correspondiente al valor de makespan 1.815, índice 265, está libre, y por lo tanto, puede ser introducido en la población. En estas condiciones la solución actual con valor 1.800 es sustituida por la nueva solución con valor de makespan 1.845, que es peor que el anterior.

A modo de ilustración, se debe resaltar como en el ejemplo anterior los valores de makespan 1.977 y 2.079, correspondientes con los índices 397, 499 respectivamente, están ocupados y sin embargo no son valores existentes en la actual configuración de matriz base. Esto se debe a que el borrado de la lista, es decir considerar a todos los valores como libres (color verde) sólo se hace cuando se mejorar el makespan más bajo de la población o existe una regeneración de

la población. Por lo tanto, pueden existir valores de makespan que en su momento fueron considerados pero que después fueron sustituidos por otra solución, y no por ello su valor ha sido liberado de la lista. Es el caso de los índices 397 y 499. Es normal que el número de valores ocupados en la lista sea superior al de Agentes-Solución de la matriz base. Esto implica que no sólo se impide que se repita un valor de makespan existente, sino cualquier valor de makespan ya analizado desde la última mejora del makespan o regeneración de la población.

Una vez finalizada la ilustración del funcionamiento del procedimiento de aplicación del operador de cruce hay que indicar que existe una situación en el que deliberadamente se replica la secuencia de un Agente-Solución. En el caso de que después de aplicar el operador de cruce el valor de makespan generado sea menor que el menor valor de makespan de la población dicha solución de implanta en un Agente-Solución de la matriz base elegido aleatoriamente dentro de la diagonal de la matriz base. Esta secuencia sustituirá a la secuencia que hubiese hasta ese momento en el Agente-Solución elegido al azar. La réplica se hace, de forma excepcional, para facilitar la participación en la evolución genética de un individuo considerado excelente hasta ese momento. Se sitúa en la diagonal de la matriz porque así se evitar que sea un vecino del Agente-Solución en curso, y porque por otro lado la ubicación de éste es indiferente.

Por último, es bueno recordar, para disponer de una idea de conjunto, que para poder realizar todo este procedimiento de aplicación del operador genético es necesario que se inicie la operación de cruce, y que para que esta se inicie hace falta que el Agente-Solución supere un sorteo vinculado al “factor de riesgo” y explicado en el apartado 8.3.2.2.4.

8.3.2.2.9. El Operador de Mutación

El operador de mutación actúa de forma análoga al proceso de mutación genética de los seres vivos. La mutación provoca el cambio de algún gen del individuo sobre el que se actúa. El operador de mutación, junto al operador de cruce, son dos herramientas tradicionales de exploración de las soluciones en busca del óptimo. Suele emplearse en combinación con el operador de cruce para evitar convergencia prematura de la población por la existencia de un individuo dominante.

El operador de mutación se suele caracterizar por ser mucho más simple que el operador de cruce, ya que suele conseguir su propósito con una sencilla operación. Existen dos enfoques diferentes en cuanto a la forma de decidir si aplicar el operador de mutación. Hay autores que en una primera decisión determinan si una solución en su conjunto debe mutar, y con una

segunda decisión determina que gen se modifica. Otros autores realizan la toma de decisión sobre cada gen, estableciendo en ese momento si se realiza un cambio o no sobre dicho gen.

Así mismo, por lo que se refiere a la forma de aplicar la mutación, existen también dos enfoques diferenciados. Por un lado, están los autores que al aplicar el operador realizan una simple modificación en el valor del gen o genes. Por otro lado, existen autores que aplican un cambio en la posición de los individuos. Por ejemplo, (Zhong, W. C. et al. 2004) optan por el primer enfoque, mientras que (Ruiz, R. 2003) opta por el segundo. Hay que tener en cuenta que la presentación de las soluciones escogida condiciona en muchos casos el tipo de aplicación del operador de mutación. Una representación ordinal impide la aplicación de la mutación por modificación del valor del gen, ya que se pueden producir soluciones con secuencias no factibles.

El operador de mutación propuesto se basa en intercambio de posiciones, y la toma de decisiones sobre su aplicación es mixta, y se explicará con detenimiento posteriormente. Dicho operador consiste en una adaptación propia del operador de mutación por intercambio adyacente analizado por (Ruiz, R. 2003). En este caso se permite que el intercambio de posiciones se realice entre individuos próximos aunque no necesariamente adyacentes. En el operador de mutación por intercambio en proximidad consiste que dado un gen cuya posición en la secuencia es la "i", selecciona a otro gen con posición "j" situado en una posición aleatoria entre $[i+1, i+3]$. En el cálculo de la posición del gen "j" se supone que la secuencia es cíclica, y que si existen n trabajos la posición n+1 de la secuencia es la posición 1.

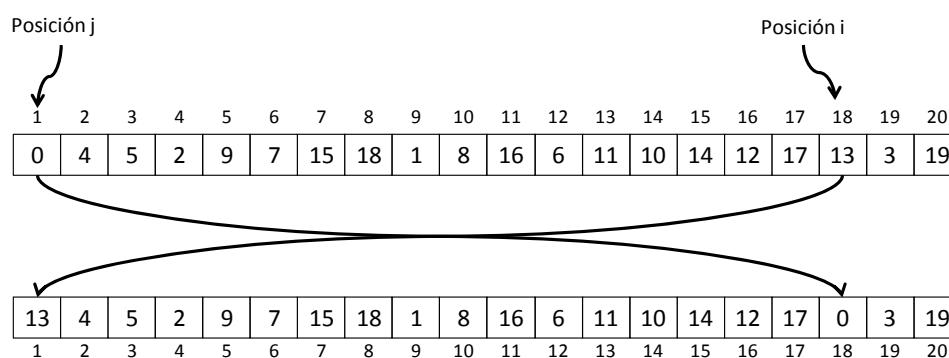


Figura 8-23. Mutación por intercambio en proximidad (elaboración propia).

En la Figura 8-23 se puede ver un ejemplo de aplicación del operador comentado para una solución con 20 trabajos. En este caso dado el gen "i", siendo éste el que ocupa la posición 18, se ha elegido aleatoriamente al gen "j", en este caso el que ocupa la posición 1 ($i+3$). La información del gen 18 se ha transferido al gen de posición 1 y vice-versa.

Por lo que se refiere a la manera de tomar la decisión sobre su aplicación, ésta es híbrida como se ha dicho anteriormente. Ello quiere decir que existe un doble proceso de toma de decisiones. Existe un nivel de decisión general que afecta al conjunto de todos los genes, y un nivel de toma de decisiones que se realiza sobre cada gen. La primera decisión que se toma es la que afecta al conjunto de la solución. Esta decisión se adopta eligiendo un número al azar mediante una distribución uniforme entre 0 y 1. Si dicho número es menor que el valor de un parámetro previamente establecido, al que se le domina tradicionalmente “**factor de mutación**”, entonces se aplica un segundo nivel de toma de decisiones. Este segundo nivel tiene como fin decidir sobre que gen se aplica el operador de mutación. Para ello se le asigna a cada gen un valor idéntico, que coincide con una cuota alícuota de la unidad, y posteriormente se realiza una selección aleatoria de un valor basado en una distribución uniforme comprendida entre 0 y 1. Si el valor obtenido aleatoriamente es inferior al valor asignado entonces se aplicará el operador de mutación sobre el gen en curso en otro caso no. En la Ilustración 8-9 se muestra el pseudocódigo donde se desarrolla el procedimiento para la aplicación del operador de mutación.

```
// Procedimiento para aplicación del Operador de Mutación

Si (sorteo(0,1)<factor_mutación) {
  Para i=1 hasta N {
    Si (sorteo (0,1)< (1/N)) operador_mutación(i)
  }
}
```

Ilustración 8-9. Pseudocódigo correspondiente al procedimiento para la aplicación del operador de mutación (elaboración propia).

Un valor bajo en el parámetro “factor de mutación” implica reducidas posibilidades de aplicarlo, mientras que un valor alto implica mayor posibilidad de hacerlo. En general, los Algoritmos Genéticos suelen asignarle un valor a dicho parámetro bastante menor que el factor de cruce. En todo caso, su valor debe ajustarse para alcanzar su fin, es decir regular la presión genética para evitar la convergencia prematura favoreciendo la exploración de espacio de soluciones. La asignación de los valores de dicho parámetro en SMAGA se realizará posteriormente mediante un ajuste paramétrico.

Otro aspecto a tener en cuenta durante la aplicación del operador de mutación es como se realiza el relevo generacional, de la misma forma que se planteó en el operador de cruce. En este caso se trata de decidir si la nueva secuencia, la secuencia mutada, debe sustituir o no a la secuencia actual. Con el fin de seguir manteniendo la filosofía establecida para el relevo

generacional descrito en el apartado correspondiente, pero al mismo tiempo no dificultar en exceso el proceso de mutación, se ha definido un conjunto de restricciones muy similares, pero no iguales, a las mencionadas. Estas son:

1. Se verifica que el makespan de obtenido con la secuencia mutada generada con el operador de mutación del Agente-Solución en curso es diferente a la de todos sus vecinos. Se entiende por vecinos de un Agente-Solución de la matriz básica a los Agentes-Solución de la izquierda, de la derecha, de arriba y de abajo. Si el valor del makespan se repite en algún caso se descarta al descendiente y no se realiza el relevo generacional.
2. Si se ha pasado la restricción del punto anterior, se verifica que el valor del makespan de la nueva solución no se encuentre repetido en el conjunto de la población, y no sólo en los vecinos. Esta medida se aplica exactamente igual que en el caso del operador de cruce.
3. Si se ha pasado el filtro del punto anterior, se verifica si el valor del makespan de la solución actual es menor o igual al *valor corregido* del makespan de la mejor solución de la población, y entonces sólo se realiza el relevo si la secuencia mutada presenta un makespan mejor que el de la secuencia actual. Si el valor del makespan de la solución actual es mayor que el *valor corregido* del makespan de la mejor solución de la población, entonces sólo se realiza el relevo generacional si el makespan de la nueva solución es menor que el valor corregido del makespan de la solución actual. En ambos casos se debe entender que el valor corregido es la multiplicación del valor indicado por la variable "**Factor Margen**", ya mencionada en el caso del operador de cruce. Un resumen válido para esta última restricción es que si la solución pertenece a un grupo de élite sólo se cambia si la nueva solución es mejor, sino se permite el cambio aunque empeore la solución actual siempre que lo haga dentro de un margen de proximidad razonable.

En la Ilustración 8-10 se resume, mediante un pseudocódigo, la aplicación del conjunto de restricciones descrito anteriormente. Dicho código se corresponde con la ejecución de la función "operador de mutación" indicada en la Ilustración 8-9.


```

// Aplicación de la lógica de Mutación
descendiente=nulo

descendiente=operar(progenitor)
Mk=Calcular makespan (descendiente)

Si (existe_en_vecindario(Mk)==cierto) retorna
Si (Esta_en_lista(Mk)==cierto) retorna
Si (makespan_actual<=makespan_poblacion*Factor_Margen){
    Si (Mk>= makespan_actual) retorna
} sino{
    Si (Mk>= makespan_actual*Factor_Margen) retorna
}
}
//SI se ha llegado hasta aquí se cambia la secuencia actual por la nueva secuencia.
Sustituye_secuencia (progenitor, descendiente)
    
```

Ilustración 8-10. Pseudocódigo correspondiente al método de relevo generacional del operador de mutación (elaboración propia).

8.3.2.2.10. Aprendizaje Genético

Se denomina aprendizaje genético a la etapa del Algoritmo Genético planteado en la cual se realiza un análisis exploratorio del espacio de soluciones en el entorno de una solución dada. En cada generación se desarrolla dicha actividad para cada uno de los Agentes-Solución que componen la matriz base. Este aprendizaje ya se planteaba en el algoritmo desarrollado por (Zhong, W. C. et al. 2004), y lo que se ha hecho es rescatar dicha idea para incorporarla a la propuesta actual. La propuesta de (Zhong, W. C. et al. 2004) consiste en generar una población inicial a partir de la secuencia de un Agente-Solución dado y aplicar sobre ella el algoritmo SMAGA simplificado. Se trata de ejecutar un Algoritmo Genético dentro de otro, donde además ambos algoritmos son muy similares. En la Figura 8-24 se muestra un resumen de las etapas en la que se divide el proceso de aprendizaje genético o SMAGA reducido.

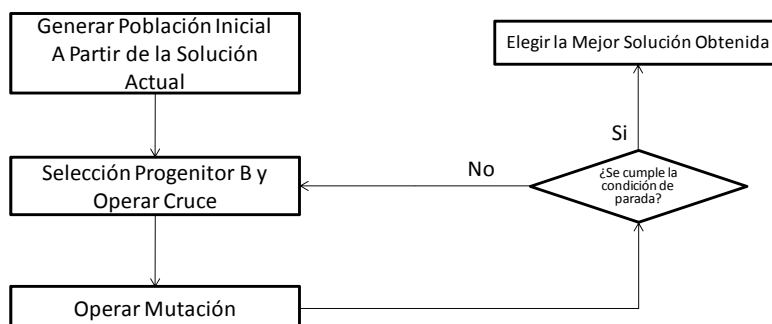


Figura 8-24. Esquema general de los procesos correspondientes al procedimiento de aprendizaje genético (basado en (Zhong, W. C. et al. 2004))

El algoritmo SMAGA simplificado que se aplica en esta etapa, y al que denominaremos procedimiento de aprendizaje genético, se ha adaptado para trabajar con:

- Una población más reducida formada por individuos muy similares genéticamente, y que proceden todos de una misma secuencia cuyo espacio de soluciones se desea explorar.
- Sin una etapa de aprendizaje genético. El proceso de aprendizaje genético no vuelve a incluir a su vez otro proceso de aprendizaje genético.
- Con un conjunto de operadores de generación de la población, cruce y mutación sobre los que se han realizado modificaciones que sólo afectan a esta etapa. Normalmente estas modificaciones han sido en un sentido tendente a reducir complejidad y/o restricciones.

Las modificaciones realizadas se han introducido debido a que el proceso de aprendizaje tiene un objetivo exploratorio en un espacio de soluciones reducido. En este aspecto es menos ambicioso, y por lo tanto busca un enfoque primado por una ejecución de corta duración, y con la mayor libertad de acción posible. Este enfoque debería implicar la eliminación de algunas restricciones que se han planteado hasta el momento. También ha influido en la modificación el aspecto relacionado con el esfuerzo computacional. Hay que tener en cuenta este procedimiento se podría ejecutar tantas veces como Agentes-Solución hay en una población, y tantas generaciones como se produzcan. Esto podría implicar multiplicar por el tamaño de la población el tiempo de ejecución del algoritmo.

Con el fin de reducir el número de Agentes-Solución sobre los que se aplica el procedimiento de aprendizaje genético, y por consiguiente reducir el esfuerzo computacional dedicado a esta tarea, se han impuesto unas condiciones que deben ser verificadas antes de proceder su aplicación. En concreto se debe verificar que el valor del makespan de la solución en curso es menor o igual al *valor corregido* del makespan de la mejor solución existente en la población. Se entiende por valor corregido a multiplicar el valor del makespan por "factor margen". La variable "factor margen" es el mismo parámetro definido para los operadores de cruce y mutación. Con este filtro se impide que soluciones que no se encuentra entre las mejores, hasta ese momento, inicien un proceso de aprendizaje. Esta acción se realiza bajo la suposición de que a partir secuencias malas es difícil llegar a secuencias buenas, lo cual es difícil de afirmar cuando el número de generaciones es bajo, pero más acertado cuando el número de generaciones es elevado.

Una vez descrito el criterio para la aplicación del procedimiento se va a proceder a explicar brevemente cada unas las acciones o módulos en los que se ha dividido, tal y como se ha

hecho con el resto del algoritmo. Para ello se va a seguir el enfoque conceptual comentado basado en la Figura 8-24. La propuesta de procedimiento de aprendizaje genético que se realiza en este trabajo se diferencia de la de (Zhong, W. C. et al. 2004) en la implementación que se realiza de las diferentes funcionalidades del algoritmo en cada caso.

Generar Población

El nuevo Algoritmo Genético establece su propia población, que se estructura también en Agentes-Solución que forman una matriz de tamaño sLxSL. Para diferenciarla de la matriz base o principal a esta nueva matriz se le denomina matriz de aprendizaje. El valor establecido para sL es 4 en este caso. Se ha considerado que este tamaño permite alcanzar una velocidad de ejecución alta al mismo tiempo que se examina un conjunto elevado de soluciones próximas a la solución original.

Para originar las secuencias que van a representar a las soluciones de los dieciséis nuevos Agentes-Solución que forman parte de la matriz de aprendizaje no se emplea el algoritmo voraz como se hacía en el caso general. En este caso, se elige la secuencia que representa al Agente-Solución para el cual se realiza el aprendizaje genético, y se realizan diversas modificaciones sobre la misma hasta dotar a cada Agente-Solución de la matriz de aprendizaje de una secuencia. El procedimiento empleado sería el siguiente:

1. En la primera columna de la matriz de aprendizaje el elemento de la primera fila es una copia la secuencia del Agente-Solución para el cual se realiza el proceso de aprendizaje genético. Para obtener la secuencia de la fila 2 se realiza un desplazamiento a la derecha de la secuencia empleada en la fila 1, ver Figura 8-25. Para obtener la secuencia de la fila 3 se realiza un desplazamiento a la izquierda de la secuencia empleada en la fila 1, ver Figura 8-26. Para obtener la secuencia de la fila 4 se realiza un intercambio entre las posiciones pares e impares de la secuencia empleada en la fila 1, ver Figura 8-27.

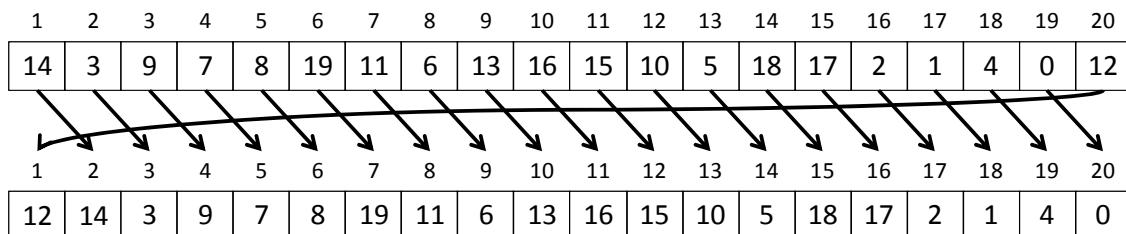


Figura 8-25. Desplazamiento a derechas de la secuencia original para la obtención del elemento de la fila 2 (elaboración propia).

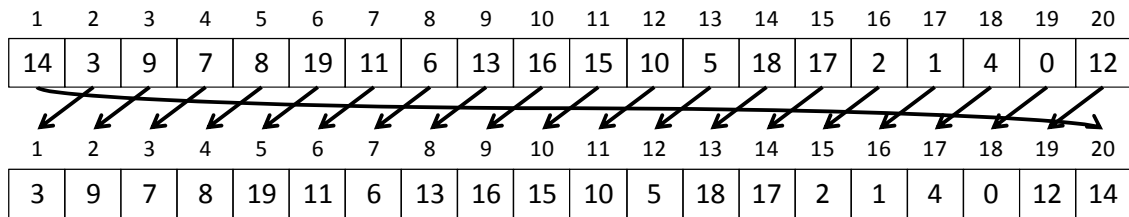


Figura 8-26. Desplazamiento a izquierdas de la secuencia original para la obtención del elemento de la fila 3 (elaboración propia).

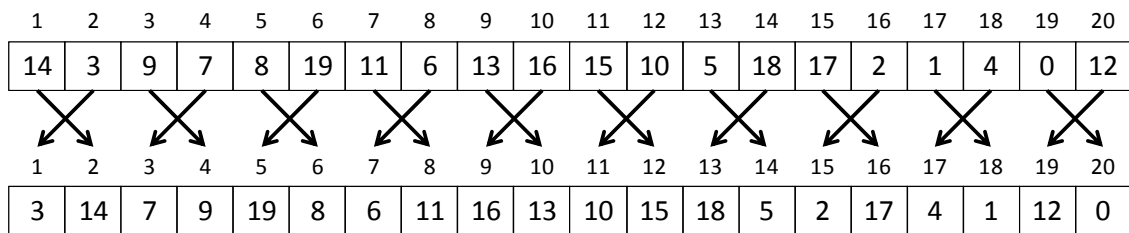


Figura 8-27. Intercambio de posiciones pares e impares de la secuencia original para la obtención del elemento de la fila 4 (elaboración propia).

- En el resto de columnas de la matriz de aprendizaje el elemento de la primera fila se obtiene mediante un intercambio aleatorio entre dos posiciones de la secuencia de la columna 1, fila 1, ver Figura 8-28. Para obtener las secuencias del resto de filas de cada columna se realizan las mismas operaciones que en la columna uno, es decir: Para obtener la secuencia de la fila 2 se realiza un desplazamiento a la derecha de la secuencia empleada en la fila 1. Para obtener la secuencia de la fila 3 se realiza un desplazamiento a la izquierda de la secuencia empleada en la fila 1. Para obtener la secuencia de la fila 4 se realiza un intercambio entre las posiciones pares e impares de la secuencia empleada en la fila 1. Siempre tomando como referencia la fila 1 de cada columna.

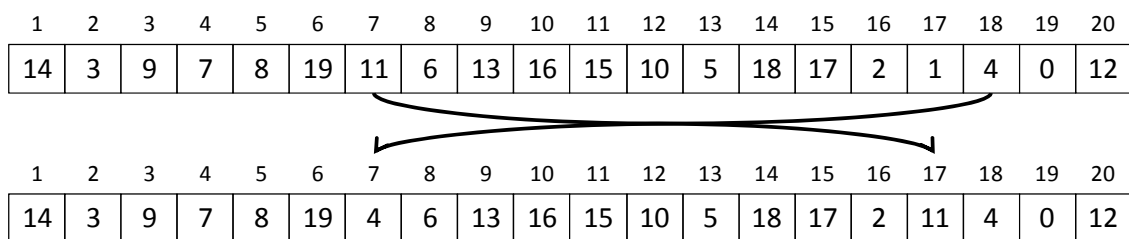


Figura 8-28. Intercambio de dos posiciones de la secuencia original para la obtención del elemento de la fila 1, en las columnas 2, 3 y 4 (elaboración propia).

Con esta población inicial lo que pretende es obtener un conjunto de soluciones muy similares a la solución que se está analizando, y de la cual se desea explorar un espacio de soluciones próximo. No importa si converge en un óptimo local, ya que en realidad, ese es el objetivo.

Operador de Cruce

En el caso del proceso de aprendizaje genético no se da opción a que los Agentes aprendan cual puede ser el operador de cruce más recomendable en función de su experiencia. Esto es así debido a que se prevee una aplicación en la que se obtendrán pocas generaciones, y por lo tanto la experiencia que puede acumular la población es reducida, y quizás poco significativa. Lo más interesante en este caso es aplicar un operador rápido en su ejecución y que por sus características tienda a converger rápidamente a un óptimo local. En este caso se ha optado por aplicar el operador NCO, ya que de los dos descritos parece ser un candidato que encaja bastante bien con los requerimientos.

El operador de cruce NCO que se emplea tiene un funcionamiento idéntico en cuanto a la generación de soluciones. No obstante, en lo que a la aplicación del mismo en relación con el relevo generacional existe una diferencia: No se gestiona ninguna lista de valores ocupados, y por lo tanto, los valores de makespan de la población, y en general las secuencias de las soluciones, se pueden repetir. El resto de condiciones que se deben cumplir para realizar el relevo generacional, descritas en el apartado 8.3.2.2.8, se deben cumplir.

Operador de Mutación

El operador de mutación que se emplea tiene un funcionamiento idéntico al descrito en el apartado 8.3.2.2.9 en cuanto a la generación de modificaciones en la secuencia. No obstante, en lo que a la aplicación del mismo en relación con el relevo generacional existe una diferencia. No se gestiona ninguna lista de valores ocupados, y por lo tanto, los valores de makespan de la población, y en general las secuencias de las soluciones, se pueden repetir. El resto de condiciones para realizar el relevo generacional, descritas en el apartado 8.3.2.2.9, se deben cumplir.

Condición de Finalización

La condición de finalización o fin del algoritmo del procedimiento de aprendizaje se ha establecido en función de la capacidad de mejora de las soluciones encontradas. El procedimiento finalizará cuando se contabilice un número de generaciones determinado sin

mejorar la mejor solución alcanza en un instante dado. Dicho valor será establecido en la fase de parametrización.

Relevo Generacional

Se debe entender por relevo generacional a la decisión que se debe tomar una vez ha finalizado el procedimiento de aprendizaje genético, y se ha obtenido una solución, posiblemente diferente de la original. Una vez el procedimiento de aprendizaje genético ha terminado éste proporcionará una secuencia que se corresponderá con la solución de menor makespan encontrada durante la ejecución del mismo. Dicha solución podrá ser mejor, igual que la solución inicial, ya que el proceso de aprendizaje debe garantizar que el makespan nunca empeorará. En todo caso, se debe decidir si la solución obtenida sustituye a la original o no.

Igual que ocurría en los procesos de cruce y mutación se han establecido una serie de condiciones para realizar el relevo generacional. Se entiende que si la mejor solución encontrada en el proceso de aprendizaje genético cumple una serie de condiciones, ésta ocupará el lugar de la actual solución. Las condiciones son las mismas que se establecieron para el operador de cruce general, es decir:

1. Se verifica si el makespan de obtenido con la secuencia de la nueva solución generada con el proceso de aprendizaje genético del Agente-Solución en curso es diferente al de todos sus vecinos. Si el valor del makespan se repite en algún caso se descarta la solución encontrada, y no se realiza el relevo generacional.
2. Si se ha pasado el filtro del punto anterior, se verifica que el valor del makespan de la nueva solución generada es menor o igual al *valor corregido* del makespan de la solución actual. Se entiende por valor corregido a multiplicar el valor del makespan por "factor margen". La variable "factor margen" es el mismo parámetro definido para los operadores de cruce y mutación.
3. Si se ha superado la restricción del punto anterior, se verifica que el valor del makespan de la nueva solución no se encuentre repetido en el conjunto de la población, y no sólo en los vecinos. Para ello se emplea la lista de valores ocupados que ya ha sido comentada y empleada en los operadores de cruce y mutación.

A su vez, en este procedimiento se produce la actualización de dos variables que se emplean en diferentes etapas del algoritmo. En primer lugar, el factor "Riesgo Mínimo" vuelve a tomar

su valor original para el Agente-Solución en curso, sólo por el hecho de encontrar, en el proceso de aprendizaje, una solución diferente que la actual. Esto se hace así porque se considera que si se ha podido encontrar una solución diferente, es necesariamente porque es mejor que la actual, y eso significa que el material genético que se contiene en dicha secuencia tiene en potencia capacidad de generar soluciones mejores. Por este motivo se restituye el valor original de “riesgo mínimo” para favorecer la aplicación del operador de cruce. En segundo lugar, si se produce una sustitución que lleve acarreada la mejora de la mejor solución de la población se borra la lista de valores ocupados.

8.3.2.2.11. Operador de Reinicio de la Población

La operación de reinicio de la población es aquella por la cual gran parte de las secuencias que representan a las soluciones de la población son reemplazadas por otras nuevas. El objetivo de este operador es crear un cambio sustancial en la composición de la población que permita que el algoritmo salga de un óptimo local en el que puede haberse situado. Los Algoritmos Genéticos suelen tender a homogeneizar la composición de la población llegando a centrar su exploración en un área muy reducida del espacio de soluciones. Para evitar esta situación, entre otras acciones suele ser recomendable aplicar un operador de reinicio de la población.

Este operador se aplica cuando se detecta que la mejor solución disponible en el conjunto de la población no se ha modificado en un número determinado de iteraciones. A dicho número de iteraciones se le ha denominado “**Generaciones sin Mejora**”, considerándolo un parámetro que debe introducir el usuario. Dicho parámetro será seleccionado en la etapa de parametrización del algoritmo. Una vez que se ha alcanzado el valor umbral de generaciones sin mejorar, el operador de reinicio, se aplica a todos los elementos de la población de forma secuencial, tal y como se representa en la Figura 8-8. Se trata de ir analizando a cada uno de los Agentes-Solución, y si cumple los requisitos establecidos, se genera una nueva secuencia que sustituye a la actual en dicho Agente-Solución.

El procedimiento por el cual se obtiene la nueva secuencia es el mismo que se estableció para la generación de la población inicial, descrito en el apartado 8.3.2.2.3. Pero como ya se ha dicho, antes de obtener una secuencia es necesario saber si el Agente-Solución en cuestión cumple los requisitos. Estos son los siguientes:

1. Si el makespan de la solución del Agente-Solución es mayor que el valor corregido del makespan de la mejor solución de la población, se genera siempre una nueva secuencia, y se sustituye la actual por la nueva.

2. En caso de no cumplirse la restricción anterior, se realizará un sorteo en el cual el Agente-Solución tiene un 70% de posibilidades de ser ganador. Si el Agente-Solución gana se genera una nueva secuencia, y se sustituye la actual secuencia por la nueva. En cualquier otro caso, no se realiza ningún cambio en el Agente-Solución en curso.

Hay que recordar que si se inicia el proceso de reinicio de la población, independientemente de cuantos individuos sean sustituidos, la lista de valores ocupados que se emplea en el operador de cruce y de mutación se borra.

```
// Aplicación del operador de reinicio de la población

Si (generacion_actual==Generaciones_sin_mejorar) {
    Borrar_lista_ocupado()
    //Se recorre toda la matriz básica
    Para i=1 hasta L
        Para j=1 hasta L {
            Sustituye=falso
            descendiente=nulo
            Si (makespan_actual>makespan_poblacion*Factor_Margen) sustituye=cierto
            Sino Si (aleatorio (0,1)<0.7) sustituye=cierto
            Si (sustituye==cierto){
                descendiente=Aplica_algoritmo_inicial()
                Cambiar_secuencia (solución_actual, descendiente)
            }
        }
    }
}
```

Ilustración 8-11. Pseudocódigo para la aplicación del operador de reinicio de la población (elaboración propia).

La Ilustración 8-11 contiene el pseudocódigo correspondiente a la aplicación del operador de reinicio. Con las restricciones impuestas para realizar el cambio de secuencia, se pretende reservar algunas de las soluciones de mejor calidad para no iniciar una nueva fase con la misma información genética que se tenía en el instante inicial. En general, en un proceso de sustitución se debe verificar que no elimina a la mejor solución de la población. Esto también ocurre en el procedimiento de reinicio para prevenir la pérdida del mejor makespan de toda la población.

8.3.2.3 Implementación del Algoritmo

Para la implementación del algoritmo SMAGA ha sido necesario combinar dos elementos fundamentales: Un Lenguaje de Programación y una Plataforma de Agentes.

El algoritmo SMAGA ha sido implementado utilizando el lenguaje de programación JAVA para su desarrollo, en concreto se ha utilizado el entorno de desarrollo de libre distribución ECLIPSE versión 3.4, disponible en <http://www.eclipse.org/platform>. Se ha seleccionado el lenguaje de programación JAVA por ser en el cual están desarrollando la mayoría de las Plataformas de Agentes que se han encontrado. Esta elección permite la posibilidad de realizar cambios futuros en la Plataforma de Agentes con una menor inversión de tiempo.

En cuanto a la Plataforma de Agentes seleccionada esta ha sido JADE, en concreto se ha empleado la versión 3.5. Esta Plataforma, como ya se comentó en el capítulo 4, es una de las más completas en cuanto a funcionalidad proporcionada, al mismo tiempo que cumple los estándares internacionales que regulan el desarrollo de aplicaciones con Agentes.

El proceso de desarrollo se puede considerar como relativamente rápido dentro de la dificultad. Se destaca como la comprensión del funcionamiento asíncrono de los Agentes ha sido el aspecto en el que ha sido necesario invertir más tiempo. Durante el proceso de implementación se desarrollaron varias versiones parciales (borradores) del algoritmo SMAGA, cada una de las cuales suponía una mejora sobre la anterior.

8.4 Desarrollo del Diseño Experimental

Una vez diseñado e implementado el algoritmo SMAGA propuesto para realizar la Programación de la Producción Predictiva, en un Taller de Flujo Híbrido con Tiempos de Cambio de Partida Dependientes de la Secuencia, y que por lo tanto está disponible el objeto de análisis, es cuando el experimentador está en condiciones de seguir desarrollando los diferentes pasos de que consta el diseño experimental. Se recuerda que el primer paso ya se ha desarrollado, dado que se trataba de entender el problema, sin el cual no se hubiera podido implementar el algoritmo SMAGA. El siguiente paso consiste en establecer los parámetros que potencialmente pueden influir en el proceso, aspecto que se desarrollará en el apartado 8.4.1, posteriormente se establecerá un plan experimental, aspecto que se abordará en el apartado 8.4.2, y el último paso del proceso de experimentación consistirá en el análisis de los resultados obtenidos, que se realizará en el apartado 8.4.4.

8.4.1 Parametrización

En este apartado se desarrolla en segundo paso de los cinco de que consta el diseño de experimentos, tal y como se definió en el apartado 8.2.3. Se trata de de identificar aquellos factores que se consideran relevantes o que potencialmente pueden influir en la función objetivo que se desea estudiar. En este caso los factores se han expresado en forma de parámetros.

Dado que durante la descripción del algoritmo se han introducido un amplio conjunto de parámetros lo primero que se presenta es un resumen de los mismos. Se puede decir que éstos se dividen en dos grupos: Parámetros fijos y parámetros ajustables.

Los parámetros fijos son aquellos para los cuales se ha decidido establecer un único valor, y que por lo tanto no es necesario examinar alternativas posibles. En la Tabla 8-10 se muestra una recopilación de aquellos parámetros que se han considerados fijos. En la tabla se muestra el nombre del parámetro, el proceso al que está vinculado y el valor asignado.

Parámetros Fijos	Proceso al que está Vinculado	Valores Establecidos
Tamaño de la Matriz	Establecimiento de la población	Matriz base (LxL): 6 x 6 Matriz Aprendizaje (sL x SL): 4 x 4
Cuota de Dominio	Algoritmo de generación de soluciones (Kauffman)	0,8
Evolución de Riesgo Mínimo	Aplicación del operador de cruce.	Permite primar las soluciones nuevas frente al resto.
Valores Iniciales del Factor de Riesgo	Restricción en la aplicación del operador de cruce.	(0,2, 0,4, 0, 6 y 0,8) elegidos aleatoriamente
Valor de la reducción del Factor de Riesgo	Restricción en la aplicación del operador de cruce.	0,05 en cada generación
Valor Inicial del Factor Límite	Selección del operador de cruce.	0,5
Valor de Incremento del Factor Límite	Selección del operador de cruce.	0,01
Cota Inferior del Factor Límite	Selección del operador de cruce.	0,1
Cota Superior del Factor Límite	Selección del operador de cruce.	0,9
Longitud de la Lista de Valores Ocupados	Relevo generacional	500
Condición de Finalización del Aprendizaje Genético	Aprendizaje genético	0,5*GsM (GsM está definido en la tabla de parámetros ajustables)

Tabla 8-10. Conjunto de parámetros fijos (elaboración propia).

De la misma forma en la Tabla 8-11 se muestra una recopilación de todos los parámetros que se han considerado ajustables, y que por lo tanto necesitan de un análisis para identificar cual es la respuesta del algoritmo SMAGA frente a diversos valores. En el caso de la tabla de parámetros ajustables se detalla el nombre del mismo, el proceso con el que tiene relación, y el objetivo que se persigue con éste.

Parámetros Ajustables	Proceso al que está Vinculado	Objetivo
Generador Dinámico	Aprendizaje en la aplicación del Algoritmo de generación de soluciones (Algoritmo Kauffman).	Permitir o no el aprendizaje
Evolución del Factor Límite	Aplicación de la selección del operador de cruce. Aprendizaje en la aplicación del operador de cruce (NCO / SB2OX)	Permitir o no el aprendizaje
Evolución de Riesgo Mínimo	Aplicación del operador de cruce.	Permite primar las soluciones nuevas frente al resto.
Riesgo Mínimo (RM)	Restricción en la aplicación del operador de cruce.	Establece un valor mínimo, es decir, la probabilidad más baja que se puede asignar para permitir o no un cruce.
Generaciones sin Mejora (GsM)	Se aplica en el operador de Reinicio	Indica cada cuantas generaciones sin mejorar el mejor resultado se reinicia la población.
Factor de Mutación (FdM)	Restricción en la aplicación del operador de mutación.	Establece un valor relacionado con la probabilidad en base a la cual se permitir o no mutar a una solución.
Factor Margen (FmG)	Aplicación del operador de cruce y mutación.	Restringe el ámbito de exploración de las nuevas soluciones respecto las soluciones existentes.

Tabla 8-11. Conjunto de parámetros ajustables (elaboración propia).

Antes de considerar los valores que se analizarán para cada parámetro ajustable es necesario precisar, que con el fin de realizar un estudio más claro, se han considerado dos versiones del algoritmo SMAGA, en función de los valores asignados a dicho conjunto de parámetros: SMAGA-1 y SMAGA-2. Esta división se justifica debido a que en la fase de análisis comparativo, que se realizará posteriormente, se desea contrastar los resultados que se obtengan con la nueva propuesta en su versión "sin aprendizaje", y en la versión completa con el empleo de las capacidades de "aprendizaje". Desde el punto de vista del paradigma de los Sistemas Multiagente, se puede decir que la no incorporación de los procesos de aprendizaje en el

algoritmo SMAGA equivale, en este caso concreto, al desarrollo de un software basado en Agentes, pero no a un software basado en un Sistema Multiagente^{ix}. Aunque en ambos casos se diseñan las propuestas en base a las características elementales de los Agentes, la diferencia fundamental es que los primeros no contemplan, ni en el diseño ni en la implementación, las conductas sociales propias de la pertenencia a un grupo, que es precisamente una de las características que se deseaba incorporar a la propuesta que se presenta.

Se denomina algoritmo SMAGA-1 a la versión que considerará las opciones indicadas en la columna correspondiente de la Tabla 8-12 . Se trata de un algoritmo que no aplica ninguna de las capacidades de aprendizaje incorporadas al Sistema Multiagente, excepto el proceso de aprendizaje genético, que es implícito al algoritmo y, que siempre se realiza. Por este motivo los parámetros “Generador Dinámico”, “Evolución del Factor Límite” y “Evolución del Riesgo Mínimo” están siempre desactivados, al mismo tiempo que el parámetro “Riesgo Mínimo” (**RM**) no necesita tener asignado ningún valor, al no realizarse modificaciones en la variable “factor riesgo”. Se denomina SMAGA-2 a la versión del algoritmo que aplica todas las capacidades de aprendizaje, por lo que los parámetros “Generador Dinámico”, “Evolución del Factor Límite” y “Evolución del Riesgo Mínimo” estarán siempre activados. Así mismo, el parámetro “Riesgo Mínimo” (**RM**) debe disponer de un valor, permitiéndose que la probabilidad de aplicación del operador de cruce se reduzca, bien hasta el 20%, o bien hasta el 40%. SMAGA-1 será considerado como un sistema basado en Agentes, situado en un escalón inferior o menos desarrollado desde el punto de vista de la tecnología de Agentes, que SMAGA-2 que se considerará un Sistema Multiagente.

Valores de los Parámetros Ajustables	SMAGA-1	SMAGA-2
Generador Dinámico	Desactivado	Activado
Evolución del Factor Límite	Desactivado	Activado
Evolución de Riesgo Mínimo	Desactivado	Activado
Riesgo Mínimo (RM)	-	Valores
Generaciones sin Mejora (GsM)	Valores	SMAGA-1
Factor de Mutación (FdM)	Valores	SMAGA-1
Factor Margen (FmG)	Valores	SMAGA-1

Tabla 8-12. Estructura de los factores ajustables para los algoritmo SMAGA-1 y SMAGA-2 (elaboración propia).

^{ix} La diferencia entre un algoritmo basado en agentes y uno basado en un Sistema multiagente se describe en el capítulo 4.

Además de los parámetros que diferencian a ambos algoritmos, existen otros parámetros, que deben ajustarse. Como se puede observar en la Tabla 8-12, SMAGA-1 dispone de 3 parámetros en los que se indica “Valores”, lo que significa que en el desarrollo del plan experimental se deben establecer los niveles que se desean analizar de los mismos. El algoritmo SMAGA-2 dispone también de los 3 parámetros mencionados, pero en este caso se ha elegido la estrategia de asignarle los mismos valores, en cada conjunto experimental, que en su momento sean seleccionados para SMAGA-1 por su idoneidad. De esta forma se podrá considerar, en la fase de evaluación comparativa, que la única diferencia que existe entre SMAGA-1 y SMAGA-2 es su capacidad de aprendizaje, que es el objetivo con el que se realizó la diferenciación entre ambos. Otra opción, que no se contempló en este trabajo, hubiera sido la parametrización completa de SMAGA-2, independiente de SMAGA-1. Esta última opción no fue la contemplada porque se pensó que no mostraría con claridad cual es el peso de los diferentes grados de incorporación de la tecnología de Agentes en el algoritmo. **Hay que tener siempre en cuenta que uno de los objetivos fundamentales de este capítulo es identificar la validez del uso de Sistemas Multiagente, y no tanto el refinamiento del propio algoritmo propuesto.**

8.4.2 Plan de Experimentación y Condición de Finalización

Establecer y ejecutar el plan experimental constituye el tercer paso en el diseño de experimentos, y supone la determinación de las condiciones en las que posteriormente se van a ejecutar los experimentos.

En este caso el plan experimental está condicionado por el enfoque que se ha realizado del estudio centrándose en el análisis de 3 factores para el algoritmo SMAGA-1, de un solo factor para el algoritmo SMAGA-2. El resto de parámetros de cada algoritmo indicados en la Tabla 8-10 se toman un valor único determinado en función del algoritmo.

Para el algoritmo SMAGA-1 se ha establecido un plan factorial (2^3) de forma que se examinarán cada uno de los tres factores que se desea analizar GsM, FdM, y FmG, cada uno de los cuales dispondrá de dos niveles (Gsm: 50, 200; FdM: 0.1, 0.3; FmG: 1.05, 1.2). El parámetro “Generaciones sin Mejora” (**GsM**) que provocará un reinicio de la población si han pasado, bien 50, o bien 200 generaciones sin que el valor del mejor makespan no se haya mejorado. El parámetro “Factor de Mutación” (**FdM**) que establece el porcentaje de éxito del operador de mutación, o bien en el 10%, bien en el 30% de las ocasiones. Y el parámetro “Factor Margen” (**FmG**) permitirá establecer el margen empleado en el operador de cruce, de mutación o de aprendizaje genético en el relevo generacional situándolo, bien alrededor del 5%, o bien alrededor del 20% del valor de la solución estudiada. Dicho plan se ejecutará de forma independiente para cada uno de los 16 conjuntos experimentales establecidos en la Tabla 8-7.

Debiéndose determinar que parámetros son significativos, y en su caso, los valores con los que mejor makespan promedio se obtiene para cada uno de los conjuntos experimentales.

En el caso del algoritmo SMAGA-2, dado que sólo existe un factor a analizar, RM, ya que el resto toman valores fijados con anterioridad, se ha planteado un simple análisis comparativo entre los dos niveles o valores de dicho factor. Seleccionando el valor que mejor resultado proporcione en términos de makespan promedio. En todo caso, e igual que se ha planteado para SMAGA-1, se ejecutará el algoritmo con cada uno de los dos valores para los 16 conjuntos experimentales de forma independiente.

La Tabla 8-13 es una réplica de la Tabla 8-12 en la que se han incluido los valores que se desean analizar para cada factor o parámetro.

Valores de los Parámetros Ajustables	SMAGA-1	SMAGA-2
Generador Dinámico	Desactivado	Activado
Evolución del Factor Límite	Desactivado	Activado
Evolución de Riesgo Mínimo	Desactivado	Activado
Riesgo Mínimo (RM)	-	0.2, 0.4
Generaciones sin Mejora (GsM)	50, 200	SMAGA-1
Factor de Mutación (FdM)	0.1, 0.3	SMAGA-1
Factor Margen (FmG)	1.05, 1.2	SMAGA-1

Tabla 8-13. Niveles asignados a los parámetros ajustables para los algoritmo SMAGA-1 y SMAGA-2 (elaboración propia).

En ambos casos, parámetros fijos o ajustables, los valores elegidos podían haber sido múltiples, y seguramente cada diseñador hubiera elegido su propio conjunto de valores. No obstante, en función de la descripción realizada del algoritmo, en donde se especifica con detalle la función de cada parámetro, se considera que se han seleccionado valores razonables. Será en todo caso el análisis de los resultados obtenidos el que determine la bondad de dichos valores.

Cabe recordar que tanto para el algoritmo SMAGA-1, como para el algoritmo SMAGA-2, cada conjunto experimental consta de 3 grupos, y cada grupo consta de 5 ejemplares. También en ambos casos, se ha decidido realizar dos repeticiones de cada ejecución, lo que supone un total de 3 ejecuciones por ejemplar y combinación de niveles de los factores. Esto implica que **para el caso de SMAGA-1 se realizarán** 16 (conjuntos experimentales) x 3 (grupos) x 5 (ejemplares) x 8 (tratamientos) x 3 (ejecuciones)= **5.760 ejecuciones**. En **el caso de SMAGA-2 se realizan** 16 (conjuntos experimentales) x 3 (grupos) x 5 (ejemplares) x 2 (tratamientos) x 3 (ejecuciones)= **1.440 ejecuciones**.

Aunque ya se mencionó en el apartado 8.2.3.2, se insiste en que la variable de respuesta que se desea evaluar es la variable IPSOVEPT, que tendrá en cuenta el IPSOVEP promedio para un conjunto experimental completo. Por lo tanto, **el resultado final será la identificación de los parámetros que resulten significativos para cada conjunto experimental, y la identificación, en su caso, del valor que proporciona un IPSOVEPT más bajo, y por lo tanto un makespan más reducido.**

Finalmente un aspecto que debe considerarse con la misma relevancia que el tratamiento de los parámetros, y que no ha sido comentado hasta este momento es el de la duración de la ejecución del algoritmo. Se trata de seleccionar el número de iteraciones que se van a ejecutar, que en este caso coincide con el número de generaciones. **Después de realizar diversas evaluaciones se ha decidido fijar la duración de las ejecuciones en 5.000 iteraciones.** Dadas las características del algoritmo, donde cabe recordar que existe un sistema de matrices que contiene a la población, puede suponer alrededor de 250.000 ejecuciones de algún operador de cruce o mutación, y alrededor de 10.000 sustituciones de las soluciones de la población. Hay que tener en cuenta, que debido a la influencia del azar, los datos sobre las ejecuciones de los operadores y sobre los cambios en las soluciones pueden variar sensiblemente, por lo que las cifras son simplemente orientativas. Lo importante es que se ha observado en los estudios previos que con estos valores existe un equilibrio aceptable entre la duración de la ejecución y la estabilidad de los resultados. **Se ha observado que los resultados obtenidos, en ciertas pruebas, no presentaban una mejora significativa por el incremento del número de ejecuciones respecto del fijado para este trabajo.**

8.4.3 Procedimiento de Ejecución los Experimentos

Para el desarrollo de los experimentos se han utilizado cuatro ordenadores tipo PC con procesador Intel Core 2 a 2.66 GHz, con memoria de 2 GByte, y sistema operativo Windows XP Service Pack 3. Cada conjunto experimental se ha ejecutado de forma aleatoria en uno de los cuatro ordenadores. El tiempo de ejecución de cada ejemplar es variable, y podemos encontrar ejecuciones que tardan 1 minuto y otras que llegan a tardar más de 10 minutos. **El tiempo promedio de ejecución es de unos 5 minutos, lo que ha implicado dedicar más de 1 mes a la realización de los experimentos.** Obviamente, este tiempo se puede considerar muy elevado con respecto a otras propuestas que obtienen resultados en segundos. Existen varias alternativas que permitirían reducir el tiempo de ejecución no obstante esta línea de trabajo no ha sido considerada en este trabajo. En este sentido se ha considerado aceptable el tiempo en relación con la tecnología empleada en la que existe una sobre carga adicional en la computación de procesos.

Al final de la ejecución de cada ejemplar el Programa desarrollado ha incorporado los resultados en un fichero de texto. En dicho fichero se han escrito diversos datos, algunos de los cuales han sido empleados para el análisis de los resultados, otros como simples elementos de identificación, y otros se han obtenido con espíritu de realizar otros análisis que sirvan de complemento a este trabajo. Los datos recogidos han sido los siguientes:

- Valor de IPSOVEP del ejemplar.
- Diferencia entre el valor más alto y el más bajo alcanzado durante la ejecución del algoritmo en el factor límite que afecta a la probabilidad de selección de operador de cruce. Se dispone de la media y de la desviación típica del promedio de los Agentes de la población.
- Porcentaje de veces que la etapa dominante al final de la ejecución del algoritmo ha sido seleccionada como parámetro para el algoritmo de inicio o reinicio de la población.
- Valor del mejor makespan obtenido en el ejemplar.
- Identificación del ejemplar y del conjunto experimental ejecutado.
- Identificación del valor de los parámetros ejecutados.
- Fecha y hora de inicio y fin de la ejecución del ejemplar.
- Secuencia correspondiente al mejor makespan alcanzado.

Este proceso se ha repetido para cada una de las réplicas de forma idéntica. Una vez finalizadas todas las ejecuciones de cada conjunto experimental los datos se han consolidado en una hoja Excel en la cual se dispuso el valor de IPSOVEPT para cada combinación de niveles de los factores, en cada repetición.

Tanto para el algoritmo SMAGA-1, como SMAGA-2 los datos se analizan mediante Statgraphics Plus para Windows versión 5.0 (ver <http://www.statgraphics.com>). En el caso de SMAGA-1, se ha creado un diseño de experimentos mediante la opción de "Screening", con una variable de respuesta y 3 factores, eligiendo un diseño factorial de 2^3 , con dos réplicas. La codificación de los factores sigue la misma nomenclatura expresada hasta ahora es decir GsM, para el parámetro "Generaciones sin Mejora", FdM, para el parámetro "Factor de Mutación", y FmG, para el parámetro "Factor Margen". En el caso de SMAGA-2, se ha creado un diseño de experimentos mediante la opción de "Multi-Factor Categorical", con una variable de respuesta y dos factores, eligiendo un diseño factorial de personalizado, con dos réplicas. La codificación

de los factores sigue la misma nomenclatura expresada hasta ahora es decir RM, para el factor “Riesgo Mínimo”, y Conjunto, para el factor “Conjunto Experimental”.

8.4.4 Análisis de los resultados

Para realizar el análisis de los resultados obtenidos, es decir, de los valores IPSOVEPT de cada conjunto experimental, se emplea el análisis de la Varianza mediante ANOVA. Tal y como se indicó en el apartado 8.2.3, antes de estudiar el significado de los resultados, es requisito imprescindible comprobar el cumplimiento de las distintas hipótesis, para lo cual en esta tesis se aplicarán fundamentalmente métodos gráficos.

Como ya explicó en el apartado 8.2.3, se empleará un “gráfico de probabilidad normal (Y_i, F_i)”, para el análisis de la normalidad, el “gráfico de residuos (X_t, \bar{u}_t) y (\hat{Y}_t, \bar{u}_t)”, para el análisis de la homocedasticidad y la linealidad, y el “gráfico de residuos (t, \bar{u}_t)”, para el análisis de la autocorrelación. Así mismo se utilizará la matriz de correlaciones para analizar la multicolinealidad.

A modo de ejemplo se desarrollará el contraste de las hipótesis para la ANOVA correspondiente al conjunto experimental “SSD10_P13_20”. El conjunto de gráficos correspondientes al resto de conjuntos experimentales se encuentran en el anexo III.

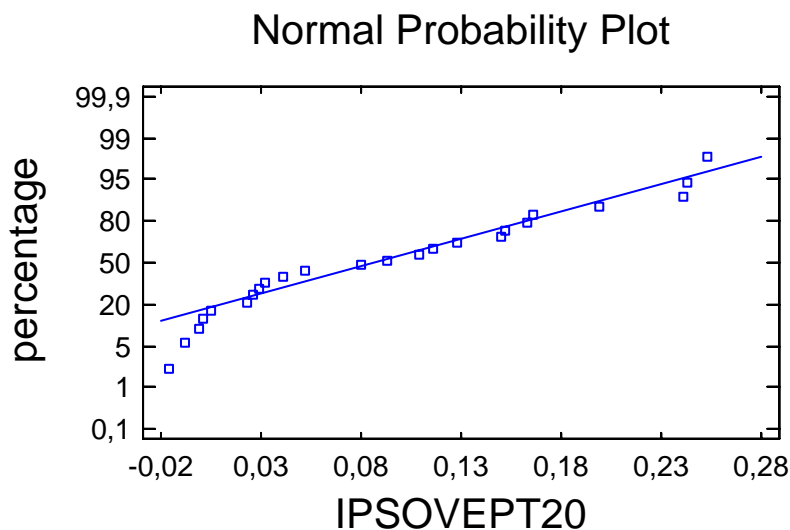


Figura 8-29. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPSOVEPT del conjunto SSD10_P13_20 en relación a las frecuencias relativas acumuladas.

Como se puede observar en la Figura 8-29 es difícil descartar la hipótesis de que la variable se distribuye como una normal.

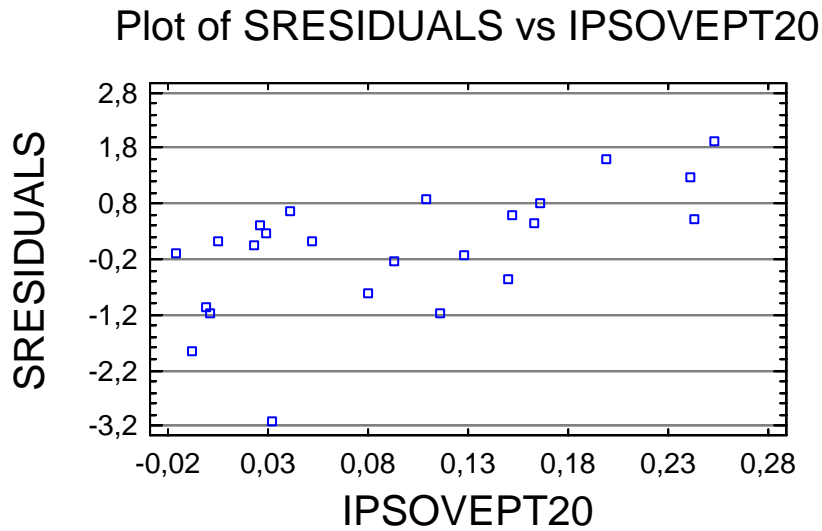


Figura 8-30. Residuos estudentizados frente a la variable IPSOVEPT del conjunto experimental SSD10_P13_20.

Como se puede observar en Figura 8-30 existe una estructura que puede considerarse aleatoria y libre de tendencia con respecto a la variable IPSOVEPT. No obstante, viendo la gráfica y siendo estricto aún no puede descartar la no linealidad. En todo caso, para analizar mejor esta última hipótesis, es necesario verificar que no existe ningún efecto debido a los factores, y así confirmar que la varianza de los errores estocásticos es similar.

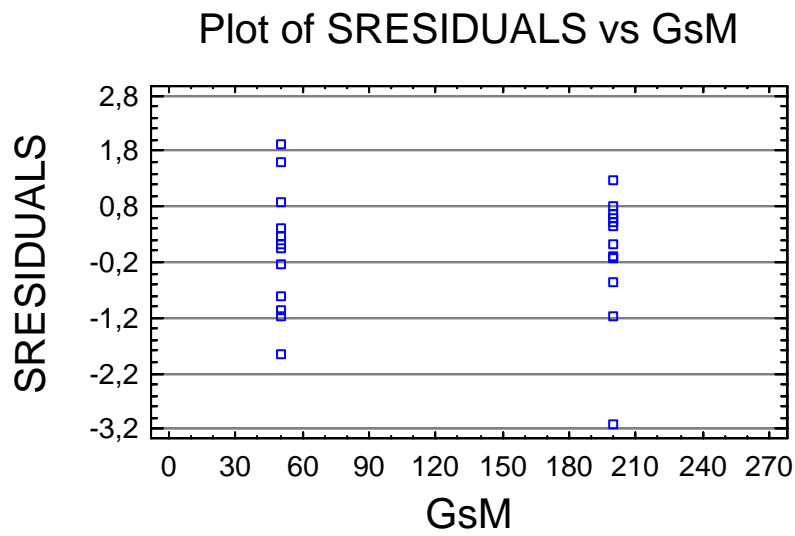


Figura 8-31. Residuos estudentizados frente a los niveles del factor GsM.

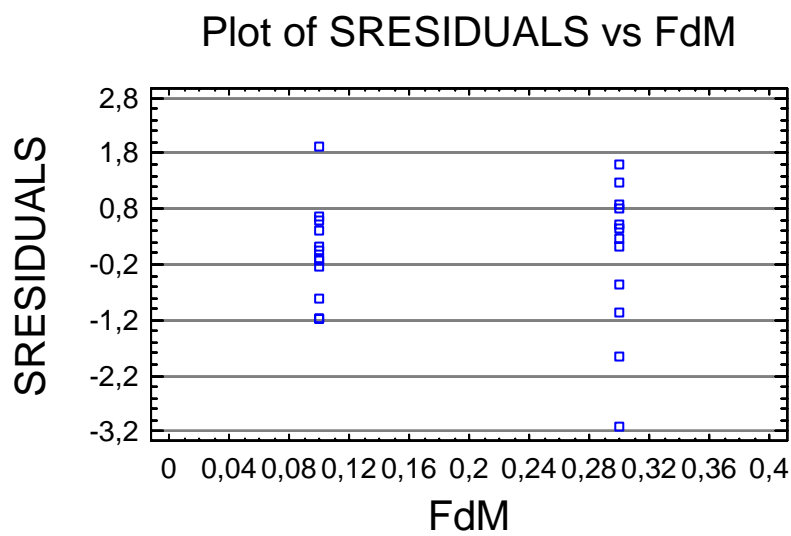


Figura 8-32. Residuos estudentizados frente a los niveles del factor FdM.

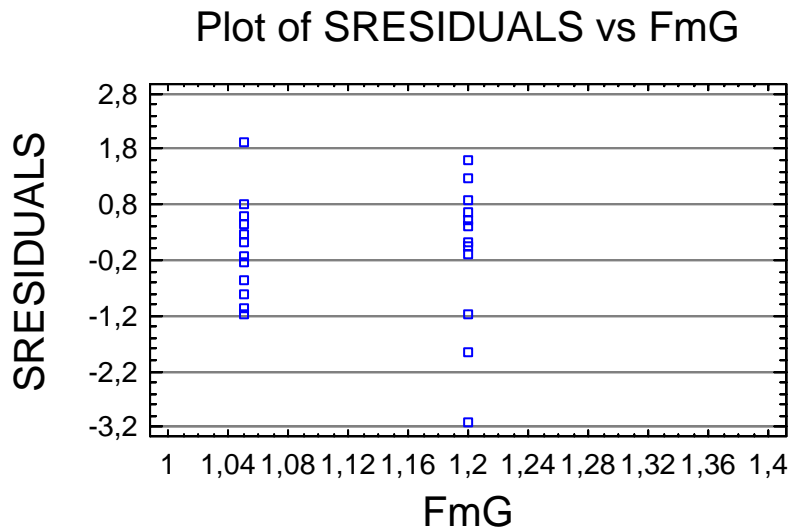


Figura 8-33. Residuos estudentizados frente a los niveles del factor FmG.

Los gráficos de los residuos contra cada variable exógena, de la Figura 8-31, Figura 8-32 y Figura 8-33, permite descartar razonablemente que exista alguna variable que cause un incumplimiento en la hipótesis de homocedasticidad, ya que se puede observar una cierta aleatoriedad en la distribución de los residuos para cada nivel.

La hipótesis más importante en el ANOVA es la de independencia, es decir, que no exista un factor oculto que esté afectando de una forma sistemática a las pruebas. En principio, dado que las pruebas se ejecutan aleatoriamente en los ordenadores, y que el entorno de computación se reinicia para cada algoritmo, es difícil que existan problemas de independencia. Analizando la matriz de correlación y graficando los residuos frente al número u orden de ejecución de las pruebas se puede ver si existe estructura en el residuo, y por tanto una falta de independencia. En la Tabla 8-14 se muestra la matriz de correlaciones y en la Figura 8-34 muestra este gráfico de análisis de los residuos.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
(1) average	1,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
(2) A:GsM	0,0000	1,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
(3) B:FdM	0,0000	0,0000	1,0000	0,0000	0,0000	0,0000	0,0000	0,0000
(4) C:FmG	0,0000	0,0000	0,0000	1,0000	0,0000	0,0000	0,0000	0,0000
(5) AB	0,0000	0,0000	0,0000	0,0000	1,0000	0,0000	0,0000	0,0000
(6) AC	0,0000	0,0000	0,0000	0,0000	0,0000	1,0000	0,0000	0,0000
(7) BC	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	1,0000	0,0000
(8) block	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	1,0000

Tabla 8-14. Matriz de correlaciones para el conjunto experimental SSD10_P13_20.

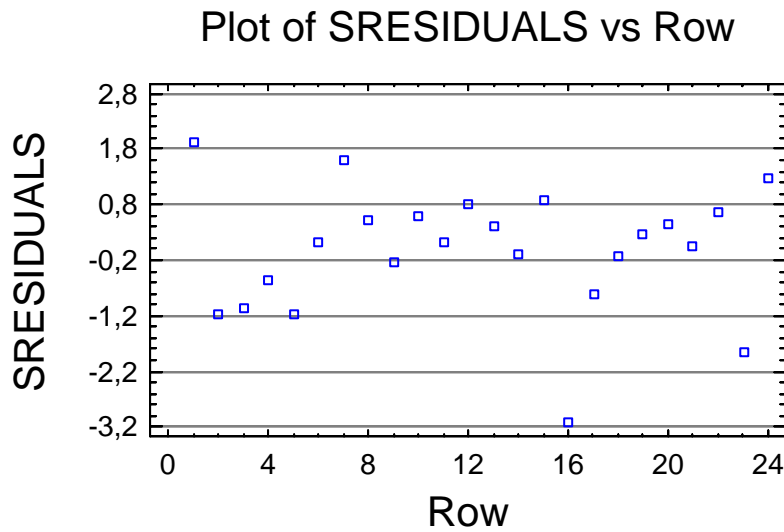


Figura 8-34. Residuos estudentizados frente al orden de ejecución.

Se puede observar en la Tabla 8-14 sólo existe un valor superior a 0,5, que indica correlación, en la diagonal por lo que se puede descartar la multicolinealidad. Así mismo en la Figura 8-34 se puede ver como no existe ningún tipo de patrón en el residuo, por lo que no hay razones para no asumir esta hipótesis. Hemos visto como se han cumplido todas las hipótesis para el experimento.

Llegados a este punto podemos analizar los resultados del experimento mediante la tabla del ANOVA que se muestra en la Tabla.

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
A:Gsm	0,0181661	1	0,0181661	4,26	0,0568
B:Fdm	0,00597277	1	0,00597277	1,40	0,2550
C:Gmf	0,00629737	1	0,00629737	1,48	0,2431
AB	0,0184553	1	0,0184553	4,33	0,0551
AC	0,00122302	1	0,00122302	0,29	0,6001
BC	0,0415501	1	0,0415501	9,74	0,0070
blocks	0,0131768	2	0,00658841	1,54	0,2454
Total error	0,0639671	15	0,00426447		
Total (corr.)	0,168809	23			

Tabla 8-15. ANOVA correspondiente al conjunto experimental SSD10_P13_20.

Observando la Tabla 8-15 vemos que disponemos de 24 datos y como vemos, tenemos 15 grados de libertad residuales. ANOVA realiza un contraste de hipótesis sobre las medias y cuando el p-value es menor que un α determinado (la probabilidad de rechazar la hipótesis

nula cuando es cierta) significa que realmente existen diferencias estadísticamente significativas entre las medias de los niveles o variantes para un factor o interacción. En este caso se establece un $\alpha=0,05$, lo que significa que sólo es significativa la combinación de factores BC, es decir FdM y GmF.

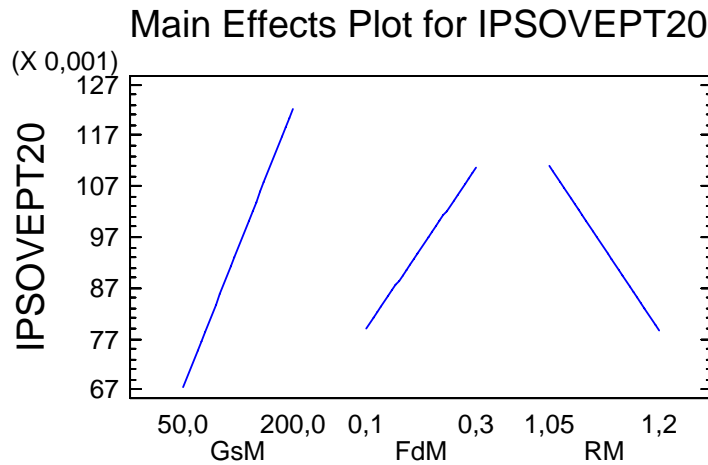


Figura 8-35. Gráfico de efectos principales de los factores del conjunto experimental SSD10_P13_20.

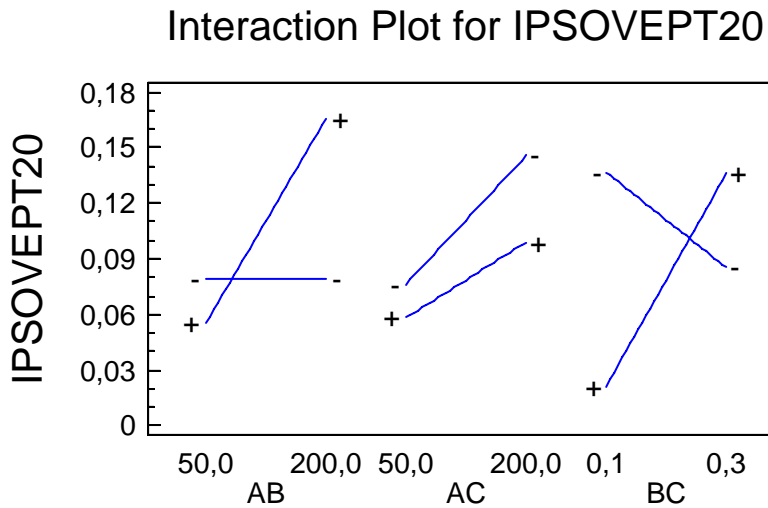


Figura 8-36. Gráfico de interacción entre los factores del conjunto experimental SSD10_P13_20.

Analizando la gráfica de la Figura 8-36 se deduce que el valor adecuado son para el factor FdM es 0.1, en combinación con el valor más alto del parámetro GmF, es decir, con 1.2. El valor del factor GsM, se puede establecer libremente. No obstante, el valor elegido para GsM será 50, ya que con este valor se obtienen mejores resultados teniendo en cuenta todas las interacciones, aunque no se pueda afirmar que éstas sean significativas. Por lo tanto, los parámetros quedarían GsM=50, FdM=0.1 y FmG=1.2.

Repetiendo el proceso descrito para el resto de conjuntos experimentales se obtiene la combinación de parámetros para cada uno de ellos que se muestra en la Tabla 8-16. Dicha tabla contiene los valores que se asignarían a los factores analizados en el experimento para alcanzar la mejor respuesta promedio en cada conjunto experimental. Como se explica a continuación algunos factores son significativos y otro o lo son, pero en cualquier caso el algoritmo requiere de un valor para su ejecución.

La Tabla 8-16 se debe interpretar teniendo en cuenta que la columna “Factores Significativos” contiene el nombre de aquellos factores, que según el análisis ANOVA, resultan ser significativos, de forma que si se expresan separados por una barra quiere decir que presentan interacción entre ellos, y si se expresan de forma individual es que son significativos por sí solos.

Parámetros SMAGA-1	Factores Significativos	Generaciones sin Mejora (GsM)	Factor de Mutación (FdM)	Factor Margen (FmG)
SSD10_P13_20	FdM FmG	50	0.1	1.2
SSD10_P13_50	-	50	0.1	1.2
SSD50_P13_20	-	200	0.3	1.05
SSD50_P13_50	FdM FmG	200	0.3	1.05
SSD100_P13_20	FmG	50	0.1	1.2
SSD100_P13_50	FmG	50	0.1	1.2
SSD125_P13_20	FmG	200	0.1	1.2
SSD125_P13_50	FmG	200	0.1	1.2
SSD10_P3_20	GsM, GsM FdM	50	0.1	1.2
SSD10_P3_50	GsM, FmG	50	0.1	1.2
SSD50_P3_20	GsM	50	0.3	1.2
SSD50_P3_50	GsM, FdM, FmG	50	0.3	1.2
SSD100_P3_20	FmG	50	0.3	1.2
SSD100_P3_50	GsM	50	0.3	1.2
SSD125_P3_20	-	50	0.1	1.2
SSD125_P3_50	FmG	50	0.1	1.2

Tabla 8-16. Valores seleccionados para los parámetros de los diferentes conjuntos experimentales del algoritmo SMAGA-1 (elaboración propia).

Por ejemplo, en la fila correspondiente al conjunto experimental SSD10_P3_20 se indica que el factor GsM es significativo por sí sólo, y además presenta una interacción con el factor FdM. Así mismo, para identificar de una forma más rápida los factores que son significativos se ha sombreado en color gris la casilla correspondiente al valor seleccionado para el mismo.

La selección del valor de los factores significativos se ha realizado en función de cual es el que menor valor de IPSOVEPT produce empleando las gráficas de efectos principales o de interacción según los casos. La selección del valor de los parámetros no significativos hubiera podido ser cualquiera dentro de los valores analizados, y de hecho se hubiera podido elegir cualquiera de las siguientes estrategias:

- Unificar al máximo posible los valores de cada factor intentado que el conjunto de valores elegidos hubiese sido el mínimo. Por ejemplo el factor GsM sólo es significativo para el valor 50, pudiéndose por lo tanto establecer este valor para cualquier conjunto experimental.
- Elegir los valores al azar dentro de los valores analizados.
- Seleccionar la combinación de valores que menor IPSOVEPT produzca aunque el resultado alcanzado no se pueda demostrar que se obtiene por dicha combinación.

En este caso se ha dudado entre la primera y la última estrategia. La primera presentaba la ventaja de simplificar el funcionamiento del algoritmo, y la segunda podía ofrecer la posibilidad de alcanzar buenos resultados estableciendo valores que en muchas ocasiones estaban cerca de ser significativos. Dado que nada hace pensar que tenga que realizarse una Parametrización general para cualquier caso se ha optado por la última estrategia.

A partir de los valores seleccionados para el algoritmo SMAGA-1 se puede ejecutar el algoritmo SMAGA-2. En este caso los parámetros “Generador Dinámico”, “Evolución del Factor Límite” y “Evolución del Riesgo Mínimo” deben estar siempre activados, al mismo tiempo que el parámetro “Riesgo Mínimo” (**RM**) necesita tener asignado ningún valor. La experimentación para el algoritmo SMAGA-2 es sencilla. Tal y como se comentó en el “Plan Experimental”, ésta consiste en un análisis experimental para un solo factor, RM, que puede tomar dos valores posibles: 0.2 ó 0.4. En la Tabla 8-17 se muestran los valores de los parámetros del algoritmo SMAGA-2 para cada conjunto experimental.

Como se puede observar en la Tabla 8-17 los conjuntos experimentales con la misma configuración de taller y de tiempo de cambio de partida, pero con diferentes número de trabajos se han podido agrupar. Por ejemplo, el conjunto experimental SSD10_P13_20 y

SSD10_P13_50 utilizan los mismos valores y se han representado como SSD10_p13. Esto es así, debido a que nunca se ha dado el caso de que un mismo factor sea significativo y con valores diferentes en ambos conjuntos (20 y 50 trabajos). De esta manera las 16 combinaciones de parámetros se han reducido a ocho.

Parámetros SMAGA-2	SSD10_P13	SSD50_P13	SSD100_P13	SSD125_P13
Generador Dinámico	Activado	Activado	Activado	Activado
Evolución del Factor Límite	Activado	Activado	Activado	Activado
Evolución de Riesgo Mínimo	Activado	Activado	Activado	Activado
Generaciones sin Mejora (GsM)	50	200	50	200
Factor de Mutación (FdM)	0.1	0.3	0.1	0.1
Factor Margen (FmG)	1.2	1.05	1.2	1.2
Parámetros SMAGA-2	SSD10_P3	SSD50_P3	SSD100_P3	SSD125_P3
Generador Dinámico	Activado	Activado	Activado	Activado
Evolución del Factor Límite	Activado	Activado	Activado	Activado
Evolución de Riesgo Mínimo	Activado	Activado	Activado	Activado
Generaciones sin Mejora (GsM)	50	50	50	50
Factor de Mutación (FdM)	0.1	0.3	0.3	0.1
Factor Margen (FmG)	1.2	1.2	1.2	1.2

Tabla 8-17. Resumen de los valores asignados a los parámetros del algoritmo SMAGA-2 para cada conjunto experimental (elaboración propia).

Para realizar el análisis de los resultados obtenidos, es decir, de los valores IPSOVEPT de cada conjunto experimental, del algoritmo SMAGA-2, se emplea el análisis de la Varianza mediante ANOVA con el análisis previo de las condiciones correspondientes, que se puede consultar en el Anexo III. Para realizar el análisis se han agrupado por un lado todos los conjuntos experimentales con configuración entre 1 y 3 máquinas por etapa, y por otro lado, todas las configuraciones con una configuración de 3 máquinas por etapa. Del análisis de la variable de respuesta se obtienen los valores indicados en la Tabla 8-18 para la primera agrupación y en la Tabla 8-19 la segunda.

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
MAIN EFFECTS					
A:Conjunto	7,10645	7	1,01521	14,04	0,0000
B:RM	0,0749496	1	0,0749496	1,04	0,3162
INTERACTIONS					
AB	0,35273	7	0,05039	0,70	0,6742
RESIDUAL	2,31365	32	0,0723015		
TOTAL (CORRECTED)	9,84778	47			

All F-ratios are based on the residual mean square error.

Tabla 8-18. ANOVA correspondiente IPSPPVEPT de la configuración P13.

Como se puede ver en la ANOVA de la Tabla 8-18 se indica que el factor RM no es significativo, mientras si lo es el Conjunto Experimental.

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
MAIN EFFECTS					
A:Conjunto	26,3796	7	3,76851	19,69	0,0000
B:RM	1,85508	1	1,85508	9,69	0,0039
INTERACTIONS					
AB	1,44438	7	0,20634	1,08	0,4000
RESIDUAL	6,1255	32	0,191422		
TOTAL (CORRECTED)	35,8045	47			

All F-ratios are based on the residual mean square error.

Tabla 8-19. ANOVA correspondiente IPSPPVEPT de la configuración P3.

En la ANOVA de la Tabla 8-19 se observa como tanto el factor RM, como el factor Conjunto Experimental es significativo. Y atendiendo a los valores de la Figura 8-37 se establecerá 0.4 como valor del factor RM para los conjuntos experimentales P3.

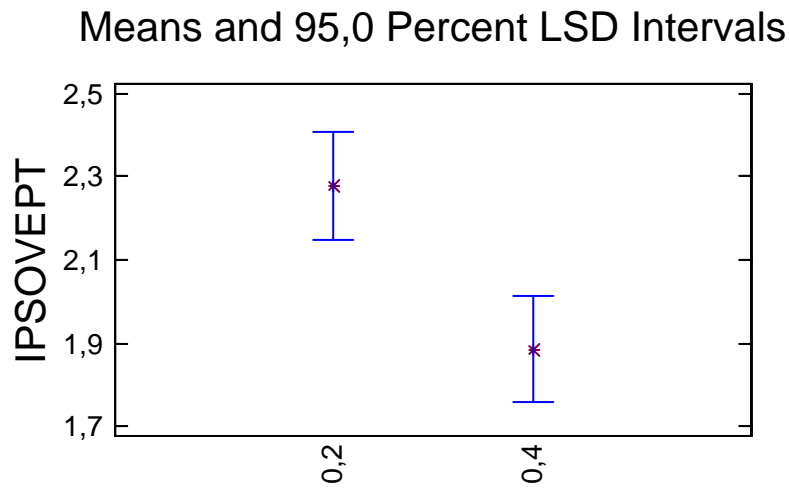


Figura 8-37. Gráfico del efecto principal del factor RM de la configuración P3.

8.5 Evaluación Comparativa

Como ya se ha expuesto, especialmente en el capítulo 3, casi no existen métodos que aporten soluciones competitivas al tipo de problema propuesto por lo que se propone emplear adaptaciones de heurísticas y Metaheurísticas utilizadas en la literatura para resolver problemas similares. La evaluación competitiva que se realiza en este trabajo se basa en gran medida en el trabajo desarrollado en (Ruiz, R. 2003), cuyo autor seleccionó, programó y publicó los resultados obtenidos para un conjunto de heurísticas y metaheurísticas adaptadas al problema del Taller de Flujo Híbrido con Tiempos de Cambio de Partida Dependientes de la Secuencia. Además, el autor empleó los mismos conjuntos experimentales que se han empleado en este trabajo, de hecho esta fue una de las razones importantes por las que se seleccionó el conjunto de datos de Taillard como ya se comentó en el apartado 8.2.3.1. El número de iteraciones de los algoritmos en el trabajo de Ruiz fueron de 5.000 evaluaciones el makespan, que coincide con las iteraciones realizadas para el algoritmo SMAGA-1 y SMAGA-2. Dada la coincidencia del desarrollo experimental con el empleado en este trabajo para el ajuste de los parámetros tan sólo es necesario indicar los algoritmos con los que se realizará el análisis comparativo, para posteriormente inicial el análisis de los resultados.

El primer método que se incluye es el denominado **GAH**, que fue introducido por el propio Ruiz, en (Ruiz, R. 2003) y que no necesita ningún tipo de modificación ya que figura entre los mejores procedimientos para el problema que se analiza.

El primer método que Ruiz adaptó fue el procedimiento de recocido simulado de (Osman, I. H. & Potts, C. N. 1989) que se adaptó sustituyendo tan sólo el cálculo del makespan, a este método se le llamó **SAOPH**. El algoritmo basado en búsqueda tabú de (Widmer, M. & Hertz, A. 1989) también se adaptó. En este caso se modificó la inicialización, que se realiza por la heurística NEH adaptada a este problema (NEHH), de igual manera, la evaluación de la solución en cada paso del algoritmo se hace con las funciones de cálculo del makespan adaptadas. A la adaptación de este algoritmo se denominó **SpiritH**. Se modificó la heurística original NEH de (Nawaz, M. et al. 1983). El cambio que se realizó estriba en abordar la asignación, y el cálculo del makespan para el problema ampliado, en vez de calcular el makespan como en el Taller de Flujo estándar. La heurística NEH adaptada para el problema que nos ocupa se denominará **NEHH**. El Algoritmo Genético de (Reeves, C. R. 1995) se adaptó modificando la función de evaluación, así como la inicialización, que ahora se realiza mediante la heurística NEHH en vez de utilizar la heurística NEH estándar. Ruiz se refiere a este método como **GAReevH**. De manera similar se modificaron los Algoritmos Genéticos de (Chen, C. L., Vempati, V. S., & Aljaber, N. 1995), (Murata, T., Ishibuchi, H., & Tanaka, H. 1996) y (Ponnanbalam, S. G., Aravindan, P., & Chandrasekaran, S. 2001) que son referidos como **GACHenH**, **GAMITH** y **GAPACH** respectivamente, y para los que simplemente se cambiaron las funciones de evaluación de los individuos. En total se han reflejado los datos obtenidos en (Ruiz, R. 2003)

para 8 algoritmos. Esos ocho algoritmos servirán para contrastar los resultados obtenidos para las propuestas SMAGA-1 y SMAGA-2 que se realizan en esta tesis.

En la Tabla 8-20 y Tabla 8-21 se muestran los resultados promedio obtenidos en todos los conjuntos experimentales para cada uno de los 10 algoritmos empleados. En la Tabla 8-20 se muestran los resultados para el caso del Taller de Flujo Híbrido con Tiempos de Cambio de Partida Dependientes de la Secuencia entre 1 y 3 máquinas por etapa (tipo P13), y en la Tabla 8-21 cuando dicho taller siempre dispone de 3 máquinas por etapa (tipo P3). Para el caso de SMAGA-1 y SMAGA-2 se han aprovechado los resultados obtenidos en la Parametrización de los algoritmos sólo para las ejecuciones con los valores seleccionados para los diversos parámetros en cada conjunto experimental. En ambas tablas, el color rojo se utiliza para destacar el mejor resultado, el azul para indicar el segundo mejor resultado, y el negro para el resto de resultados.

Resultados	GAH	SAOPH	SpiritH	GARevvH	NEHH	GACHenH	GAPACH	GAMITH	SMAGA-1	SMAGA-2
SSD10_P13_20	0,1467	1,0933	2,8533	0,6967	2,5900	3,0300	8,2967	2,7100	0,0213	0,0417
SSD10_P13_50	0,2200	1,6067	2,7967	0,7300	2,6333	3,7900	10,3933	5,0500	0,8515	0,4434
SSD50_P13_20	0,5933	3,6900	6,5100	2,5267	6,0233	6,7167	17,2300	7,0033	0,8300	0,8761
SSD50_P13_50	0,7933	3,6367	6,1100	2,2933	18,2833	8,9100	21,4233	10,5267	0,9201	0,9957
SSD100_P13_20	1,1900	6,1700	9,9567	4,5567	8,8933	11,0633	25,7433	9,9067	1,2152	1,275
SSD100_P13_50	0,8233	5,3767	8,0233	3,3033	6,8733	13,9067	31,8767	15,0867	0,7263	0,7283
SSD125_P13_20	1,3333	7,4267	11,0433	4,8533	9,6100	12,1167	28,5933	11,7533	1,3766	1,1714
SSD125_P13_50	0,7133	5,8767	9,1767	3,6000	7,8767	15,9767	35,5633	15,9200	0,7604	0,5061

Tabla 8-20. Valor de IPSOVEPT para cada uno de los conjuntos experimentales de tipo P13 para los diez algoritmos seleccionados (elaboración propia).

Resultados	GAH	SAOPH	SpiritH	GARevvH	NEHH	GACHenH	GAPACH	GAMITH	SMAGA-1	SMAGA-2
SSD10_P3_20	2,1433	6,8100	10,9433	3,1967	7,6433	7,0233	17,1133	12,7467	1,7744	1,3981
SSD10_P3_50	1,7200	10,4867	11,2000	2,7000	5,4133	4,7300	18,2933	11,6033	0,1285	0,1703
SSD50_P3_20	3,1033	9,5867	12,3533	5,1000	10,3667	8,2467	18,5133	10,8967	2,8240	2,4393
SSD50_P3_50	1,9000	10,2700	10,1700	3,2633	6,2500	4,3833	17,0067	11,2033	1,7773	1,7207
SSD100_P3_20	3,3000	11,8833	12,9867	6,2000	12,6733	8,7033	20,1333	12,0733	2,6651	2,5837
SSD100_P3_50	2,4600	10,1133	10,2267	4,1033	7,0933	4,9333	17,0967	11,5733	2,2260	1,9767
SSD125_P3_20	3,6700	12,3133	14,1100	7,1867	14,2633	8,9600	21,0300	13,3867	3,2635	2,6329
SSD125_P3_50	2,9133	10,3667	10,7733	4,7067	7,8167	5,8133	17,7900	11,9900	2,5469	2,1792

Tabla 8-21. Valor de IPSOVEPT para cada uno de los conjuntos experimentales de tipo P3 para los diez algoritmos seleccionados (elaboración propia).

Para el análisis comparativo se ha seguido la misma metodología que para el ajuste paramétrico. De esta manera se ha diseñado un plan experimental con dos factores y una variable de respuesta. Los dos factores analizados han sido el tipo de algoritmo empleado (**ALG**), y el conjunto experimental sobre el que se aplica (**Conjunto**). Para simplificar el análisis los algoritmos analizados han sido sólo los mejores, que para el tipo de problemas P13 son tres algoritmos, ya para el tipo de problemas P3 son dos, como se justificará en cada caso. En cuanto al conjunto experimental se analizan los 16 planteados para el ajuste paramétrico. No obstante, el análisis se divide en dos bloques uno para los 8 conjuntos experimentales del tipo P13, y otro para los 8 conjuntos experimentales del conjunto P3. La variable de respuesta es la misma que en el ajuste paramétrico, **IPSOVEPT**.

El objetivo del análisis es identificar si existe algún algoritmo que se diferencie del resto en cada conjunto experimental, es decir identificar si el factor ALG es significativo. Para ello se empleará el análisis de la varianza mediante ANOVA.

Los datos se analizan mediante Statgraphics Plus para Windows versión 5.0 (ver <http://www.statgraphics.com>). En ambos casos, tipo P13 y tipo P3, se ha creado un diseño de experimentos mediante la opción de "Multi-Factor Categorical", con una variable de respuesta y dos factores, eligiendo un diseño factorial de personalizado, con dos réplicas. La codificación de los factores sigue la misma nomenclatura expresada hasta ahora es decir ALG, para el factor "Algoritmo", y Conjunto, para el factor "Conjunto Experimental". Para el análisis se han utilizado los datos de obtenidos en cada iteración, a partir de los cuales se han obtenido los promedios de la Tabla 8-20 y Tabla 8-21.

Statgraphics sólo permite análisis experimentales con un mínimo de 2 factores, como es lógico, por este motivo se incorporado el factor "Conjunto", ya que en realidad se tienen datos para poder decir a priori que el tipo de configuración de taller es significativa.

8.5.1 Análisis de los resultados para el Caso P13.

El caso P13 representa al conjunto de resultados correspondientes a los Talleres de Flujo Híbrido que disponen entre 1 y 3 máquinas por etapa. Se trata de una configuración que puede presentar algunos cuellos de botella debido a que existen etapas con gran capacidad de proceso y a continuación otras con mejor capacidad. Donde además puede existir una gran descompensación entre las capacidades de etapas contiguas. Obviamente, esto ocurre al tratarse de unos datos experimentales que no reflejan una realidad concreta, ya que se entiende que en el proceso de diseño del sistema productivo de una empresa real este inconveniente se hubiera suavizado.

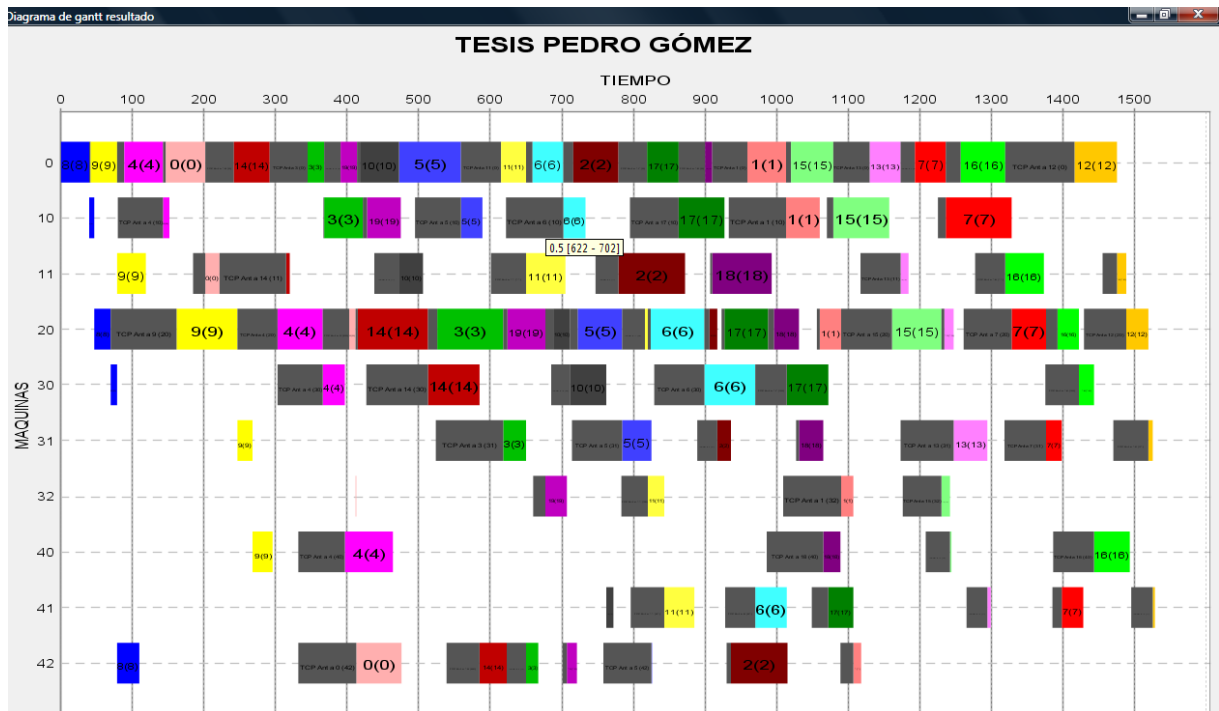


Figura 8-38. Diagrama Gantt de un Programa Productivo para un taller de flujo híbrido del tipo P13 (elaboración propia).

En la Figura 8-38 se muestra un diagrama Gantt correspondiente a un Programa Productivo con 20 trabajos con una máquina en la etapa cero (0), dos máquinas en la etapa uno (10, 11), una máquina en la etapa dos (20), tres máquinas en la etapa tres (30, 31, 32) y tres máquinas en la etapa cuatro (40, 41, 42). En color gris aparecen los cambios de partida y los tiempos de operación en otros colores. Se puede ver con claridad como la máquina 0 y la máquina 20, que son únicas en sus correspondientes etapas, actúan de cuello de botella.

Para una mejor visualización de los resultados de la Tabla 8-20 estos se han representado gráficamente en la Figura 8-39. En esta gráfica se han incluido los resultados de todos los algoritmos considerados menos los del algoritmo GAPACH, que presenta el peor valor de IPSOPEPT para cualquier conjunto experimental con una diferencia sustancial, que llega en el caso del conjunto SDD125_P13_20 a la máxima diferencia con respecto al mejor algoritmo, siendo 70,27% peor.

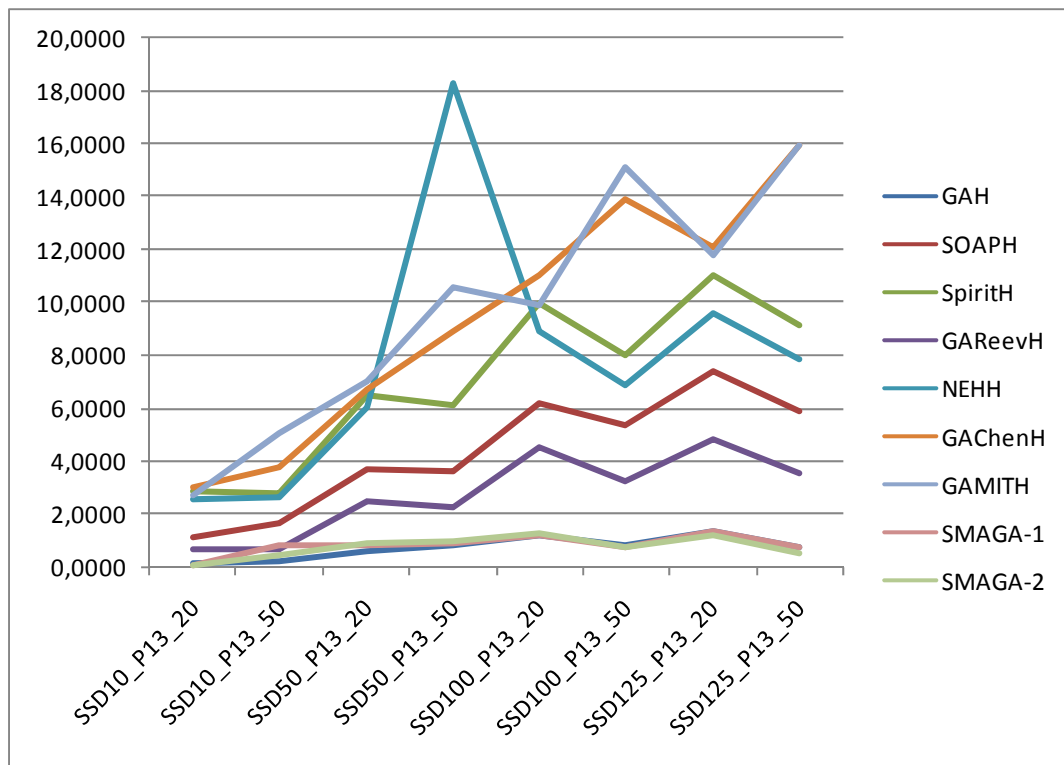


Figura 8-39. Representación gráfica de los valores IPSOVEPT para los conjuntos experimentales P13.

En general hay 3 aspectos que son llamativos a primera vista. En primer lugar las diferencias relativas entre los algoritmos son muy similares para cualquier conjunto experimental. Si bien, es cierto, que el algoritmo NEHH muestra un empeoramiento brusco en el caso del conjunto SDD50_P13_50, en que pasa a ser el peor algoritmo después de GAMITH. Y en alguna otra ocasión, algoritmos con resultados muy similares intercambian posiciones en el ranking, por ejemplo los algoritmos GACHenH y GAMITH. El segundo aspecto destacable es como los algoritmos más competitivos, es decir todos menos GAPACH, GACHenH y GAMITH, presentan mejores resultados para los casos de 50 trabajos que para los casos de 20 trabajos en los conjunto experimentales con tiempos de cambio de partida elevados (SSD100 y SSD125). Y el tercer aspecto destacable, es que los algoritmos GAH, SMAGA-1 y SMAGA-2 se encuentran en todos los casos en 3 mejores algoritmos, excepto para el conjunto experimental SSD10_P13_50, en el que el algoritmo GAReevH se sitúa en tercera posición y SMAGA-1 en cuarta. Este dominio se produce, en la mayoría de las ocasiones, con clara diferencia con respecto al resto de algoritmos analizados.

Por lo tanto, se puede afirmar que para los conjuntos experimentales analizados siempre se elegiría obtener un nuevo Programa Productivo Predictivo empleando el algoritmo GAH,

SMAGA-1 o SMAGA-2. En cualquier caso se ha realizado una representación detallada de los resultados sólo para los algoritmos mencionados en la Figura 8-40.

Del análisis de los datos de la Tabla 8-22 y de la Figura 8-40 se puede decir que las nuevas propuestas realizadas, SMAGA-1 y SMAGA-2, serían seleccionados en la mitad de las ocasiones por ser el mejor algoritmo estudiado, y que por lo tanto, estos algoritmos representan una aportación interesante en la resolución del problema establecido. No obstante, para el caso de talleres entre 1 y 3 máquinas por etapa la diferencia entre cualquiera de los tres algoritmos con el mejor para cada conjunto experimental es muy reducida encontrándose por debajo de 0,03 en todos los casos, excepto para el ya mencionado conjunto SSD10_P13_50 donde SMAGA-1 es no se encuentra entre los 3 primeros y la diferencia llega a 0.6315. En la Tabla 8-22 se indican los valores obtenidos para la diferencia IPSOVEPT entre GAH, SMAGA-1 y SMAGA-2 y el mejor algoritmo, que siempre es una de ellos (identificado por “-”). Se puede observar como en muchos casos la diferencia es menor a 0,01.

	GAH	SMAGA-1	SMAGA-2
SSD10_P13_20	0,1254	-	0,0204
SSD10_P13_50	-	0,6315	0,2234
SSD50_P13_20	-	0,2367	0,2828
SSD50_P13_50	-	0,1267	0,2023
SSD100_P13_20	-	0,0252	0,0850
SSD100_P13_50	0,0970	-	0,0020
SSD125_P13_20	0,1620	0,2052	-
SSD125_P13_50	0,2072	0,2543	-

Tabla 8-22. Diferencias entre el mejor IPSOVEPT y el de cada algoritmo en problemas P13 (elaboración propia).

Desde el punto de vista de la aplicación de los algoritmos GAH, SMAGA-1 y SMAGA-2 al cálculo de Programas Productivos se puede decir que no existe una ventaja clara de ninguno de ellos y que podrían formar un conjunto más o menos homogéneo respecto en la calidad de sus Programas Productivos. De hecho se ha realizado un análisis ANOVA con dos factores (ver Anexo III), los algoritmos GAH, SMAGA-1 y SMAGA-2 y los 8 conjunto experimentales, frente a la variable de respuesta IPSOVEPT, y ningún factor ha resultado significativo. Este dato se puede contrastar en la Figura 8-40.

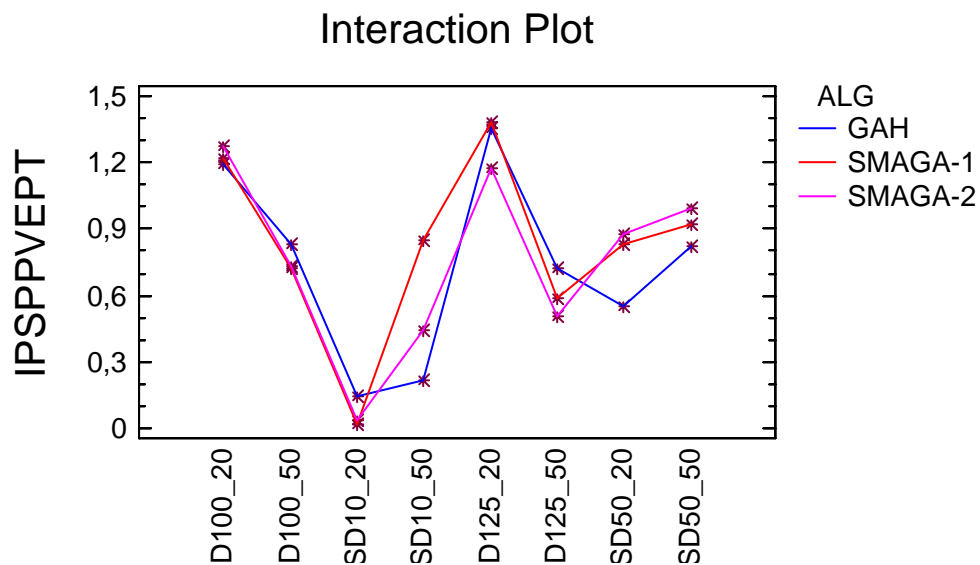


Figura 8-40. Representación gráfica de los valores IPSPVEPT de los algoritmos GAH, SMAGA-1 y SMAGA-2 para los conjuntos experimentales P13.

Desde el punto de vista del análisis de la respuesta obtenida con SMAGA-1 y SMAGA-2 se puede decir, que es satisfactorio, ya que se sitúan en el entorno del algoritmo más competitivo que se ha encontrado en la literatura para el problema propuesto. Pero también es importante destacar que el algoritmo SMAGA-2 no resulta ser significativamente mejor que SMAGA-1. Dicho de otra forma, la incorporación del conjunto de características propias de los sistemas Multiagente que se introducen en el algoritmo SMAGA-2 no son determinantes. Aunque es cierto, sin ser significativo, que cuanto mayor es el Tiempo de Cambio de Partida la posición relativa de SMAGA-2 con respecto a sus dos competidores mejora.

Se debe resaltar que en la ejecución de los algoritmos, SMAGA-1 ha mejorado en 78 ocasiones el mejor valor del makespan conocido hasta el momento. Por su parte, SMAGA-2 lo ha superado en 52 ocasiones. En total de los 120 ejemplares empleados en el tipo de problema P13, SMAGA-1 dispone del mejor makespan alcanzado para 70 de ellos, y SMAGA-2 para 16 ejemplares. Los mejores valores obtenidos mediante la ejecución de SMAGA-1 y SMAGA-2 y el correspondiente valor IPSO, asociado a los ejemplares, se muestra en el anexo II.

8.5.2 Análisis de los resultados para el Caso P3.

El caso P3 representa al conjunto de resultados correspondientes a los Talleres de Flujo Híbrido que disponen de 3 máquinas por etapa. Se trata de una configuración que por su

topología no debería presentar problemas de cuellos de botella debido a que el número de máquinas es similar en cada etapa. Aunque igualdad de máquinas no implica igualdad de capacidades sí que reduce el riesgo de disponer de un taller excesivamente desequilibrado. La Figura 8-41 muestra un Programa Productivo para un Taller de Flujo Híbrido con 3 máquinas por etapa y 20 trabajos, y en ella se puede apreciar como los trabajos están mucho más agrupados que para el mismo caso, que el caso en el que se dispone entre 1 y 3 máquinas por etapa, que se muestra en la Figura 8-38.

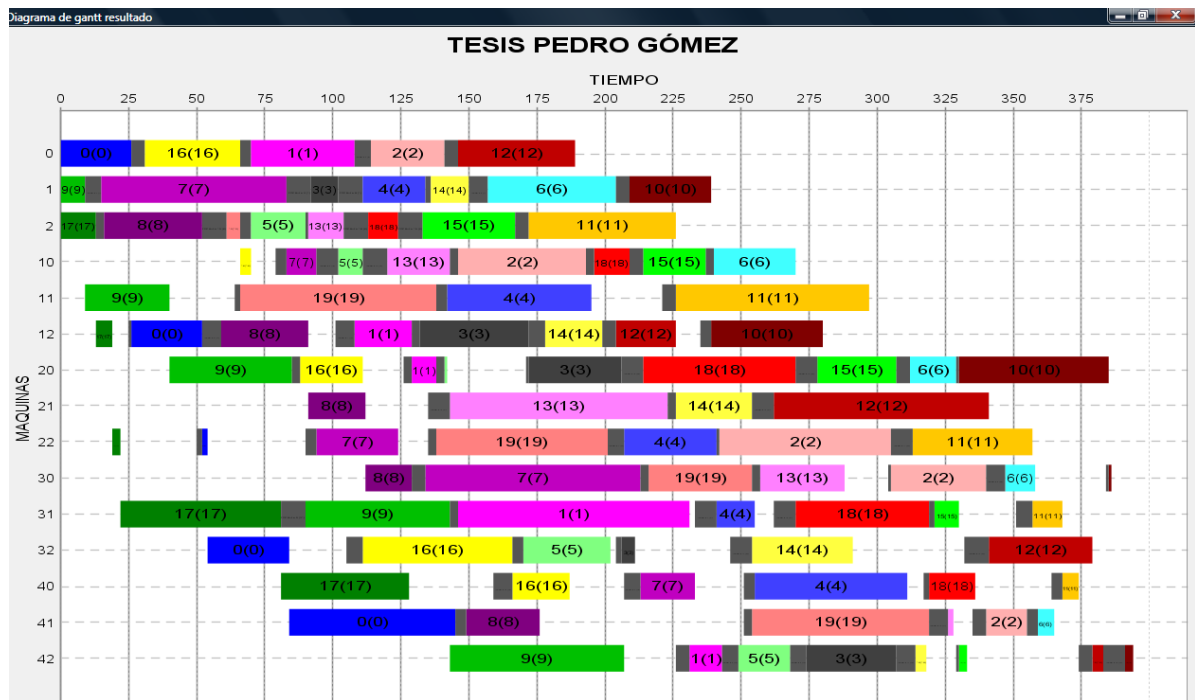


Figura 8-41. Diagrama Gantt de un Programa Productivo para un taller de flujo híbrido del tipo P3 (elaboración propia).

Para una mejor visualización de los resultados de la Tabla 8-21 estos se han representado gráficamente en la Figura 8-42. En esta gráfica se han incluido los resultados de todos los algoritmos considerados.

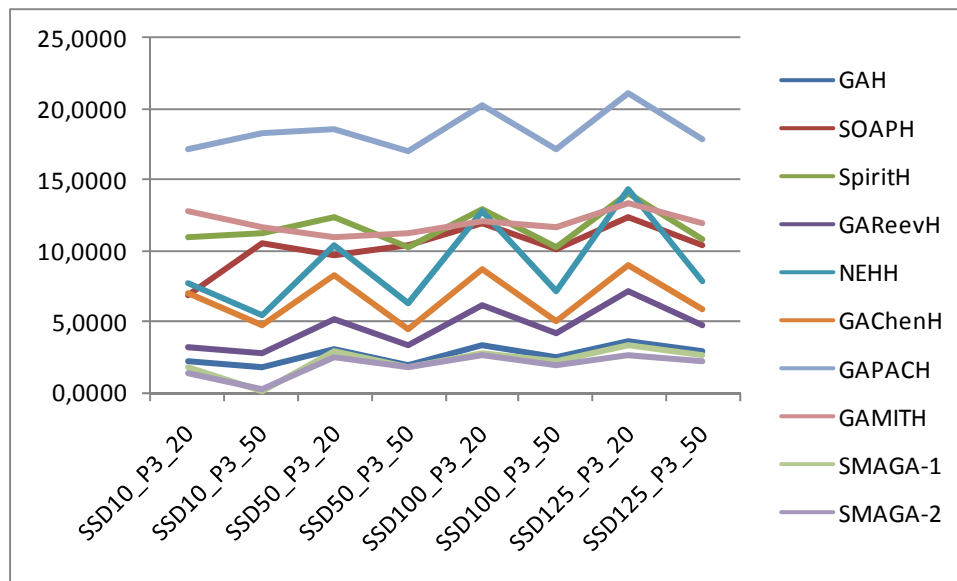


Figura 8-42. Representación gráfica de los valores IPSOVEPT para los conjuntos experimentales P3.

En general hay 3 aspectos que reseñables. En primer lugar, igual que en el caso P13, las diferencias relativas entre los algoritmos es muy similar para cualquier conjunto experimental. Si bien, es cierto que el algoritmo NEHH muestra un empeoramiento brusco para los conjuntos con 20 trabajos, en que incrementa bastante su valor IPSOVEPT y pierde puestos en el ranking de algoritmos, sin llegar a ser el peor. Y en alguna otra ocasión, aunque en menor medida que el caso P13, algoritmos con resultados muy similares intercambian posiciones en el ranking. En este caso queda prácticamente reducido a SpiritH y GAMITH. El segundo aspecto destacable que en el caso SSD100 y SSD125 todos los algoritmos presentan un mejor IPSOVEPT en el caso de 50 que en de 20 trabajos. Para el caso de SSD50 se mantiene esta tendencia excepto para el algoritmo SOAPH y GAMITH. Y en el caso SSD10 se mantiene la tendencia excepto para el algoritmos SOAPH, SpritH y GAPACH. Y el tercer aspecto destacable, es que los algoritmos GAH, SMAGA-1 y SMAGA-2 se encuentran en todos los casos en 3 mejores algoritmos, en la mayoría de las ocasiones con clara diferencia con respecto al resto de algoritmos analizados, y el algoritmo GAPACH es en todos los casos el peor también a una amplia distancia del resto.

Para un mejor análisis de los mejores algoritmos los resultados de los mismo se han representado en la Figura 8-43.

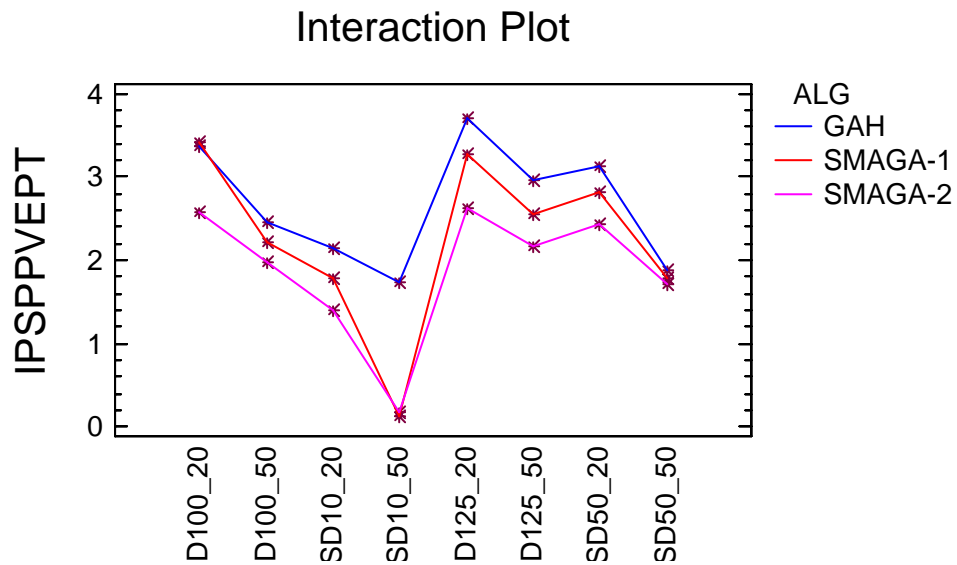


Figura 8-43. Representación gráfica de los valores IPSOVEPT de los algoritmos GAH, SMAGA-1 y SMAGA-2 para los conjuntos experimentales P3.

En este caso, los algoritmos SMAGA-1 y SMAGA-2 son siempre los dos mejores. Por lo tanto, se puede afirmar que para los conjuntos experimentales analizados siempre se elegiría obtener un nuevo Programa Productivo Predictivo empleando el algoritmo SMAGA-1 o SMAGA-2. Pero en el caso P3 el dominio de SMAGA-2 frente a SMAGA-1 es bastante significativo, siendo el mejor en 7 de los 8 conjuntos experimentales. Sólo en el conjunto SSD10_P3_50 SMAGA-1 es mejor que SMAGA-2. En todo caso, para poder disponer de una idea clara de cual es la diferencia de GAH, SMAGA-1 y SMAGA-2 con respecto al valor IPSOVEPT del mejor algoritmo (identificado por “-”) en cada conjunto experimental se puede ver la Tabla 8-23.

	GAH	SMAGA-1	SMAGA-2
SSD10_P3_20	0,7452	0,3763	-
SSD10_P3_50	1,5915	-	0,0418
SSD50_P3_20	0,6641	0,3847	-
SSD50_P3_50	0,1793	0,0565	-
SSD100_P3_20	0,7163	0,0814	-
SSD100_P3_50	0,4833	0,2493	-
SSD125_P3_20	1,0371	0,6306	-
SSD125_P3_50	0,7341	0,3676	-

Tabla 8-23. Diferencias entre el mejor IPSOVEPT y el de cada algoritmo en problemas P3 (elaboración propia).

A tenor de lo observado en la Tabla 8-23 el algoritmo SMAGA-2 domina al resto, En todo caso se ha realizado el análisis de la varianza con los algoritmos GAH, SMAGA-1 y SMAGA-2 como niveles u opciones del factor ALG, que se puede ver en el Anexo III. Una vez comprobadas las hipótesis básicas de normalidad, homocedasticidad y linealidad, se deduce que el valor adecuado son para el factor ALG es SMAGA-2 en cualquier caso, siendo además significativo el tipo de configuración de taller. Sin embargo, no se puede establecer ninguna relación entre algoritmo y conjunto, lo que implica que SMAGA-2 es siempre el mejor.

En este tipo de problemas, P3, el planteamiento de un Sistema Multiagente que incorpore las características introducidas en la propuesta implementada con SMAGA-2 sí supone una ventaja que permite superar en todos los casos al mejor algoritmo conocido GAH. Aunque los problemas del tipo P13 y P3 son en ambos casos complejidad NP-Completa, como se indicó en el capítulo 3, es reseñable que en todo caso siempre es más sencillo programar un taller en el cual existen una o varias etapas identificadas como “cuello de botella” al disponer de una sólo máquina, que aquel taller que se encuentra mejor balanceado. Sin que suponga una evidencia científica, el hecho de que el algoritmo SMAGA-2 se comporte comparativamente mejor en los problemas de tipo P3, este dato puede servir de punto de partida para una línea de investigación futura que intente analizar en profundidad el comportamiento de SMAGA-2 en diferentes grados de complejidad en función de la configuración del taller.

Se debe resaltar que en la ejecución de los algoritmos, SMAGA-1 ha mejorado en 57 ocasiones el mejor valor del makespan conocido hasta el momento. Por su parte, SMAGA-2 lo ha superado en 37 ocasiones. En total de los 120 ejemplares empleados en el tipo de problema P3, SMAGA-1 dispone del mejor makespan alcanzado en 46 de ellos, y SMAGA-2 en 25 ejemplares. Los mejores valores obtenidos mediante la ejecución de SMAGA-1 y SMAGA-2 y el correspondiente valor IPSOPEP, asociado a los ejemplares, se muestra en el anexo II.

8.6 Conclusiones

En este capítulo se ha tratado el problema de Programación de la Producción con un enfoque Predictivo en base a la definición del mismo realizada en los capítulo 5 y 7, y en el marco de un Proceso de Negocio de Programación Predictiva que tiene como objetivo dar soporte a la Programación de la producción en un entorno de incertidumbre para la Industria Cerámica. No sólo se ha empleado el trabajo realizado en el capítulo 7 para definir el entorno de trabajo sino que se referenciado el modelo matemático, desarrollado en el capítulo, para centrar el problema en un taller concreto, el Taller de Flujo Híbrido con Tiempos de Cambio de Partida Dependientes de la Secuencia. Una vez establecido el marco de trabajo se ha diseñado un

nuevo Algoritmo Genético hibridado entre un Algoritmo Genético y el paradigma de los Sistemas Multiagente que permita ofrecer una solución a dicho problema.

Desde el principio se persiguió la idea de que la tecnología basada en Sistemas Multiagente podía por su forma de enfocar los problemas **aportar una visión que permitiese superar en términos eficiencia** los algunos de los resultados que se habían publicado hasta la fecha. Por este motivo, para el diseño e implementación del nuevo algoritmo se han empleado una metodología y una Plataforma de desarrollo propia del desarrollo de Sistemas Multiagente. Además, **con el fin de diferenciar la posible aportación de los Sistemas Multiagente, se dividió el proceso en dos pasos**, que aunque se han trabajado casi simultáneamente, han proporcionado dos algoritmos diferenciados, SMAGA-1 y SMAGA-2. El algoritmo SMAGA-1 basa su diseño en la tecnología Multiagente pero que no incorpora el conjunto de características propias de los Agentes en “sociedad”, y que por lo tanto no aprovechan una condición que les es propia, “el trabajo en equipo”. Sin embargo, SMAGA-2 si incorpora características que le permiten sacar un mayor partido al trabajo conjunto en la sociedad.

Una vez diseñados e implementados los algoritmo SMAGA-1 y SMAGA-2 se ha realizado un exhaustivo análisis experimental dividido en dos fases. En la primera fase se ha realizado el ajuste del conjunto de parámetros, que así lo requerían, de ambos algoritmos. Con la peculiaridad, que de forma deliberada, SMAGA-2 emplea siempre los mismos valores que ha establecido SMAGA-1, a excepción de los que le son propios que son los que se han parametrizado de forma específica. En la segunda fase, la versión más competitiva de SMAGA-1 y SMAGA-2 se ha empleado para ser comparada con algunos de los mejores algoritmos que se han encontrado en la literatura para este tipo de problema.

En la fase de ajuste paramétrico se establecieron los valores de los parámetros para el conjunto de configuraciones más próximas a las empleadas en las empresas de la Industria Cerámica. El número de ejecuciones de los algoritmos fue muy elevado, aunque basados en un sencillo plan experimenta. **Se destaca como en esta fase SMAGA-1 ha establecido un nuevo valor mínimo del makespan en 106 ejemplares, y SMAGA-2 en 41 ejemplares de un total de 240 del juego de datos empleado.**

En la fase de análisis comparativo se identificaron los algoritmos GAH, SMAGA-1 y SMAGA-2 como los más competitivos para el tipo de problemas P13 y P3. Aunque en el caso de los problemas de tipo P13 no existe un predominio claro de ninguno de estos 3 algoritmos en ninguno de los conjuntos experimentales analizados. En el caso de los problemas de tipo P3 se puso de manifiesto el predominio del algoritmo SMAGA-2 en todos los conjuntos experimentales. En esta fase el algoritmo SMAGA-2, y en menor medida el algoritmo SMAGA-1, se han revelado como altamente competitivos. Además el algoritmo **SMAGA-2 ha alcanzado**

el mejor resultado promedio en 9 de los 16 conjuntos experimentales, y siempre entre los casos más complejos, mientras que el algoritmo SMAGA-1 lo es en 3 ocasiones.

A partir de los resultados comparativos obtenidos se ha preparado la Plataforma IPSU-MAS para realizar el proceso de “Programación Predictiva de la Programación” mediante la integración de los algoritmos seleccionados, y su lógica de aplicación, en dicha Plataforma.

Como conclusión final, y resumida del capítulo, se puede considerar que tanto **SMAGA-1, como SMAGA-2 han supuesto una aportación relevante como algoritmos para la Programación Predictiva de la Producción** en un Taller de Flujo Híbrido con Tiempos de Cambio de Partida Dependientes de la Secuencia, y pueden ser empleados como elementos activos de la Plataforma IPSU-MAS.

8.7 Bibliografía

Brucker, P., Heitmann, S., & Hurink, J. 2003. Flow-shop problems with intermediate buffers. Or Spectrum, 25(4): 549-574.

Chen, C. L., Vempati, V. S., & Aljaber, N. 1995. An Application of Genetic Algorithms for Flow-Shop Problems. European Journal of Operational Research, 80(2): 389-396.

Coello, C. A. C. & Pulido, G. T. 2005. Multiobjective structural optimization using a microgenetic algorithm. Structural and Multidisciplinary Optimization, 30(5): 388-403.

Conway, R. W., Maxwell, W. L., & Miller, L. W. 1967. Theory of Scheduling. Addison-Wesley Publishing Company.

Davis, L. Applying Adaptative Algorithms to Epistatic Domains. Aravin K.J. Proceeding of 9th International Joint Conference of Artificial Intelligence (IJCAI) , 162-164. 1985. Los Angeles, Morgan Kaufmann.

Goldberg, D. E. 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley.

Goldberg, D. E. and Lingle, Jr. R. Alleles, loci, and the traveling salesman problem. Grenfenstette J.J. Proceeding of the First International Conference on Genetic Algorithms and their Applications , 154-159. 1985. Hillsdale, Lawrence Erlbaum Associates.

Holland, J. H. Adaptation in Natural and Artificial Systems. Report of the Systems Analysis Research Group SYS--1/92, University of Dortmund, Department of Computer Science. 1975. University Michigan Press, Ann Arbor.

Liu, J., Zhong, W. C., & Hao, L. C. 2007. *An organizational evolutionary algorithm for numerical optimization. Ieee Transactions on Systems Man and Cybernetics Part B-Cybernetics*, 37(4): 1052-1064.

Liu, J., Zhong, W. C., & Jiao, L. C. 2006. *A multiagent evolutionary algorithm for constraint satisfaction problems. Ieee Transactions on Systems Man and Cybernetics Part B-Cybernetics*, 36(1): 54-73.

Liu, J., Zhong, W. C., Jiao, L. C., & Liu, F. 2004. *Multiobjective optimization based on coevolutionary algorithm. Rough Sets and Current Trends in Computing*, 3066: 774-779.

Marin, J. & Sole, R. V. 1999. *Macroevolutionary algorithms: A new optimization method on fitness landscapes. Ieee Transactions on Evolutionary Computation*, 3(4): 272-286.

Mattfeld, D. C. 1996. *Evolutionary Search and The Job Shop; Investigations on Genetic Algorithms and Production Scheduling. Production and Logistics: Berlin: Springer verlag.*

Murata, T., Ishibuchi, H., & Tanaka, H. 1996. *Genetic algorithms for FlowShop Scheduling Problem. Computers and Industrial Engineering*, 30(4): 1061-1071.

Naderi, B., Zandieh, M., & Roshanaei, V. 2009. *Scheduling hybrid flowshops with sequence dependent setup times to minimize makespan and maximum tardiness. International Journal of Advanced Manufacturing Technology*, 41(11-12): 1186-1198.

Nawaz, M., Enscore, E. E., & Ham, I. 1983. *A Heuristic Algorithm for the M-Machine, N-Job Flowshop Sequencing Problem. Omega-International Journal of Management Science*, 11(1): 91-95.

Osman, I. H. & Potts, C. N. 1989. *Simulated Annealing for Permutation Flowshop Scheduling. Omega-International Journal of Management Science*, 17(6): 551-557.

Pavon, J. & Gomez-Sanz, J. 2003. *Agent oriented software engineering with INGENIAS. Multi-Agent Systems and Applications II, Proceedings*, 2691: 394-403.

Ponnanbalam, S. G., Aravindan, P., & Chandrasekaran, S. 2001. *Constructive and Improvement Flow Shop Scheduling Heuristics: An extensive Evaluation. Production Planning and Control*, 12(4): 335-344.

Reeves, C. R. 1995. *A Genetic Algorithm for Flowshop Sequencing. Computers & Operations Research*, 22(1): 5-13.

Ruiz, R. 2003. *Técnicas Metaheurísticas para la Programación Flexible de la Producción. Departamento de Estadística e Investigación Operativa. Universidad Politécnica de Valencia.*

Ruiz, R., Maroto, C., & Alcaraz, J. 2005. *Solving the flowshop scheduling problem with sequence dependent setup times using advanced metaheuristics - Discrete optimization. European Journal of Operational Research*, 165(1): 34-54.

Syswerda, G. 1996. *Scheduling Optimization using Genetic Algorithms*. In L. Davis (Ed.), *Handbook of Genetic Algorithms*: 332-349. London: International Thomson Computer Press.

Taillard, E. 1993. *Benchmarks for Basic Scheduling Problems*. *European Journal of Operational Research*, 64(2): 278-285.

Vallada, E., Ruiz, R., and Maroto, C. *Synthetic and Real Benchmarks for Complex Flow-Shops Problems*. Informe Técnico. 2003. Grupo de Investigación Operativa (GIO). Universidad Politécnica de Valencia (Spain).

Watson, J. P., Barbulescu, L., Whitley, L. D., & Howe, A. E. 2002. *Contrasting structured and random permutation flow-shop scheduling problems: Search-space topology and algorithm performance*. *Inform Journal on Computing*, 14(2): 98-123.

Widmer, M. & Hertz, A. 1989. *A New Heuristic Method for the Flow-Shop Sequencing Problem*. *European Journal of Operational Research*, 41(2): 186-193.

Zhang, R. & Wu, C. 2009. *A Decomposition-Based Optimization Algorithm for Scheduling Large-Scale Job Shops*. *International Journal of Innovative Computing Information and Control*, 5(9): 2769-2780.

Zhong, W. C., Liu, J., & Jiao, L. C. 2005. *Job-shop scheduling based on Multiagent Evolutionary Algorithm*. *Advances in Natural Computation, Pt 3, Proceedings*, 3612: 925-933.

Zhong, W. C., Liu, J., Xue, M. Z., & Jiao, L. C. 2004. *A multiagent genetic algorithm for global numerical optimization*. *Ieee Transactions on Systems Man and Cybernetics Part B-Cybernetics*, 34(2): 1128-1141.

Zhou, Z. F. & Harris, K. D. M. 2008. *Counteracting stagnation in genetic algorithm calculations by implementation of a micro genetic algorithm strategy*. *Physical Chemistry Chemical Physics*, 10(48): 7262-7269.

Zuo, X. Q., Mo, H. W., & Wu, J. P. 2009. *A robust scheduling method based on a multi-objective immune algorithm*. *Information Sciences*, 179(19): 3359-3369.

**Algoritmos para la Programación
Reactiva de la Producción en un Taller
de Flujo Híbrido**

Capítulo

9

Tabla de contenido

9. Capítulo 9 593

9.1 Introducción 593

9.2 Ámbito de Aplicación y Metodología 594

9.2.1 Identificación del Problema 594

9.2.2 Diseño del Experimento 597

9.2.2.1 Entorno Experimental: Identificación de los Factores 599

9.2.2.1.1 Política de Reprogramación 599

9.2.2.1.2 El Juego de Datos 602

9.2.2.1.3 Identificación de los Parámetros de los Algoritmos 604

9.2.2.2 Definición de los Factores 605

9.2.2.3 Plan de Experimentación..... 607

9.3 Descripción de los Algoritmos Analizados. 610

9.4 Desarrollo del Diseño Experimental..... 614

9.4.1 Procedimiento de Ejecución los Experimentos 614

9.4.2 Análisis de los resultados 615

9.4.2.1 Análisis de los resultados en situación de Adelanto de las Operaciones.. 616

9.4.2.1.1 Casos con Programación Predictiva SMAGA-2 616

9.4.2.1.2 Casos con Programación Predictiva SMAGA-2 Modificada 621

9.4.2.2 Análisis de los resultados en situación de Retraso de las Operaciones 622

9.4.2.2.1 Casos con Programación Predictiva SMAGA-2 623

9.4.2.2.2 Casos con Programación Predictiva SMAGA-2 Modificada 628

9.5 Conclusiones 629

9.6 Bibliografía 631

Índice de figuras

Figura 9-1. Diagrama de Funcional del Proceso de Negocio de “Programación Reactiva” (elaboración propia). 595

Figura 9-2. Diagrama de flujo del proceso de selección del trabajo y operación afectados por un evento..... 600

Figura 9-3. Ejemplo de selección de operaciones para retraso o adelanto de la misma..... 601

Figura 9-4. Representación de los diferentes ámbitos en los que se realiza en análisis de los factores..... 607

Figura 9-5. Ejemplo en la implementación del algoritmo Right Shift (RS). 611

Figura 9-6. Ejemplo en la implementación del algoritmo Affected Operations Rescheduling (AOR). 611

Figura 9-7. Ejemplo en la implementación del algoritmo Same Sequence Different Schedule (SSDS)..... 612

Figura 9-8. Gráfico del efecto de la interacción entre el factor CEX y el factor ALG (izquierda) y entre el factor K y ALG (derecha) del bloque Adelanto-Predictiva SMAGA-2-P13_20.... 617

Figura 9-9. Gráfico del efecto de la interacción entre el factor CEX y el factor ALG (izquierda) y entre el factor K y ALG (derecha) del bloque Adelanto-Predictiva SMAGA-2-P13_50.... 618

Figura 9-10. Gráfico del efecto de la interacción entre el factor CEX y el factor ALG (izquierda) y entre el factor K y ALG (derecha) del bloque Adelanto-Predictiva SMAGA-2-P3_20..... 618

Figura 9-11. Gráfico del efecto de la interacción entre el factor CEX y el factor ALG (izquierda) y entre el factor K y ALG (derecha) del bloque Adelanto-Predictiva SMAGA-2-P3_50..... 618

Figura 9-12. Gráfico del efecto de la interacción entre el factor VA y el factor ALG del bloque Adelanto-Predictiva SMAGA-2-P3_50. 619

Figura 9-13. Gráfico del efecto de la interacción entre el factor CEX y el factor ALG (izquierda) y entre el factor VA y ALG (derecha) del bloque Retraso-Predictiva SMAGA-2-P13_20 (K=0). 624

Figura 9-14. Gráfico del efecto de la interacción entre el factor CEX y el factor ALG (izquierda) y entre el factor VA y ALG (derecha) del bloque Retraso-Predictiva SMAGA-2-P13_20 (K=0.01). 624

Figura 9-15. Gráfico del efecto de la interacción entre el factor CEX y el factor ALG (izquierda) y entre el factor K y ALG (derecha) del bloque Retraso-Predictiva SMAGA-2-P13_50 (K=0). 625

Figura 9-16. Gráfico del efecto de la interacción entre el factor CEX y el factor ALG (izquierda) y entre el factor K y ALG (derecha) del bloque Retraso-Predictiva SMAGA-2-P13_50 (K=0.005). 625

Figura 9-17. Gráfico del efecto de la interacción entre el factor CEX y el factor ALG con K=0 (izquierda) y con K=0.01 (derecha) del bloque Retraso-Predictiva SMAGA-2-P3_20. 626

Figura 9-18. Gráfico del efecto de la interacción entre el factor CEX y el factor ALG (izquierda) y entre el factor K y ALG (derecha) del bloque Retraso-Predictiva SMAGA-2-P3_50 con K=0.
..... 627

Índice de tablas

Tabla 9-1. Conjunto de baterías de eventos simples.	601
Tabla 9-2. Resumen de los conjunto experimentales empleadas en el diseño experimental..	603
Tabla 9-3. Niveles asignados a los factores en el experimento de Programación Reactiva.	609
Tabla 9-4. Resumen del análisis de los resultados del bloque experimental P13 y P3 para el adelanto en la finalización de las operaciones.....	620
Tabla 9-5. Resumen del análisis de los resultados del bloque experimental P13 y P3 para el adelanto en la finalización de las operaciones.....	622
Tabla 9-6. Resumen del análisis de los resultados del bloque experimental P13 y P3 para el retraso en la finalización de las operaciones.....	627
Tabla 9-7. Resumen del análisis de los resultados del bloque experimental P13 y P3 para el retraso en la finalización de las operaciones.	629

9.1 Introducción

En este capítulo se realiza un análisis experimental sobre una selección de algoritmos y con el fin de analizar su comportamiento en relación con el proceso de Programación Reactiva de la Producción. Esta tarea forma parte del tercer paso de la metodología propuesta en el capítulo 6, y con su realización se debe dar por finalizado el mismo. En el proceso experimental se va a configurar un entorno de trabajo similar a aquel en el que se pretende que trabaje la plataforma IPSU-MAS. En este entorno se va a **estudiar el comportamiento de diversos algoritmos para la Reprogramación parcial y completa de un Programa Productivo**, algunos de los cuales se basan en propuestas elaboradas en esta tesis.

En este capítulo se plantea una estructura similar a la que se propone habitualmente en el desarrollo del Diseño de Experimentos. Una vez se identifica el problema a resolver el diseño de experimentos determina que es necesaria la identificación de los factores, después de la cual se establecerá un plan de experimentación. En el proceso de Programación Reactiva de la Producción, en un entorno con Incertidumbre, se considera que tanto los **algoritmos de reprogramación** como **las condiciones del contorno** pueden influir significativamente en los resultados, y que por lo tanto en ellos residen los factores a analizar.

Debido a que el problema de la Reprogramación se origina por la existencia de eventos, un entorno incierto, es fundamental establecer el método para **simular eventos** y las **políticas de reprogramación** que se aplicarán en dichas situaciones. Los eventos pueden afectar a los trabajos en cualquier de sus operaciones, pero quizás no en todas con la misma probabilidad ni con la misma duración. Así mismo es necesario considerar que pueden existir **eventos que aceleren o que retrasen las operaciones** en curso. Todas estas son cuestiones que se debe definir. También es importante considerar que posiblemente el grado de **degradación del Programa Productivo en curso** pueda influir en el resultado final.

Por lo que se refiere a los algoritmos que se proponen para ser analizados se considera importante disponer, tanto de los algoritmos que ante un evento vuelvan a programar los trabajos aún no finalizados sin tener en cuenta el programa anterior (**reprogramación completa**), como de los algoritmos que tengan como referencia el programa anterior y que sólo alteren los trabajos afectados (**reprogramación parcial**). Se emplean algoritmos basados en heurísticas y en metaheurísticas, algunos de uso generalista y otros de carácter más específico. Aunque desde el punto de vista del diseño de la plataforma IPSU-MAS el fin de este capítulo es la obtención de un criterio de selección que permita aplicar el mejor método en

cada circunstancia **se ponen de manifiesto que el verdadero interés de este trabajo reside en la comprensión de las reacciones que cada tipo de procedimiento provoca en los resultados del problema**. Por este motivo se ha planteado un número de algoritmos reducido. Adicionalmente, hay que subrayar que no es fácil encontrar propuesta para la reprogramación aplicada a casos de empresas industriales.

Por lo que refiere al diseño del plan experimental se debe poner de manifiesto la dificultad para conciliar los intereses relativos a la obtención de un modelo sencillo y fácil de interpretar con los relativos al cumplimiento de las condiciones marcadas por el tipo de análisis estadístico que se sigue. En concreto se avanza que con el fin de buscar una distribución de la varianza homogénea se ha tenido que dividir el análisis en múltiples bloques independientes.

El trabajo desarrollado en el presente capítulo se organiza de la siguiente forma: En primer, en el apartado 9.2 lugar se expondrá el ámbito de aplicación y la metodología con la que se desarrollará la propuesta, en este apartado se definirán el problema y se realizará el diseño del experimento. En segundo lugar, en el apartado 9.3 se describirán brevemente los algoritmos propuestos. En el apartado 9.4, se abordará el desarrollo del experimento y el análisis de los resultados. Y finalmente, en el apartado 9.5 se comentarán brevemente las conclusiones del capítulo.

9.2 Ámbito de Aplicación y Metodología

En este apartado se definirá el problema que se desea abordar así como la metodología que se va a seguir para ello.

9.2.1 Identificación del Problema

Según se comentó en el capítulo 7, “Análisis, Diseño e Implementación de una Plataforma Software basada en Multiagentes para la Programación de la Producción Predictivo-Reactiva”, de esta tesis, se propone integrar un conjunto de métodos y criterios que permitan afrontar el problema de la Programación de la Producción, en un entorno de incertidumbre en coordinación con el Proceso de Planificación de la Producción. Una vez definidos y diseñados dichos métodos se implementarán en una plataforma software. En el mencionado capítulo se identificaron los requerimientos de la plataforma mediante la aplicación de la propuesta CIMOSA. Dentro del amplio conjunto de responsabilidades que se la atribuyeron a dicha plataforma se destacan dos grandes “Procesos de Negocio (BP): El “BP-Programación Predictiva”, y el “BP-Programación Reactiva”.

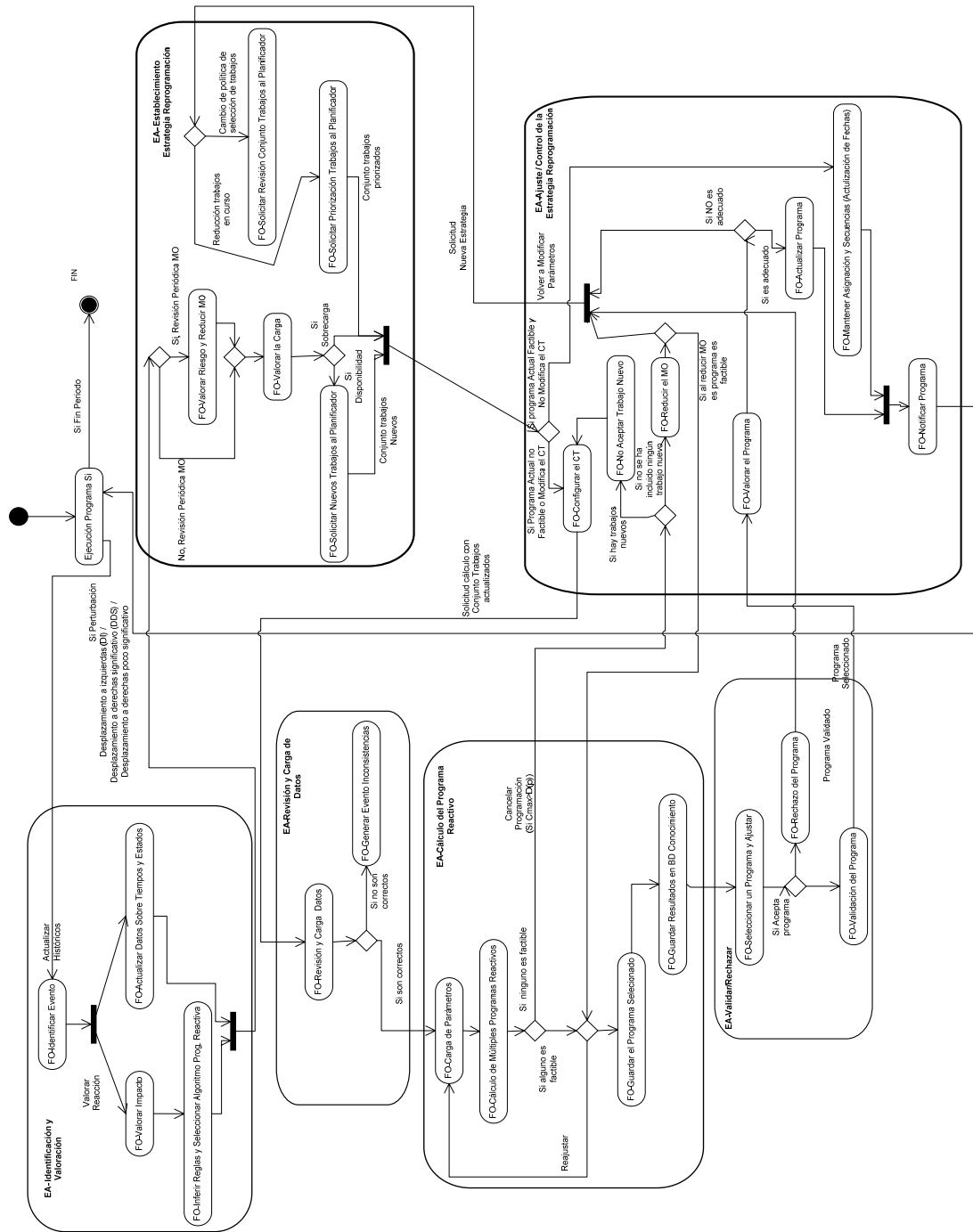


Figura 9-1. Diagrama de Funcional del Proceso de Negocio de "Programación Reactiva" (elaboración propia).

En la Figura 9-1 se muestra el diagrama funcional del proceso completo, dentro del cual se describen las responsabilidades de lo que CIMOSA denomina Actividades Empresariales, que son:

- EA-5: Identificación y valoración.
- EA-6: Establecimiento de la estrategia de Reprogramación
- EA-7: Ajustes /Control de la estrategia de Reprogramación.
- EA-8: Revisión y carga de datos.
- EA-9: Cálculo del Programa Reactivo.
- EA-10: Validar/Rechazar.

El objetivo de este capítulo es la selección de uno o varios algoritmos que puedan formar parte del núcleo del Proceso de Negocio para la Programación Reactiva. El algoritmo o algoritmos que se seleccionen deben asumir las funciones atribuidas a la actividad empresarial EA-9 en la que se sitúa la función asociada al cálculo del Programa Reactivo. Las acciones comprendidas en la actividad empresarial EA-6 están orientadas al establecimiento de los parámetros que determinarán la función objetivo, y que se han denominado parámetros estratégicos o estrategia de Reprogramación. En la EA-7 se realiza el ajuste de los parámetros estratégicos del algoritmo que se decida utilizar y la validación final del Programa Productivo. La EA-10, es una actividad intermedia, cuyo cometido es el registro y transferencia de los resultados obtenidos a la Actividad Empresarial AE-7.

Se desea resaltar que aunque lo que en capítulos anteriores se denominó “Gestión del Programa Productivo” es una función que asumirán el conjunto de procesos que forman la plataforma, recaerá de forma especial en el Proceso de Negocio “BP-Programación Reactiva”. Dado que el objetivo de este capítulo es el análisis de diversos algoritmos que permitan el cálculo de un Programa Reactivo el desarrollo del mismo se centrará en la EA-9, por lo que se aislará el proceso de Reprogramación para evitar influencia de la estrategia en el rendimiento de los algoritmos.

El problema Reactivo al que hará frente la actividad EA-9, y por lo tanto, el algoritmo que se seleccione en este capítulo, fue definido en un lenguaje natural y matemático en el capítulo 5. En concreto en el apartado 5.2.2.3 se comentan las hipótesis que se plantean para el modelo reactivo, y en el apartado 5.4.3 y 5.4.4 se desarrolla el modelo matemático que define sin ambigüedades el problema se pretende resolver en este capítulo. Cabe recordar que la función objetivo es:

$$[8.1] \quad \text{Minimizar } Z = \left\{ \max_i^{NX} \{c(i, RX)\} + \kappa \cdot \left(\sum_i^{NX} (at(i) \cdot af(i)) \right) \right\}$$

Nos referiremos a la variable “Z” de la función objetivo como “makespan reactivo penalizado”.

En los apartados indicados del capítulo 5 se describe el significado de la nomenclatura empleada. Así mismo cabe recordar que basándose en la notación establecida por (Conway, R. W., Maxwell, W. L., & Miller, L. W. 1967) para identificar los problemas de secuenciación, el Problema Reactivo Determinista que se pretende resolver podría ser representado como $n/r/F/k_1, \dots, k_r\text{-parallel/setup, preasig}/B$. Es decir se trata de un Taller de Flujo con r etapas, y k_1 máquina en paralelo en la primera etapa, k_2 en la segunda y k_r en la última. Donde existe un Tiempo de Cambio de Partida o Setup Dependiente de la Secuencia y una serie de trabajos preasignados a máquinas (preasig), cuya función objetivo es B (balance), un balance entre el makespan y el adelanto de las fechas de inicio de ciertos trabajos.

El objetivo es por lo tanto seleccionar un conjunto de algoritmos para la Programación Reactiva e identificar si existe alguna influencia de los mismos, y en que condiciones, con el fin de incluirlos en la actividad EA-9 de la plataforma IPSU-MAS.

9.2.2 Diseño del Experimento

Tal y como se definió en el apartado 8.2.3 del capítulo 8 , el Diseño de Experimentos es una metodología basada en herramientas matemáticas y estadísticas cuyo objetivo es ayudar al experimentador a seleccionar la estrategia experimental óptima que permita alcanzar, y evaluar los resultados obtenidos, con el mínimo coste y la máxima fiabilidad en las conclusiones.

La aplicación del diseño de experimentos requiere considerar las etapas que se indican a continuación:

- Comprender el problema y definir claramente el objetivo.
- Identificar los factores que potencialmente podrían influir en la función objetivo, y los valores que éstos pueden tomar. Entre estos valores se buscará la información necesaria.
- Establecer una estrategia experimental, llamada plan de experimentación.
- Efectuar los experimentos con los valores de los factores decididos en el punto tres para obtener los valores de las respuestas estudiadas.
- Responder las preguntas planteadas, sea directamente, sea utilizando un modelo matemático. Si es necesario, volver a la etapa uno.

Se puede decir que los pasos uno a tres están más ligados al propio diseño del experimento, y los pasos cuatro y cinco se encuentran vinculados a la ejecución y análisis del mismo. Dentro de este apartado, aunque organizados en subapartados, se desarrollarán los aspectos relacionados con los tres primeros pasos.

En el primer paso se debe establecer que el problema que se desea abordar en la experimentación es el de identificar el conjunto de factores significativos y determinar, para cada combinación de factores significativos, el algoritmo con el que se obtenga los mejores resultados promedio posibles para el problema especificado en el apartado 9.2.1. La variable de respuesta o endógena que se empleará para evaluar los resultados será la variable IPSPVPEP ligada a la función objetivo que los algoritmos desean minimizar definida en la ecuación [8.1]. Esta medida proporciona el Incremento Porcentual Sobre el Programa Predictivo del Valor Examinado Promedio (IPSPVPEP) de un problema o ejemplar dado. Esta medida se puede expresar como:

$$IPSPVPEP = \frac{\text{makespan_reactivo_penalizado} - \text{makespan_predictivo}}{\text{makespan_reactivo_penalizado}} * 100$$

La variable “makespan_predictivo” representa al valor del makespan obtenido para el Programa Predictivo del ejemplar que se desea evaluar. La variable “makespan_reactivo_penalizado” representa al mejor valor obtenido para la función objetivo establecida para la Programación Reactiva en dicho ejemplar. Así por ejemplo, para un ejemplar con un makespan inicial o predictivo de 1.714 uds. que sufra una Reprogramación obteniendo un nuevo makespan de 1.724 y dos trabajos adelantados y una K=10, alcanza un makespan reactivo penalizado de 1.744, y un IPSPVPEP de 1,720.

La identificación de los factores será la consecuencia del análisis del entorno experimental. En este caso son tres los aspectos que definen dicho entorno: La política de Reprogramación, la configuración del sistema, y el valor de ciertos parámetros del modelo de Programación Reactiva. El entorno experimental se describe posteriormente en el apartado 9.2.2.1. Por lo que se refiere al paso tres, el plan experimental, se define en el apartado 9.2.2.3.

Un aspecto que pertenece al diseño, aunque se emplee en la fase de análisis, es la selección de una herramienta que permita establecer conclusiones claras y fiables. En este caso se ha considerado que existe una variable explicada (dependiente o endógena) cuyo valor es función de diversas variables explicativas (independientes o exógenas). Donde además la variable explicada es métrica y las variables explicativas son no métricas. En este contexto, habitual por otra parte en este tipo de comparativa de algorítmicos, el uso de un test paramétrico es

fundamental. Uno de los test paramétricos más potentes, y ajustados a las características de este caso, es el del análisis de la varianza o ANOVA. El uso del análisis de la varianza implica la verificación de una serie de hipótesis mediante un conjunto de técnicas, que ya han sido indicadas en el apartado 8.2.3 del capítulo 8.

Es importante, recordar que la información que se obtenga después de la fase de análisis formará parte de la EA-5 en la cual se sitúa la FO-Inferir reglas y seleccionar algoritmo de Programación Reactiva, ver la Figura 9-1.

9.2.2.1 Entorno Experimental: Identificación de los Factores

Para acabar de definir el objetivo del experimento que se va a realizar es necesario definir con claridad el entorno en el que va a llevar a cabo, y al cual quedarán circunscritas las conclusiones.

9.2.2.1.1. Política de Reprogramación

Como se ha descrito en capítulos anteriores, especialmente en los capítulos 5 y 7, la Reprogramación de la Producción o Programación Reactiva se origina cuando se detecta un evento que, una vez analizado, es considerado susceptible de activar este proceso. Esta forma de actuar implica la existencia de un sistema de gestión de eventosⁱ que actúa de acuerdo con una política determinada, que establece el instante y tipo de actuación. Sin embargo, la gestión de los eventos implica la aplicación de una estrategia, y como se ha comentado desde un inicio, en este experimento tan sólo se pretende identificar los mejores algoritmos para la EA-9. Por este motivo, en el proceso experimental se aislará cualquier decisión sobre como actuar ante un evento que no sea la de decidir que algoritmo se debe aplicar. El funcionamiento conjunto del Proceso de Negocio de Reprogramación, es decir la combinación de estrategia y algorítmica, se realizará en el capítulo 10.

Por este motivo, se ha decidido simular eventos que afecten a una única operación de un trabajo en curso y que impliquen:

ⁱ La funcionalidad relacionada con la gestión de los eventos se ubica en la EA-5 “Identificación y Valoración” del BP-Programación Reactiva.

- La generación de Programa Reactivo debido a un retraso en la fecha de finalización de una operación de un trabajo en curso, o bien,
- la generación de un Programa Reactivo debida a un adelanto en la fecha de finalización de una operación de un trabajo en curso.

Se ha considerado la posibilidad de analizar el comportamiento de los algoritmos ante una sola modificación del Programa Productivo, o ante un conjunto sucesivo de Reprogramaciones. No obstante, debido a las características de función objetivo y del problema se ha considerado que un sistema de Reprogramaciones no es un problema diferente al problema de una sola Reprogramación, y que adicionalmente complicaría el análisis experimental. Por este motivo, tanto para el caso de retraso, como el de adelantos en los tiempos de operación se genero siempre un único evento.

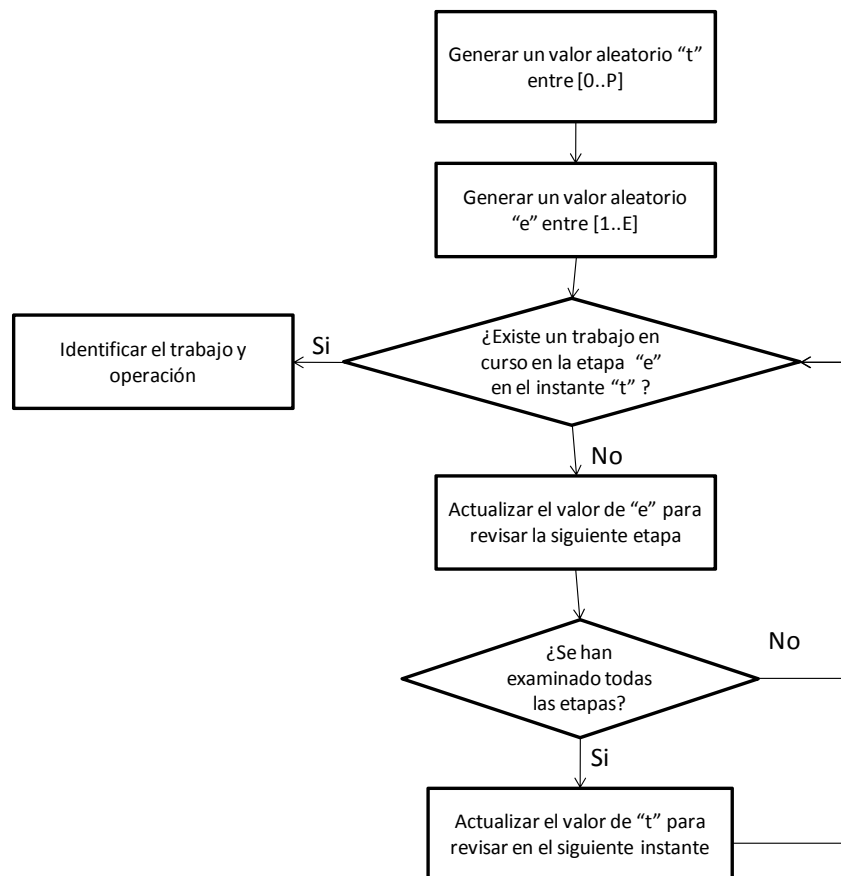


Figura 9-2. Diagrama de flujo del proceso de selección del trabajo y operación afectados por un evento.

La selección del trabajo y operación afectadas por el evento se realiza mediante el diagrama de flujo de la Figura 9-2 . La operación afectada es seleccionada determinando aleatoriamente un

instante en el Programa Productivo mediante una distribución uniforme dentro del periodo programado. También se determinará de forma aleatoria una etapa. Con estos datos se buscará una operación que esté en proceso en dicho instante y etapa, si se encuentra ésta será la seleccionada. En otro caso, primero se busca en otras etapas en el mismo instante. Si no existe ninguna operación en proceso en el instante determinado, éste se incrementa en una unidad y se vuelve a intentar buscar una operación a partir de la etapa determinada inicialmente. El instante de Reprogramación se incrementa en una unidad hasta encontrar una operación o llegar al final del programa en curso.

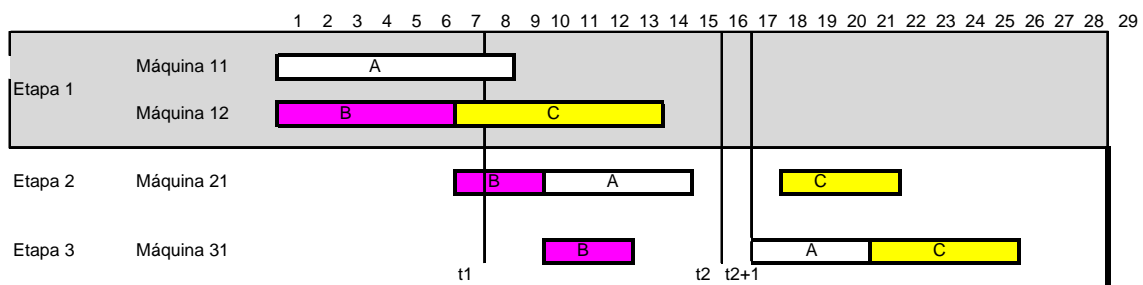


Figura 9-3. Ejemplo de selección de operaciones para retraso o adelanto de la misma.

Para el caso de la Figura 9-3 si se establece como instante de Reprogramación el “t1” y como etapa inicial la tres, al no existir ninguna operación activa se busca en la siguiente etapa, la primera. En ésta se detecta como en la primera máquina el trabajo “A” está en proceso, y dicha operación se selecciona para ser retrasada o adelantada. Si por el contrario, se hubiese establecido como instante de Reprogramación inicial el instante “t2”, y como etapa inicial la dos, al no existir ninguna operación en proceso en el instante “t2” este se incrementa en una unidad. En el instante “t2+1” se buscaría una operación en la etapa dos, y tampoco hay. Al examinar para “t2+1” la etapa tres se detecta que la operación tres del trabajo “A” está activa y se selecciona.

Una vez determinado el proceso de selección de operaciones es necesario definir un espectro lo más completo posible de casos, que se denominarán “Baterías de Eventos”, y que se muestran en la Tabla 9-1.

	Desviaciones del 10%	Desviaciones del 50%	Desviaciones del 100%
Retraso	BAT_RS_10	BAT_RS_50	BAT_RS_100
Adelanto	BAT_AS_10	BAT_AS_50	BAT_AS_80

Tabla 9-1. Conjunto de baterías de eventos simples.

Cada algoritmo será experimentado en cada una de las situaciones que representa cada una de las seis baterías definida en la Tabla 9-1. Se analizará el comportamiento de cada algoritmo siempre en casos de aparición de un solo evento durante la ejecución del Programa Productivo. Para ello se han definido tres posibles situaciones de retraso, y tres posibles situaciones de adelanto con respecto a las fechas de finalización previstas de la operación.

Para las situaciones de retraso en la fecha de finalización de una operación se han planteado tres baterías de eventos (BAT_RS_10, BAT_RS_50, BAT_RS_100) que representan incrementos del 10%, del 50% o del 100% del tiempo de proceso previsto inicialmente. Así mismo, para las situaciones de adelanto en la fecha de finalización de una operación se han planteado tres baterías de eventos (BAT_AS_10, BAT_AS_50, BAT_AS_80) que representan una reducción del 10%, del 50% o del 80% del tiempo de proceso previsto inicialmente. En este último caso, no se ha considerado una reducción del 100% del tiempo de proceso, ya que ello equivaldría a eliminar el trabajo en curso. Además para el caso de adelantos se ha considerado que sólo se puede adelantar la fecha de finalización de una operación hasta alcanzar el instante de reprogramación. Por ejemplo, en la Figura 9-3 si se selecciona la operación uno del trabajo "A", en ningún caso se puede reducir más de una unidad, que es lo que le queda para finalizar desde el instante "t2".

Con la simulación de eventos propuesta se cubren un abanico de situaciones próximas a lo que pudiera darse en una situación real dentro del entorno de la Industria Cerámica. Esto permite evaluar las diferentes alternativas independientemente de la estrategia seguida para la "Gestión del Programa Productivo".

9.2.2.1.2. El Juego de Datos

Para la realización experimento es necesario disponer de dos tipos de datos diferenciados. Por un lado, igual que en el caso de la Programación Predictiva, es necesario disponer de la información relativa a la configuración del taller (número de etapas, máquinas en cada etapa), tiempos de proceso de los trabajos en las diferentes máquinas, y los Tiempo de Cambio de Partida correspondiente. Y por otro lado, se debe disponer de un Programa Productivo Predictivo que sea objeto de Reprogramación.

En cuanto a los datos relativos a la configuración del taller y los tiempos de ejecución y Tiempo de Cambio de Partida se empleará el juego de datos definido en el apartado 8.2.3.1 del capítulo 8. Se recuerda que se emplean dieciséis agrupaciones que se denominaron Conjuntos Experimentales.

Ejemplares	Entre 1..3 Máquinas por Etapa y 20 Trabajos.	Entre 1..3 Máquinas por Etapa y 50 Trabajos.	3 Máquinas por Etapa y 20 Trabajos.	3 Máquinas por Etapa y 20 Trabajos.
	P13_20	P13_50	P3_20	P3_50
Tiempos cambio 10%	SSD10_P13_20	SSD10_P13_50	SSD10_P3_20	SSD10_P13_50
Tiempos cambio 50%	SSD50_P13_20	SSD50_P13_50	SSD50_P3_20	SSD50_P13_50
Tiempos cambio 100%	SSD100_P13_20	SSD100_P13_50	SSD100_P3_20	SSD100_P13_50
Tiempos cambio 125%	SSD125_P13_20	SSD125_P13_50	SSD125_P3_20	SSD125_P13_50

Tabla 9-2. Resumen de los conjunto experimentales empleadas en el diseño experimental.

Para el conjunto de datos indicado en la Tabla 9-2 se ha realizado una agrupación de los ejemplares de cada conjunto experimental. Se presentan tres grupos, y cinco ejemplares por grupo en cada conjunto experimental. Atendiendo a la subdivisión realizada en grupos de ejemplares y conjunto experimentales se han introducido otras dos variables. La variable **IPSPVPEPG** tendrán en cuenta el IPSPVPEP promedio para un todas los ejemplares de un grupo, y la variable **IPSPVPEPT** tendrá en cuenta el IPSPVPEP promedio para un conjunto experimental completo.

En cuanto a los Programas de Producción Predictivos sobre los que se realizarán las Reprogramaciones se ha considerado que el resultado de la función objetivo debería ser robusto y no depender de un determinado tipo de Programa Productivo Predictivo. Por este motivo se ha decidido que exista un conjunto de Programas Productivos Predictivos para cada ejemplar, para que de esta manera que no exista siempre una misma situación de partida que pudiera condicionar los resultados. En el apartado dedicado al Plan Experimental se realizará una matización sobre este aspecto.

Se dispondrá de tres Programas Productivos para cada ejemplar. Se ha decidió que estos tres Programas de Producción sean: El Programa Productivo correspondiente a la mejor secuencia obtenida en cada ejemplar a partir de un algoritmo predictivo, más dos programas obtenidos a partir de un intercambio entre dos elementos de dicha secuencia elegidos de forma aleatoria. El Programa Predictivo se realizará con el algoritmo SMAGA-2 obtenido en el capítulo anterior, y se ejecutará con un límite de 10.000 generaciones.

Para cada ejemplar se analizará el valor promedio de la variable de respuesta obtenido a partir de cada uno de los tres Programas Predictivos de la Producción.

El objetivo es encontrar los factores significativos, y dentro de estos los parámetros más adecuados para cada uno de los dieciséis conjuntos experimentales. Uno de los factores, que posteriormente se definirán, será el tipo de algoritmo. Este factor permitirá determinar de forma comparativa el mejor algoritmo para cada conjunto experimental y para cada combinación del resto de factores.

9.2.2.1.3. Identificación de los Parámetros de los Algoritmos

Durante la ejecución de los diferentes algoritmos de Reprogramación será necesario establecer el valor de algunos parámetros que se comentan a continuación:

- El valor de la variable “K” de la función objetivo. Dicha variable toma valores en base a la capacidad que el programador considera que tiene para hacer frente en un corto periodo de tiempo a trabajos previstos inicialmente para un instante más lejano en el tiempo. En la industria cerámica esta capacidad de reacción está muy condicionada por la capacidad de los proveedores afectados por dichos cambios, que básicamente son los proveedores de fritas, esmaltes y colores.
- Valor de la “Ventana de Avance (VA)”. La VA es un valor que permite definir un periodo que se situará entre t y $t+VA$. Siendo t el instante en el que se revisa el Programa Productivo. Por lo tanto, el valor de VA determina el periodo o ventana de tiempo en la cual se penalizará que los trabajos que sitúen, en el nuevo Programa Productivo, su primera operación dentro de dicho periodo, siempre y cuando dicha fecha de inicio se hayan adelantado con respecto al programa en curso. Se recomienda revisar el modelo de Programación Determinista Reactivo del apartado 5.4.3 del capítulo 5.

Por lo que se refiere a la variable “K” se considera que la capacidad de reacción del programador no evoluciona dinámicamente durante el transcurso de un programa. Por lo tanto, el valor de “K” puede ser considerado constante durante todo el periodo productivo. No obstante, dado que pueden existir diversas posibilidades se analizarán diversos valores de “K”. En este caso se desea contemplar tres posibles valores. Se establecerá una “K” nula para una situación que sirva de referencia en la que no se consideren penalizaciones. Una “K” baja relacionada con una gran flexibilidad y poca penalización en el adelanto excesivo de las fechas de inicio de las operaciones. Y por otro lado, un valor de “K” alto relacionado con una baja flexibilidad y gran penalización en dicho adelanto de fechas de inicio. Se ha decidido que el valor “K” sea proporcional al makespan del Programa Predictivo inicial.

En relación a la VA sería conveniente considerarla como un factor a analizar, y estudiar el comportamiento de los algoritmos frente a diversos valores. En base a lo observado en las empresas de la Industria Cerámica, se debería considerar dos valores. Uno correspondiente a una ventana reducida y otro asociado a una ventana de mayor duración. Dado que en este estudio experimental se van a emplear un juego de datos estándar, y que se desconoce el periodo que se considera en para cada Programa Productivo se ajustará la consideración anterior para realizarla en base al makespan obtenido en cada Programa Productivo Predictivo.

9.2.2.2 Definición de los Factores

La experimentación que se propone implica la segmentación de la misma en varios experimentos parciales y complementarios. Esta segmentación se detallará en el apartado dedicado al Plan Experimental. En cualquier caso, cada experimento parcial contemplará los factores que están relacionados con los parámetros del algoritmo, con el algoritmo empleado, y con el entorno, que se describen a continuación.

Por lo que se refiere a los parámetros, hay que considerar el valor de la variable “K” y el de la variable “Ventana de Avance (**VA**)”. Para el parámetro “K” se utilizarán penalizaciones correspondientes al 0, 1% y 2% del makespan del Programa Predictivo (0, 0,5% y 1% en el caso de que la cantidad de trabajos programados sea 50). Para el parámetro “VA” se utilizarán ventanas correspondientes al 20% y 40% del makespan del Programa Predictivo (10% y 20% en el caso de que la cantidad de trabajos programados sea 50). En ambos casos, los valores indicados se refieren siempre a los Programas Productivos con 20 trabajos, como se indica, para el caso de 50 trabajos los valores se dividirán por la mitad con el fin de establecer valores absolutos similares.

Se sospecha que existe una relación no lineal entre los factores “K” y “VA” que puede afectar negativamente en el proceso de análisis experimental. En todo caso, durante la fase de análisis se prestará especial atención, y si es necesario, se resolverá esta dificultad añadida.

Uno de los factores más relevantes que se desea analizar es el algoritmo (**ALG**) empleado para la Programación Reactiva o Reprogramación. Este factor tendrá cinco niveles, ya que serán cinco las opciones empleadas. Los algoritmos que se analizarán son la heurística “**Right Shift (RS)**” empleada por (Conway, R. W. et al. 1967; Sadeh, N. M., Otsuka, S., & Schnellbach, R. 1993), “**Same Sequence Different Schedule (SSDS)**” de elaboración propia, “**Affected Operations Rescheduling (AOR)**” propuesto por (Abumaizar, R. J. & Svestka, J. A. 1997), y las metaheurísticas para la Reprogramación completa “**Scheduling MultiAgent Genetic Algorithm**

V.2 (SMAGA-2)” con restricciones en el avance de las operaciones, y **SMAGA-2** sin restricciones, ambas de elaboración propia. Las heurísticas seleccionadas son algoritmos específicos para la Programación Reactiva y todos ellos serán descritos brevemente posteriormente. La Metaheurística **SMAGA-2**, con y sin restricciones, se empleará como método para la Reprogramación completa del programa pendiente de ejecución, y su empleo será comentado posteriormente.

En relación con el entorno se manejan dos factores: El “Conjunto Experimental (**CEX**)”, y la “Batería de eventos (**BAT**)”.

El primer factor relacionado con el entorno es el “Conjunto Experimental (**CEX**)” que refleja la composición del taller y que está vinculado al juego de datos. Aunque este factor podría llegar a tener hasta 16 niveles, debido a la segmentación que se va a realizar del análisis sólo se manejará un máximo de 4 niveles en cada experimento.

Por lo que se refiere al segundo de los factores relacionados con el entorno se consideró inicialmente emplear la “Batería de Eventos (**BAT**)” como un factor con el que poder valorar el efecto de la duración del evento sobre el resultado final. Esto implicaba que si el factor resultaba significativo, al menos, en relación con el algoritmo empleado se podrían tomar diferentes decisiones en cuanto a la solución a aplicar en base a la duración del evento. Por un lado, dicha diferenciación sería buena porque mejoraría los resultados, pero por otro supondría un nivel mayor de dificultad en el proceso de toma de decisiones. Aunque el proceso decisional es perfectamente automatizable mediante la plataforma IPSU-MAS, y en este sentido no habría ninguna dificultad en disponer de múltiples soluciones aplicadas en base a condicionantes complejos, sin duda un proceso decisional sencillo facilitaría la comprensión del experimento y su implantación en una Empresa. Para resolver este dilema se decidió realizar los experimentos diferenciando los resultados de los tres tipos de duración, tanto para los eventos que implicaban retraso, como para los que implicaban adelanto de las operaciones, y en función de los resultados establecer la necesidad o no de mantenerlo como parámetro.

Una vez realizado un primer análisis se vio como desde el punto de vista ingenieril la duración del evento, tanto para el caso de retraso como adelanto de las operaciones, no influía en relación con el algoritmo seleccionado. Descartado por lo tanto, un comportamiento del resultado final condicionado por el factor “**BAT**” se decidió emplear el promedio de los tres niveles para el caso de adelanto en la fecha de finalización de las operaciones, y el promedio de los tres niveles para el caso de retraso en la fecha de finalización de las operaciones.

9.2.2.3 Plan de Experimentación

El establecimiento y ejecución del plan experimental constituye el tercer paso en el diseño de experimentos, y supone la determinación de las condiciones en las que posteriormente se van a ejecutar los experimentos.

Como se ha comentado previamente se ha decidido dividir el proceso experimental en varios bloques complementarios. Hubiera resultado más sencillo y más rápido no realizar dicha división, no obstante esta decisión está basada en la necesidad mantener homogéneas las varianzasⁱⁱ entre los diferentes subgrupos de individuos formados por la combinación de factores. La división realizada refleja situaciones experimentales excluyentes.

La división se ha realizado de forma que el decisor, en este caso el programador de la producción, obtenga las conclusiones que le puedan afectar, analizando sólo uno o dos bloques. En la Figura 9-4 se muestra una representación gráfica de la división. Como se ve en primer lugar se diferencia entre los datos obtenidos generando retrasos o adelantos sobre la fecha de finalización de las operaciones (Adelanto o Retraso), en segundo lugar se diferencia el tipo de Programa Predictivo inicial (SMAGA-2 o SMAGA-2 Modificado), en tercer lugar la configuración del taller (P13 o P3), y en cuarto y último lugar el número de trabajos considerados en cada periodo (T20 o T50). En una situación práctica concreta el programador conocerá de ante mano el Programa Predictivo empleado inicialmente, la configuración del taller, y el número de trabajos, y por lo tanto su ámbito de análisis quedará reducido a dos bloques concretos. En función del evento que se produzca en un instante dado deberá consultar el bloque relacionado con el adelanto (destacados con relleno en la Figura 9-4) o con el retraso.

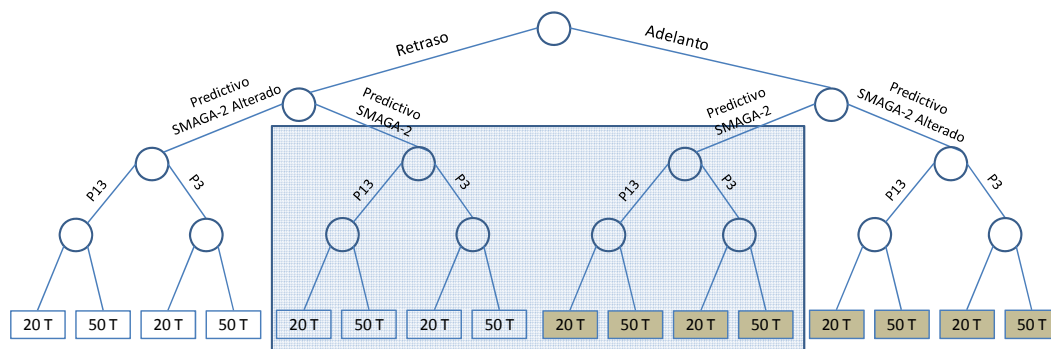


Figura 9-4. Representación de los diferentes ámbitos en los que se realiza en análisis de los factores.

ⁱⁱ La existencia de una varianza heterogénea viola una de las condiciones más importantes que se exigen para el análisis basado en la ANOVA.

En relación a la división realizada en base al Programa Predictivo inicial, diferenciando entre SMAGA-2, y SMAGA-2 Modificado, es necesario indicar que esta separación es consecuencia de una reflexión basada en un primer análisis de los resultados. Como se indico en el apartado dedicado al “Juego de Datos”, inicialmente se pensó efectuar las Reprogramaciones partiendo de tres tipos de Programas Predictivos iniciales. Uno de ellos sería SMAGA-2, y los otros dos serían sendas alteraciones en la secuencia obtenida en el primero. Aunque inicialmente se planteó realizar una media de los resultados y trabajar con dicho valor, cuando se inició el análisis de los resultados se observó la posible existencia de un factor que afectaba a los resultados, y que no había sido considerado en el modelo inicial.

Después de realizar diversas conjeturas se detectó como el Programa Predictivo inicial presenta una influencia sumamente relevante, y que por lo tanto, no tenía sentido alguno trabajar con las medias de los resultados obtenidos, sino más bien considerarlos de forma diferenciada. Aunque se intentó incluir este aspecto como un factor experimental adicional su inclusión produjo siempre una situación en la que se violaban las condiciones necesarias para realizar una ANOVA. Por este motivo se decidió separar los casos en bloques diferentes. En relación con este hecho se ha considerado que el análisis en profundidad se debe realizar sobre las ramas que emplean como algoritmo predictivo SMAGA-2.

SMAGA-2 sin modificaciones es una solución que representa a los algoritmos de mayor eficiencia para el problema estudiado, y que en general no sería lógico plantearse una situación en la que el Programa Predictivo inicial no sea “bueno” en términos de makespan. En todo caso, aunque se ha profundizado en el comportamiento de los factores en relación a la variable independiente sea realizado un resumen de los resultados obtenidos.

Se ha establecido un plan de experimentación de forma que para cada bloque se examinarán cada uno de los cuatro factores que se desea analizar CEX, K, VA y ALG. Cada uno de los cuales dispondrá de diversos niveles (CEX: 4 niveles; K: 3 niveles máximo; VA: 2 niveles; ALGⁱⁱⁱ: 5 niveles como máximo). La Tabla 9-3 muestra los valores que se desean analizar para cada factor.

ⁱⁱⁱ En los experimentos relacionados con adelanto en la fecha de finalización de las operaciones sólo se emplean los algoritmos SSDS y SMAGA-2.

Factores	Valores
Conjunto Experimental (CEX)	SSD10_X_Y; SSD50_X_Y; SSD100_X_Y; SSD125_X_Y X= P13 o P3 Y=20 ó 50
K	(0,0; 0,01; 0; 0,02)*Makespan Predictivo (conf. de 20 trabajos) (0,0; 0,005; 0; 0,01)*Makespan Predictivo (conf. de 50 trabajos)
Ventana de Avance (VA)	(0,2 ; 0,4)*Makespan Predictivo (conf. de 20 trabajos) (0,1;0,2)*Makespan Predictivo (conf. de 50 trabajos)
Algoritmo (ALG)	Retrasos: RS; SSDS; AOR; SMAGA-2s; SMAGA-2 Adelantos: SSDS; SMAGA-2

Tabla 9-3. Niveles asignados a los factores en el experimento de Programación Reactiva.

Se podían haber considerado otras opciones, y seguramente cada diseñador hubiera elegido su propio conjunto de valores para cada factor. No obstante, en función de la descripción realizada del experimento, en donde se especifica con detalle la función de cada factor, se considera que se han seleccionado valores razonables. Será en todo caso el análisis de los resultados obtenidos el que determine la bondad de dichos valores.

La combinación de factores no se aplica de la misma forma a todos los bloques experimentales. En el caso de los algoritmos, que serán descritos en el siguiente apartado, conviene hacer una aclaración. En el caso de retraso en las operaciones los algoritmos RS, AOR y SMAGA-2s no permiten realizar avances en la ejecución de las operaciones, y por lo tanto no incurrirán en ningún tipo de penalización, por lo que no se ha considerado el valor de K superior a cero para dichos casos. En el caso de los adelantos en las operaciones, sólo es razonable emplear el algoritmo SSDS y SMAGA-2 ya que el resto de algoritmos nunca aceptan el adelanto de operaciones, y por lo tanto, nunca mejoraran el Programa Productivo Predictivo. Así mismo los experimentos que parten del Programa Predictivo SMAGA-2 sólo realizan una ejecución al ser este bastante estable, mientras que los que parten de un Programa Predictivo SMAGA-2 Modificado ejecutan 2 Programas Predictivos diferentes. Por este motivo el número de combinaciones es desigual en cada bloque.

Para los 4 bloques con retraso y Programa Predictivo SMAGA-2 la combinación de factores es 4 (CEX) x 1 (K) x 2 (VA) x 3 (ALG) + 4 (CEX) x 3 (K) x 2 (VA) x 2 (ALG), es decir 72 tratamientos. Para los 4 bloques con retraso y Programa Predictivo SMAGA-2 Modificado es el doble ya que se realiza una vez con cada programa inicial, es decir 144 tratamientos. Para los 4 bloques con adelanto y Programa Predictivo SMAGA-2 la combinación es 4 (CEX) x 3 (K) x 2 (VA) x 2 (ALG), es decir 48 tratamientos. Y finalmente, para los 4 bloques con adelanto y Programa Predictivo SMAGA-2 Modificado es el doble que el anterior ya que se realiza una vez con cada Programa

Predictivo, es decir 96 tratamientos. Por lo tanto hay un total de 360 (tratamientos bloque) x 4 (bloques), es decir 1.440.

Cabe recordar que cada conjunto experimental consta de tres grupos, y cada grupo consta de 5 ejemplares. Se ha decidido realizar 50 ejecuciones por ejemplar y combinación de niveles de los factores. **Esto implica que se realizarán 3 (grupos) x 5 (ejemplares) x 1.440 (tratamientos) x 50 (ejecuciones) = 1.080.000 ejecuciones.**

9.3 Descripción de los Algoritmos Analizados.

Con los algoritmos propuestos se pretende examinar propuestas para la Programación Reactiva que implican una modificación parcial del Programa Productivo, y propuestas que emplean la fuerza más propia de la Programación Predictiva realizando una modificación completa del Programa Productivo pendiente de procesar. En ningún caso se pretende realizar un barrido exhaustivo ni de algoritmos de Reprogramación parcial, ni de programación completa. El objetivo es más modesto. Tan sólo se pretende realizar una aproximación a las soluciones de tipo reactivo para el problema propuesto, que permita disponer de una situación de partida para futuras investigaciones, al mismo tiempo que se aprovechará la información que se obtenga para implementar una funcionalidad concreta a la AE-9 de la plataforma IPSU-MAS.

Las propuestas para la Programación Reactiva parcial son en todos los caso de tipo heurístico, y relativamente sencillas de implementar. Como ya se comentó en el capítulo 3, las propuestas de carácter Reactivo parcial suelen estar muy adaptadas a cada al problema concreto que se aborda, y difícilmente se encuentran propuesta de carácter general. Entre las de carácter general se han empleado dos de las más mencionadas en la literatura son el enfoque “Right Shift” (**RS**) y la el algoritmo “Affected Operation Rescheduling” (**AOR**). Así mismo, se ha querido realizar una portación heurística para la Programación Reactiva aplicable de forma específica al problema propuesto, y que complemente a las heurísticas generalistas. Esta se ha denominado “Same Sequence Different Schedule” (**SSDS**).

La técnica Right-Shift (RS) (Conway, R. W. et al. 1967;Sadeh, N. M. et al. 1993) es una heurística que consiste en mover toda la secuencia hacia el futuro, hasta que el evento es superado en el tiempo. La secuencia se respeta y queda tal cual fue establecida inicialmente, permitiendo introducir tiempos muertos.

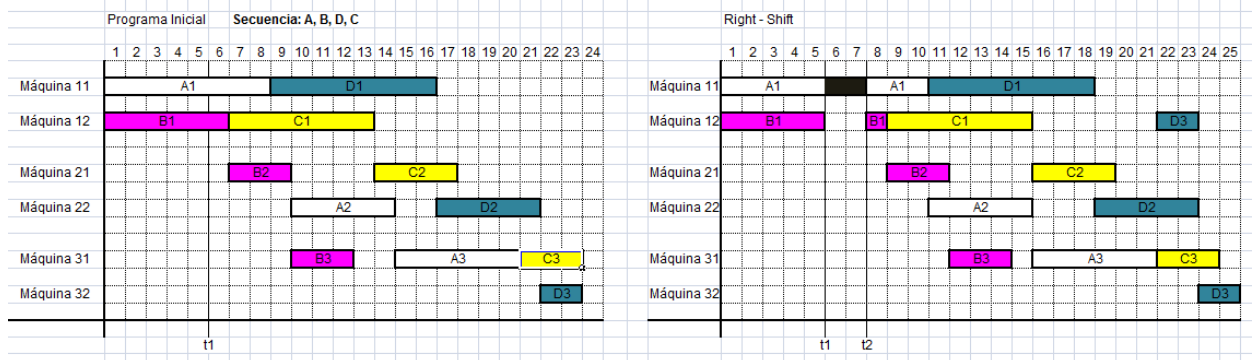


Figura 9-5. Ejemplo en la implementación del algoritmo Right Shift (RS).

Según se muestra en la Figura 9-5, dado un Programa Predictivo en ejecución (izquierda) en el que en el instante “ t_1 ” se produce un evento que implica prolongar la operación 1 del trabajo A, “A1”, en dos unidades de tiempo hasta el instante “ t_2 ”, la solución propuesta por RS adaptada al caso del Taller de Flujo Híbrido es la del gráfico de la derecha. El trabajo “A1” se prorroga 2 unidades de tiempo, y los trabajos que se ejecuta en el resto de máquinas de la etapa 1, máquina 11 y 12, donde se hace realiza la operación afectada también se prorrogan 2 unidades. En este caso, el trabajo “B1” no se interrumpiría durante 2 unidades de tiempo sino que a su finalización habría que esperar 2 unidades de tiempo antes de empezar el siguiente. Una vez desplazados los trabajos de las máquinas de la operación afectada, el resto de operaciones se modifican convenientemente.

Affected Operation Rescheduling (AOR) (Abumaizar, R. J. et al. 1997; Hasle, G. & Smith, S. F. 1994; Leon, V. J., Wu, S. D., & Storer, R. H. 1994) es una técnica heurística en la cual sólo las operaciones que se ven afectadas por los efectos de un evento son modificadas. El concepto básico de esta técnica es el desplazamiento hacia delante de las fechas de inicio de algunas operaciones manteniendo las secuencias de inicio de cada máquina.

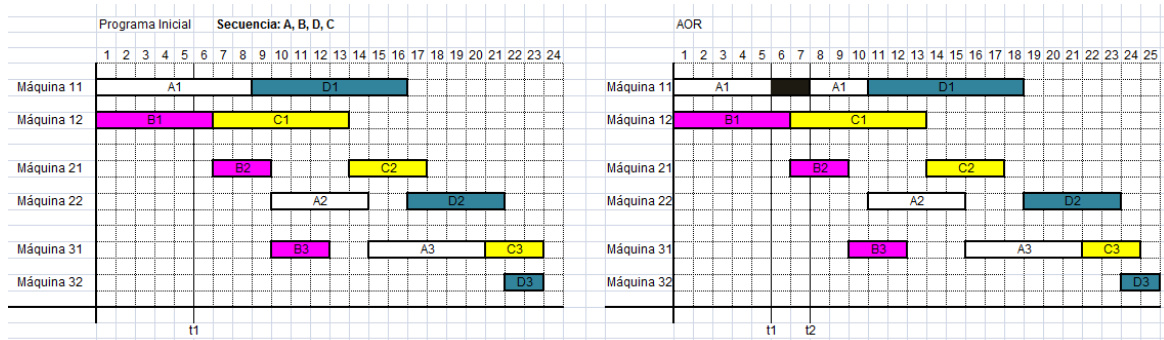


Figura 9-6. Ejemplo en la implementación del algoritmo Affected Operations Rescheduling (AOR).

Según se muestra en la Figura 9-6, dado un Programa Predictivo en ejecución (izquierda) en el que en el instante “t1” se produce un evento que implica prolongar la operación 1 del trabajo A, “A1”, en dos unidades de tiempo hasta el instante “t2”, la solución propuesta por AOR adaptada al caso del Taller de Flujo Híbrido es la del gráfico de la derecha. La operación afectada inicialmente “A1” se prorroga 2 unidades de tiempo desplazando en esa etapa sólo al trabajo “D1” que se iniciaba en el instante 9 y ahora se inicia en el 11. En etapas posteriores las operaciones afectadas del trabajo A y D se retrasan el tiempo necesario. Obsérvese como la operación “A2” comenzaba en el instante 10 y después de Reprogramar comienza en el instante 11, es decir que el retraso inicial de 2 unidades no se traslada a todas las operaciones, tan sólo se ajustan los inicios y finales, según las fechas de finalización de los trabajos en las etapas precedentes.

Se puede considerar que RS es un caso especial de AOR en el que se inicialmente no hay una operación afectada sino tantas como recursos existen en la misma etapa. Se podrá observar como tanto en RS, como en AOR, la asignación de los trabajos a las máquinas permanece inalterada.

El método “Same Sequence Different Schedule” (SSDS), de elaboración propia, propone la modificación de la fecha de fin de la operación afectada manteniendo la secuencia, pero permitiendo reasignar los trabajos pendientes en los recursos disponibles en cada etapa sin restricciones. Como ya se ha mencionado en varias ocasiones, la resolución del problema de Programación de la Producción en un Taller de Flujo Híbrido implica abordar dos aspectos: secuenciación y asignación. La heurística SSDS se puede ver como una modificación de AOR, ya que igual que esta última SSDS propone actuar sobre la operación afectada directamente, y sobre las afectadas indirectamente como consecuencia de la reparación de la primera, respetando la secuencia original.

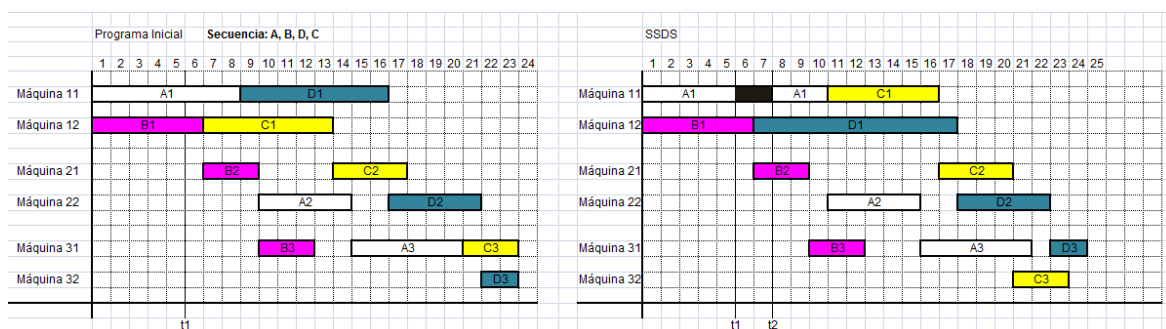


Figura 9-7. Ejemplo en la implementación del algoritmo Same Sequence Different Schedule (SSDS).

A diferencia de AOR, SSDS no mantiene la asignación trabajo-máquina que inicialmente se estableció, permitiendo la búsqueda de nuevas combinaciones trabajo-máquina dentro de la misma etapa. Este tipo de Reprogramación, es la más intuitiva, y podría ser bastante adecuada, a priori, para aplicar en los talleres cerámicos en el caso de tener que aplicar cambios al Programa Productivo en curso.

En la Figura 9-7 se muestra una Reprogramación empleando la heurística SSDS. En este caso se retrasa 2 unidades la operación "A1", y se analiza la secuencia inicial "A, B, D, C" reasignando de forma parcial todos los trabajos que en la secuencia se encuentren por detrás de "A", cuya operación 1 no haya sido realizada total o parcialmente, es decir la sub-secuencia "D, C". Como se ve, en este caso se respeta la secuencia inicial pero la asignación de recursos cambia, y por ende el reajuste de las fechas de inicio y fin del resto de operaciones. En concreto al retrasar el final de la operación "A1" hasta el instante 11 la operación "D1" se ha asignado a la máquina 12, y entonces la operación "C1" se ha asignado a la máquina 11. La diferencia fundamental con AOR y RS es la posibilidad de modificar los recursos asignados inicialmente.

Además de las heurísticas para la Reprogramación parcial de la Producción, también se han considerado un procedimiento metaheurísticos para la Reprogramación completa de la Producción. La razón que justifica su inclusión es que a priori parece razonable que, por lo que se refiere a makespan, una Reprogramación completa sea más eficiente que una Reprogramación parcial. Por lo tanto, se espera que cuando el valor de K sea cero el mejor algoritmo sea el de Reprogramación completa. Por lo que el procedimiento de Reprogramación completa puede servir de punto de referencia para el resto de situaciones y algoritmos.

En cuanto al algoritmo seleccionado no tiene relevancia, ya que se considera que cualquier algoritmo que proporcione buenos resultados es válido para obtener la referencia esperada. Se ha seleccionado **SMAGA-2**, ya que se demostró en el capítulo 8 que se trata de un buen Algoritmo Predictivo, y que por lo tanto se puede emplear para la Reprogramación completa, y también porque el amplio conocimiento que se tiene sobre la implementación del mismo facilita realizar algunas modificaciones necesarias. Efectivamente, para utilizar el algoritmo SMAGA-2 en la Programación Reactiva completa requiere modificar el algoritmo para que re programe en cada ocasión un número de trabajos variable, manteniendo sin alterar aquellos que ya se habían ejecutado o que estaban en proceso. Pero además, se desea considerar dos versiones de este algoritmo. La primera versión, que se denominará SMAGA-2s, incluirá una restricción que impedirá que se consideren movimientos de trabajos que pudieran suponer una penalización, es decir avance de trabajos cuya primera operación pendiente se realice en el nuevo programa dentro de la Ventana de Avance. La segunda versión, se denominará SMAGA-2, y consistirá en una implementación sin restricciones del algoritmo. Ambos algoritmos se ejecutarán con un límite de 5.000 iteraciones.

Con las dos versiones de Reprogramación completa, con y sin restricciones, incluyen alternativas que pueden mostrar su bondad en situaciones diferentes. Por un lado, una versión SMAGA-2 puede ser la mejor opción a efectos de makespan, aunque si se penalizan los adelantos puede reducir su efectividad. Por otro lado, se dispone de SMAGA-2s, que aunque no alcance un makespan mejor que SMAGA-2, nunca se incurrirá en penalizaciones, y por lo tanto puede ser interesante cuando el valor de K sea superior a cero.

9.4 Desarrollo del Diseño Experimental

Una vez implementados todos los algoritmos seleccionados para realizar la Programación de la Producción Reactiva, en un Taller de Flujo Híbrido con Tiempos de Cambio de Partida Dependientes de la Secuencia, y que por lo tanto estar disponible el objeto de análisis, es cuando el experimentador está en condiciones de seguir desarrollando los diferentes pasos de que consta el diseño experimental.

9.4.1 Procedimiento de Ejecución los Experimentos

Para el desarrollo de los experimentos se han empleado 4 ordenadores tipo PC con procesador Intel Core 2 a 2.66 GHz, con memoria de 2 GByte, y sistema operativo Windows XP Service Pack 3. De forma que cada conjunto experimental se ha ejecutado de forma aleatoria en uno de los cuatro ordenadores. Cada ejemplar de cada conjunto experimental se ha Programado Predictivamente, y con dicho programa de referencia se ha realizado un experimento para una determinada combinación de factores, volviendo a recuperar el programa de referencia para realizar otra repetición con la misma combinación, y así hasta cincuenta ejecuciones por combinación y Programa de Predictivo de referencia. Una vez realizados los experimentos para cada Programa Predictivo y combinación de parámetros se ha guardado el resultado en un fichero. En dicho fichero se han escrito diversos datos, algunos de los cuales han sido empleados para el análisis de los resultados, otros se han empleado como simples elementos de identificación, y otros se han obtenido con espíritu de realizar futuros análisis que sirvan de complemento a este trabajo. Los datos almacenados han sido los siguientes:

- Valor de IPSPPVEP del ejemplar.
- Valor del makespan obtenido para el Programa Predictivo del ejemplar.
- Identificación del ejemplar y del conjunto experimental ejecutado.
- Identificación del valor de los parámetros ejecutados.

- Fecha y hora de fin de la ejecución del ejemplar.

Una vez finalizados todas las ejecuciones de cada conjunto experimental los datos se han consolidado en una hoja Excel en la cual se dispuso el valor de IPSPPVEPT para cada combinación de niveles de los factores.

Los datos se han exportado a Statgraphics Plus para Windows versión 5.0 (ver <http://www.statgraphics.com>). Previamente se ha creado un diseño de experimentos mediante la opción de "Multi-Factor Categorical", con una variable de respuesta y un máximo de cuatro factores (dependiendo del bloque), realizando un diseño personalizado con diferentes niveles por factor, con una réplica no aleatoria para el caso de los bloques correspondientes al Programa Predictivo SMAGA-2 Modificado y sin réplica en caso de partir del Programa Predictivo SMAGA-2. La codificación de los factores sigue la misma nomenclatura expresada hasta el momento.

9.4.2 Análisis de los resultados

En general el análisis de los resultados implica poder dar respuesta satisfactoria a las siguientes cuestiones:

- ¿Cuál es el algoritmo que mejor se comporta en general?
- ¿Existe alguna interacción entre el factor ALG y cualquier otro factor? Si es así, ¿En que afecta dicha interacción a los algoritmos con mejor comportamiento?, ¿Cuáles son los algoritmos con mejor comportamiento en cada combinación de interacciones significativas?
- ¿Existe alguna interacción significativa entre los factores CEX, K y VA interesante? ¿Cuál es efecto y como afecta a las decisiones del programador?

A partir de estas cuestiones básicas se propondrán otras más concretas según sea necesario. En todo caso, las respuestas a estas y otras cuestiones estarán basadas en el análisis de los datos obtenidos en la experimentación. Algunas cuestiones podrán encontrar respuesta mediante la simple observación de las tablas de resultados y otras necesitarán de un estudio estadístico riguroso. En el anexo IV se muestran todos los resultados, así como el análisis ANOVA de todos los bloques con su correspondiente verificación de las hipótesis básicas de normalidad, homocedasticidad y linealidad para lo cual se ha empleado el mismo tipo de gráficas que en el capítulo 8.

A continuación se abordarán las cuestiones planteadas en para cada uno de los bloques en los que se ha dividido el experimento para posteriormente realizar un resumen de las más relevantes.

9.4.2.1 Análisis de los resultados en situación de Adelanto de las Operaciones

Como ya se ha comentado con anterioridad en este capítulo, durante el análisis preliminar de los resultados obtenidos, se detectó la existencia de un comportamiento muy diferente de la función objetivo en relación con los factores considerados dependiendo de si el Programa Predictivo de partida era un programa con un makespan lo más próximo al óptimo posible o si era un Programa Predictivo con un makespan no tan bueno como el anterior. Por este motivo se ha diferenciado el análisis de los resultados dependiendo del Programa Predictivo de partida.

Así mismo, también se debe hacer hincapié en el hecho de que de todos los métodos propuestos para la Reprogramación sólo dos de ellos son capaces de proponer un programa alternativo cuando la fecha de finalización de una operación se adelanta. Tanto el método RS, como los métodos AOR y SMAGA-2s son incapaces de desplazar a la izquierda el diagrama Gantt de las operaciones debido a su forma de actuar. Por este motivo, sólo para el caso de adelanto en las operaciones se estudian dos algoritmos: SSDS y SMAGA-2. No obstante, se entienden que este tipo de situaciones con adelanto ocurren con mucha menos frecuencia en la Industria Cerámica que las situaciones de retraso.

9.4.2.1.1. Casos con Programación Predictiva SMAGA-2

En este apartado se ha realizado un estudio estadístico en profundidad para intentar profundizar en el análisis dando cobertura a todas las cuestiones generales planteadas, partiendo siempre de la hipótesis de que el Programa Predictivo inicial alcanza unos valores de makespan muy buenos.

Han sido 4 los casos que partiendo de un Programa Predictivo altamente competitivo han sido posteriormente sometidos a eventos que les ha supuesto una alteración de alguna de las operaciones en curso provocando una reducción de la misma. El Programa Predictivo empleado ha sido SMAGA-2, y los casos estudiados se han diferenciado por su configuración de taller y por su número de trabajos. Estos bloques experimentales están caracterizados por el conjunto experimental seleccionado han sido los denominados como: P13_20, P13_50, P3_20 y P3_50. Que se corresponden con las 4 alternativas descritas para la duración de los

Tiempos de Cambio de Partida dependientes de la secuencia y la configuración los talleres con una configuración de entre 1 y 3 máquinas por etapas y 20 trabajos por periodo (SSD10_P13_20, SSD50_P13_20, SSD100_P13_20, y SSD125_P13_20), o bien la configuración de entre 1 y 3 máquinas por etapas y 50 trabajos por periodo (SSD10_P13_50, SSD50_P13_50, SSD100_P13_50, y SSD125_P13_50), o bien con 3 máquinas por etapas y 20 trabajos por periodo (SSD10_P3_20, SSD50_P3_20, SSD100_P3_20, y SSD125_P3_20) , o bien con 3 máquinas por etapas y 50 trabajos por periodo (SSD10_P3_50, SSD50_P3_50, SSD100_P3_50, y SSD125_P3_50).

Tanto para el caso P13_20^{iv}, como para el P13_50^v, P3_20 y P3_50 se muestra en los resultados de la ANOVA (ver anexo IV) que el factor principal ALG es significativo. Y en todos los casos en promedio los mejores resultados se obtienen con el algoritmo SSDS. No obstante, la ANOVA también indica como siempre es significativa la interacción del factor ALG tanto con CEX, como con K. Y en el caso P3_50 también lo es la interacción entre VA y ALG. A continuación se muestran las gráficas de interacción entre los diferentes factores para los bloque experimentales analizados.

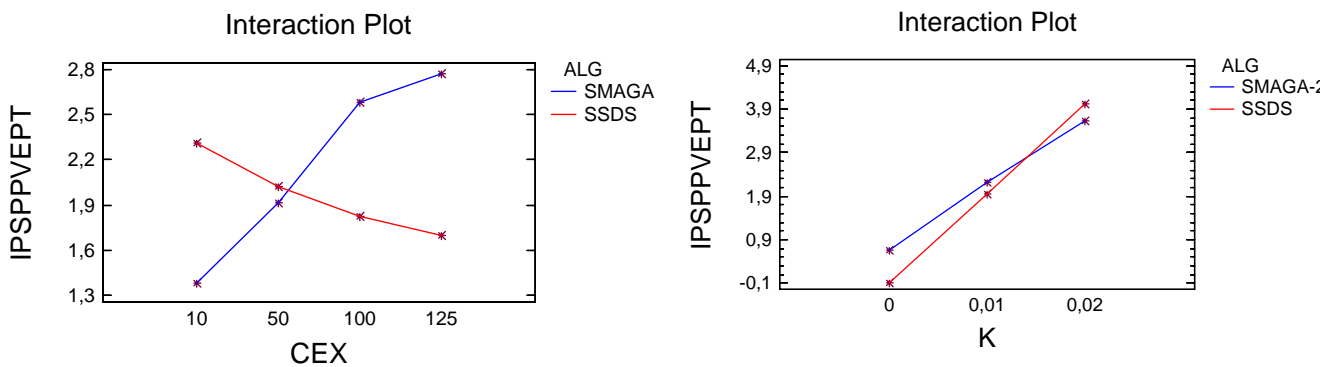


Figura 9-8. Gráfico del efecto de la interacción entre el factor CEX y el factor ALG (izquierda) y entre el factor K y ALG (derecha) del bloque Adelanto-Predictiva SMAGA-2-P13_20.

^{iv} Para este bloque experimental, y para el bloque P3_20, los factores (y valores) estudiados son: CEX (10, 50, 100, 125)^{iv}, K (0, 0.01, 0.02), VA (0.2, 0.4), ALG (SSDS y SMAGA-2).

^v Para este bloque experimental, y para el bloque P3_50, los factores (y valores) estudiados son: CEX (10, 50, 100, 125)^v, K (0, 0.005, 0.01), VA (0.1, 0.2), ALG (SSDS y SMAGA-2).

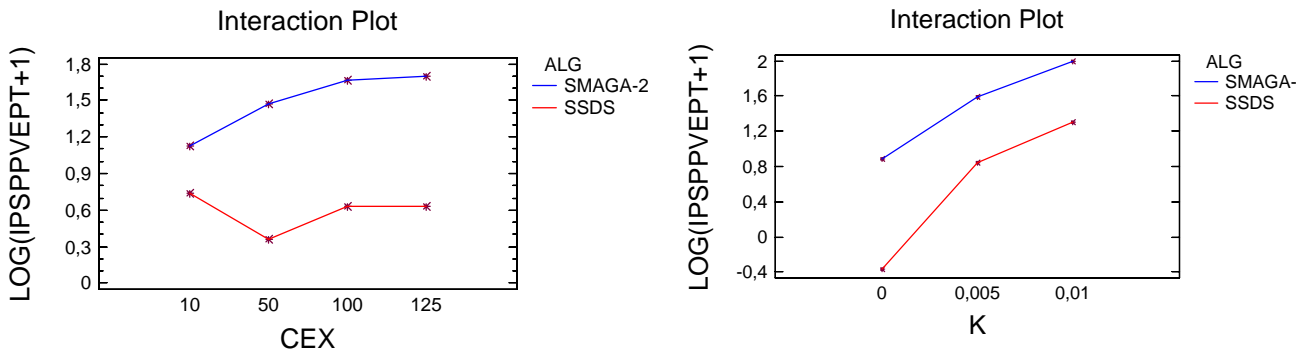


Figura 9-9. Gráfico del efecto de la interacción entre el factor CEX y el factor ALG (izquierda) y entre el factor K y ALG (derecha) del bloque Adelanto-Predictiva SMAGA-2-P13_50.

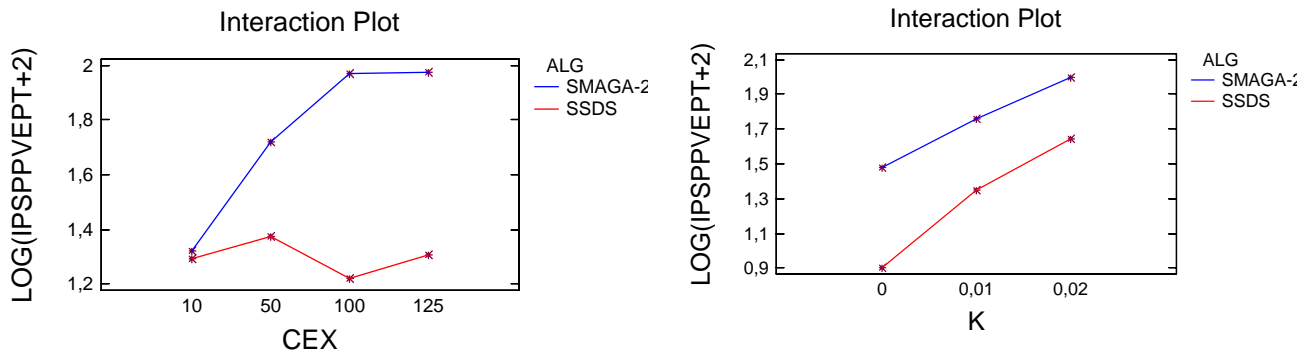


Figura 9-10. Gráfico del efecto de la interacción entre el factor CEX y el factor ALG (izquierda) y entre el factor K y ALG (derecha) del bloque Adelanto-Predictiva SMAGA-2-P3_20.

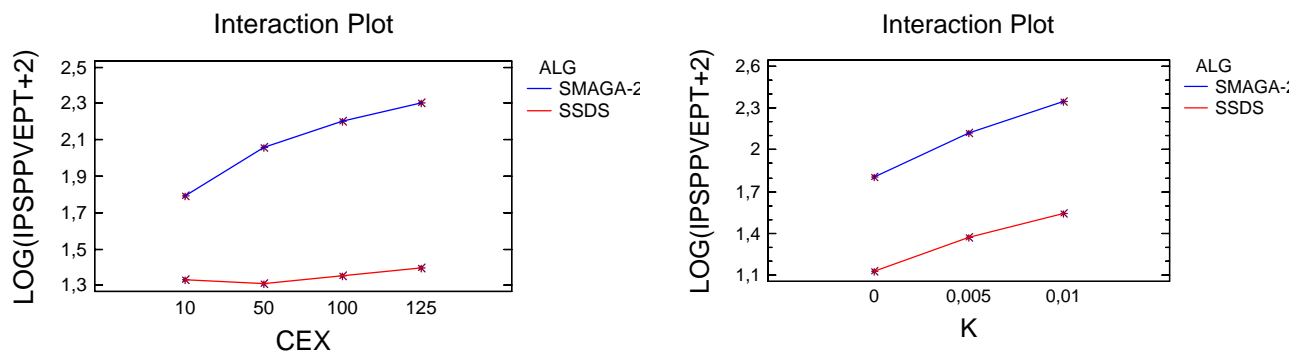


Figura 9-11. Gráfico del efecto de la interacción entre el factor CEX y el factor ALG (izquierda) y entre el factor K y ALG (derecha) del bloque Adelanto-Predictiva SMAGA-2-P3_50.

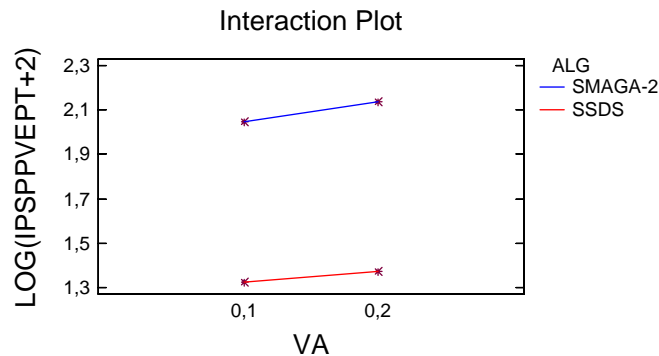


Figura 9-12. Gráfico del efecto de la interacción entre el factor VA y el factor ALG del bloque Adelanto-Predictiva SMAGA-2-P3_50.

En la Figura 9-8 (izquierda) se puede observar como existe una tendencia creciente del algoritmo SMAGA-2 en relación con el Tiempo de Cambio de Partida, mientras que la tendencia es decreciente para SSDS. Esto podría indicar **que cuando los Tiempos de Cambio de Partida son elevados es mejor reasignar recursos sin modificar la secuencia (que era inicialmente buena), que reprogramar por completo los trabajos pendientes**. En esa misma figura, a la derecha, se observa como ambos algoritmos empeoran cuando el valor de K crece, lo que es lógico, ya que ello implica el incremento de la penalización por trabajos adelantados dentro de VA. No obstante, se observa como parece que dicho crecimiento presenta una pendiente más suave para SSDS que para SMAGA-2, lo que **se deduce que con fuertes penalizaciones sería mejor usar SSDS**.

En el lado izquierdo de la Figura 9-9, Figura 9-10 y Figura 9-11 se puede observar como existe una tendencia creciente del algoritmo SMAGA-2 en relación al Tiempo de Cambio de Partida, mientras que SSDS es más irregular, y fluctúa dentro de un margen, sin presentar una tendencia creciente. **En todos los casos indicados es siempre más interesante reasignar recursos que reprogramar**. Así mismo, se observa como en general los valores obtenidos en el bloque P13_50 son menores que en bloque P3_20, que a su vez son menores que en bloque P3_50. Sin embargo, los valores obtenidos en el primer bloque, P13_20, son en general superiores a los tres bloques recién comentados. Por lo que se refiere a la parte derecha de la figura, ambos algoritmos crecen con respecto al incremento de K, pero en estos casos el valor de SSDS es siempre menor que el de SMAGA-2.

La ANOVA muestra que para el caso P13_20, el caso P3_20 y el caso P3_50 también son significativos los factores principales CEX, K y VA. En cuanto a CEX y K se espera que sean significativos en todos los casos, y no sólo en este bloque, ya que se intuye que existe una gran diferencia entre los resultados que se pueden alcanzar con diferentes Tiempos de Cambio de

Partida, así como cuando se aplican diversos grados de penalización por adelanto de trabajos dentro de VA. Por lo tanto, el factor CEX y K posibilitan el aumento o reducción de la función objetivo de forma directa. Sin embargo, la significación de VA, y su interacción con otros factores es más incierta. En los casos P13_20, P3_20 y P3_50 es significativo y en el P13_50 no es significativo. En principio, VA puede afectar al resultado, en el caso de adelantos, ya que al aumentar la ventana sobre la cual está prohibido adelantar los trabajos aumentan el riesgo de ser penalizados en el caso de reprogramación. Por lo explicado, **se puede deducir como es normal que K y VA por separado o ambos en combinación condicionen el resultado**. De hecho la ANOVA ha confirmado que la interacción entre el factor K y el factor VA es significativa para P13_20, P13_50, P3_20 y P3_50.

Además en el único caso en el que se ha podido confirma la significación en la interacción entre ALG y VA ha sido en el P3_50, en el resto de casos no se puede decir que existan diferencias en el resultado elegido dependiendo de la combinación de ambos factores. Aunque en este caso existe una significación estadística, que nos llevaría a pensar que deberíamos de tener en cuenta cual es el valor de VA antes de elegir el algoritmo a aplicar, desde el punto de vista de la ingeniería no sería relevante.

En la Tabla 9-4 se proporciona un resumen de los factores significativos para cada bloque experimental, así como el criterio de selección del algoritmo que proporciona mejores resultados en cada ocasión.

Bloque Experimental	Factores principales significativos	Factores Combinados significativos	Mejor Algoritmo
P13_20	CEX, K, VA, ALG	CEX ALG, K ALG, K VA	K=0→SSDS K>0 y SSD100 SSD125→SSDS K>0 y SSD10 SSD50→SMAGA-2 (excepto con en SSD50 con VA=0.4 →SSDS)
P13_50	K, ALG	CEX ALG, K ALG, K VA	SSDS
P3_20	CEX, K, VA, ALG	CEX K, CEX ALG, K ALG, K VA, VA ALG	SSD10 y K=0→SSDS SSD50 SSD100 SSD125→SSDS K=0,02 y SSD10→SMAGA-2 K=0,01 y VA=0,2 y SSD10→SMAGA-2 K=0,01 y VA=0,4 y SSD10→SSDS
P3_50	CEX, K, VA, ALG	CEX K, CEX ALG, K ALG, K VA, VA ALG	SSDS

Tabla 9-4. Resumen del análisis de los resultados del bloque experimental P13 y P3 para el adelanto en la finalización de las operaciones.

En base a las observaciones realizadas se pueden establecer algunas conclusiones generales, que son:

- El algoritmo SSDS proporciona unos resultados mejores en la gran mayoría de los casos.
- Existe una fuerte interacción entre el algoritmo (ALG) y el conjunto experimental (CEX). En este contexto, SMAGA-2 sólo es competitivo cuando los Tiempos de Cambio de Partida son reducidos (SSD10 e incluso SSD50), y el número de trabajos también reducido (20 trabajos).
- Los peores valores del mejor algoritmo se obtiene para el bloque P13_20. Este bloque se ha mostrado como uno de los que más distancia existe entre el Programa Predictivo y el Programa Reactivo, y por lo tanto más difíciles de reprogramar.
- La interacción entre el algoritmo y el parámetro K ha mostrado que en todos los bloques SSDS y SMAGA-2 incrementan su valor cuando se incrementa el valor de K, y en todo los casos menos en el bloque P13_20 lo hacen con una pendiente muy similar. Este último dato, refuerza la conclusión de que en los bloques P13_20, P3_20 y P3_50 el algoritmo SSDS es mejor algoritmo en base a la obtención de un buen makespan en la función objetivo, ya que parece que la penalización por adelantar trabajos es similar a la de SMAGA-2. Por lo tanto, en los casos indicados la reasignación de recursos es preferible al reordenamiento de la secuencia de trabajos.

9.4.2.1.2. Casos con Programación Predictiva SMAGA-2 Modificada

También han sido 4 los casos que partiendo de un Programa Predictivo mejorable, es decir con un makespan que podría mejorar de forma significativa, han sido posteriormente sometidos a eventos que les ha supuesto una alteración de alguna de las operaciones en curso provocando una reducción de la misma. El Programa Predictivo empleado ha sido SMAGA-2, pero a la secuencia obtenida se le ha realizado un intercambio entre dos posiciones de la misma, por lo que se le ha denominado SMAGA-2 Modificado. Los bloques experimentales: P13_20, P13_50, P3_20 y P3_50. Estos bloques contienen los mismos ejemplares que los indicadas en el apartado anterior.

Se ha considerado que el hecho de partir con un Programa Predictivo elaborado de forma que inicialmente no sea el mejor posible no es un punto de partida deseable, y por lo tanto, debe ser poco empleado. Por este motivo, se ha realizado un estudio básico de los resultados, con el

simple objetivo de detectar cual es el mejor algoritmo posible en las diversas circunstancias, pero sin entrar en una análisis factorial sobre la influencia e interacciones de otros factores.

En la Tabla 9-5 se proporciona un resumen del criterio de selección del algoritmo que proporciona mejores resultados para cada bloque experimental.

Bloque Experimental	Mejor Algoritmo
P13_20	SMAGA-2
P13_50	SMAGA-2
P3_20	SMAGA-2
P3_50	SMAGA-2, excepto: K=0.01 y SSD100 SSD125 → SSDS y K=0.01 y SSD50 y VA=0.1 → SSDS

Tabla 9-5. Resumen del análisis de los resultados del bloque experimental P13 y P3 para el adelanto en la finalización de las operaciones.

Como se puede observar el criterio de selección no está en absoluto relacionado con el de los casos cuyo algoritmo predictivo es un Programa Productivo con makespan reducido. En este caso el algoritmo que predomina es SMAGA-2, es decir, **en la mayoría de los casos es mejor reprogramar de nuevo frente a la opción de reasignar aprovechando la secuencia existente.**

9.4.2.2 Análisis de los resultados en situación de Retraso de las Operaciones

Tal y como se ha realizado anteriormente, se va a diferenciar entre los resultados obtenidos partiendo de un Programa Predictivo con un makespan próximo al óptimo, y los obtenidos con un Programa Predictivo con un makespan alejado del primero.

En el caso de retraso en las operaciones se emplearán 5 algoritmos diferentes: RS, AOR, SSDS, SMAGA-2 y SMAGA-2s. El retraso en la fecha de fin de las operaciones es uno de los eventos más habituales, se cree que mucho más que el adelanto, por lo que las conclusiones de este apartado se consideran especialmente interesantes para el programador de la producción.

Han que destacar que se sabe de antemano que no todos los métodos propuestos son sensibles a los factores K y VA. En concreto el algoritmo RS y AOR nunca se ven afectados por los valores que puedan tomar dichos factores. El algoritmo SMAGA-2s sólo se ve afectado por el valor de VA, ya que contra más grande se dicho valor menor será el número de trabajos que podrá reprogramar. Y los algoritmos SSDS y SMAGA-2 se ven afectados por los valores que

tomen los factores comentados. En concreto SSDS y SMAGA-2 alcanzan sus mejores valores con K igual a 0, valores que no pueden ser superados con valores de K superiores a 0, es más el resultado obtenido debería ser peor cuanto más alto sea el valor de K.

9.4.2.2.1. Casos con Programación Predictiva SMAGA-2

Las consideraciones realizadas en el apartado 9.4.2.1.1 son válidas en este. Los bloques experimentales que se estudian son los mismos que en dicho apartado. Estos son: P13_20, P13_50, P3_20 y P3_50.

Debido a las dificultades encontradas en el proceso de tratamiento estadístico de los datos para que el análisis ANOVA cumpliera la condición de igualdad en la varianza entre grupos se ha procedido analizando inicialmente, en todos los bloques experimentales, sólo el caso en el que K toma el valor 0. Esto ha permitido superar la condición indicada sobre las varianzas. Así mismo ha permitido simplificar el estudio no analizando el resto de valores de K siempre que los algoritmos que dependen de éste, que empeoran con su incremento, se han mostrado claramente dominados por otros, cuando el valor de K es 0. Hay que tener en cuenta que fundamentalmente el estudio experimental pretende conocer cual es el comportamiento de los diferentes algoritmos, siempre y cuando sean susceptibles de ser empleados. En este aspecto, ha sido necesario ampliar el estudio para valores superiores a 0, pero sólo para el valor inmediatamente superior, con el fin de analizar algunos conjuntos experimentales de los bloques P13_20, P13_50 y P3_20.

Tanto para el caso P13_20^{vi}, como para el P13_50^{vii}, P3_20 y P3_50 se observa en los resultados de la ANOVA (ver anexo IV) que el factor principal ALG es significativo siempre. Aunque no existe un algoritmo claramente dominador en las diferentes situaciones. Por este motivo hay que interpretar cuidadosamente las interacciones. En este aspecto la ANOVA muestra como también es siempre significativa la interacción entre el factor ALG y el factor CEX. Y en todos los casos, menos en el P3_20, también lo es la interacción entre VA y ALG.

^{vi} Para este bloque experimental, y para el bloque P3_20, los factores (y valores) estudiados son: CEX (10, 50, 100, 125), K (0, 0.01, 0.02), VA (0.2, 0.4), ALG (RS, SSDS, AOR, SMAGA-2s y SMAGA-2). Del factor K no se ha tenido en cuenta el valor 0.02.

^{vii} Para este bloque experimental, y para el bloque P3_50, los factores (y valores) estudiados son: CEX (10, 50, 100, 125), K (0, 0.005, 0.01), VA (0.1, 0.2), ALG (RS, SSDS, AOR, SMAGA-2s y SMAGA-2). Del factor K no se ha tenido en cuenta el valor 0.01.

A continuación se muestran las gráficas de interacción entre los diferentes factores para los bloque experimentales analizados. En los casos en los que se ha estudiado por separado los resultados obtenidos con $K=0$ y $K>0$ se muestran ambas gráficas.

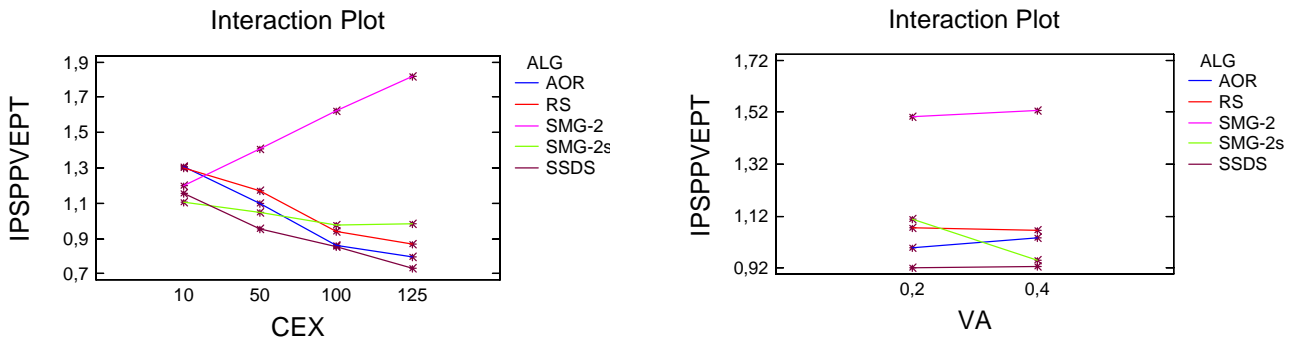


Figura 9-13. Gráfico del efecto de la interacción entre el factor CEX y el factor ALG (izquierda) y entre el factor VA y ALG (derecha) del bloque Retraso-Predictiva SMAGA-2-P13_20 ($K=0$).

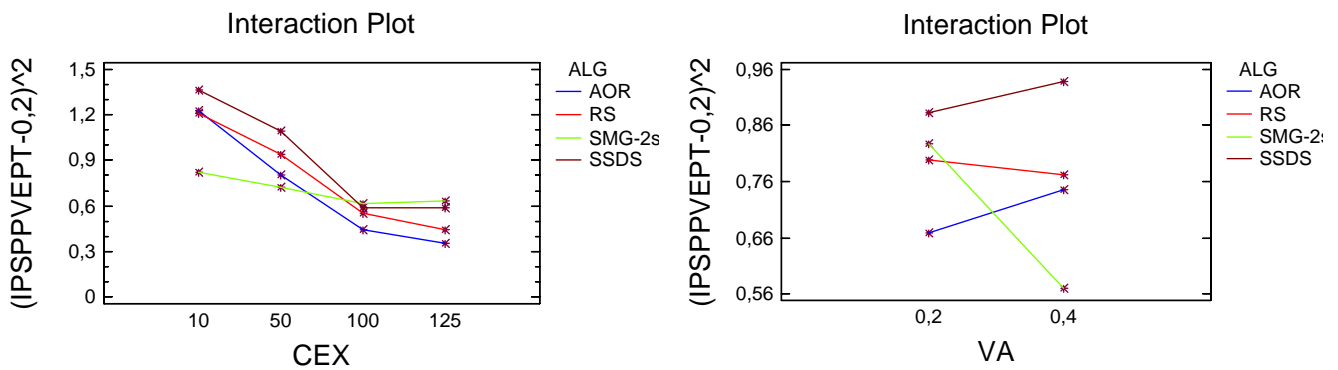


Figura 9-14. Gráfico del efecto de la interacción entre el factor CEX y el factor ALG (izquierda) y entre el factor VA y ALG (derecha) del bloque Retraso-Predictiva SMAGA-2-P13_20 ($K=0.01$).

El análisis de los resultados obtenidos para el bloque experimental P13_20 nos presenta la posibilidad de emplear hasta 3 algoritmos diferentes según la combinación de parámetros. En la Figura 9-13 (izquierda) se puede observar como en el bloque experimental P13_20 el método SSDS obtiene el mejor valor para todos los conjuntos experimentales menos para el de Tiempo de Cambio de Partida más reducido (SSDS10_P13_20) en el que el algoritmo SMAGA-2s es un poco mejor. En todo caso, SMAGA-2s se va mostrando como un algoritmo muy poco recomendable, que se puede descartar. Dado que el procedimiento SSDS es dependiente del valor de K se ha analizado, en la Figura 9-14 (izquierda), lo que ocurre cuando este valor aumenta. Cuando el valor de K es 0.01 el método SSDS deja de ser competitivo, situándose entre los peores. Con este valor son los algoritmos SMAGA-2s, para Tiempos de Cambio de Partida pequeños, y AOR, para Tiempos de Cambio de Partida grandes, los que presentan

mejores resultados. No es necesario analizar lo que ocurriría con $K=0.02$ ya que ni SMAGA-2s, ni AOR dependen del valor de K .

En cuanto a la relación entre ALG y VA se observa como existe una cierta tendencia a que los algoritmos que son buenos en su interacción con el factor CEX son también buenos en esta interacción. También es relevante como SMAGA-2s es mejor cuando mayor es el valor de VA, es decir, cuanto menos opciones tiene de modificar la secuencia y más se parece en su forma de operar al procedimiento SSDS. El resto de métodos presentan muy pequeñas diferencias que pueden ser debidas al azar implícito en la simulación.

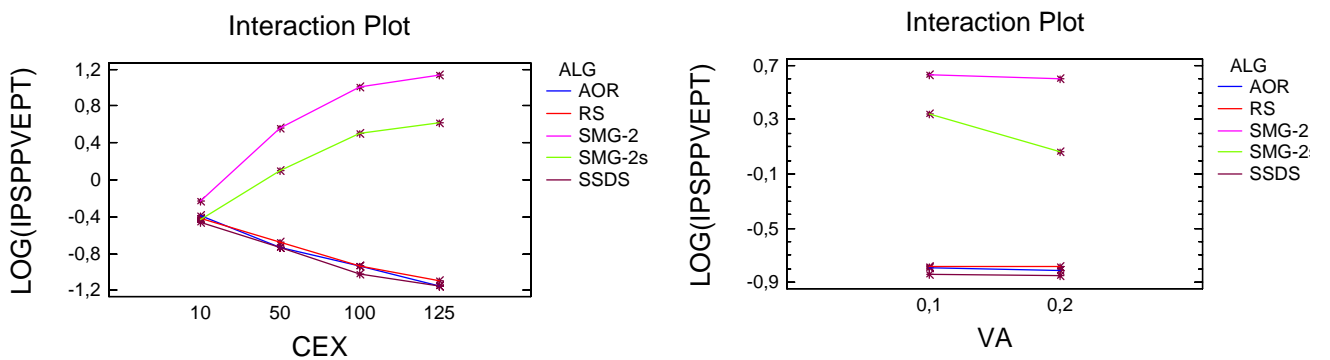


Figura 9-15. Gráfico del efecto de la interacción entre el factor CEX y el factor ALG (izquierda) y entre el factor K y ALG (derecha) del bloque Retraso-Predictiva SMAGA-2-P13_50 ($K=0$).

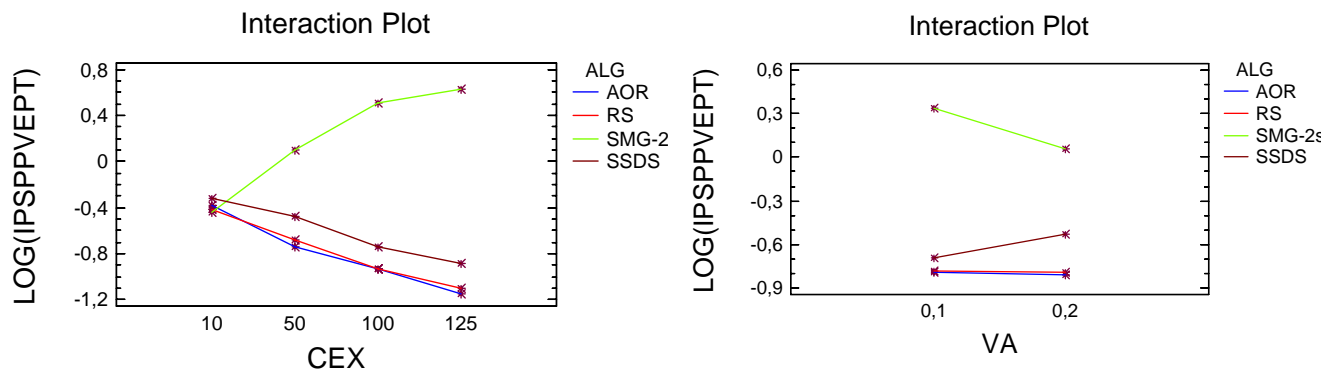


Figura 9-16. Gráfico del efecto de la interacción entre el factor CEX y el factor ALG (izquierda) y entre el factor K y ALG (derecha) del bloque Retraso-Predictiva SMAGA-2-P13_50 ($K=0.005$).

En la Figura 9-15 (izquierda) se puede observar como para el bloque P13_50 los algoritmos con Reprogramación completa (SMAGA-2 y SMAGA-2s) son los peores en para todos los conjuntos experimentales. El resto de métodos se muestran muy similares desde el punto de

vista ingenieril. Si se observa la parte derecha de la gráfica se puede ver como ocurre exactamente lo mismo en relación al factor VA. Sin embargo, cuando el valor de K se incrementa a 0.005 el método SSDS deja de ser competitivo, dejando al algoritmo AOR y RS como las mejores alternativas, tal y como se puede ver en la Figura 9-16. Como ya se ha dicho con anterioridad estos dos algoritmos seguirían siendo las mejores opciones incluso con $K=0.01$.

Por lo que respecta a la interacción entre el factor ALG y el factor VA, tanto la Figura 9-16, como la Figura 9-17 muestran con claridad como los algoritmo RS, AOR no dependen del valor de VA, así como SMAGA-2 para el caso de $K=0$. Así mismo se mantiene el comportamiento observado en SMAGA-2s proporcionando resultados inversamente proporcionales al valor de VA.

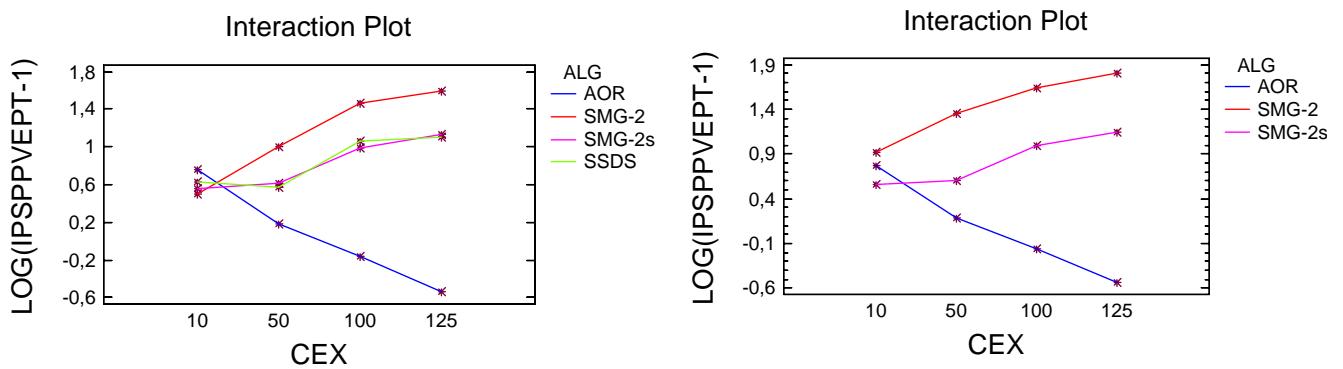


Figura 9-17. Gráfico del efecto de la interacción entre el factor CEX y el factor ALG con $K=0$ (izquierda) y con $K=0.01$ (derecha) del bloque Retraso-Predictiva SMAGA-2-P3_20.

En el caso del bloque experimental P3_20 la única interacción significativa es la que existe entre el factor CEX y el factor ALG. En la Figura 9-17 se puede ver como el algoritmo AOR es el mejor para todos los conjuntos experimentales, excepto el conjunto con Tiempo de Cambio de Partida más reducido (SSD10_P3_20), para cualquier valor de K. Sin embargo, el conjunto experimental SSD10_P3_20 presenta mejores resultados con el método SMAGA-2 si el valor de K es 0, y con el método SMAGA-2s si K es mayor de 0.

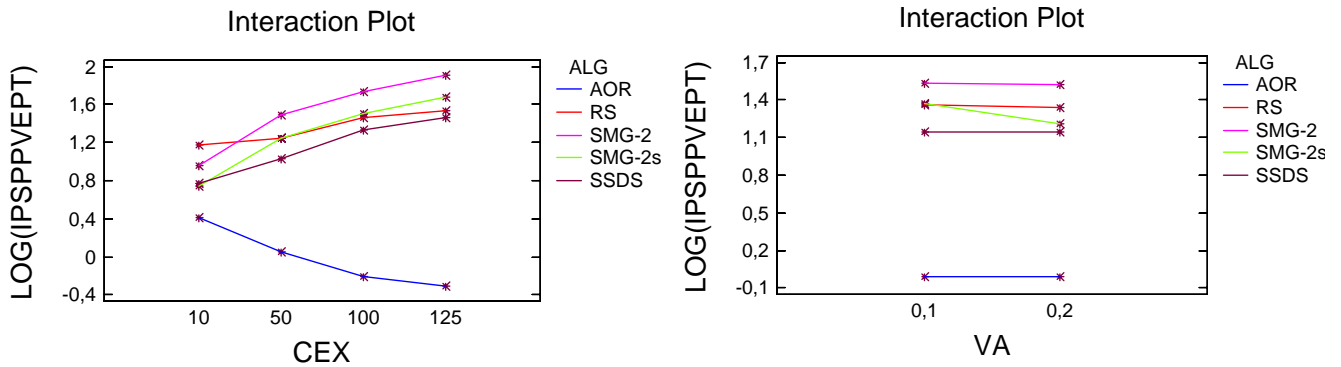


Figura 9-18. Gráfico del efecto de la interacción entre el factor CEX y el factor ALG (izquierda) y entre el factor K y ALG (derecha) del bloque Retraso-Predictiva SMAGA-2-P3_50 con K=0.

En el último bloque experimental, P3_50, la Figura 9-18 (izquierda) muestra como existe un dominio absoluto del método AOR para cualquier conjunto experimental, y que debido a su independencia de K se puede decir que esto será así para cualquier valor de K. En la parte derecha de la figura se observa como AOR continua siendo el mejor algoritmo con respecto a VA, y como SMAGA-2s mantiene la tendencia observada en el resto de bloques experimentales.

Bloque Experimental	Factores principales significativos	Factores Combinados significativos	Mejor Algoritmo
P13_20 (K=0 y K=0.01)	CEX, ALG	CEX ALG, VA ALG	SSDS10→ SMAGA-2s K=0 y SSD50 SSD100 SSD125→SSDS K>0 y SSD50 SSD100 SSD125→AOR
P13_50 (K=0 y K=0.005)	CEX, VA(sólo con K=0), ALG	CEX ALG, VA ALG	SSD10 y VA=0.2→SMAGA-2s SSD10 y K=0 y VA=0.1 →SSDS SSD10 y K>0 y VA=0.1 →RS SSD50→AOR SSD100 SSD125 y K=0 →SSDS SSD100 y K>0 →RS SSD125 y K>0→AOR
P3_20 (K=0 y K=0.01)	CEX, ALG	CEX ALG	SSD50 SSD100 SSD125→AOR SSD10→SMAGA-2s (excepto: K=0 y VA=0.4→SMAGA-2)
P3_50 (K=0)	CEX, VA, ALG	CEX ALG, VA ALG	AOR

Tabla 9-6. Resumen del análisis de los resultados del bloque experimental P13 y P3 para el retraso en la finalización de las operaciones.

En la Tabla 9-6 se proporciona un resumen de los factores significativos para cada bloque experimental, así como el criterio de selección del algoritmo que proporciona mejores resultados en cada ocasión.

A tenor de lo observado se pueden extraer varias conclusiones generales para el bloque de retrasos en las operaciones:

- Se observa como los métodos de Reprogramación parcial suelen ser los más adecuados, especialmente cuando el problema es más complejo (mayor número de trabajos combinado con mayor Tiempo de Cambio de Partida) y el número de máquinas de cada etapa está más equilibrado. En concreto AOR es un O de los métodos que mejores resultados suele dar en este tipo de problemas.
- Se ha detectado que en los problemas más sencillos (menor número de trabajos combinado con menores Tiempos de Cambio de Partida) y con el número de máquinas desigual entre etapas, los algoritmos de Reprogramación completa pueden dar buen resultado. Entre estos el algoritmo SMAGA-2s es el mejor.
- Se ha detectado un resultado inversamente proporcional del algoritmo SMAGA-2s respecto al valor de VA. Que indica, en general, la falta de adecuación de la filosofía de Reprogramación completa a este tipo de problemas cuando se parte de un Programa Predictivo altamente competitivo.

9.4.2.2.2. Casos con Programación Predictiva SMAGA-2 Modificada

Las consideraciones realizadas en el apartado 9.4.2.1.2 son válidas en este. Los bloques experimentales que se estudian son los mismos que en dicho apartado. Estos son: P13_20, P13_50, P3_20 y P3_50.

Como se puede observar, en la Tabla 9-7, tal y como ocurría en el caso de los adelantos en las operaciones, el criterio de selección de algoritmo presentado difiere completamente al del caso en el que el algoritmo predictivo es un Programa Productivo con makespan reducido. En este caso predominan los programas que fuerzan una Reprogramación completa, SMAGA-2 y SMAGA-2s, lo que implica que siempre es mejor Reprogramar de nuevo frente a cualquier otra opción.

Bloque	Mejor Algoritmo
Experimental	
P13_20	$K < 0.02 \rightarrow \text{SMAGA-2}$ $K = 0.02 \text{ y } \text{SSD125} \rightarrow \text{SMAGA-2}$ $K = 0.02 \text{ y } \text{SSD10} \text{SSD50} \rightarrow \text{SMAGA-2s}$ $K = 0.02 \text{ y } \text{SSD100} \text{ y } \text{VA} = 0.2 \rightarrow \text{SMAGA-2s}$ $K = 0.02 \text{ y } \text{SSD100} \text{ y } \text{VA} = 0.4 \rightarrow \text{SMAGA-2}$
P13_50	$K = 0 \rightarrow \text{SMAGA-2}$ $K = 0.01 \rightarrow \text{SMAGA-2s}$ $K = 0.005 \rightarrow \text{SMAGA-2s}$ (excepto: $\text{SSD100} \text{SSD125}$ y $\text{VA} = 0.2 \rightarrow \text{SMAGA2}$)
P3_20	$K < 0.02 \rightarrow \text{SMAGA-2}$ $K = 0.02 \text{ y } \text{SSD125} \rightarrow \text{SMAGA-2}$ $K = 0.02 \text{ y } \text{SSD10} \text{SSD50} \rightarrow \text{SMAGA-2s}$ $K = 0.02 \text{ y } \text{SSD100} \text{ y } \text{VA} = 0.2 \rightarrow \text{SMAGA-2s}$ $K = 0.02 \text{ y } \text{SSD100} \text{ y } \text{VA} = 0.4 \rightarrow \text{SMAGA-2}$
P3_50	$K = 0 \rightarrow \text{SMAGA-2}$ $K > 0 \rightarrow \text{SMAGA-2s}$

Tabla 9-7. Resumen del análisis de los resultados del bloque experimental P13 y P3 para el retraso en la finalización de las operaciones.

9.5 Conclusiones

En este capítulo se ha tratado el Problema de Programación de la Producción con un enfoque Reactivo en base a la definición del mismo realizada en los capítulo 5 y 7, y en el marco de un Proceso de Negocio Reactivo que tiene como objetivo dar soporte a la Programación de la Producción de empresas de la Industria Cerámica en un entorno de Incertidumbre. Se ha empleado la labor realizada en el capítulo 5 para referenciar el modelo matemático, y en el capítulo 7 para concretar el problema sobre el Taller de Flujo Híbrido con Tiempos de Cambio de Partida Dependientes de la Secuencia. Una vez establecido el marco de trabajo se han seleccionado algunos de los algoritmos existentes en la literatura que mejor encajan en el problema planteado, y se ha diseñado una nueva heurística (SSDS) que permita ofrecer una solución a dicho problema. Entre los algoritmos seleccionados se encuentran tanto propuestas de Reprogramación parcial como completa, entre ellas la propuesta realizada en el capítulo anterior SMAGA-2.

El diseño experimental ha seccionado los experimentos en **16 bloques experimentales independientes**. Estos se han dividido en base a tres criterios: **Configuración del problema** (P13_20, P13_50, P3_20 y P3_50), **tipo de Programación Predictiva** (makespan competitivo o makespan no competitivo), **tipo de evento** (adelanto o retraso en las operaciones). En cada

bloque el experimento se han comparado los algoritmos (**factor ALG**) que se han considerado atendiendo al posible efecto de diversos factores, algunos correspondientes a decisiones del programador como el peso de la penalización (**factor K**) o el tamaño de la Venta de Avance (**factor VA**), y otros relativos al Tiempo de Cambio de Partida (**CEX**) de cada configuración. **En todos, los casos se han contemplado eventos de diversa duración cuyos resultados se han promediado.**

El análisis comparativo es relevante y complejo, sin embargo a modo de resumen **se puede decir que la Reprogramación parcial predomina sobre la Programación completa cuando el Programa Predictivo es competitivo en relación a su makespan. La excepción a esta norma se encuentra** en los problemas con un número reducido de trabajos del conjunto P13 (P13_20 y a veces P13_50), ya sea con retraso o con adelanto en las operaciones, en cuyo caso los procedimientos de Reprogramación completa predominan. **El caso opuesto se da cuando el Programa Predictivo no parte con un buen valor de makespan.** Sin considerarlo un corolario, simplemente un resumen aproximado, se podía decir, que tanto cuando el Programa Productivo que se encuentra lejos del su mejor Programación, posiblemente por una modificación parcial continuada, o bien cuando el sistema productivo no presenta claros cuellos de botella es preferible volver a realizar los cálculos del Programa pendiente de ejecutar sin considerar el Programa en curso, y cuando el Programa a sufrido una pequeña variación respecto del original y el sistema productivo cuenta con algún cuello de botella en alguna etapa un pequeño ajuste es más adecuado.

Como conclusión final, y resumida del capítulo, **se puede considerar que los resultados obtenidos en la experimentación son relevantes, así como la heurística SSDS propuesta en este capítulo.** Y en general las deducciones obtenidas son de gran utilidad en la Programación Reactiva de la Producción para un Taller de Flujo Híbrido con Tiempos de Cambio de Partida Dependientes de la Secuencia, y pueden ser empleados como elementos activos de la plataforma IPSU-MAS.

Con este capítulo finaliza el diseño, a partir de los resultados comparativos obtenidos se ha preparado la plataforma IPSU-MAS para realizar el proceso de “Programación Reactiva de la Programación” mediante la integración de los algoritmos seleccionados, y su lógica de aplicación, en dicha plataforma. En el siguiente capítulo se procederá a realizar una prueba de funcionamiento de la propuesta en su conjunto cumplimentando así el último paso de la metodología propuesta en el capítulo 6.

9.6 Bibliografía

Abumaizar, R. J. & Svestka, J. A. 1997. *Rescheduling job shops under random disruptions. International Journal of Production Research*, 35(7): 2065-2082.

Conway, R. W., Maxwell, W. L., & Miller, L. W. 1967. *Theory of Scheduling*. Addison-Wesley Publishing Company.

Hasle, G. & Smith, S. F. 1994. *Directing an opportunistic scheduler: an empirical investigation of reactive scenario. Artificial intelligence in reactive scheduling: 1-11*. UK: Chapman and Hall.

Leon, V. J., Wu, S. D., & Storer, R. H. 1994. *Robustness Measures and Robust Scheduling for Job Shops. Iie Transactions*, 26(5): 32-43.

Sadeh, N. M., Otsuka, S., and Schnellbach, R. *Redictive and reactive scheduling with micro-boss scheduling and control system*. 1993. Centre for Integrated Manufacturing Decision Systems, Carnegie Melon University.

**Aplicación de la Propuesta de
Programación de la Producción
Predictivo/Reactiva a una Empresa
Cerámica**

Capítulo

10

Tabla de contenido

10.	Capítulo 10	641
10.1	Introducción	641
10.2	Descripción General de la Empresa Cerámica	642
10.2.1	La Empresa	642
10.2.2	Descripción de la Planta Seleccionada	643
10.2.3	Los Productos Fabricados.....	645
10.2.4	Procedimiento Actual para la Gestión del Programa Productivo	647
10.3	Configuración de la Plataforma IPSU-MAS para la Gestión del Programa Productivo en la Empresa Cerámica.....	651
10.3.1	Criterios para el Establecimiento del Programa Productivo Predictivo en la Plataforma IPSU-MAS.....	651
10.3.2	Política de Revisión del Programa Productivo en Curso en la Plataforma IPSU-MAS	654
10.3.2.1	Criterios aplicables a la Revisión del Conjunto de Trabajos en Curso.....	655
10.3.2.2	Simulación y Gestión de Eventos Internos y Externos	660
10.3.3	Programación en la Planta: Restricciones y M.O.	663
10.3.4	Aproximación a la Realidad: Generación de Trabajos.....	667
10.4	Análisis Experimental	672
10.4.1	El Juego de Datos	672
10.4.2	Decisión Pre-Experimental: Parametrización de SMAGA-1 y SMAGA-2	673
10.4.3	Decisión Pre-Experimental: Criterios de Aplicación de Algoritmo Reactivos....	676
10.4.4	Diseño del Experimento	677
10.4.5	Desarrollo del Experimento y Análisis de los Resultados.....	682

10.4.5.1	Análisis de los Resultados Sin Considerar Turnos de Trabajo	682
10.4.5.2	Análisis de los Resultados Considerando Turnos de Trabajo	689
10.5	Conclusiones.....	695
10.6	Bibliografía	696

Índice de figuras

Figura 10-1. Proceso Productivo (Giret-Boggino, A. S. 2005). 643

Figura 10-2. Representación del Sistema Productivo simplificado de la Empresas Seleccionada. 644

Figura 10-3. Diagrama Gantt correspondiente a dos Programas Productivos. 649

Figura 10-4. Control del proceso de Modificación del Programa (elaboración propia). 655

Figura 10-5. Representación del proceso de generación de Reprogramaciones. 661

Figura 10-6. Diagrama Gantt correspondiente a un taller con tres etapas con desplazamiento de la fecha de inicio de las operaciones entre ellas. 664

Figura 10-7. Diagrama Gantt de múltiples Programas correspondiente a un taller con tres etapas con desplazamiento de la fecha de inicio de las operaciones entre ellas. 665

Figura 10-8. Incremento de la Tasa de Productividad obtenida para los diferentes enfoques de Programación Predictiva-Reactiva sin considerar turnos de trabajo expresada como porcentaje de mejora con respecto a los valores obtenidos por la Empresa. 684

Figura 10-9. Número de Trabajos Cancelados obtenidos para los diferentes enfoques de Programación Predictiva-Reactiva sin considerar turnos de trabajo. 688

Figura 10-10. Incremento de la Tasa de Productividad obtenida para los diferentes enfoques de Programación Predictiva-Reactiva considerando turnos de trabajo expresada como porcentaje de mejora con respecto a los valores obtenidos por la Empresa. 691

Figura 10-11. Trabajos Cancelados obtenidos para los diferentes enfoques de Programación Predictiva-Reactiva considerando turnos de trabajo expresada como porcentaje de mejora con respecto a los valores obtenidos por la Empresa. 694

Índice de tablas

Tabla 10-1. Tipos de Producto en la fabricación de baldosas cerámicas.....	646
Tabla 10-2. Cuadro resumen de los criterios establecidos para la ampliación/reducción del conjunto de trabajos en curso.....	659
Tabla 10-3. Conjunto de baterías de eventos.	662
Tabla 10-4. Plan de la Empresa.	669
Tabla 10-5. Resumen de los conjunto experimentales empleadas en el diseño experimental.	673
Tabla 10-6. Niveles asignados a los parámetros ajustables para los algoritmo SMAGA-1 y SMAGA-2.	674
Tabla 10-7. Valores seleccionados para los parámetros de los diferentes conjuntos experimentales del algoritmo SMAGA-1.....	675
Tabla 10-8. Resumen de los valores asignados a los parámetros del algoritmo SMAGA-2 para todos los conjuntos experimentales.	675
Tabla 10-9. Resumen de los valores asignados a los parámetros del algoritmo SMAGA-1 y SMAGA-2 para cualquier conjunto experimental.	676
Tabla 10-10. Tasa de Productividad obtenida para los diferentes enfoques de Programación Predictiva-Reactiva sin considerar turnos de trabajo expresada en m ² /minuto.	682
Tabla 10-11. Incremento de la Tasa de Productividad obtenida para los diferentes enfoques de Programación Predictiva-Reactiva sin considerar turnos de trabajo expresada como porcentaje con respecto a los valores obtenidos por la Empresa.	683
Tabla 10-12. Rendimiento obtenido para los diferentes enfoques de Programación Predictiva-Reactiva sin considerar turnos de trabajo.....	685
Tabla 10-13. Incremento de la Tasa de Productividad del prevista con el Programa inicial obtenida para los diferentes enfoques de Programación Predictiva-Reactiva sin considerar turnos de trabajo expresada como porcentaje con respecto a los valores obtenidos por la Empresa.....	686
Tabla 10-14. Diferencia entre el incremento de la Tasa de Productividad alcanzado con el Programa finalmente ejecutado y el previsto al inicio del periodo sin considerar turnos de trabajo.	687
Tabla 10-15. Número de Trabajos Cancelados obtenidos para los diferentes enfoques de Programación Predictiva-Reactiva sin considerar turnos de trabajo.	687
Tabla 10-16. Tasa de Productividad obtenida para los diferentes enfoques de Programación Predictiva-Reactiva considerando turnos de trabajo expresada en m ² /minuto.....	689
Tabla 10-17. Incremento de la Tasa de Productividad obtenida para los diferentes enfoques de Programación Predictiva-Reactiva considerando turnos de trabajo expresada como porcentaje de mejora con respecto a los valores obtenidos por la Empresa.	690

Tabla 10-18. Rendimiento obtenido para los diferentes enfoques de Programación Predictiva-Reactiva sin considerar turnos de trabajo..... 692

Tabla 10-19. Incremento de la Tasa de Productividad del prevista con el Programa inicial obtenida para los diferentes enfoques de Programación Predictiva-Reactiva considerando turnos de trabajo expresada como porcentaje con respecto a los valores obtenidos por la Empresa. 692

Tabla 10-20. Diferencia entre el incremento de la Tasa de Productividad alcanzado con el Programa finalmente ejecutado y el previsto al inicio del periodo considerando turnos de trabajo. 693

Tabla 10-21. Número de Trabajos Cancelados obtenidos para los diferentes enfoques de Programación Predictiva-Reactiva considerando turnos de trabajo. 693

10.1 Introducción

En este capítulo se pondrá en práctica la propuesta descrita, y compilada en la plataforma IPSU-MAS, mediante la simulación de los resultados que se alcanzarían **empleando datos de una Empresa característica del sector cerámico español**. Esta prueba de funcionamiento forma parte del último paso de la metodología descrita en el capítulo 6. El objetivo que se persigue es **contrastar la validez de los planteamientos realizados**, verificando en primer lugar el **funcionamiento conjunto de los procesos** de Programación Predictivos y Reactivos, y **comparando en segundo lugar los resultados obtenidos** con respecto a los alcanzados con el procedimiento tradicionalmente empleados en dicha Empresa.

Como toda experiencia práctica que desea desarrollarse en un **entorno realista**. El trabajo que se realiza en este capítulo debe necesariamente dar un papel relevante a los elementos y decisiones directamente vinculadas a las aportaciones realizadas en capítulos anteriores, sin olvidar que existen otros elementos y decisiones, que soportan a los primeros y sin los cuales el sistema no funcionaría en su conjunto. Se debe tener en cuenta que la Programación de la Producción se sitúa en el ámbito decisional del corto o muy corto plazo, y que en este trabajo se pretende que las decisiones de este ámbito estén coordinadas, hasta cierto punto, con las decisiones que se tomen en el ámbito de la Planificación de la Producción con una visión a más largo plazo. Por este motivo, **para que la plataforma IPSU-MAS funcione correctamente, es necesario determinar aspectos como la parametrización de los algoritmos Predictivos y Reactivos, la política de revisión del Programa en curso, pero también otros como la política de selección de nuevos trabajos, o el número de turnos de las diferentes secciones**, etc. Todos estos aspectos se irán desgranando a lo largo de este capítulo.

Para llevar a cabo el trabajo previsto **es necesario adaptar los planteamientos generales a la situación concreta de la empresa seleccionada**. Para ello ha sido necesario alcanzar un amplio conocimiento de la estructura, proceso decisional táctico-operativo y configuración de la dicha empresa.

El capítulo se ha dividido en tres grandes bloques. En el apartado 10.2 se realiza una presentación y descripción de la Empresa sobre la cual se va realizar el contraste de la propuesta. En este mismo apartado se concretarán aspectos del problema que se desea abordar. En el apartado 10.3 se comentan aspectos de la implementación necesarios para hacer posible que la plataforma IPSU-MAS ofrezca soluciones adecuadas. En el apartado 10.4 se planteará el diseño de un experimento y su análisis para poder poner de manifiesto la

bondad de la propuesta. Por último se realizará una breve reflexión a modo de conclusión del capítulo.

10.2 Descripción General de la Empresa Cerámica

En este apartado se va a dar a conocer cual es la relevancia de la Empresa con la que se va a contrastar la propuesta que se ha presentado, así como su configuración productiva y de producto, y una aproximación a su manera de afrontar la gestión de la Programación de la Producción.

10.2.1 La Empresa

La Empresa seleccionada para la aplicación de la propuesta realizada en esta tesis, e implementada en la plataforma IPSU-MAS, es una de las compañías productoras de pavimento y revestimiento cerámico más importantes de España. Por razones de confidencialidad, su nombre no se explicitará, y algunos de los datos serán modificados con el objeto de no hacer públicas informaciones de gran valor estratégico para la misma. En todo caso, es importante destacar, no sólo su relevante posición en el ranking por volumen producido, sino también su larga y continuada apuesta por la innovación en todos los campos que le son propios. En este sentido, se desea subrayar el gran número de proyectos de investigación e innovación en los que la Empresa seleccionada ha participado junto a diversos miembros del Centro de Investigación de Gestión en Ingeniería de la Producción (CIGIP), así como colaboración continuada en la elaboración de proyectos fin de carrera y trabajos de investigación.

La Empresa seleccionada está presente en el mercado desde hace más de treinta años, y en este tiempo ha ido afianzando su posición como líder en productos de calidad. La Empresa seleccionada pertenece a un grupo empresarial que cuenta con más de quinientos trabajadores, y dispone de varias plantas productivas en España, todas ellas ubicadas en la provincia de Castellón.

Su posición en el mercado se basa en una amplia red de tiendas que ofrecen directamente sus productos a los clientes finales, lo que no excluye a su canal tradicional de venta a través de mayoristas, o venta directa a los grandes clientes. Con todo ello está presente en más de 150 países y cuenta con millones de clientes.

10.2.2 Descripción de la Planta Seleccionada

Para confirmar la validez de la propuesta presentada en los capítulos anteriores se ha elegido la planta de producción de pavimento. Algunos de los datos obtenidos son fruto del trabajo realizado dentro del proyecto interdisciplinar “Implementación y Aplicación de Sistemas Multi-Agente en Entornos Industriales con Restricciones Temporales” financiado por el Vicerrectorado de Investigación, Desarrollo e Innovación de la Universidad Politécnica de Valencia (UPV). Este proyecto es una colaboración del Grupo de Tecnología Informática e Inteligencia Artificial, GTI-IA y el Centro de Investigación Gestión e Ingeniería de la Producción, CIGIP, de la UPV.

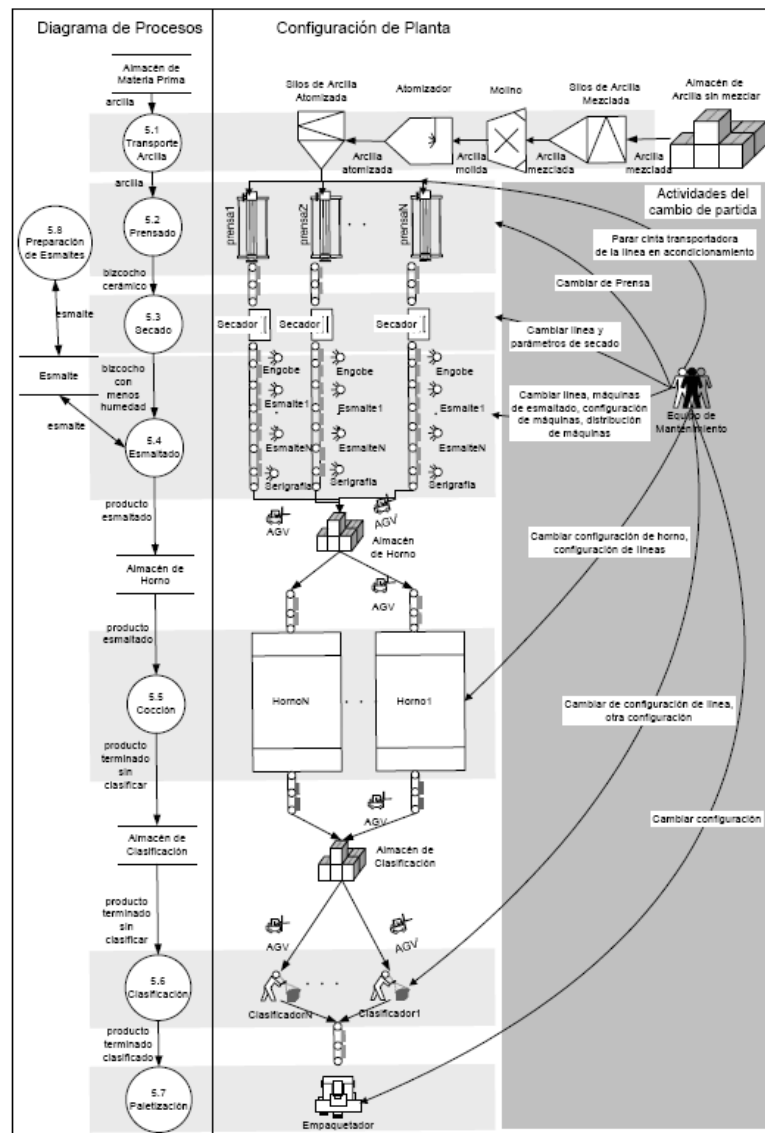


Figura 10-1. Proceso Productivo (Giret-Boggino, A. S. 2005).

En general, el proceso de producción de una baldosa cerámica es complejo, y fue descrito con detalle en el capítulo 2. A modo de recordatorio se han resumido las operaciones más relevantes que se realizan en la planta de pavimento en la Figura 10-1.

Dentro de las actividades que se realizan en la planta se ha considerado que son tres las operaciones que caracterizan al sistema productivo. Estas son las operaciones de prensado-Esmaltado, Horneado, y Clasificación-Embalaje. Estas son las tres operaciones más complejas que se realizan en el proceso productivo de una baldosa, y que en todo caso condicionan la programación del taller. El resto de operaciones se suelen hacer en un segundo plano, y muchas de ellas son realizadas por proveedores externos, como es el caso de la preparación de esmaltes.

En base a las tres operaciones mencionadas, la planta de pavimentos se puede considerar un Taller de Flujo Híbrido en el que los materiales, en diversos estados de terminación, circulan desde la línea de Prensado y Esmaltado, hasta la maquinaria de Clasificación y Embalado, pasando por el Horno. En este caso, se han considerado seis líneas de Prensado-Esmaltado que alimentan a tres Hornos, que a su vez alimentan a seis máquinas de Clasificación y Embalado. En la Figura 10-2 se ha representado la distribución del taller. En el momento de realizar este documento la Empresa disponía de algunos recursos adicionales en la misma planta empleados para fabricar productos de gran tamaño. La configuración seleccionada suponía en dicho instante aproximadamente el 80% de la capacidad productiva de la misma, y fue elegida porque, desde el punto de vista del programador, constituía un grupo homogéneo de recursos.

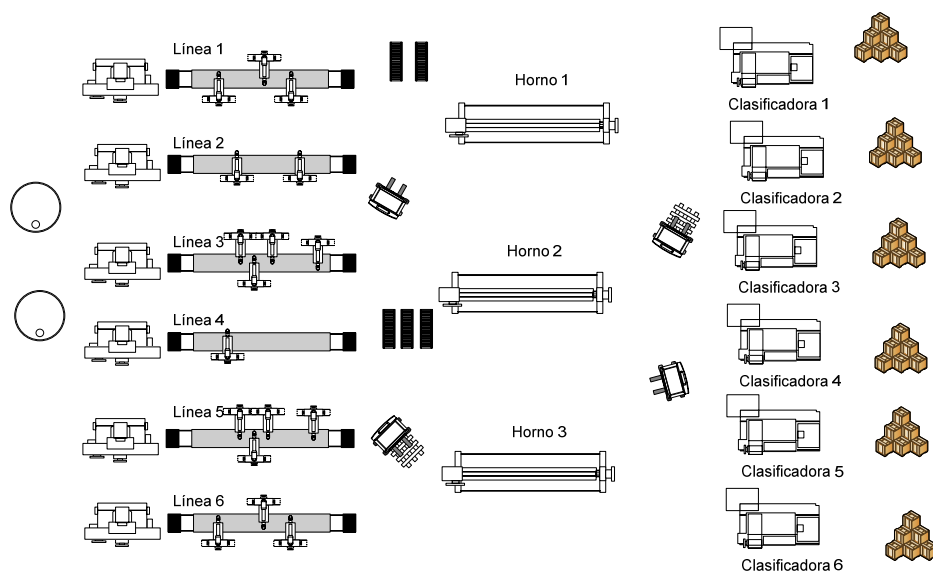


Figura 10-2. Representación del Sistema Productivo simplificado de la Empresas Seleccionada.

Se ha determinado que el proceso productivo esté compuesto por tres etapas que se corresponden con Prensado-Esmaltado, Hornos y Clasificación-Embalaje. El material que finaliza su operación en la primera etapa, o su segunda etapa, queda almacenado en estanterías móviles que son trasladadas entre el final de una etapa y el principio de otra mediante un sistema denominado "Automated Guided Vehicle" (AGV). En todo caso, la disponibilidad de elementos de almacenamiento y desplazamiento son ampliamente superados por las necesidades existentes, por lo que no se consideran una restricción.

Las capacidades de las etapas están equilibradas. *Las líneas de Prensado-Esmaltado trabajan entre un 50% y un 60% de su capacidad, y a dos turnos y cinco días a la semana.* El resto de tiempo se dedica a mantenimiento, y sobre todo a reajustar la maquinaria necesaria para adaptarla a las necesidades técnicas del lote siguiente. *Cada Horno es alimentado por el equivalente a dos líneas, y además trabajan a tres turnos, y siete días a la semana.* Por su parte la maquinaria de Clasificación y Embalaje trabaja a demanda, siendo en todo caso su capacidad muy superior a la demanda existente.

La fabricación de pavimento cerámico se caracteriza por incurrir en altos costes, temporales y económicos, debidos a los Cambios de Partida. Como ya se mencionó en el capítulo 2, y se recalcó en el capítulo 5, los cambios de partida en la Industria Cerámica se caracterizan por ser Dependientes de la Secuencia. Se recuerda que existen dos tipos de coste, uno que viene motivado por el cambio en el diseño del producto, y otro que se debe al cambio en las dimensiones del mismo. El coste cuyo origen es un cambio en el diseño tiene un impacto relativamente bajo en las líneas de Prensa-Esmaltado y muy escaso en el resto de etapas. Sin embargo, el impacto debido al cambio de formato (dimensiones) es muy elevado en la etapa de Prensas-Esmaltado y medio-bajo en el resto de etapas. Por este motivo, cuando se produce un cambio de lote que implica una modificación del formato, el recurso afectado puede estar parado durante un periodo de entre 4 y 8, o incluso más horas.

Según la disposición y forma de operar se puede decir que la planta de pavimentos es un Taller de Flujo Híbrido con Tiempos de Cambio de Partida Dependientes de la Secuencia. En concreto, con tres etapas, y 6-3-6 recursos (máquinas) por etapa.

10.2.3 Los Productos Fabricados

Las características que definen al producto final son las siguientes: Pasta, Formato, Esmalte, Calidad, Tono, Calibre.

Existen dos tipos básicos de pasta. La pasta blanca y la roja. En la Tabla 10-1 se muestra una clasificación de los tipos de producto existentes en la Empresa. Como se puede observar, en el caso de pavimento cerámico, existen dos tipos de producto comercializables, el porcelánico y el gres. La baldosa de porcelánico se realiza con pasta roja, y presenta una porosidad muy baja. La baldosa de gres se realiza con pasta blanca, y su porosidad es ligeramente superior a la anterior. En ambos casos el acabado final se realiza mediante esmaltado.

Tipo de Pasta	Porosidad	Uso	Tipo	Acabado Final
Pasta Blanca	Porosidad Alta	Revestimiento	-	Esmaltado
	Porosidad Media	Pavimento	NO EXISTE COMERCIALMENTE	
	Porosidad muy Baja	Pavimento	Porcelánico.	Esmaltado
Pasta Roja	Porosidad Alta	Revestimiento	-	Esmaltado
	Porosidad Media	Pavimento	Gres	Esmaltado
	Porosidad muy Baja	Pavimento	NO EXISTE COMERCIALMENTE	

Tabla 10-1. Tipos de Producto en la fabricación de baldosas cerámicas.

Desde el punto de vista comercial el producto se puede clasificar con respecto a su calidad, al menos, en dos categorías: Primera y Segunda. Existe un porcentaje, inherente al proceso productivo, de desperfectos que en general son clasificados como productos de segunda calidad en función de los defectos visuales y su frecuencia (roturas en las esquinas, grietas, manchas y pequeños cráteres, arañazos, etc.), y que no pueden ser vendidos como de primera calidad.

Debido también a la variabilidad del proceso productivo, sobre todo en la etapa de cocción en el Horno, existen piezas de un mismo lote productivo que siendo de primera calidad poseen tonos y calibres diferentes, lo que implica que una vez clasificados deben ser vendidos agrupados según tonos y calibres. A la hora de planificar la cantidad a producir de cada uno de los ítems se tiene en cuenta tanto el porcentaje de rechazos por no ser de primera calidad como las distribuciones de probabilidad por calibres y tonos.

Existen piezas especiales y determinados formatos que son comprados directamente a proveedores, por lo que no realizan la Planificación de Producción de los mismos. La Empresa seleccionada subcontrata a otros fabricantes la producción de aquellos productos de bajo valor añadido, principalmente revestimiento de pasta roja. En ningún caso, debido a políticas internas de la propia empresa, se subcontrataría la producción de productos de alto valor añadido, como son los porcelánicos (principalmente porcelánico de gran formato y esmaltados), o los revestimientos de pasta blanca de alto valor.

Para el contraste de la propuesta se han considerado un conjunto de productos representativos de la Empresas seleccionada, cuyos nombres han sido modificados. Un resumen de los productos se encuentra relacionado en el anexo V.

10.2.4 Procedimiento Actual para la Gestión del Programa Productivo

Después de varias reuniones con los diversos responsables de Planificación/Programación de la Producción se ha recopilado información suficiente como para tener una idea de conjunto de como se realiza el proceso de Programación de la Producción, y como se coordina este con el área de Planificación de la Producción.

La Planificación de la Producción se hace a dos meses vista, y en ella se especifica que formato y que cantidad (m^2) se va a producir en cada línea de cada referencia para *un periodo próximo al semanalⁱ*, que será el que se corresponda con el Programa Productivo. El Plan Maestro de Producción se “congela” para las tres semanas más próximas, lo que significa que la Planificación de tres semanas no se varía por restricciones productivas que hacen costosos los cambios de producción en tan corto plazo de tiempo. Aunque en la realidad este plan se modifica por requerimientos no previstos anteriormente, aunque nunca se modifica para la semana siguiente, si no como muy temprano a dos semanas vistas.

Se debe reseñar que se ha observado que existe una similitud muy alta en cuanto a la estructura de la información manejada entre el Plan Maestro y el Programa de Producción. Esto se debe a que ya durante la Planificación de la Producción se realiza una pre-asignación de productos a líneas. Como se indica en el capítulos 8 en los algoritmos propuestos, SMAGA-1

ⁱ En apartados posteriores se especificará un programa Productivo con lanzamiento semanal y duración bi-semanal.

o SMAGA-2, no se realiza ningún tipo de asignación previa y en el mismo proceso de cálculo se resuelve la asignación y la secuenciación. En este sentido el planificador realiza casi un Programa Productivo en el sólo es necesario por fechas de inicio y fin a las operaciones. Para ello el planificador ha tenido en cuenta que el tamaño de lote mínimo de producción es de 1.500 m² para todos los productos. No se puede fabricar por debajo de la cantidad indicada ya que desde el punto de vista de costes de producción no sería rentable.

El Programa Productivo con la pre-asignación de trabajos a máquinas es el documento de mayor detalle al que se ha tenido acceso. El establecimiento de la secuencia, y asignación de fechas es una tarea, se basa más en la experiencia que en métodos basados en técnicas propias de Programación de la Producción. Se intuye, que existen dos motivos que hacen que esto sea así, en primer lugar existe personal con amplia experiencia en dicho trabajo, y en segundo lugar en la actualidad existe un exceso de capacidad de la planta.

El Programa Productivo se puede decir que se basa en un procedimiento de asignación de los lotes a las líneas de Prensado-Esmaltado. Este procedimiento de asignación tiene como objetivo que cada línea contenga aquellos lotes que en conjunto minimizan los Tiempos de Cambio de Partida. Una vez asignados los trabajos a las líneas estos se ordenan intentando minimizar el Tiempo de Cambio de Partida de la secuencia. Como ya se ha comentado, cada línea alimenta sólo a un horno, pero a un horno le pueden llegar trabajos de varias líneas. En nuestro caso concreto los hornos reciben trabajos de 2 líneas. En los hornos se mezclan los trabajos atendiendo primero al trabajo que llega antes. En la etapa de clasificación, se suele aplicar la regla de atender al que llega antes, pero en todo caso es importante recordar que en esta etapa existe una capacidad superior a la que se necesita.

Un aspecto que debe resaltarse es que al existir un desplazamiento entre la fechas de inicio de la operación del primer trabajo de la secuencia entre la primera etapa y la segunda, y entre la segunda y la tercera, la Empresa realiza varios Programas de forma simultánea en la planta como se ve en la Figura 10-3. Es más, para esta aplicación se ha establecido un desplazamiento mínimo entre las diferentes etapas de tres días, que permite acumular un pequeño stock de producto intermedio que garantiza una ocupación más elevada de las máquinas que en el caso de no existir dicha restricción. Dicha restricción permite que el uso de los recursos se concentre más en el tiempo, balanceando así el tiempo de flujo del conjunto de los trabajos en las diferentes etapas.

Antes de la aprobación de un Programa Productivo se realiza una reunión entre los responsables de las operaciones en la planta que asumen el papel de programadores, y que conocen la situación real de los recursos, y los planificadores. En esta reunión se suele aprobar el Programa a nivel de pre-asignación de trabajos a recursos. A partir de ese momento se

considera que el Programa Productivo aprobado está o estará en marcha hasta la próxima reunión, es decir hasta la siguiente semana.

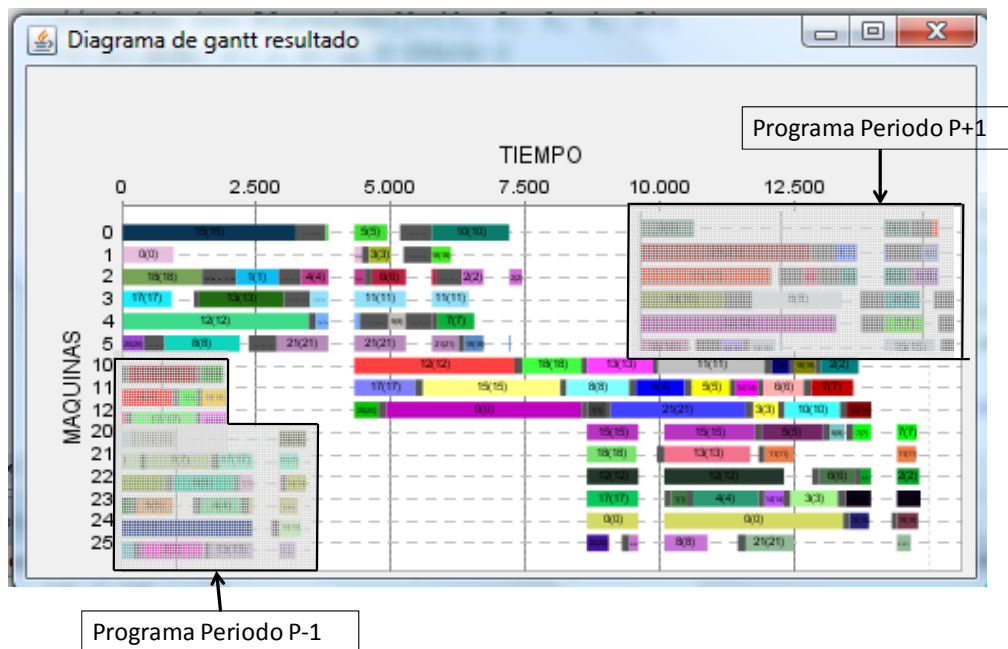


Figura 10-3. Diagrama Gantt correspondiente a dos Programas Productivos.

En caso de incidencias relevantes las soluciones se aportan siempre desde el lado en el que estas se producen. Por ejemplo, si existe la necesidad de introducir un trabajo urgente es Planificación quien cambia su plan más próximo sin que ello afecte nunca al Programa Productivo en curso. Si por el contrario, se trata de una avería es el responsable de la planta quien modifica el Programa en curso según su experiencia intentando cumplir los plazos iniciales. Es excepcional que se produzca una reunión para abordar un conflicto que afecte a un Programa en curso o próximo a estarlo, y eso es así básicamente por falta de procedimientos y herramientas orientados a dotar de Agilidad al Proceso de Toma de Decisiones.

Se puede decir, que la forma de actuar de la Empresa, relativas a los adelantos o retrasos en la fecha de fin de una operación, es similar a aplicar el algoritmo AOR comentado en el capítulo 9. Lo normal es que los responsables adelanten o retrasen el conjunto de operaciones afectadas dependiendo de cual sea el tipo de evento (adelanto/retraso), manteniendo en todo caso la asignación y secuencia preestablecida.

La Empresa seleccionada se ha planteado realizar un proceso de puesta en marcha de una nueva aplicación de Programación de la Producción coordinada con la Planificación. Los objetivos de este proyecto, ordenados de mayor a menor importancia, son:

- **Aumentar la fiabilidad del Plan.**
- **Mantener el Plan Actualizado.**
- **Desvincular la secuenciación de los jefes de fábrica.**
- **Secuenciar mejor a largo plazo.**
- **Avanzar los plazos de entrega.**

El proyecto tiene como objetivo resolver el problema de comunicación/coordinación actual entre Planificación y Programación de la Producción cambiando la situación actual en la que el plan pierde fiabilidad, ya que no tiene capacidad de respuesta adecuada ante los incidentes, para alcanzar una situación que permita su mantenimiento y aumente la Tasa de Productividad existente. Actualmente los fallos en la planta se detectan mediante partes informativos, y el objetivo es que se detecten en el Software de Gestión Integral de la Empresa mediante la captura de datos en planta, y otras facilidades. La Empresa se propone que con las mejoras la Programación de la Producción:

- Siga trabajando en periodos semanales pero mejore el número de trabajos entregados fuera de plazo.
- Facilite la respuesta a los clientes cuyos pedidos puedan suponer la incorporación de un trabajo urgente no planificado.
- Facilite la Reprogramación por caídas o problemas en los recursos de la planta.

Como se puede ver la línea de trabajo que la Empresa se ha propuesto encaja en muchos de sus objetivos con la propuesta realizada en esta tesis. En el análisis experimental que posteriormente se propondrá se podrán confrontar las dos formas de abordar el problema de Programación de la Producción, es decir, la que actualmente está empleando la Empresa y la que podría emplear si pusiera en marcha la propuesta presentada.

10.3 Configuración de la Plataforma IPSU-MAS para la Gestión del Programa Productivo en la Empresa Cerámica.

En este apartado se abordan todos aquellos aspectos necesarios para la implantación de la plataforma IPSU-MAS en la Empresa seleccionada. Como ya se ha comentado en la introducción, además de las aportaciones realizadas en esta tesis será necesario apoyarse en un conjunto de elementos y decisiones complementarios que permitan el funcionamiento de la plataforma IPSU-MAS en su conjunto. En este aspecto, en este apartado lo que se propone es dar un paso más en el diseño del sistema para adaptar la plataforma al caso concreto que se estudia.

El apartado se divide en dos bloques. El primer bloque está compuesto por el punto 10.3.1 y el punto 10.3.2 en los que se detallan aspectos que quedaron pendientes de abordar en la fase de diseño, y que tiene que ver con pequeños procesos o criterios para la Toma de Decisiones. El segundo bloque está compuesto por los puntos 10.3.3 y 10.3.4 en los que se abordan aspectos más relacionados con la configuración del sistema en relación con el entorno concreto que se afronta. En cualquier caso, ambos enfoques son necesarios para la puesta en marcha de la plataforma.

10.3.1 Criterios para el Establecimiento del Programa Productivo Predictivo en la Plataforma IPSU-MAS

A partir del trabajo realizado en la fase de análisis de requerimientos, del capítulo 7, y del trabajo desarrollado en la fase de diseño de la plataforma, en ese mismo capítulo, se detallarán aquellos aspectos vinculados con la implementación del sistema en un caso concreto. Para facilitar la comprensión de apartado se establecerán las oportunas relaciones entre los aspectos desarrollados en relación con la implementación, y los ya establecidos en las fases de análisis y diseño.

La plataforma IPSU-MAS, tal y como se describió en el capítulo 7, está compuesta por dos grandes bloques o Procesos de Negocio: El Proceso de Negocio de Programación Predictiva, y el Proceso de Negocio de Programación Reactiva. En la fase de diseño estos 2 procesos se asociaron a los flujos de trabajo “Establecer producción predictiva” y “Reajustar producción reactivamente”. En ambos casos es necesario realizar una revisión funcional y establecer cual es la configuración más adecuada de los mismos.

El Proceso de Negocio para la Programación Predictiva es un elemento cuyo funcionamiento se basa en la recepción de peticiones de Programación Predictiva, para lo cual se emplea la

interacción “obtener planificación periódica” identificada en el modelo de objetivos y tareasⁱⁱ y se desarrollado en el modelo de interacciones, y la entrega de Programas factibles a planta, que se basa en la interacción “reasignar producción” identificada y desarrollada en los mismo modelos. Pero, tal y como se definió en los capítulos 5 y 7, este no es un proceso pasivo sino activo, en el que se busca que los Programas generados provoquen una ocupación en el taller lo más próxima posible a la capacidad que se ha considerado inicialmente. Por este motivo, según el diseño realizado en el capítulo 7, el Proceso de Negocio no sólo realiza cálculos de Programas Productivos sino que propone cambios en el conjunto de trabajos seleccionados para el periodo productivo, si se observa una posibilidad de mejorar la Tasa de Productividad del taller, en base a la interacción “modificar planificación”. Para su correcto funcionamiento este Proceso de Negocio necesita disponer de:

- Información sobre el Margen Operativo establecido y Duración del Periodo Productivo junto con el calendario del mismo. En base a estos datos la Actividad Empresarial “EA-3 Ajuste” podrá calcular la capacidad productiva considerada y proponer cambios en el conjunto de trabajos seleccionados para el periodo mediante la mencionada interacción, “modificar planificación”, y excepcionalmente mediante la interacción “notificar inconsistencias”.
- Información sobre el conjunto de trabajos seleccionados para el periodo productivo considerado. Esta información se obtiene inicialmente mediante la interacción “obtener planificación periódica”, pero puede ser modificado en la actividad “EA-3 Ajuste” mediante las comentadas interacciones.
- Los algoritmos disponibles para ejecutar la Actividad Empresarial de “EA- 2 Cálculo del Programa Predictivo”, y obviamente que este o estos estén implementados y accesibles. Los algoritmos que se deben emplear son aquellos que se identificaron como los más eficientes en función de la configuración del sistema analizado (conjunto experimental) en el capítulo 8 de esta tesis. La aplicación IPSU-MAS dispondrá de los algoritmos SMAGA-1, SMAGA-2 y GAH implementados, así como la información relativa al conjunto de parámetros para su funcionamiento más eficiente en cada situación.

Además de la información que se ha indicado el Proceso de Negocio para la Programación Predictiva necesita disponer de un canal de comunicación para los Agentes que forman parte

ⁱⁱ El modelo de objetivos y tareas, así como el modelo de interacciones fueron definidos en el capítulo 6 durante la fase de diseño de la plataforma IPSU-MAS.

del modelo de Agentes, que permita iniciar las diferentes interacciones definidas durante el diseño de la plataforma en el capítulo 7.

Así mismo, como se ha visto la actividad Empresarial “EA-3 Ajuste” puede generar mediante un evento una notificación de solicitud de reducción del conjunto de trabajos inicial al Dominio de Planificación de la Producción, o bien puede poner en marcha la aplicación de un criterio para la ampliación de dicho conjunto. Ello supondría la puesta en marcha de la interacción “modificar planificación” o excepcionalmente la de “notificar inconsistencias”. Sin embargo, no sería posible que el Proceso de Negocio funcionase, si no se definiesen los criterios de reducción y ampliación del conjunto de trabajos inicial que se manejarán en dichos protocolos. Este nivel de detalle fue deliberadamente evitado en la fase de diseño, para simplificar la misma. Estos aspectos se definirán a continuación.

La reducción del número de trabajos del Programa inicial es debida a la inclusión de más trabajos de los que pueden realizarse con un Programa factible en base a la capacidad productiva considerada. Este hecho se puede dar en el instante inicial, cuando sólo se tienen en cuenta los trabajos propuestos por el planificador para el periodo analizado, o bien en una fase más avanzada del proceso de Programación Predictiva, cuando se han incluido algunos trabajos adicionales a los propuestos inicialmente por el planificador. En cualquier caso, como se sugería en el capítulo 5 y se reflejó en el capítulo 7, la decisión sobre el trabajo o trabajos a cancelar o a añadir debe ser del planificador.

Dado que esta tesis se ha centrado en las funciones del programador de la producción, y no se han desarrollado modelos de actuación para el planificador, se ha decidido implementar en la plataforma IPSU-MAS una decisión basada en el azar para ambas situaciones. Siempre que se detecte un exceso de trabajos en el proceso de Programación Predictiva se realizará una reducción de un trabajo elegido aleatoriamente. Y siempre que se detecte la necesidad de ampliar el número de trabajo se generará un nuevo trabajo aleatoriamente, con los mismos criterios que se generaron los trabajos iniciales. Estos criterios se aplicarán en cada caso hasta alcanzar el número de trabajos deseado. De esta forma, cuando el Agente programador ejecute la actividad empresarial “EA-3 Ajuste”, y detecte un desajuste y lance el protocolo correspondiente a la interacción “modificar planificación”, el Agente planificador contestará con un nuevo conjunto de trabajo a programar (el mismo que inicialmente se estableció menos un trabajo quitado aleatoriamente entre los que no han realizado operaciones de ningún tipo) en el acto “reasignar planificación” del protocolo, o con un nuevo conjunto de trabajos a programar (el mismo que inicialmente se estableció más un trabajo añadido aleatoriamente) en el acto “asignar trabajo nuevo” del protocolo.

Por su lado, el Proceso de Negocio para la Programación Reactiva de la Producción debe quedar cubierta la funcionalidad asociada a la Actividad Empresarial concebida para el “EA-9

Calculo del Programa Reactivo”. Esta actividad empleará los algoritmos propuestos en el capítulo 9 de esta tesis. La selección del algoritmo más adecuado para cada ocasión se basará en los factores analizados y los resultados obtenidos en dicho capítulo. No obstante, este Proceso de Negocio contiene una Actividad Empresarial para el “EA-6 Establecimiento de la Estrategia de Reprogramación” y otra Actividad Empresarial para el “EA-7 Ajuste/Control de la Estrategia de Reprogramación” que deben ser comentadas. El establecimiento y revisión de la programación en curso necesitan de la revisión detallada de algunos aspectos que se abordarán en el apartado 10.3.2.

Así mismo, en el apartado 10.3.3 se analizarán la existencia simultanea de varios Programas Productivos en la Empresa, y como esta cuestión puede afectar las restricciones de los algoritmos predictivos y reactivos, así como a la plataforma IPSU-MAS.

10.3.2 Política de Revisión del Programa Productivo en Curso en la Plataforma IPSU-MAS

La política de revisión del Programa en curso de la plataforma IPSU-MAS puede tener su origen en dos tipos de causas de naturaleza bien diferentes. La primera puede ser el propio interés del programador o planificador de aprovechar claras oportunidades de mejorar la Tasa de Productividad incorporando nuevos trabajos al Programa en curso. Se puede decir que se trata de un evento interno. Por el contrario, pueden existir eventos que se generan de forma ajena al programador, eventos externos, y que pueden implicar un reajuste del Programa en curso debido a un adelanto o retraso en la fecha de fin de una operación de un trabajo en curso. Ambos requieren de una revisión diferenciada.

En la fase de diseño ya se identificó un flujo de trabajo denominado “ajustar la producción reactivamente”, en el que se contemplan los dos casos. Si se trata de un evento interno se establece una interacción entre programador y planificador en la que se emplea un protocolo para solicitar una modificación del conjunto de trabajos. Si se trata de un evento externo primero se establece una interacción entre operario y programador denominada “notificar problema” donde el primero le indica al segundo que alguna operación no se ha ajustado a lo previsto, en cuyo caso se genera una segunda interacción, esta vez entre programador y planificador, denominada “notificar inconsistencias” donde se le avisa al éste último de la necesidad de modificar el plan. En los apartados 10.3.2.1 y 10.3.2.2 se detallan aspectos que no se incluyeron en la fase de análisis ni de diseño de la plataforma y que son necesarios especificar para su implementación.

10.3.2.1 Criterios aplicables a la Revisión del Conjunto de Trabajos en Curso

En caso de que exista un evento interno la actividad “EA- 5 Identificación y Valoración” se inicia como consecuencia de la interacción “notificar problema”, en caso de que sea un evento interno esa misma actividad será iniciada por un temporizador previamente programado. En cualquier caso, será en la actividad “EA-6 Establecimiento de la Estrategia de Reprogramación” donde se emplee la interacción “notificar inconsistencia” si se trata de un evento externo, o “modificar planificación” si se trata de un evento interno. La actividad “EA-7 Ajuste/Control de la Estrategia de Reprogramación” contienen la lógica que permite “Gestionar el Programa Productivo”, tal y como se estableció en el capítulo 7, y que será la que regule si se ha de volver a realizar la actividad “EA-6” de nuevo o no. Sólo al finalizar la actividad “EA-7” se inicia la interacción “reasignar producción” para comunicar el nuevo Programa en curso.

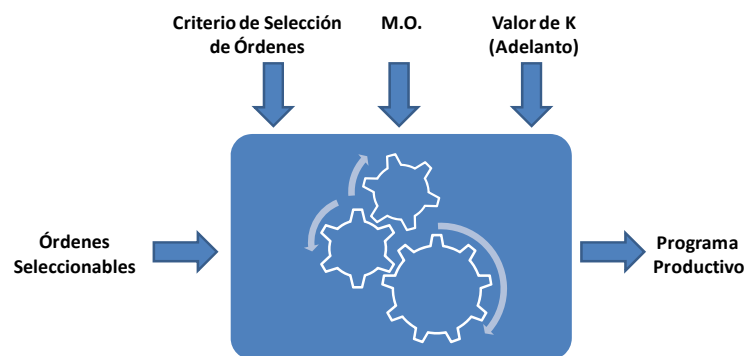


Figura 10-4. Control del proceso de Modificación del Programa (elaboración propia).

La estrategia asociada a la “Gestión del Programa Productivo” se traduce en gran medida, en la gestión de los parámetros Margen Operativo, el valor de K (asociado a la penalización por adelanto de trabajo dentro de la ventana de adelanto), y el criterio de selección de trabajos que deben ser programados. Dichos parámetros se establecerán en la EA-6 y se revisarán en la EA-7. En la Figura 10-4 se muestra una representación de los parámetros que son necesarios tener en cuenta en el proceso de modificación de un Programa Productivo según la propuesta que se realiza, y cuyos valores estarán determinados por la estrategia que se establezca.

Durante la ejecución del Programa en curso se puede interpretar que cuando se produce un evento se ejecuta el Proceso de Negocio relacionado con la Programación Reactiva que debe contar con información respecto de los trabajos que puede manejar. En función de las decisiones que hay que tomar en este Proceso de Negocio, los trabajos se pueden clasificar en:

- Trabajos en curso. Conjunto de trabajos que actualmente están programados para realizar sus operaciones en el taller.

- Trabajos que en algún momento pertenecieron al Programa pero que fueron cancelados por la imposibilidad de cumplir la fecha de entregaⁱⁱⁱ, DS().
- Trabajos que nunca han sido programados pero que están en la lista trabajos candidatos a ser incorporados si se añaden nuevos trabajos.

La Reprogramación puede conllevar la ampliación o reducción del conjunto de trabajos en curso. Si en el proceso de Reprogramación se considera necesario ampliar el conjunto de trabajos en curso los nuevos trabajos procederán siempre o bien del conjunto DS() o bien del conjunto de trabajos candidatos. Pero se debe tener en cuenta que existe una diferencia sustancial entre los trabajos pertenecientes al conjunto DS() y los trabajos candidatos. Los trabajos DS() fueron aceptados por el programador, inicialmente en el Programa Predictivo o por una modificación del Margen Operativo durante un proceso de Reprogramación, lo que supuso en su momento una valoración positiva del cumplimiento de las restricciones y la aceptación de *un compromiso de ejecución*. Mientras que los trabajos candidatos aún no han sido valorados, ni se ha alcanzado un compromiso para su ejecución. Por este motivo cuando se produce un evento y se valora en conjunto de trabajos no pertenecientes al Programa en curso se debe dar una prioridad absoluta a los pertenecientes al conjunto DS() frente a los candidatos por lo que se refiere a la posibilidad de incluirlos en un nuevo Programa. Este problema no se da en el proceso de selección de trabajos para la Programación Predictiva ya que en ese caso no existen trabajos cancelados.

En este trabajo se empleará el criterio que acepta que debido a que los trabajos pertenecientes al conjunto DS() ya fueron valorados y sobre todo a que existió un compromiso de realización su incorporación al Programa se realiza en cuanto sea posible y de forma automática. Esto significa que en todas las ocasiones en las que se inicie una Reprogramación, antes de finalizar el proceso, la plataforma IPSU-MAS intentará incorporar a todos los trabajos del conjunto DS() por orden inverso al de su cancelación antes de analizar la incorporación de cualquier trabajo candidato. Este proceso no requiere la modificación del Margen Operativo ya que sólo se puede cancelar un trabajo en curso cuando el Margen Operativo se ha reducido a cero como se dijo en el capítulo 7. No obstante, formalmente se requiere el inicio de la interacción “modificar planificación”, aunque el planificador actuará, tal y como, se ha comentado, es decir dando prioridad a los trabajos cancelados, siempre que no haya reasignado el trabajo a otra planta en cuyo caso ya se da por cumplido el compromiso.

ⁱⁱⁱ Según la nomenclatura del capítulo 5 el conjunto de trabajos que han sido cancelados se denominan DS().

Por el contrario, según el criterio que se desea establecer para la plataforma IPSU-MAS pero que podría ser diferente en otros planteamientos, el proceso de Reprogramación sólo considerará la posibilidad de reducir el conjunto de trabajos en curso si el Margen Operativo se ha reducido a cero, y aún así, ha sido imposible obtener un Programa factible. En caso de tener que cancelar algún trabajo, al igual que el Proceso de Negocio Predictivo, se debe realizar una notificación al Dominio del Planificador mediante la interacción “modificar programación” para que éste seleccione dicho trabajo. No obstante, de la misma forma que se planteó para el Proceso de Negocio Predictivo en la plataforma IPSU-MAS, la selección se realizará de forma aleatoria, ya que la interacción “modificar planificación” emplea el mismo protocolo que en el caso anterior.

Queda pendiente el establecimiento del criterio mediante el cual son seleccionados los trabajos candidatos, en caso de que una vez incorporados los trabajos cancelados existentes sea necesario añadir nuevos trabajos al Programa en curso. Esta decisión depende en gran medida del instante de tiempo en el que se valore. La incorporación de un trabajo está vinculada a la gestión que el programador realice del Margen Operativo. Todos los Programas Predictivos se realizarán en base a un valor de Margen Operativo inicial fijo y establecido por el programador. Según se determinó en el capítulo 7, el Margen Operativo se modificará durante la ejecución del Programa en curso debido a dos causas diferentes. La primera debido a la aparición de una perturbación que afecte a las fechas programadas de las operaciones. Esta causa es la que origina la necesidad de contar con dicho margen de tiempo. La segunda causa por la que se modifica el Margen Operativo es debido a una decisión del programador, cuando este observa una descompensación entre el riesgo existente de que se produzca una perturbación y el Margen Operativo disponible para atenderla. Esta segunda causa es la que puede dar lugar a la incorporación de trabajos candidatos al Programa en curso, si el programador reduce el Margen Operativo y transforma dicho tiempo en tiempo productivo. El proceso de valoración de riesgos puede ser sumamente sensible al tipo de empresa en el que se implante, e incluso a las circunstancias que se produzcan en cada periodo, y en todo caso, debe atender a dos cuestiones básicas:

- ¿Cuándo se puede incorporar un trabajo candidato?
- ¿Cuál es el trabajo candidato que se añade al Programa en curso?

Ambas cuestiones están relacionadas con dos Operaciones Funcionales que es necesario configurar en la plataforma IPSU-MAS. Establecer cuando se puede incorporar un trabajo candidato implica determinar la política de Reprogramación y la valoración del riesgo. La determinación de estas operaciones es en sí mismo un interesante campo de investigación. En todo caso, los criterios establecidos, ya que es necesario determinar alguno, se han basado en la experiencia obtenida en la observación del proceso. En cualquier caso, el análisis que se

planteado en este capítulo no está comprometido por esta decisión, que en todo caso se admite que puede ser mejorable.

La política de Reprogramación elegida es la de revisión periódica, lo que implica que la plataforma IPSU-MAS quedará programada para chequear la posibilidad de incluir un trabajo candidato cada cierto tiempo. Se ha fijado realizar dos revisiones. La primera revisión se realizará una vez finalizada la primera operación (Prensado-Esmaltado) para el 60% de los trabajos. La segunda se realizará cuando acabe la primera operación de todos los trabajos del Programa en curso. Esta decisión se justifica en base a que:

- La primera operación es una de las más sensibles a los eventos, ya que el Tiempo de Cambio de Partida es en esta etapa el más elevado, y cualquier cambio puede reducir la eficiencia del Programa en curso.
- Se ha perseguido evitar que el Programa esté sometido a un “estrés” adicional al que va a sufrir por causa de los eventos no deseados que impliquen adelantos o retrasos en las operaciones. Por este motivo se ha reducido a dos las revisiones.

En cuanto a la valoración del riesgo se ha considerado que en la primera revisión no existirá riesgo, y por lo tanto se autorizará la incorporación de nuevos trabajos, si el Margen Operativo en ese momento es igual o superior al 50% del inicial. En la segunda revisión se exige que el Margen Operativo existente sea superior al 10% del inicial para autorizar la incorporación de nuevos trabajos. Tanto el procedimiento de revisión como de valoración pueden influir en el número de trabajos autorizados, y por lo tanto en la Tasa de Productividad final. Esta decisión se ha tomado teniendo en cuenta que, al seguir los eventos una distribución uniforme, es lógico considerar que si quedan un 40% de trabajos por ejecutarse en la primera etapa se puede necesitar también un 40% del Margen Operativo. En cualquier caso se ha decidido que se debe dejar un 10% para el resto de Operaciones. El 10% asignado al resto de operaciones se ha planteado después de analizar el comportamiento real las operaciones de Horneado y Clasificación-Embalaje.

El proceso de revisión y el análisis de riesgo elegido se han mostrado eficaces en los test previos al experimento realizado. En todo caso, se insiste en su amplio margen de mejora.

Por lo que se refiere a la decisión sobre el trabajo candidato que se debe incorporar, esta está ligada a la política de selección de trabajos que será “ad-hoc”. Es obvio, que cada empresa establecerá un criterio diferente para conformar la lista de trabajos candidatos, y que éste estará más vinculado a decisiones “cualitativas” que “cuantitativas”. En este caso, dado que no existe un cliente real al que atender, el trabajo a incorporar se ha configurado seleccionando

aleatoriamente de la el tipo de producto y su tamaño. Lo único que se ha exigido es que el nuevo trabajo siga el patrón usual en a Empresa para evitar así anomalías.

Un resumen de los criterios aplicados en la plataforma IPSU-MAS para modificación del conjunto de trabajos en curso se muestra en la Tabla 10-2. **Por lo que se refiere a la incorporación de trabajos al Programa en curso se puede resumir diciendo que cada vez que se re programe se intentará reincorporar al Programa a los trabajos que han sido cancelado en algún momento (DS()), mientras que en momentos puntuales se planteará la incorporación de trabajos candidatos siempre que no exista ningún trabajo perteneciente a DS().** En ambos casos la condición de incorporación es que exista capacidad disponible para realizar el trabajo completo pero mientras que para los trabajos del conjunto DS() no se revisa el Margen Operativo para el caso de los trabajos candidatos se realiza una modificación previa del mismo.

Modificación del Conjunto de Trabajos en Curso	Criterio a Aplicar	Instante de Aplicación
Reducción	Sólo se produce sino queda Margen Operativo (MO=0) disponible y es imposible realizar un Programa factible.	Se aplica cuando se produce el evento, es decir de forma instantánea.
Ampliación mediante Trabajos Cancelados	Se produce si una vez realizada la Reprogramación se detecta que la capacidad productiva considerada permite rescatar algún trabajo cancelado.	Se aplica cuando se produce cualquier tipo de evento que implica realizar una Reprogramación.
Ampliación mediante Trabajos Candidatos	Se produce si una vez realizada la Reprogramación se detecta que la capacidad productiva considerada permite incluir algún trabajo adicional y todos los trabajos cancelados ya se han incorporado previamente.	Se aplica de forma programada, al final de ciertos periodos establecidos.

Tabla 10-2. Cuadro resumen de los criterios establecidos para la ampliación/reducción del conjunto de trabajos en curso.

Con esta información (K, Margen Operativo, Trabajos) el programador debe establecer el nuevo Programa Productivo teniendo en cuenta el criterio de selección de trabajos candidatos, el dato sobre Margen Operativo actual, y el valor de la penalización (K) por adelanto de trabajos.

Por lo que respecta a el valor de K se debe indicar que es un parámetro del sistema que puede influir en el algoritmo seleccionado para la Reprogramación, y cuyo valor será establecido por el programador en función de su experiencia, y la capacidad de reacción ante cambios inmediatos en el Programa, teniendo en cuenta tanto sus propias capacidades como la de los proveedores. Hay que recordar, que la relación entre valor de K y algoritmos a emplear fue establecida en el capítulo 9.

10.3.2.2 Simulación y Gestión de Eventos Internos y Externos

Como se ha descrito en capítulos anteriores, especialmente en los capítulos 5, 7 y 9, la Reprogramación de la Producción o Programación Reactiva se desencadena cuando se detecta un evento que, una vez analizado, es considerado susceptible de activar dicho proceso. Esta forma de actuar implica la existencia de un sistema de gestión de eventos^{iv} que actúa de acuerdo a una política ya definida ante una perturbación externa, en la que se establece el instante y el criterio de actuación. Se debe entender que una perturbación es externa cuando es ajena a la voluntad del gestor del Programa. En general, todos los eventos que se producen en el taller que implican un adelanto o retraso de una operación se deben considerar perturbación externa.

Además de las perturbaciones externas, el sistema de gestión de eventos debe ser capaz de canalizar las perturbaciones internas. Éstas últimas son las que se ocasionan al solicitar una revisión del Programa en curso con objeto de decidir posteriormente si es posible o no incorporar algún trabajo candidato al Programa en curso.

Tanto para las perturbaciones externas como internas es fundamental definir cual será la política de gestión del Programa en curso que condicionará el entorno sobre el cual intervendrá la EA correspondiente al cálculo del Programa Reactivo. En el caso de la plataforma IPSU-MAS, dicha política quedo definida en el apartado anterior, ver Tabla 10-2.

Debido a la necesidad de compatibilizar perturbaciones con dos respuestas diferentes en su aplicación temporal la plataforma IPSU-MAS trabajará con un sistema híbrido que permita considerar Reprogramaciones periódicas y Reprogramaciones guiadas por eventos. Las

^{iv} La funcionalidad relacionada con la gestión de los eventos se ubica en la EA-5 "Identificación y Valoración" del BP-Programación Reactiva.

primeras vinculadas a revisiones el Programa o perturbaciones internas, y las segundas a adelanto o retrasos de las operaciones o perturbaciones externas.

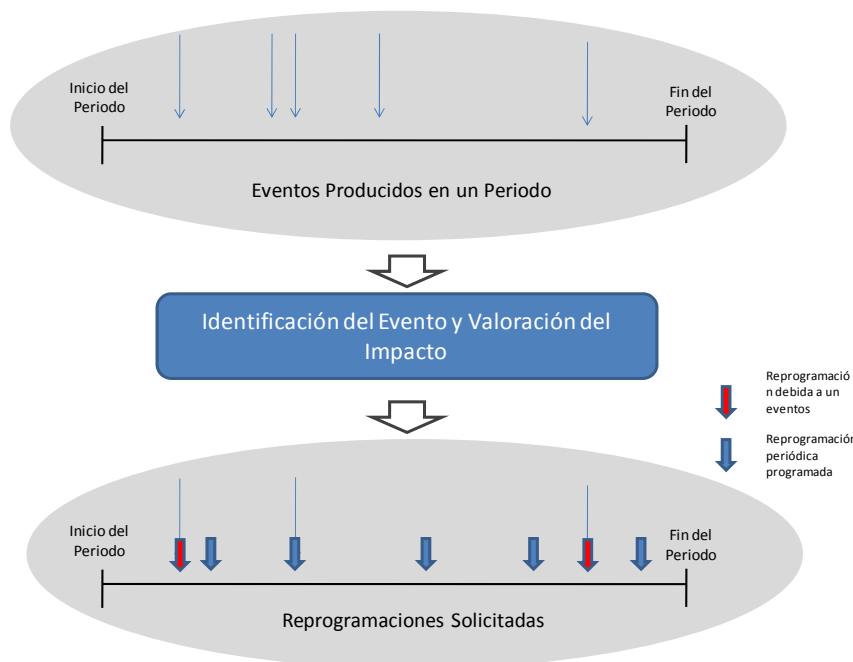


Figura 10-5. Representación del proceso de generación de Reprogramaciones.

En la Figura 10-5 se muestra como durante un periodo pueden surgir diversos tipos de eventos externos, como la “AE-5 Identificación y Valoración” analiza las perturbaciones externas recibidas, y como algunas de estas (rojas), junto con las perturbaciones internas programadas (azul), se convierten en solicitudes de Reprogramación. Se puede observar como en el caso de que un evento externo esté próximo a una revisión periódica prevista se pueden unificar los dos eventos, generando una única solicitud de Reprogramación.

Dado que la plataforma IPSU-MAS no trabajará directamente sobre un entorno real se debe simular la generación de eventos externos. Un evento quedará definido en IPSU-MAS por tres variables: tipo (adelanto/retraso), duración e instante en el que se produce. La generación de eventos consistirá, por lo tanto, en la generación de una batería de eventos repartidos en diversos instantes de tiempo a lo largo del periodo productivo, cada uno con una duración y un efecto de adelanto o retraso. No obstante, para que una batería de eventos adquiera la consideración de tal es necesario que cada evento o perturbación que vinculada a una operación que se vea afectada por éste.

Para generar una batería de eventos en primer lugar se calculará en número de eventos deseados. Posteriormente se asignará un instante de ocurrencia a cada evento. Para realizar esta asignación temporal se dividirá el Programa en dos sub-periodos. Un sub-periodo cubrirá el Programa desde su inicio hasta la fecha de fin del último trabajo en la primera operación. Y un segundo periodo de cubrirá el Programa Productivo desde la finalización del primer sub-periodo hasta el final del Programa. En cualquier instante de cada sub-periodo del Programa se insertará un evento mediante una distribución uniforme en dicho periodo. El 90% de los eventos se generarán en el primer sub-periodo, y el 10% en el segundo, reflejando el comportamiento del proceso de fabricación.

Una vez los eventos estén repartidos a lo largo del tiempo se determina de que tipo será cada uno de ellos, al mismo tiempo que su duración. Este último paso se llevará a cabo en base al tipo de combinación que se desee simular en cada momento. En este caso la plataforma IPSU-MAS permite la simulación de 21 combinaciones que se muestran en la Tabla 10-3.

	Desviaciones del 20%	Desviaciones del 50%	Desviaciones del 100%(80%)
Composición 100%_00%	BAT_MS20_100_00	BAT_MS50_100_00	BAT_MS100_100_00
Composición 80%_20%	BAT_MS20_80_20	BAT_MS50_80_20	BAT_MS100_80_20
Composición 60%_40%	BAT_MS20_60_40	BAT_MS50_60_40	BAT_MS100_60_40
Composición 50%_50%	BAT_MS20_50_50	BAT_MS50_50_50	BAT_MS100_50_50
Composición 40%_60%	BAT_MS20_40_60	BAT_MS50_40_60	BAT_MS100_40_60
Composición 20%_80%	BAT_MS20_20_80	BAT_MS50_20_80	BAT_MS100_20_80
Composición 00%_100%	BAT_MS20_00_100	BAT_MS50_00_100	BAT_MS100_00_100

Tabla 10-3. Conjunto de baterías de eventos.

Existen 3 posibles duraciones de un evento diferentes: una del 20%, otra del 50% y otra del 100% (80% para los adelantos) de la duración de la operación afectada. Y existen 7 posibles combinaciones de tipos de eventos en base a la combinación del porcentaje de eventos que producen adelanto o retaso en una operación, estos son: 100%_00% (todos los eventos son adelantos), 80%_20% (80% de los eventos son adelantos y el 20% son retrasos), 60%_40%, 50%_50%, 40%_60%, 20%_80%, 00%_100%.

La vinculación entre el evento y la operación afectada se realiza en el momento en el que el simulador dispara el evento. La forma de actuar consiste en seleccionar de forma aleatoria una operación, entre las que están previstas finalizar próximas a la fecha asignada al evento en cualquier máquina, simulando un retraso o adelanto de la fecha de finalización de la misma. En todo caso, siempre se mantendrá que el 90% de las veces la operación afectada será la de la primera etapa y el 10% en del resto de etapas.

En cuanto al número de eventos que contendrá una batería se ha pretendido este sea relativamente alto para acentuar el efecto del proceso de Reprogramación. Por este motivo, todas las configuraciones están formadas por un número de eventos igual al 40% del número de trabajos del Programa Predictivo.

Con la simulación de eventos propuesta, tanto en cuanto a su configuración como en cuanto a su variedad, se cubre un amplio espectro de situaciones próximas a lo que pudiera darse en una situación real dentro del entorno de la Industria Cerámica que permite evaluar las diferentes alternativas independientemente de la estrategia seguida para la "Gestión del Programa Productivo".

10.3.3 Programación en la Planta: Restricciones y M.O.

El enfoque que se realiza de la Programación de la Producción se basa en la existencia de un conjunto de trabajos iniciales que deben de realizarse en un determinado periodo. Aunque dicho conjunto de trabajos puede cambiar debido a la revisión de Programa en curso, siempre debe respetarse la restricción de que todos los trabajos deben estar terminados antes de una fecha previamente establecida D_p . La fecha D_p se corresponde con la fecha de fin del Programa P en curso. Esta restricción, ya introducida en el modelo matemático del capítulo 5, encaja perfectamente con la aplicación de la filosofía predictiva-reactiva. Lo que se expresó en los modelos matemáticos Predictivos y Reactivos del capítulo 5, de la misma forma que en [10-1].

$$[10-1] \quad D_p = d(i) \geq \max_{i=1}^{NX} \{c(i, RX)\}$$

Si un Programa P se inicia en el instante S_p , el programador dispone del periodo $D_p - S_p$ para realizar todas las operaciones de todos los trabajos. Como se puede ver en la Figura 10-6 el periodo $D_p - S_p$ se corresponde con el tiempo máximo de flujo que tiene disponible un Programa P.

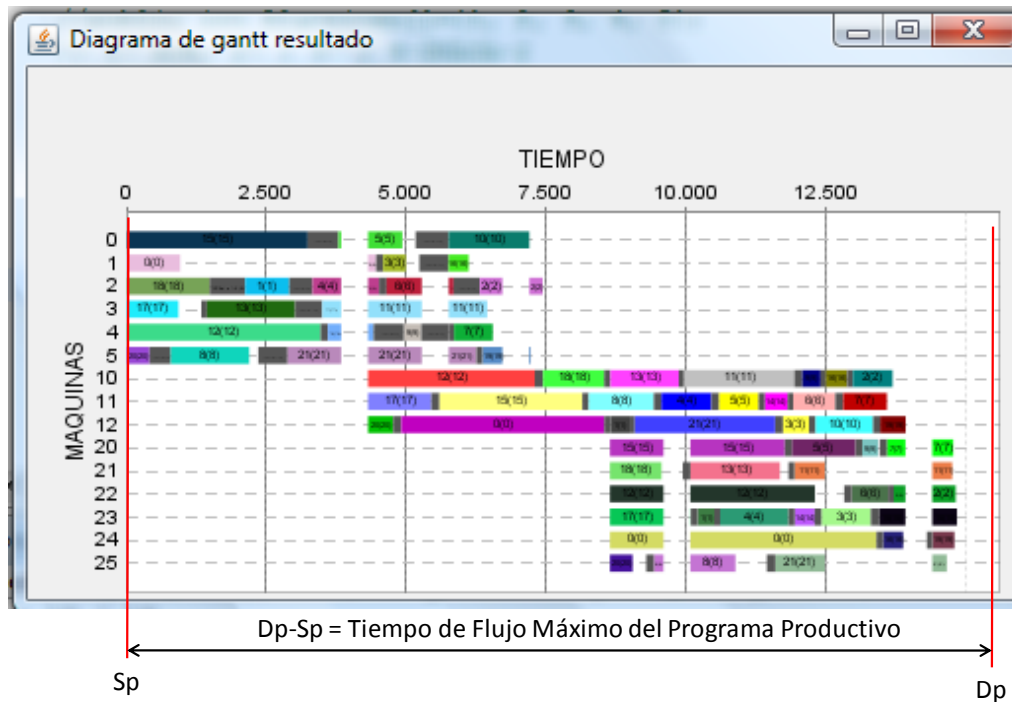


Figura 10-6. Diagrama Gantt correspondiente a un taller con tres etapas con desplazamiento de la fecha de inicio de las operaciones entre ellas.

La restricción que obliga a realizar un Programa en el que todos los trabajos terminen antes de la fecha D_p , potencia su efecto cuando a su vez se desea realizar múltiples Programas de forma simultánea, con el objeto de aprovechar los recursos disponibles y aumentar la Tasa de Productividad. Como ya se comentó la Empresa seleccionada mantiene Programas de múltiples periodos, normalmente semanales, en ejecución concurrente, aunque siempre en diferentes etapas.

Para poder comprender la necesidad de mantener múltiples Programas se muestra el diagrama Gantt de la Figura 10-6 en el que se puede observar un taller con tres etapas, y seis líneas de Prensado-Esmaltado en la primera etapa (0..5), tres Hornos (10..12) en la segunda etapa, y seis Clasificadoras-Embaladoras (20-25) en la última etapa. Como se puede ver la segunda operación no se inicia hasta pasado un tiempo del inicio de la primera (instante 4.320), y con la tercera operación ocurre lo mismo con respecto a la segunda. Esta situación provoca que los hornos estén ociosos hasta el instante 4.320, y que las clasificadoras estén

ociosas hasta el instante 8.460. Se verifica por lo tanto, como efectivamente, si sólo se considera un Programa Productivo se producirá el desajuste comentado.

Una empresa debería atacar esta situación para intentar que el tiempo ocioso fuese el mínimo posible, que es precisamente lo que se propone con los diferentes algoritmos presentados. No obstante, el lector debe de tener en cuenta que en el caso del Taller de Flujo, y en concreto cuando los tiempos de ocupación de la máquina (proceso y Cambio de Partida), y por lo tanto el desplazamiento entre las fechas de inicio y fin de las diversas operaciones es elevado, suele ser necesario que en un mismo instante de tiempo se estén ejecutando Programas diferentes sobre etapas diferentes. Como se ve en la Figura 10-7 se trata de considerar, que se puede dar el caso, de que mientras que en la etapa uno se ejecuta ya el Programa correspondiente al periodo P, en la etapa dos y/o tres se puede estar ejecutando a su vez el Programa del periodo P-1.

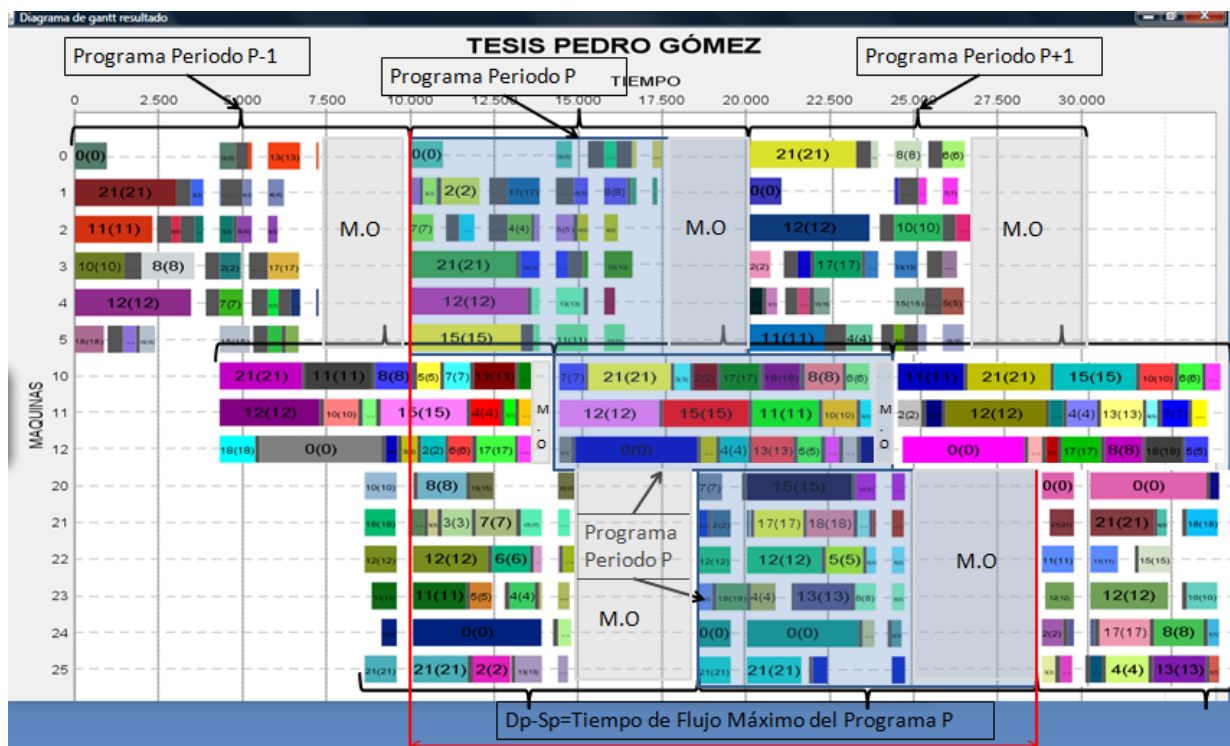


Figura 10-7. Diagrama Gantt de múltiples Programas correspondiente a un taller con tres etapas con desplazamiento de la fecha de inicio de las operaciones entre ellas.

La plataforma IPSU-MAS debe está preparada para soportar esta forma de trabajo y debe garantizar que no se producen solapamientos entre Programas Productivos de periodos contiguos en una misma etapa. Si los solapamientos se evitan exclusivamente no permitiendo que el Programa del periodo P se inicie en una etapa determinada hasta que se haya acabado

el Programa P-1 se podría provocar que el lanzamiento de Programas no fuese periódico debido a algún desajuste que condujera esperas excesivas entre Programas Productivos sucesivos. Sin embargo, además de respetar el tiempo de flujo máximo para cada Programa se debe de respetar la existencia de una periodicidad en el lanzamiento de los mismos. En la Empresa seleccionada la periodicidad en el lanzamiento suele ser semanal, aunque el tiempo de flujo máximo es aproximadamente de dos semanas debido al desacople entre etapas, que como ya se señaló suele ser de 3 días.

En definitiva, la plataforma IPSU-MAS se configurará para que los Programas se puedan lanzar cada semana, con una duración máxima de cada Programa de dos semanas, y con un retraso o desplazamiento mínimo entre etapas de 3 días. Para ello se exigirá que se cumpla una restricción adicional en el Programa Productivo, y es que el tiempo de flujo máximo de los trabajos de un Programa determinado en una etapa sea de una semana, es decir, igual al periodo de lanzamiento según refleja en [10-2]. Con esta restricción se garantiza que no existe solapamiento entre Programas y que se puede mantener la cadencia de lanzamiento. En la Figura 10-7 se muestran tres Programas Productivos (P-1, P y P+1) dos ellos tienen un tiempo de flujo máximo en cada etapa idéntico (indicado por una llave) que es igual a la cadencia de lanzamiento. Y además, como se señala en el Programa P, el tiempo de flujo máximo del Programa completo es $D_p - S_p$.

$$[10-2] \quad s(i, r) + T \geq \max_{j=1/i \neq j}^{NX} \{c(j, r)\} \quad \forall r_v$$

Por otro lado, en la Figura 10-7 se muestra como el tiempo de flujo máximo para todos los trabajos de una etapa (indicado con una llave) está compuesto por un periodo de trabajo y un Margen Operativo, que al menos inicialmente es mayor que 0. El periodo de trabajo está determinado por la fecha de finalización del trabajo que más tarde termina en dicha etapa (makespan de la etapa), y el Margen Operativo siempre será el tiempo restante. Por lo tanto, el programador deberá decantarse por aquel Programa Productivo que le permita disponer del Margen Operativo que crea conveniente en la etapa más crítica de cara a los riesgos relacionados con los eventos. En otras palabras, el Margen Operativo de la etapa crítica determinará el makespan máximo que puede alcanzarse en dicha etapa, y por ende, éste condicionará el Programa Productivo en el resto de etapas. En el caso que nos ocupa, ya se ha indicado, que la etapa crítica en relación a los posibles eventos inesperados en la primera,

^v La ecuación sigue la nomenclatura del capítulo 5. T es el periodo de cadencia de lanzamiento.

línea de Prensas-Esmaltado. De esta forma concreta se establecerá en IPSU-MAS para el caso de la Empresa seleccionada el Margen Operativo.

En general, el programador de la Empresa seleccionada es conocedor de la máxima capacidad de carga con la que cuenta, y casi de forma intuitiva es capaz de realizar la configuración del conjunto de trabajos respetando el Margen Operativo. En todo caso, durante la experimentación de trabajar con diversos grados de carga, lo que en este caso es equivalente a decir con diversos Márgenes Operativos.

10.3.4 Aproximación a la Realidad: Generación de Trabajos

Ya se ha comentado que existen ciertos datos que maneja el programador que deberían obtenerse mediante algún tipo de comunicación/coordinación con el planificador. Uno de los datos de mayor relevancia que debe ser proporcionado por el planificador es el conjunto de trabajos a programar. Sin embargo, como se ha visto que en la plataforma IPSU-MAS sólo se ha implementado el Dominio de Programación de la Producción, y que por lo tanto las decisiones que le corresponden al planificador se han resuelto aplicando reglas sencillas. En relación al conjunto de trabajos inicial, en base al cual se realizará la Programación Predictiva, se ha decidido crear un módulo que, aplicando lo observado en la Empresa seleccionada, realice la generación de un Plan de Fabricación^{vi} que determine dicho conjunto y que simule al real.

Aunque se dispone de varios ejemplos con datos reales que podrían ser empleados como Planes de Fabricación, y por lo tanto ser utilizados en sustitución del generador, se ha creído más conveniente abordar la implementación del mismo. Los motivos por los que se considera más adecuada esta segunda opción son:

- Aunque los Planes de Fabricación disponibles son reflejo de un periodo concreto (25 semanas) en los que se identifican las combinaciones de trabajo más habituales, sin embargo no constituyen un conjunto representativo de todas las combinaciones

^{vi} Según la nomenclatura establecida en el capítulo 5 el Plan de Fabricación contiene la información básica que el programador necesita para iniciar su actividad, es decir la elaboración del Programa de Producción, y que se obtiene después de someter al CRP al Plan Maestro de Producción.

posibles. Una simulación puede incorporar situaciones que no se han observado en la muestra disponible pero que con el azar pueden darse.

- Diseñar e Implementar el simulador permite profundizar en el conocimiento relativo a la configuración de un Plan de Fabricación en dicha Empresa.
- Si se dispone de un simulador se pueden generar cuantos planes se desee ahora, y en el futuro, permitiendo así volver a generar nuevos planes si cambiasen las condiciones actuales.

Dado que el simulador que se pretende implementar debe ser un reflejo de lo que se hace en la Empresa seleccionada para la obtención del Plan de Fabricación se han seguido los siguientes criterios:

- El Plan generado debe dar lugar a un Programa Productivo factible aplicando el procedimiento de Programación de la Producción de la Empresa seleccionada.
- La configuración de las referencias empleadas en cada Plan vendrá determinada por la asignación de estos a las líneas de Prensado-Esmaltado. En concreto se realizará una asignación de formatos a líneas de tal forma todos los lotes de un mismo formato se intenten realizar en la misma línea para minimizar los Tiempos de Cambio de Partida.
- Las líneas de Prensado-Esmaltado se agruparán de 2 en 2 unidades. A cada grupo se le intentará asignar productos similares entre sí en la medida de lo posible. Esta es la forma de operar observada en la Empresa.
- Las referencias incluidas en un Programa Productivo, dentro de un mismo formato, y los tamaños de lote de dichas referencia vendrán determinadas por las ocurrencias determinadas por los datos disponibles, obtenidos mediante observación en la Empresa seleccionada.

Guiados por los criterios generales recién expuestos se ha diseñado un simulador que produce Planes de Fabricación de la siguiente forma:

1. Se establecen 3 grupos de líneas formado cada uno por una línea impar y otra línea par. Se determina el número de formatos diferentes que se desea tener en cada grupo de líneas de Prensado-Esmaltado. Para todos los grupos será el mismo número. Así

mismo, se determina la capacidad máxima, medida en m^2 , de cada línea de Prensado-Esmaltado.

2. En cada grupo de forma independiente se asigna a la línea impar un número de formatos aleatorio que sea superior al 20% e inferior al 80% del número total de formatos asignados al grupo de líneas. Y a la línea par se le asignará el resto.
3. Para cada formato, de forma independiente, se establecerá aleatoriamente el número de m^2 que se realizará. Se respetará siempre que el mínimo asignable a cada formato sea $1.500 m^2$, y que la cantidad total de todos los formatos asignados a una línea no superen la capacidad de ésta.
4. De forma independiente cada formato se dividirá en trabajos o lotes diferenciados cada uno de los cuales se corresponderá con alguna referencia de las incluidas en el anexo V. La referencia se selecciona eligiendo de forma aleatoria un producto, respetando las probabilidades reales de demanda observadas, entre los disponibles en el formato correspondiente. El número de referencias dentro de un formato dependerá del tamaño de lote asignado a cada referencia. Cada referencia dispone de un conjunto de tamaños de lote que habitualmente se emplean en dicho producto, así como la probabilidad de que dicho tamaño se emplee. Por lo tanto, una vez conocida la referencia se realiza un sorteo entre los posibles tamaños de lote. El tamaño de lote asignado a un trabajo debe ser como mínimo de $1.500 m^2$, y además la suma de todos los tamaños de lote de todas las referencias del mismo formato no pueden superar el número de m^2 asignado al formato correspondiente.

Como resultado de este proceso se obtiene un conjunto de referencias agrupado en formatos de manera que cada formato, y en consecuencia cada referencia, se encuentra asignada a una línea de Prensado-Esmaltado. Esta relación línea-trabajos es la que proporciona el planificador al programador de la Empresa seleccionada según se comentó en el apartado 10.2.4. Por este motivo dicha relación, de la cual se muestra un ejemplo en la Tabla 10-4, se denomina “Plan de la Empresa”.

Línea 0: 0 1
Línea 1: 2 3 4 5 6 7
Línea 2: 8 9 10 11
Línea 3: 12 13 14
Línea 4: 15 16 17
Línea 5: 18 19 20 21

Tabla 10-4. Plan de la Empresa.

Al mismo tiempo que se genera un determinado “Plan de la Empresa” se genera un fichero en el que se contiene la configuración del taller (etapas y máquinas por etapa), el tamaño de lote de todos los trabajos generados, tiempos de proceso de cada trabajo en cada máquina, y los Tiempos de Cambio de Partida de todos los trabajos en todas las máquinas. El formato de los ficheros de configuración es muy similar al empleado en el capítulo 8. Se emplea una estructura en la cual, para poder considerar todas las restricciones, en la primera línea aparecen tres números, el primero representa el número de trabajos (n), el segundo el número de máquinas totales (m) y el tercero el número de etapas (e). En la segunda línea se ha indicado el número de máquinas no relacionadas que existen por etapa, que en este caso son siempre 6 en la etapa 1, 3 en la etapa 2, y 6 en la etapa 3. En la tercera línea se indica el tamaño de lote de cada trabajo. Tras estas tres líneas aparece una matriz con los tiempos de proceso por lote que tiene tamaño $n \times m$, es decir, se especifican todos los tiempos de proceso para todas las máquinas, no por etapas. Al terminar estas matrices aparecen m matrices adicionales de tamaño $n \times n$, que son las matrices de los Tiempos de Cambio de Partida para cada una de las m máquinas. Todos los tiempos de proceso y de Cambio de Partida se expresan en minutos.

Aunque en este capítulo se pretende realizar una aplicación de la propuesta realizada a una Empresa concreta, se ha considerado oportuno conciliar la visión general con la particular, y en la medida de lo posible generar valores para los tiempos de proceso y de Cambio de Partida que sean asumibles no sólo por la Empresa Seleccionada sino por un amplio espectro de empresas del sector.

Para la generación de tiempos de proceso se ha considerado que un promedio adecuado para una línea de Prensado-Esmaltado es que se tarde entre 0.21 minutos y 0.27 minutos en producir un m^2 . Y tomando como referencia este dato se ha procedido de la siguiente forma para generar los tiempos de proceso:

1. A la primera de las 6 líneas de esmaltado de que consta el taller se le ha asignado un tiempo de proceso unitario (TPU_1) dentro del rango de referencia, es decir entre 0.21 y 0.27 minutos. La asignación se ha realizado mediante una distribución uniforme.
2. A cada una de las 5 restantes líneas del taller se le ha asignado aleatoriamente, mediante una distribución uniforme, un tiempo de proceso dentro del rango que oscila entre el 90% y el 110% del valor de la primera línea ($TPU_1 * [0.9..1.1]$).
3. A cada uno de los 3 Hornos del taller se le ha asignado aleatoriamente, mediante una distribución uniforme, un tiempo de proceso dentro del rango que oscila entre el 105% y el 115% del valor de la primera línea ($TPU_1 * [1.05..1.15]$).

4. A cada una de las 6 máquinas de Clasificación-Embalado del taller se le ha asignado aleatoriamente, mediante una distribución uniforme, un tiempo de proceso dentro del rango que oscila entre el 95% y el 105% del valor de la primera línea ($TPU_1 * [0.95..1.05]$).

Como se puede observar los valores de todos los recursos fluctúan tomando como referencia el valor de la primera línea de Prensado-Esmaltado, con la caracterización de que el resto de líneas presentan una diferencia máxima del +/- 10%, las máquinas de Clasificación-Embalado pueden llegar a alcanzar una diferencia de +/- 5%, y los Hornos superan dicho valor con una diferencia que varía entre el 5% y el 15%. Este procedimiento de obtención de tiempos unitarios se realiza tantas veces como trabajos existen, lo que permite obtener la matriz de tiempos de proceso para todos los trabajos en todas las máquinas. El paso final consiste en multiplicar el tiempo unitario por el tamaño de lote, para obtener el tiempo de proceso por lote, que es el que se refleja en el fichero de configuración.

De la misma forma que se han obtenido los tiempos de proceso se ha establecido un criterio para generar los Tiempos de Cambio de Partida dependientes de la secuencia. Una vez más, a partir de la observación, se tomo como referencia que el Tiempo de Cambio de Partida entre referencias del mismo formato en las líneas de Prensado-Esmaltado debía de oscilar entre 90 y 150 minutos, mientras que si se trataba de referencias de formato diferente debía de oscilar entre los 360 y los 660 minutos. Así mismo, los Tiempos de Cambio de Partida entre cualquier referencia en los Hornos y en las máquinas de Clasificación-Embalaje debían de oscilar entre los 90 y los 150 minutos. Con estos valores de referencia se han generado para cada máquina y para cada pareja de referencias los Tiempos de Cambio de Partida seleccionando de mediante una distribución uniforme valores dentro de los rangos indicados.

Con los procedimientos y criterios establecidos se ha implementado un módulo capaz de proporcionar el fichero con el "Plan de la Empresa" y el fichero de configuración correspondiente, tomando como entradas la capacidad máxima de las líneas y del número de formatos diferentes por cada grupo líneas. El fichero con el "Plan de la Empresa" sólo se emplea si se desea realizar un Programa Productivo siguiendo el método que utiliza la Empresa seleccionada, y explicado en el apartado 10.2.4.

El procedimiento seguido por la Empresa ha sido implementado y se ha empleado para calcular el Makespan que obtendría la Empresa. De esta forma para cualquier configuración que se genere siempre se podrá comparar el makespan del Programa Predictivo obtenido por la Empresa y el Makespan obtenido por la plataforma IPSU-MAS.

10.4 Análisis Experimental

Una vez presentada la Empresa seleccionada, sus características productivas, y su forma de abordar la Planificación/Programación de la Producción, así como definidos los procedimientos y los criterios que se emplearán en la plataforma IPSU-MAS para posibilitar el funcionamiento conjunto de los procesos de Programación Predictiva y de Programación Reactiva, tal y como se han propuesto en los capítulos anteriores, se desea plantear un análisis experimental que permita poner de manifiesto la viabilidad de la propuesta y su eficiencia en comparación con la empleada en la actualidad por la Empresa seleccionada.

Se considerará que la plataforma IPSU-MAS es viable si se consigue abordar con ella los problemas reales que se plantean en la Empresa seleccionada, y se obtienen resultados factibles. La eficiencia se medirá, tal y como ya se planteó en el capítulo 5, en base a la Tasa de Productividad alcanzada con los diferentes Programas Productivos obtenidos. Y en este caso, se propone realizar una comparativa entre la Tasa de Productividad obtenida con los Programas generados con la plataforma IPSU-MAS y los obtenidos con el método empleado por la Empresas tanto para la Programación Predictiva, como para la Programación Reactiva.

10.4.1 El Juego de Datos

Para confirmar la validez y eficacia de la propuesta, implementada en la plataforma IPSU-MAS, será necesario analizar su comportamiento en diversos escenarios. En este caso la configuración del taller ya está predeterminada, y no varía. Sin embargo existen otros elementos que al modificarlos configurarían escenarios diferentes. Se ha identificado que estos elementos son la combinación de trabajos del Plan de Fabricación empleado y el Margen Operativo establecido.

Un Plan de Fabricación queda determinado, a su vez, como combinación de diversas variables como son el número de formatos diferentes, el número de trabajos, el tamaño de lote de los mismos, y los tiempos de proceso y de Cambio de Partida establecidos. Por este motivo, se ha decidido crear un amplio juego de datos, que constituya una muestra representativa de los diferentes Planes de Fabricación que puede ser empleados en la Empresa seleccionada, y en general en una empresa Cerámica. En este sentido se actúa siguiendo la misma estrategia que en las experimentaciones de los capítulos 8 y 9, en los que se disponía de un amplio juego de datos que permitieron analizar diferentes casos.

	3 formatos	4 formatos	5 formatos	6 formatos
MO005 (5% M.O.)	MO_005_001	MO_005_011	MO_005_021	MO_005_031
	hasta MO_005_010	hasta MO_005_020	hasta MO_005_030	hasta MO_005_040
MO010 (10% M.O.)	MO_010_001	MO_010_011	MO_010_021	MO_010_031
	hasta MO_010_010	hasta MO_010_020	hasta MO_010_030	hasta MO_010_040
MO015 (15% M.O.)	MO_015_001	MO_015_011	MO_015_021	MO_015_031
	hasta MO_015_010	hasta MO_015_020	hasta MO_015_030	hasta MO_015_040
MO020 (20% M.O.)	MO_020_001	MO_020_011	MO_020_021	MO_020_031
	hasta MO_020_010	hasta MO_020_020	hasta MO_020_030	hasta MO_020_040

Tabla 10-5. Resumen de los conjunto experimentales empleadas en el diseño experimental.

En la Tabla 10-5 se muestran los nombres de los 160 conjuntos experimentales propuestos para realizar el análisis experimental. Como se ve éstos son fruto de la combinación de 4 posibles valores para el número de formatos diferentes por grupo de líneas (3, 4, 5 ó 6), y de 4 posibles valores para el Margen Operativo (5%, 10%, 15% ó 20% del periodo). Para cada combinación, cada celda de la tabla, se han generado 10 ficheros diferentes, lo que representa 10 combinaciones de Planes de Fabricación diferentes, aunque todos ellos con el mismo número de formatos diferentes y para el mismo Margen Operativo. Para seguir la misma nomenclatura que en el capítulo 8 denominaremos a este conjunto de 10 configuraciones “conjunto experimental”. Hay que recordar que los Tiempos de Cambio de Partida son superiores entre referencias de formatos diferentes que entre referencias del mismo formato, y por eso es relevante definir el número de formatos diferentes por línea o grupo de líneas.

Con el juego de datos obtenido se podrá analizar el comportamiento de la plataforma IPSU-MAS en función de los diferentes parámetros o criterios que se establezcan en el diseño del experimento. Este juego de datos empleado se adjunta a este trabajo en formato digital.

10.4.2 Decisión Pre-Experimental: Parametrización de SMAGA-1 y SMAGA-2

Los algoritmos SMAGA-1 y SMAGA-2 son dos métodos relevantes en el Proceso de Negocio de Programación Predictiva, y en el caso de SMAGA-2 también en el Proceso de Negocio de Programación Reactiva. En el capítulo 8 se realizó la parametrización de los mismos para el juego de datos disponible en ese caso. En este caso se dispone de un juego de datos diferente, y aunque inicialmente se intentó identificar a que juego de datos de los ya parametrizados se ajustaba mejor el juego actual, y evitar así repetir el proceso de ajuste de parámetros, los

datos obtenidos en experimentos preliminares no fueron concluyentes. Por este motivo se ha decidido volver a calcular el valor de los parámetros para los dos algoritmos. Para hacerlo se ha seguido el mismo criterio que en el capítulo 8. En primer lugar se han establecido los valores de los parámetros FmG, FdM y GsM, que determinan el funcionamiento del algoritmo SMAGA-1, y posteriormente, ya con dichos valores identificados, se ha establecido el valor del parámetro RM, que junto con los anteriores determina el funcionamiento del algoritmo SMAGA-2. En la Tabla 10-6 se muestran los posibles valores para los diferentes parámetros ajustables.

Valores de los Parámetros Ajustables	SMAGA-1	SMAGA-2
Generador Dinámico	Desactivado	Activado
Evolución del Factor Límite	Desactivado	Activado
Evolución de Riesgo Mínimo	Desactivado	Activado
Riesgo Mínimo (RM)	-	0.2, 0.4
Generaciones sin Mejora (GsM)	50, 200	SMAGA-1
Factor de Mutación (FdM)	0.1, 0.3	SMAGA-1
Factor Margen (FmG)	1.05, 1.2	SMAGA-1

Tabla 10-6. Niveles asignados a los parámetros ajustables para los algoritmo SMAGA-1 y SMAGA-2.

Tanto para el algoritmo SMAGA-1, como para el algoritmo SMAGA-2, cada conjunto experimental consta de 10 ejemplares. También en ambos casos, se ha decidido realizar una repetición de cada ejecución, lo que supone un total de 2 ejecuciones por ejemplar y combinación de niveles de los factores. Esto implica que para el caso de SMAGA-1 se realizarán 16 (conjuntos experimentales) x 10 (ejemplares) x 8 (tratamientos) x 2 (ejecuciones)= 2.560 ejecuciones. En el caso de SMAGA-2 se realizan 16 (conjuntos experimentales) x 10 (ejemplares) x 2 (tratamientos) x 2 (ejecuciones)= 640 ejecuciones.

De forma similar a como se planteó en el capítulo 8 la variable de respuesta que se utilizará será una medida comparativa respecto el Incremento Porcentual Sobre el Valor de Referencia del Valor Examinado Promedio (IPSVRVEP) de un problema o ejemplar dado. Esta medida se puede expresar como:

[10-3]

$$IPSVRVEP = \frac{Sol_referencia - Mejor_sol}{Mejor_sol} * 100$$

La variable “Actual_sol” representa al valor del makespan obtenido con el algoritmo que se desea evaluar con un ejemplar determinado. La variable “Sol_referencia” representa al valor del makespan obtenido empleando el método de la Empresa seleccionada para dicho ejemplar. Los valores de IPSVRVEP negativos implican que se ha mejorado la referencia. Atendiendo a una subdivisión realiza en conjunto experimentales se ha introducido la variable **IPSVRVEPT** tendrá en cuenta el IPSVRVEP promedio para un conjunto experimental completo.

Por lo que se refiere al número de iteraciones que se ejecutará el algoritmo se ha decidido mantener el valor empleado en el capítulo 8, es decir 5.000 iteraciones.

En las condiciones mencionadas, y una vez realizado en análisis estadístico basado en el ANOVA de los resultados que se puede ver en el anexo VI, la significación y los valores más adecuados de de los parámetros GsM, FdM y FmG se han resumido en la Tabla 10-7.

Parámetros SMAGA-1	Factores Significativos	Generaciones sin Mejora (GsM)	Factor de Mutación (FdM)	Factor Margen (FmG)
MO_005_001/040	GsM	50	-	-
MO_010_001/040	FmG	-	-	1.2
MO_015_001/040	-	-	-	-
MO_020_001/040	-	-	-	-

Tabla 10-7. Valores seleccionados para los parámetros de los diferentes conjuntos experimentales del algoritmo SMAGA-1.

Como se puede observar sólo es significativo el parámetro GsM para los 4 conjuntos experimentales con Margen Operativo del 5%, y el parámetro FmG para los 4 conjuntos experimentales con Margen Operativo del 10%. En el resto de casos no existe ningún parámetro significativo. En este sentido el algoritmo es bastante insensible a las distintas configuraciones lo que nos permite utilizarlo de forma más sencilla.

Parámetros SMAGA-2	SSD10_P13
Generador Dinámico	Activado
Evolución del Factor Límite	Activado
Evolución de Riesgo Mínimo	Activado
Generaciones sin Mejora (GsM)	50
Factor de Mutación (FdM)	0.1
Factor Margen (FmG)	1.2

Tabla 10-8. Resumen de los valores asignados a los parámetros del algoritmo SMAGA-2 para todos los conjuntos experimentales.

La Tabla 10-8 muestra la configuración de parámetros con la que se ha realizado el experimento para identificar el valor del parámetro RM del algoritmo SMAGA-2. Como se muestra, se le ha asignado a GsM el valor 50 y a FmG el valor 1.2 para todos los conjuntos experimentales. En ambos casos se ha empleado el valor que ha resultado más favorable en el conjunto experimental en el que ha resultado significativo. Así mismo se ha elegido para FdM el valor 0.1 arbitrariamente, ya que es necesario disponer de alguno.

Después de realizar el análisis de los resultados obtenidos con el algoritmo SMAGA-2, que se pueden consultar en el anexo VI, se ha observado que los valores analizados (0.2 y 0.4) para el parámetro RM no son significativos en ningún conjunto experimental. Por lo tanto, SMAGA-2 también muestra un funcionamiento similar con cualquier valor de RM.

Parámetros	SMAGA-1	SMAGA-2
Generador Dinámico	Desactivado	Activado
Evolución del Factor Límite	Desactivado	Activado
Evolución de Riesgo Mínimo	Desactivado	Activado
Generaciones sin Mejora (GsM)	50	50
Factor de Mutación (FdM)	0.1	0.1
Factor Margen (FmG)	1.2	1.2
Riesgo Mínimo (RM)	-	0.4

Tabla 10-9. Resumen de los valores asignados a los parámetros del algoritmo SMAGA-1 y SMAGA-2 para cualquier conjunto experimental.

En función de las conclusiones alcanzadas, en relación con la significación de los parámetros determinados para los algoritmos SMAGA-1 y SMAGA-2, los valores que se emplearán en esta experimentación en la plataforma IPSU-MAS son los que se muestran en Tabla 10-9.

10.4.3 Decisión Pre-Experimental: Criterios de Aplicación de Algoritmo Reactivos.

Como ya se ha comentado en el apartado anterior el juego de datos actual ha variado con respecto a la experimentación previamente realizada en los capítulos 8 y 9. Y aunque ello obligó a re-parametrizar los algoritmos predictivos SMAGA-1 y SMAGA-2, en el caso de los algoritmos reactivos esto no es así. Se ha podido verificar mediante un breve experimento preliminar como con el juego de datos preparado para el experimento actual se obtiene unos resultados muy similares, en lo referente al tipo de algoritmo a aplicar, a los que se obtenía para el conjunto P3_20_SSD50. Esto implica que en caso de adelanto en las operaciones se aplicará el procedimiento SSDS, y que en caso de retraso en las operaciones el método AOR.

Tanto en el caso de adelanto, como en el caso de retraso en la fecha de finalización de las operaciones el algoritmo aplicado no depende del valor del factor K. En este proceso experimental el valor de K hubiera podido influir en el algoritmo seleccionado, aunque finalmente no ha sido así, pero no en la función objetivo como luego se comentará no depende de dicho factor. En todo caso, el valor establecido para K es 0.01. En cuanto al valor del factor VA éste se ha establecido en 0.2, ya que frente a la opción de 0.4, el primero es más similar al empleado en la Empresa seleccionada.

El último aspecto que se ha valorado en relación a la configuración del proceso de Programación Reactiva es la posibilidad de analizar, si en un caso real, se puede considerar que después de un cierto número de Reprogramaciones el Programa resultante puede ser catalogado como mejorable en relación al Makespan alcanzable. Si puede ser catalogado como mejorable entonces el Programa sería similar al caso que en el capítulo 9 se denominó SMAGA-2 modificado, es decir aplicar la Reprogramación sobre un Programa factible pero no situado próximo al óptimo en el espacio de soluciones. Por este motivo se han planteado dos variantes en la aplicación de los algoritmos de Reprogramación. Una primera variante en la que no se acepta esta hipótesis, y en la que se aplican siempre los algoritmos SSDS o AOR después de cada evento e independientemente del Programa en curso, es decir con Reprogramación Parcial (**RP**). Y una segunda en la que se acepta dicho supuesto y se realiza un Reprogramación Completa (**RC**) cada cierto número de Reprogramaciones parciales. En concreto, y para la segunda variante, se ha decidido que sea cada 4 Reprogramaciones parciales cuando se cambia el método por una Reprogramación completa basada en SMAGA-2 que es la que le corresponde al juego de datos P3_20_SSD50.

10.4.4 Diseño del Experimento

En este apartado se trata de identificar el problema, definir claramente el objetivo de la experimentación, identificar los factores que pueden influir y establecer la estrategia experimental.

La experimentación que se plantea en este apartado tiene como objetivo poner de manifiesto la bondad de la propuesta desarrollada en este trabajo. Este contraste se desea realizar más desde un punto de vista ingenieril que analítico. **En concreto se debería mostrar tanto la eficacia del procedimiento de Programación de la Producción basada en la combinación de Programaciones Predictivas y Reactivas con un elevado grado de integración en el Sistema de Planificación y Control de Operaciones como la eficiencia del mismo en base a los resultados obtenidos en el marco de su aplicación en una Empresa seleccionada del sector cerámico.**

En cuanto a la eficacia se asumirá que en líneas generales quedará demostrada en el momento en el que se aporten los resultados, que se deberán entender consecuencia del correcto funcionamiento del método propuesto, y que en este caso además estará implementado con el software correspondiente a la plataforma IPSU-MAS. No obstante, posteriormente se propondrá un indicador que permita valorar un aspecto concreto relativo al adelanto de trabajos que podría ser conflictivo y podría impedir la puesta en marcha de la propuesta. En este sentido se debe destacar una vez más que el interés de este trabajo reside en los métodos y algoritmos empleados para la Programación de la Producción, pero que precisamente por el enfoque integrador que se ha realizado de los mismo en relación a las decisiones que se toman en el ámbito de la Planificación de la Producción, ha sido necesario incorporar algunos procesos y criterios meramente funcionales pertenecientes a este último ámbito. Estos procesos y criterios del ámbito de la Planificación de la Producción a los que se hacer referencia, nunca han formado parte de los objetivos de esta tesis, y sólo son una forma de que la plataforma sea operativa, en todo caso constituyen una línea de mejora futura.

En cuanto a la eficiencia de la propuesta se ha considerado que esta quedará contrastada si se puede poner en evidencia que la Tasa de Productividad que se alcanza con ésta es claramente superior a la que resulta empleando los actuales métodos de Programación de la Producción en la Empresa seleccionada. En este punto es donde los algoritmos de Programación Predictivos y Reactivos que se han desarrollado y evaluado tienen un peso específico más elevado. En ningún caso se trata de realizar una demostración de la eficiencia de la propuesta en general, ya que las variables del entorno hacen que este tipo de demostraciones sean bastante complicadas, al mismo tiempo que faltas de sentido, cuando se trata de procedimientos de gestión complejos.

Para poder realizar la evaluación respecto de la eficiencia de la propuesta se han seleccionado dos variables básicas que se endiente que proporcionan un reflejo adecuado de lo que se pretende mostrar. La primera variable será la Tasa de Productividad promedio, es decir el número medio de m^2 fabricados cada minuto durante un periodo productivo^{vii}.

[10-4]

$$\text{Tasa de Productividad} = \frac{m^2 \text{ Fabricados en un Periodo}}{\text{Duración del Periodo en Minutos}}$$

^{vii} Hay que incidir en que al ser la duración de los periodos constante la única forma de variar la Productividad es incrementando o disminuyendo el número de trabajos. No se admiten trabajos realizados parcialmente.

La segunda variable, se ha denominado Rendimiento, y representa el porcentaje promedio de variación la Tasa de Productividad al final de un periodo con respecto a la Tasa de Productividad prevista al inicio del mismo.

$$[10-5] \quad \text{Rendimiento} = \frac{T.\text{Productividad Alcanzada} - T.\text{Productividad Prevista}}{T.\text{Productividad Prevista}}$$

La Tasa de Productividad proporciona la información de mayor relevancia ya que es la que determina, según el enfoque planteado, cual de los dos procedimientos es el más adecuado. Sin embargo, el Rendimiento aporta información relativa a que parte de la Tasa de Productividad se ha obtenido debido a un buen proceso de Programación Predictiva, y que parte de ésta se debe al procedimiento de Programación Reactiva, en la que se incluye la política de que se aplica a la modificación del Margen Operativo e incorporación de nuevos trabajos. Así mismo, se ha querido añadir una variable adicional que indique el promedio de Trabajos Cancelados por periodo, que será empleada para valorar la viabilidad de la propuesta, y por lo tanto su eficacia. Se ha considerado que un aspecto importante que debe tenerse en cuenta es la cantidad de trabajos que son incluidos en el Programa y que posteriormente se cancelan por falta de capacidad. El valor de esta variable será relevante cuando se analice la puesta en funcionamiento de la propuesta. Un número elevado de cancelaciones podría indicar un mal funcionamiento de la política de modificación del Margen Operativo que da lugar a la incorporación de trabajos que posteriormente acaban anulándose. Si esto es así, y la propuesta se implementase, se podrían generar graves problemas de gestión que no compensarían en ningún caso el posible aumento de la Tasa de Productividad.

Así mismo, se pretende que el proceso experimental considere varios entornos de trabajo. Por un lado, parece interesante incluir en este proceso, basado en un caso real, la consideración de los turnos de trabajo frente a la hipótesis habitual en un entorno teórico de que todos los recursos están disponibles 24 horas por jornada. Por este motivo se planteará un bloque experimental sin la consideración de los turnos, y otro donde se supondrá que en la etapa de Prensado-Esmaltado y en la de Clasificación-Embalaje se trabaja a dos turnos de 8 horas y en la etapa de Hornos se trabaja a 3 turnos de 8 horas. El supuesto en el que se introducen los turnos de trabajo se ha implementado tomando como referencia la situación en la Empresa seleccionada. Por otro lado, se desea distinguir entre los resultados obtenidos cuando la plataforma IPSU-MAS trabaja sin restricciones (**SR**) e incluye la opción de incorporar nuevos trabajos en las revisiones periódicas del M.O., o por el contrario cuando se restringe (**R**) esta posibilidad. La versión restringida de la plataforma podría considerarse una situación intermedia que servirá para valorar el peso de esta opción en el proceso.

Por lo que a los objetivos de la experimentación se refiere se trabajará con 8 entornos diferenciados fruto de la combinación entre las dos variantes de Reprogramación (RP o RC) y las dos variantes en la aplicación de la inclusión de nuevos trabajos (SR o R), que se identificarán con las siglas R_RP, R_RC, SR_RP, SR_RC, que a su vez serán analizadas en dos tipos de organización del trabajo en turnos (sin turnos o con turnos).

Por lo que se refiere a los factores, en un primer análisis se detectó que los factores que potencialmente influyen en el experimento son: El Margen Operativo disponible (**MO**), el número de formatos diferentes en cada línea (**CARGA**), y la batería de eventos a la que se somete a un Programa en curso (**BAT**). El factor MO puede influir ya que un mayor margen debería permitir afrontar mejor los eventos y cancelar menos trabajos que un margen menor, y por lo tanto variar la Tasa de Productividad final. El factor CARGA puede influir en la programación, ya que éste determina la cantidad de cambios de partida de larga duración que pueden producirse en un periodo, y al generar mayor cantidad de tiempos no productivos puede afectar a la Tasa de Productividad que se alcance. El factor BAT también puede influir en la Tasa de Productividad ya que no es lo mismo quedar sometido a un conjunto de eventos que sólo adelanten las operaciones que a otro que sólo las atrase, o a uno donde se mezclen ambas.

En cualquier caso, y dado que el fin último del experimento es realizar una comparativa entre dos formas de proceder (IPSU-MAS y Empresa), y no tanto realizar una inferencia de los resultados para parametrizar el funcionamiento del sistema, se ha considerado oportuno reducir los tres factores a uno sólo. El factor seleccionado es MO, ya que es el único cuyo valor, que en todo caso, puede ser decidido por el programador de la producción. Tanto el factor CARGA como el factor BAT son consecuencia de lo que la naturaleza depare en cada momento. En el caso de la CARGA, el programador podrá intentar equilibrar las líneas de forma que todas tengan un número de formatos diferentes más o menos similar, pero dicho número vendrá determinado por la Planificación realizada. En el caso del factor BAT es bastante más intuitivo, y fácil de asumir, que los eventos son por su propia naturaleza aleatorios. No obstante, para cada nivel del factor MO se ejecutarán las 21 baterías de eventos indicadas en el apartado 10.3.2.2, así como los 120 conjuntos experimentales correspondientes a los 4 tipos de CARGA (1.5, 2, 2.5, y 3) indicados en el apartado 10.4.1., los resultados obtenidos en esas 2.520 combinaciones serán promediados. En este aspecto se asume que lo realmente interesante es detectar cual es la respuesta promedio antes los diferentes formatos y eventos que se analizan con los diferentes Márgenes Operativos, que este experimento serán del 5%, del 10%, del 15% y del 20% del makespan del periodo.

Por lo que se refiere a los valores elegidos para el factor MO, estos representan un nivel de ocupación de la planta que oscilan entre el 80% (para un MO del 20%), que se puede considerar un valor normal para una planta madura y con gran demanda, y el 95% (para un

MO del 5%), que se puede asociar a una planta sumamente saturada. Se pretende analizar, por lo tanto, diferentes opciones pero todas ellas dentro de un nivel de ocupación alto a muy alto. Esto se ha planteado así, ya que en la propuesta que se ha realizado se ha establecido que se realice un ajuste del MO que permita ir incorporando trabajos planificados para periodos posteriores al actual, lo que influirá de forma determinante en la Tasa de Productividad. Se entiende que con un MO muy elevado la plataforma IPSU-MAS superaría la Tasa de Productividad que pueda obtener la Empresa fácilmente, ya que la primera incorpora nuevos trabajos y la segunda no. Con el rango de valores propuesto en el experimento se entiende que los resultados que se obtengan serán consecuencia de un cierto equilibrio entre las aportaciones de la Programación Predictiva, la Programación Reactiva, incluida en esta última la política de revisión del MO e incorporación de nuevos trabajos. Y que si en estas condiciones la propuesta realizada supera a la de la Empresas, sin duda, también lo hará con Márgenes Operativos superiores.

El experimento ha sido planificado para realizar 3 réplicas, es decir un total de 4 ejecuciones, de cada tratamiento, es decir de los 4 posibles valores de Margen Operativo. No obstante, hay que recordar para cada Margen Operativo se han planteado 2.520 combinaciones correspondientes al cruce de 4 posibles valores de formatos diferentes por líneas y a los 21 tipos de baterías de eventos disponibles. Esto supone que en total se van a realizar 4 (valores en MO) x 2.520 (conjuntos experimentales) x 4 (lanzamientos) = 40.320 ejecuciones en cada variante de entorno, y dado que hay 8 variantes se realizan un total de 322.560 ejecuciones.

Para el desarrollo de los experimentos se han utilizado cuatro ordenadores tipo PC con procesador Intel Core 2 a 2.66 GHz, con memoria de 2 GByte, y sistema operativo Windows XP Service Pack 3. Cada conjunto experimental se ha ejecutado de forma aleatoria en uno de los cuatro ordenadores. Al final de la ejecución de cada ejemplar de la plataforma IPSU-MAS ha incorporado los resultados en un fichero de texto. Los datos recogidos han sido los siguientes:

- Tasa de Productividad alcanzada al final del periodo.
- Tasa de Productividad prevista al inicio del periodo (sólo con la Programación Predictiva).
- Número de trabajos cancelados en el periodo.
- Identificación de la configuración del experimento (conjunto experimental, batería de eventos).
- Fecha y hora de inicio y fin de la ejecución del ejemplar.

Este proceso se ha repetido para cada una de las réplicas de forma idéntica. Una vez finalizadas todas las ejecuciones de cada conjunto experimental los datos se han consolidado en una hoja Excel en la cual se dispuestó los valores de la Tasa de Productividad, Rendimiento y Trabajos Cancelados para cada combinación de niveles de los factores, en cada repetición.

Una vez más se desea recalcar que para el análisis de los datos no se desea realizar un enfoque tan analítico como en los experimentos de capítulos anteriores, ampliamente soportado en herramientas estadísticas, sino una aproximación más ingenieril que permita poner en valor aquellos aspectos aporta la solución propuesta.

10.4.5 Desarrollo del Experimento y Análisis de los Resultados

Una vez desarrollado el experimento siguiendo la estrategia expuesta en el apartado anterior se ha obtenido un amplio conjunto de datos, que una vez formateados en función de las variables y factores a analizar, se pueden resumir en una serie de tablas que serán mostradas y comentadas a continuación.

10.4.5.1 Análisis de los Resultados Sin Considerar Turnos de Trabajo

En primer lugar se procederá a analizar los resultados obtenidos bajo la hipótesis de que todos los recursos están disponibles, y se emplean las 24 horas del día. En la Tabla 10-10 se puede verificar como a medida que se incrementa el M.O. disminuye la Tasa de Productividad, independientemente del método empleado para realizar el Programa productivo. Lo que es razonable si se tiene en cuenta que ello implica una reducción de la capacidad disponible en el periodo. Así mismo, se puede comprobar como el método empleado por la Empresa es superado por cualquiera de las demás opciones.

T.Productividad (m ² /minuto)	Empresa	IPSU-MAS (R_RP)	IPSU-MAS (R_RC)	IPSU-MAS (SR_RP)	IPSU-MAS (SR_RC)
MO005	7,2449	8,8772	8,7808	8,8737	8,7795
MO010	7,1207	8,6362	8,6130	8,6957	8,6785
MO015	6,8064	8,1542	8,1544	8,2815	8,2932
MO020	6,4038	7,7153	7,7192	7,9769	8,0131

Tabla 10-10. Tasa de Productividad obtenida para los diferentes enfoques de Programación Predictiva-Reactiva sin considerar turnos de trabajo expresada en m²/minuto.

Para una mejor visualización de los resultados, en la Tabla 10-11 y en la Figura 10-8, se muestran los datos de la Tasa de Productividad como el porcentaje de incremento de la opción correspondiente de IPSU-MAS con respecto a la Empresa. Se ha seguido la fórmula [10-6].

$$[10-6] \quad \text{Incremento Productividad} = 100 * \frac{(\text{Productividad}_{\text{IPSU-MAS}} - \text{Productividad}_{\text{Empresa}})}{\text{Productividad}_{\text{Empresa}}}$$

Incremento de la T. Productividad (%)	IPSU-MAS (R_RP)	IPSU-MAS (R_RC)	IPSU-MAS (SR_RP)	IPSU-MAS (SR_RC)
MO005	22,5703	21,2383	22,5210	21,2203
MO010	21,3290	21,0019	22,1645	21,9231
MO015	19,8314	19,8347	21,7021	21,8746
MO020	20,4442	20,5063	24,5287	25,0945

Tabla 10-11. Incremento de la Tasa de Productividad obtenida para los diferentes enfoques de Programación Predictiva-Reactiva sin considerar turnos de trabajo expresada como porcentaje con respecto a los valores obtenidos por la Empresa.

Según los datos de la Tabla 10-11 existe un incremento de la Tasa de Productividad que oscila entre el 19,8% y el 25%. En la Tabla 10-11 se puede comprobar como la opción que incluye Reprogramación completa cada 4 revisiones del Programa (_RC) sólo mejora la Tasa de Productividad en los casos con mayor margen operativo, el MO015 y el MO020. También se observa como la opción que contempla la posibilidad de incluir nuevos trabajos (SR_), cuando se produce una revisión del M.O. que lo permite, produce mejores resultados siempre que el M.O. es del 10% o superior. Este último resultado es fácilmente entendible, ya que con un margen operativo pequeño parece no haber resultado factible incluir ningún nuevo trabajo. Esto puede ser debido porque la capacidad finalmente disponible haya resultado insuficiente teniendo en cuenta que había una capacidad adicional sólo del 5%. Con márgenes mayores, que posibilitan mayores capacidades, si ha sido factible aplicar esta opción.

A su vez, se cree que el hecho de que la aplicación de la Reprogramación completa sólo adecuada en el MO015 y MO020, se debe a la combinación de 2 efectos. Por un lado, se debe precisamente al hecho de que son el único caso, de los estudiados, en el que el número de trabajos nuevos es lo suficientemente elevado como para modificar lo suficiente el Programa en curso. Por otro lado, cuando el margen operativo es reducido una mala Reprogramación influye más que cuando el margen operativo es mayor. El impacto de este último efecto se podrá evaluar mejor cuando se analicen el número de trabajos cancelados en los casos RP y RC.

Es imprescindible destacar que se obtienen mejores resultados con la propuesta realizada sin incorporar nuevos trabajos que con el método de la Empresa, es más la esta opción se sitúa mucho más próxima a la opción de incorporar trabajos que a la opción empleada por la Empresa. Esto que indica que la adición de trabajos no es el único aspecto importante en la Programación Reactiva

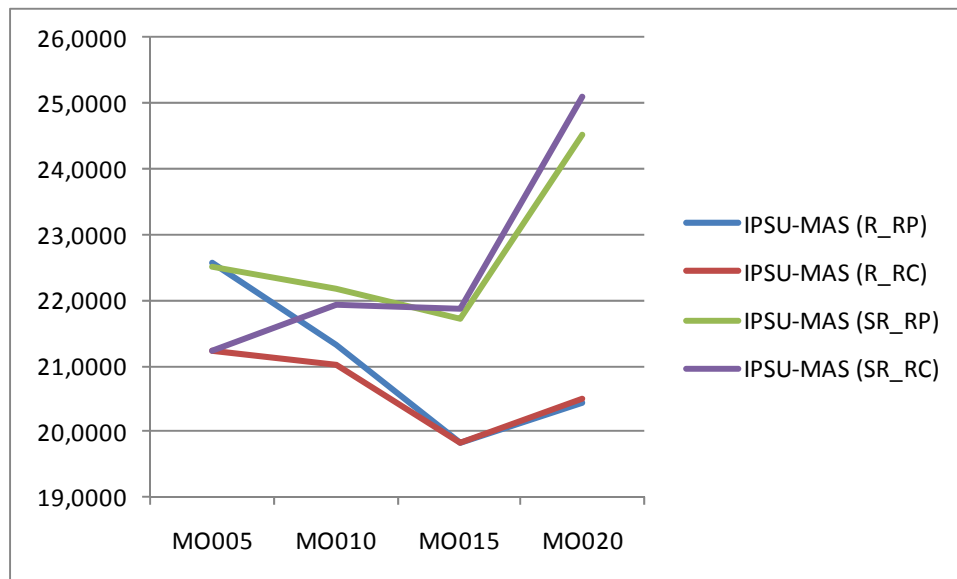


Figura 10-8. Incremento de la Tasa de Productividad obtenida para los diferentes enfoques de Programación Predictiva-Reactiva sin considerar turnos de trabajo expresada como porcentaje de mejora con respecto a los valores obtenidos por la Empresa.

Para el conjunto MO005 la diferencia máxima entre la mejor opción y la peor opción de la plataforma IPSU-MAS es del 1,35%, para el conjunto MO010 es del 1,16%, para el conjunto MO015 es del 2,04%, y para el conjunto MO020 es del 4,65%. Lo que da una idea del grado de magnitud en el impacto de las diferentes opciones de IPSU-MAS. **En el contexto de la experimentación realizada las mejoras obtenidas indican que en el conjunto MO005, en donde ninguna de las dos opciones (trabajos nuevos o Reprogramación completa) tienen efecto la variación máxima entre la mejor y la peor opción es sólo del 1,35%, pero en el caso MO020 en el cual las dos opciones tienen una influencia significativa la diferencia llega al 4,65%.**

Por lo que se refiere al Rendimiento, que nos indica el porcentaje de Tasa de Productividad final que se gana (positiva) o pierde (negativa) con respecto a la Tasa de Productividad prevista si se ejecutara sin cambios el Programa Predictivo inicial, la Tabla 10-12 nos muestra los resultados promedio relativos a los diferentes márgenes operativos. **Se observa que en este caso nuevamente los resultados, en este caso de Rendimiento, obtenidos siguiendo el**

método empleado por la Empresa son mejorados por cualquiera de las opciones de la plataforma IPSU-MAS. Y al contrario de los que ocurría con la Tasa de Productividad, los resultados de los experimentos con M.O. más amplio presentan mejores resultados, mayor Rendimiento, cualquiera que sea el método empleado. Este último aspecto se entiende cuando se considera que un M.O. más amplio permite absorber incrementos en el makespan debidos a la aparición de eventos reduciendo la necesidad de cancelar trabajos en curso.

Rendimiento	Empresa	IPSU-MAS (R_RP)	IPSU-MAS (R_RC)	IPSU-MAS (SR_RP)	IPSU-MAS (SR_RC)
MO005	-6,1374	0,0000	-1,1664	0,0773	-0,9842
MO010	-5,6206	0,0000	-0,3614	0,3339	0,3358
MO015	-4,6491	0,0000	-0,0922	0,9016	1,5074
MO020	-4,1682	0,0000	-0,0094	1,8609	3,5097

Tabla 10-12. Rendimiento obtenido para los diferentes enfoques de Programación Predictiva-Reactiva sin considerar turnos de trabajo.

En la Tabla 10-12 se observa como la opción que contempla la posibilidad de incluir nuevos trabajos (SR_), cuando se produce una revisión del M.O. que lo permite, produce mejores resultados siempre. Este efecto es fácilmente entendible, ya que la incorporación de trabajos es la única forma de mejorar la Tasa de Productividad con respecto al Programa Predictivo previsto al inicio del periodo, en cualquier otro caso la Tasa de Productividad se mantendrá o empeorará. Al cotejar los resultados de Tasa de Productividad con los resultados del Rendimiento se observa como existe una coincidencia casi completa en cuanto a este comportamiento. Sin embargo en el conjunto MO005 se puede ver como la máxima Tasa de Productividad se obtiene en un caso en el que no se incluyen nuevos trabajos (R_RP) el Rendimiento es 0, y en el caso SR_SP en donde el Rendimiento es positivo se alcanza una Tasa de Productividad menor. La única explicación que se encuentra a este hecho es que en el resultado de la Tasa de Productividad final están implicadas la Programación Predictiva y Reactiva, y aunque el Rendimiento indica que la Programación Reactiva es mejor en el caso SR_SP, ya que existe Rendimiento positivo, puede que la Tasa de Productividad obtenida con la Programación Predictiva inicial sea ligeramente mejor en el caso R_RP. Esto podría justificar la diferencia de 4 milésimas de diferencia entre ambos casos.

En la misma tabla se puede ver como la opción que incluye Reprogramación completa cada 4 revisiones del Programa (_RC) mejora la Tasa de Productividad en los casos con mayor margen operativo, que en este caso son el MO010, el MO015 y el MO020, y sólo en la opción en la que se incluyen nuevos trabajos (SR_RC). De estos sólo existe una diferencia significativa para MO015 y MO020. Si se contrasta esta información con la de la Tasa de Productividad, se comprueba que existe bastante relación entre ambos datos. La única diferencia es que la inclusión de Reprogramación completa en el caso de la Tasa de Productividad producía un

mejor resultado, no sólo en el caso de inclusión de nuevos trabajos, sino también en el caso de no incluirlos, aunque la diferencia es del orden de milésimas. Esta pequeña discrepancia nuevamente puede ser debida al efecto del Programa Predictivo inicial.

Una vez analizada la Tasa de Productividad, y el Rendimiento, se puede decir que con la plataforma IPSU-MAS siempre se obtienen mejores resultados que con el método empleado en la Empresa, y que en general, cuando existe una opción (R_RP, R_RC, SR_RP, SR_RC) cuyo resultado en términos de Tasa de Productividad indica que es significativa con respecto al resto de las opciones de la plataforma existe esa misma significación en términos de Rendimiento. Por lo tanto, se intuye que la propuesta implementada en la plataforma IPSU-MAS mejora los resultados mediante una Programación Predictiva que aporta un porcentaje de la mejora y la correcta gestión del proceso de revisión del Programa que aporta un segundo porcentaje.

A continuación se va a proceder a calcular en que porcentaje el incremento de la Tasa de Productividad final de la plataforma IPSU-MAS con respecto a la Empresa se debe a la Programación Predictiva y en que porcentaje al resto de acciones. En la Tabla 10-13 se han calculado los incrementos de Tasa de Productividad inicial, es decir la prevista si se ejecutara sin modificaciones el Programa Predictivo calculado al inicio del periodo, para las diferentes opciones de la plataforma IPSU-MAS con respecto al obtenido por el Empresa.

Incremento de la T. Productividad Inicial (%)	IPSU-MAS (R_RP)	IPSU-MAS (R_RC)	IPSU-MAS (SR_RP)	IPSU-MAS (SR_RC)
MO005	15,0111	15,1038	14,8760	14,8749
MO010	14,4670	14,5724	14,8716	14,6424
MO015	14,2319	14,3404	14,9785	14,4544
MO020	15,4575	15,5279	17,1921	15,8492

Tabla 10-13. Incremento de la Tasa de Productividad del prevista con el Programa inicial obtenida para los diferentes enfoques de Programación Predictiva-Reactiva sin considerar turnos de trabajo expresada como porcentaje con respecto a los valores obtenidos por la Empresa.

En la Tabla 10-14 se ha calculado el porcentaje del incremento de la Tasa de Productividad final que se debe al proceso de modificación del Programa en curso, es decir a la Programación Reactiva con el proceso de inclusión de nuevos trabajos. La suma de la aportación debida al Programa Predictivo de la reflejada en la Tabla 10-13 más la aportación debida al proceso de modificación del Programa reflejada en la Tabla 10-14 es la que de muestra en la Tabla 10-11.

Incremento de la T. Productividad final debido a la Programación Reactiva (%)	IPSU-MAS (R_RP)	IPSU-MAS (R_RC)	IPSU-MAS (SR_RP)	IPSU-MAS (SR_RC)
MO005	7,5592	6,1345	7,6450	6,3453
MO010	6,8621	6,4294	7,2928	7,2806
MO015	5,5995	5,4942	6,7235	7,4202
MO020	4,9867	4,9784	7,3366	9,2453

Tabla 10-14. Diferencia entre el incremento de la Tasa de Productividad alcanzado con el Programa finalmente ejecutado y el previsto al inicio del periodo sin considerar turnos de trabajo.

Como se ve en líneas generales el Proceso de Negocio asociado a la Programación Predictiva aporta entorno al 75% de lo que se mejora con respecto al proceso actual que aplica la Empresa y el otro 25% se obtiene mediante el Proceso de Negocio de Programación Reactiva.

Como ya se comentó en el diseño del experimento aunque la eficiencia pudiera ser elevada sería conveniente revisar el número de trabajos cancelados en promedio por periodo ya que este aspecto podría influir en la viabilidad de la propuesta, y su posible implementación en un entorno real. No sería admisible que el incremento en la eficiencia fuese a costa de generar un excesivo número de modificaciones en el Programa que tuviera una fuerte repercusión fuera del ámbito del programador, como es el caso de la cancelación de un trabajo. La Tabla 10-15 nos muestra los resultados promedio relativos a los diferentes márgenes operativos. Los datos indican que el número de trabajos cancelados es mínimo, aunque en este caso algunas opciones de la plataforma son peores que las de la Empresa.

Nº de Trabajos Cancelados	Empresa	IPSU-MAS (R_RP)	IPSU-MAS (R_RC)	IPSU-MAS (SR_RP)	IPSU-MAS (SR_RC)
MO005	1,1206	0,0000	0,4288	0,9969	1,3743
MO010	1,0923	0,0000	0,1017	0,9556	1,0261
MO015	0,9602	0,0000	0,0270	0,8782	0,9250
MO020	0,7596	0,0000	0,0027	0,7787	0,7638

Tabla 10-15. Número de Trabajos Cancelados obtenidos para los diferentes enfoques de Programación Predictiva-Reactiva sin considerar turnos de trabajo.

La información de la Tabla 10-15 pone de manifiesto como la opción que incluye Reprogramación completa de forma esporádica (_RC) produce en todos los casos menos en uno más trabajos cancelados que la Reprogramación parcial (_RP), aunque desde el punto de

vista del programador la diferencia entre ambas opciones no es muy relevante en los conjuntos MO010, MO015 y MO020. Este comportamiento se puede entender, si tenemos en cuenta, que ya se observó como para el conjunto MO005 el empleo de Reprogramación completa no es el más adecuado, y ello implica un peor aprovechamiento de los recursos y un aumento de la probabilidad de cancelar trabajos. Esto indica poco a poco se va consumiendo tiempo del M.O., que debido a una mala Reprogramación (caso RC) se consume más rápido que con RP, y se producen cancelaciones de los trabajos más frecuentemente. Por este motivo ahora se puede afirmar la observación que se realizó al analizar la Tasa de Productividad, en la que se indicó que el aumento de la misma con el método que incluye Reprogramación completa (RC) para los casos MO015 y MO020 podía ser debido, además del al mayor número de trabajos nuevos que se da en estos conjuntos debido a su margen, a que el efecto negativo de su aplicación se nota menos con márgenes elevados.

Sin embargo, aún es más relevante para el programador la diferencia que se produce cuando se trata posibilidad de incluir nuevos trabajos (SR_) o no (R_), ya que la primera produce un número comparativamente mucho más elevado de trabajos cancelados. Este comportamiento no es más que el efecto lógico que produce la opción de incluir nuevos trabajos. Si aparecen trabajos nuevos el taller se carga más, y cualquier evento inesperado puede tener como consecuencia la cancelación de un trabajo, y con mayor probabilidad contra menos margen operativo se tienen disponible.

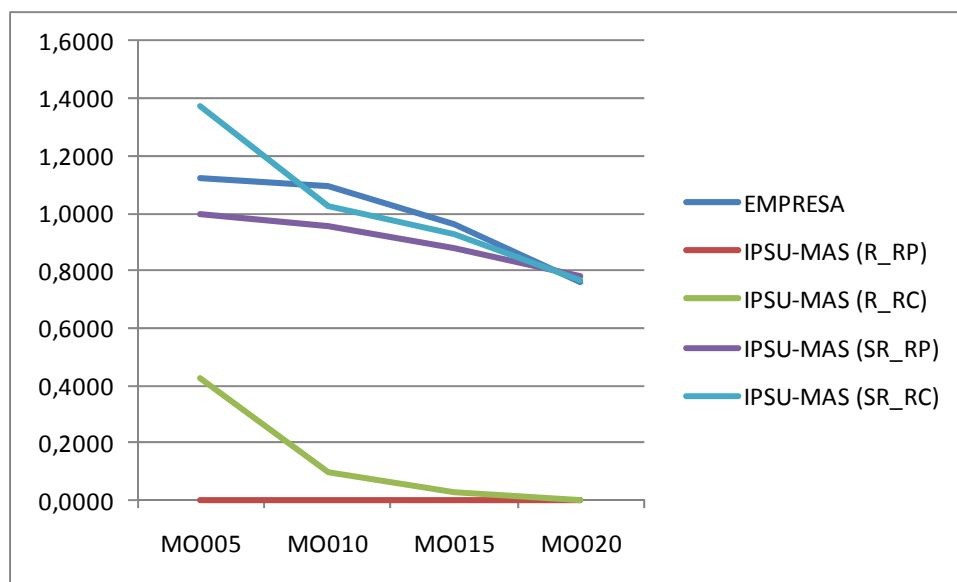


Figura 10-9. Número de Trabajos Cancelados obtenidos para los diferentes enfoques de Programación Predictiva-Reactiva sin considerar turnos de trabajo.

En la Figura 10-9 se puede ver como el número de trabajos cancelado en promedio por periodo en los casos con los que se alcanza mayor Tasa de Productividad similar a los que cancela la Empresa con su método de trabajo. Sin embargo, la plataforma ofrece alternativas en el que el número de trabajos cancelados es cero o valores muy próximos a éste, y en los que la Tasa de Productividad, aunque menor que en los mejores casos, está muy por encima de la que alcanza la Empresa. **Esto permite al programador optar por enfoques más conservadores, que eviten posible problemas de cumplimiento del Programa, u optar por enfoques más agresivos, que permitan alcanzar altas cotas de la Tasa de Productividad sin llegar a comprometer la fiabilidad del Programa.**

Si el programador se enfrentara al problema de programación de la producción con un enfoque agresivo en los conjunto MO010, MO015 y MO020 debería optar con la opción SR_RP o SR_RC, mientras que con un enfoque más conservador, en esos mismos conjunto, podría optar por la opción R_RP. En el conjunto MO005 la opción R_RP sería las más adecuada en cualquiera de los casos.

Se puede concluir el análisis subrayando que el en caso estudiado la propuesta ha proporcionado mejores resultados tanto en su Proceso Predictivo como Reactivo que los empleados por la Empresa, y que su implementación parece factible tanto por los resultados alcanzados como por que ello no supone en ningún caso un incremento del número de trabajos cancelados, es más en algunos casos es similar o menor.

10.4.5.2 Análisis de los Resultados Considerando Turnos de Trabajo

En segundo lugar se procederá a analizar los resultados obtenidos bajo la hipótesis de que no todos los recursos están disponibles en todos los casos, y se emplean 2 turnos cada uno de 8 horas al día para la etapa de Prensado-Esmaltado y para la etapa de Clasificación-Embalaje, y 3 turnos de 8 horas cada uno para la etapa de Horneado.

T. Productividad m ² /minuto	Empresa	IPSU-MAS (R_RP)	IPSU-MAS (R_RC)	IPSU-MAS (SR_RP)	IPSU-MAS (SR_RC)
MO005	4,6657	5,7014	5,3683	5,7096	5,4716
MO010	4,3363	5,3667	5,3060	5,4388	5,3696
MO015	4,0305	4,9252	4,8945	5,0958	5,0180
MO020	3,7781	4,5716	4,5628	4,9061	4,9003

Tabla 10-16. Tasa de Productividad obtenida para los diferentes enfoques de Programación Predictiva-Reactiva considerando turnos de trabajo expresada en m²/minuto.

En la Tabla 10-16 se puede comprobar que tal y como ocurría en el caso de no disponer de turnos a medida que se incrementa el M.O. disminuye la Tasa de Productividad, independientemente del método empleado para realizar el Programa Productivo.

Para una mejor visualización de los resultados, en la Tabla 10-17 y en la Figura 10-10 se muestran los datos de la Tasa de Productividad como el porcentaje de incremento de la opción correspondiente de IPSU-MAS con respecto a la Empresa. Se ha seguido la fórmula [10-6].

Incremento de la T. Productividad %	IPSU-MAS (R_RP)	IPSU-MAS (R_RC)	IPSU-MAS (SR_RP)	IPSU-MAS (SR_RC)
MO005	22,2554	15,1134	22,4310	17,3277
MO010	23,7334	22,3330	25,3966	23,7994
MO015	22,1545	21,3943	26,3868	24,4572
MO020	20,9966	20,7628	29,8488	29,6947

Tabla 10-17. Incremento de la Tasa de Productividad obtenida para los diferentes enfoques de Programación Predictiva-Reactiva considerando turnos de trabajo expresada como porcentaje de mejora con respecto a los valores obtenidos por la Empresa.

En comparación con los resultados obtenidos en apartado anterior se observa como la Tasa de Productividad se ha reducido, así como el porcentaje de mejora de la Tasa de Productividad con respecto a la Empresa por lo general ha aumentado. La primera observación es consecuencia de la reducción del tiempo de trabajo por jornada. La segunda puede deberse a que al tener en cuenta los turnos de trabajo se incluye una restricción al problema que antes no existía, cuyo efecto sobre el problema se traduce en una mayor sensibilidad a los cambios. Un pequeño cambio en un Programa en curso puede suponer la aparición de un turno de descanso, 8 horas adiciones sin realizar operaciones, dentro del periodo de ejecución de una operación. Este supuesto implicaría que la aportación con respecto al método empleado por la Empresa del proceso de Programación Reactiva, que es el que trata esos cambios en el Programa en curso, en relación con el porcentaje total de mejora es superior en este entorno que en el del apartado anterior. Esta hipótesis se verificará al tratar los resultados del Rendimiento.

Respecto a la evolución de la Tasa de Productividad alcanzada en este bloque de experimentos, en la Tabla 10-17 se puede ver como la opción que incluye Reprogramación completa cada 4 revisiones del Programa (_RC) no mejora la Tasa de Productividad en ningún caso. Aunque existe una tendencia a que la diferencia entre ambas se reduzca a medida que aumenta el M.O., siendo bastante amplia en el conjunto MO005 y muy reducida en el conjunto MO020. Como ya se vio en el apartado anterior aquí influyen dos efectos, y en ambos

los márgenes operativos amplios se ven beneficiados. Por un lado, un M.O. amplio permite incluir más trabajos, y por otro puede absorber mejor una mala Reprogramación provocando menos trabajos cancelados. También se observa como la opción que contempla la posibilidad de incluir nuevos trabajos (SR_), cuando se produce una revisión del M.O. que lo permite, produce mejores resultados siempre. En este caso la diferencia entre ambas opciones también aumenta a medida que lo hace el M.O. Este comportamiento está ligado al hecho de que nuevos trabajos incrementan la Tasa de Productividad. Sin embargo, es imprescindible destacar que aún así se obtienen mejores resultados con la propuesta realizada sin incorporar nuevos trabajos que con el método de la Empresa, es más esta opción se sitúa mucho más próxima a los valores de la opción que permite incorporar trabajos que a la opción empleada por la Empresa. Esto que indica que la adición de trabajos no es el único elemento relevante en la Programación Reactiva.

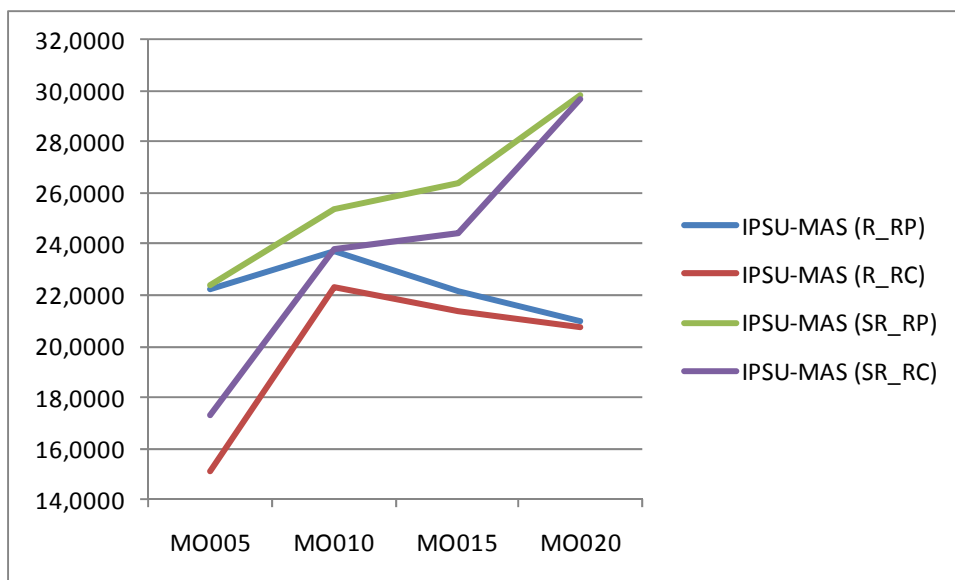


Figura 10-10. Incremento de la Tasa de Productividad obtenida para los diferentes enfoques de Programación Predictiva-Reactiva considerando turnos de trabajo expresada como porcentaje de mejora con respecto a los valores obtenidos por la Empresa.

Para el conjunto MO005 la diferencia máxima entre la mejor opción y la peor opción de la plataforma IPSU-MAS es del 7,31%, para el conjunto MO010 es del 3,06%, para el conjunto MO015 es del 4,99%, y para el conjunto MO020 es del 9,08%. Lo que da una idea del grado de magnitud en el impacto de las diferentes opciones de IPSU-MAS, y como en este caso existe una diferencia más marcada entre las diferentes opciones.

Por lo que se refiere al Rendimiento, la Tabla 10-18 nos muestra los resultados promedio relativos a los diferentes márgenes operativos. Se observa que en este caso nuevamente los

resultados obtenidos siguiendo el método empleado por la Empresa son mejorados por cualquiera de las opciones de la plataforma IPSU-MAS, excepto en la opción R-RC para el conjunto MO005. Y al contrario de los que ocurría con la Tasa de Productividad, los resultados de los experimentos con M.O. más amplio presentan mejores resultados, mayor Rendimiento, cualquiera que sea el método empleado. Este último aspecto se entiende cuando se considera que un M.O. más amplio permite absorber incrementos en el makespan debidos a la aparición de eventos reduciendo la necesidad de cancelar trabajos en curso.

Rendimiento	Empresa	IPSU-MAS (R_RP)	IPSU-MAS (R_RC)	IPSU-MAS (SR_RP)	IPSU-MAS (SR_RC)
MO005	-5,2277	0,0000	-5,4298	0,1376	-3,6956
MO010	-4,8190	0,0000	-1,2210	1,3582	-0,2182
MO015	-4,5042	0,0000	-0,6461	3,4114	1,7644
MO020	-3,5026	0,0000	-0,0143	7,3185	6,9654

Tabla 10-18. Rendimiento obtenido para los diferentes enfoques de Programación Predictiva-Reactiva sin considerar turnos de trabajo.

En consonancia con los datos de la Tasa de Productividad se produce mayor Rendimiento cuando se permite incluir nuevos trabajos (SR) que cuando no (R), y también cuando siempre se aplica Reprogramación parcial (RP), que cuando se mezcla con alguna Reprogramación completa (RP).

A continuación se va a proceder a calcular en que porcentaje el incremento de la Tasa de Productividad final de la plataforma IPSU-MAS con respecto a la Empresa se debe a la Programación Predictiva y en que porcentaje al resto de acciones. En la Tabla 10-19 se han calculado los incrementos de Tasa de Productividad inicial, es decir la prevista si se ejecutara sin modificaciones el Programa Predictivo calculado al inicio del periodo, para las diferentes opciones de la plataforma IPSU-MAS con respecto al obtenido por el Empresa.

Incremento de la T. Productividad Inicial (%)	IPSU-MAS (R_RP)	IPSU-MAS (R_RC)	IPSU-MAS (SR_RP)	IPSU-MAS (SR_RC)
MO005	15,8085	15,3039	15,8155	15,4055
MO010	17,7985	17,9048	17,7821	18,1190
MO015	16,6923	16,7202	16,7524	16,8306
MO020	16,7638	16,5550	16,7611	17,0076

Tabla 10-19. Incremento de la Tasa de Productividad del prevista con el Programa inicial obtenida para los diferentes enfoques de Programación Predictiva-Reactiva considerando turnos de trabajo expresada como porcentaje con respecto a los valores obtenidos por la Empresa.

En la Tabla 10-20 se ha calculado el porcentaje del incremento de Tasa de Productividad final que se debe al proceso de modificación del Programa en curso, es decir a la Programación Reactiva y a la inclusión de nuevos trabajos. La suma de la aportación debida al Programa Predictivo de la reflejada en la Tabla 10-19 más la aportación debida al proceso de modificación del Programa reflejada en la Tabla 10-20 es la que de muestra en la Tabla 10-17.

Incremento de la T. Productividad final debido a la Programación Reactiva (%)	IPSU-MAS (R_RP)	IPSU-MAS (R_RC)	IPSU-MAS (SR_RP)	IPSU-MAS (SR_RC)
MO005	6,4469	-0,1905	6,6155	1,9221
MO010	5,9350	4,4282	7,6144	5,6804
MO015	5,4623	4,6742	9,6344	7,6266
MO020	4,2327	4,2079	13,0876	12,6871

Tabla 10-20. Diferencia entre el incremento de la Tasa de Productividad alcanzado con el Programa finalmente ejecutado y el previsto al inicio del periodo considerando turnos de trabajo.

En este caso la portación del Proceso de Negocio de Programación Reactiva es mucho más variable que en el apartado anterior, y en algún caso supone hasta el 43% de la mejora, y en otra puede suponer incluso un empeoramiento. En todo caso, dentro de la opción que mejores resultados ofrece en cuanto a la Tasa de Productividad, la SR_RP, el proceso de Reprogramación aporta alrededor del 29% de la mejora en los conjuntos MO005 y MO010, aproximado el 36% en el MO015 y cercano al 43% en el MO020. Por lo que se ha observado, la Programación Reactiva es más relevante cuando se tiene en cuenta los turnos, que cuando no. Y en el caso de disponer de turnos, es más relevante cuando el M.O. es mayor.

Nº de Trabajos Cancelados	Empresa	IPSU-MAS (R_RP)	IPSU-MAS (R_RC)	IPSU-MAS (SR_RP)	IPSU-MAS (SR_RC)
MO005	1,0451	0,0000	1,6257	0,9952	2,0450
MO010	0,9321	0,0000	0,3252	0,9009	1,2648
MO015	0,7997	0,0000	0,1574	0,8277	1,0331
MO020	0,6482	0,0000	0,0030	0,6626	0,6198

Tabla 10-21. Número de Trabajos Cancelados obtenidos para los diferentes enfoques de Programación Predictiva-Reactiva considerando turnos de trabajo.

Como ya se comentó en el diseño del experimento aunque la eficiencia pudiera ser elevada sería conveniente revisar el número de trabajos cancelados en promedio por periodo ya que este aspecto podría influir en la viabilidad de la propuesta y su posible implementación en un

entorno real. La Tabla 10-21 nos muestra los resultados promedio relativos a los diferentes márgenes operativos. **Los datos indican que el número de trabajos cancelados son en general asumibles, aunque en este caso algunas opciones de la plataforma son peores que las de la Empresa.**

La información de la Tabla 10-21 pone de manifiesto como la opción que incluye Reprogramación completa de forma esporádica (_RC) produce en todos los casos, menos en uno, más trabajos cancelados que la Reprogramación parcial (_RP). Aunque la diferencia disminuye cuando aumenta el M.O. y pasa a ser casi despreciable para MO020. Además la opción RC produce un aumento de cancelaciones que en comparación con los que obtiene la Empresa son elevados, mientras que la opción RP mejoran o están a la par que los de la Empresa. Como ya se ha dicho esto es debido a que poco a poco se va consumiendo tiempo del M.O., que debido a una mala Reprogramación (caso RC) se consume más rápidamente que con RP, y se producen cancelaciones de los trabajos más frecuentemente. Por este motivo se justifica que la Tasa de Productividad empleando la opción RC o RP se aproximen cuando el M.O. es amplio y se produzca una diferencia más acusada y favorable a la opción RP cuando el M.O. es reducido.

Si se compra la opción de incluir trabajos nuevos (SR) frente a la de no incluirlos (R) se puede observar como existe un incremento sustancial de los trabajos cuando se permite la incorporación de nuevos trabajos. Que este caso, debido al nivel que se alcanza, se puede dudar de la viabilidad de la opción SR_RC.

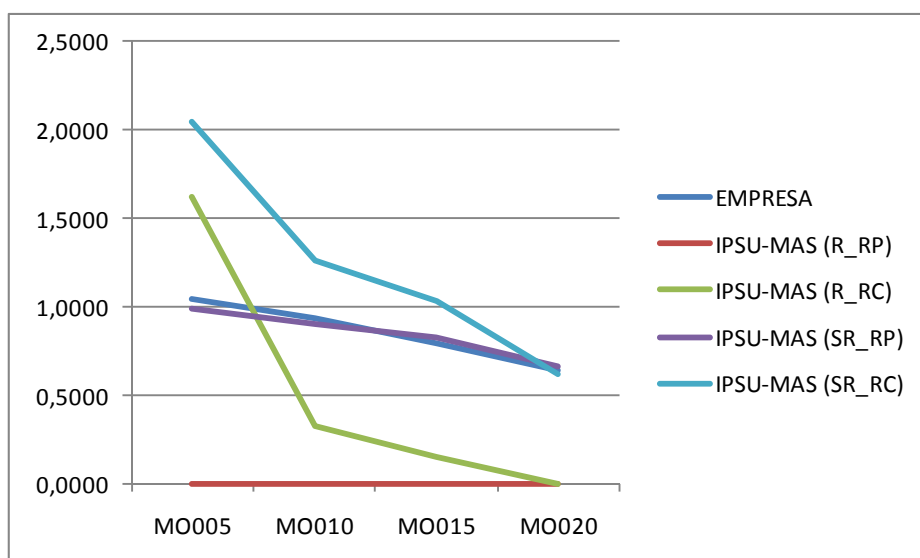


Figura 10-11. Trabajos Cancelados obtenidos para los diferentes enfoques de Programación Predictiva-Reactiva considerando turnos de trabajo expresada como porcentaje de mejora con respecto a los valores obtenidos por la Empresa.

En la Figura 10-11 se puede ver como el número de trabajos cancelados en promedio por periodo en los casos con los que se alcanza mayor Tasa de Productividad, es decir la opción SR_RP, similar a los que cancela la Empresa con su método de trabajo. Sin embargo, igual que ocurría en el apartado anterior, hay que recordar que la plataforma ofrece alternativas en el que el número de trabajos cancelados es cero o valores muy próximos a éste, y en los que la Tasa de Productividad, aunque menor que en los mejores casos, está muy por encima de la que alcanza la Empresa. **Esto permite al programador optar por enfoques más conservadores, que eviten posible problemas de cumplimiento del Programa, u optar por enfoques más agresivos, que permitan alcanzar altas cotas de la Tasa de Productividad sin llegar a comprometer la fiabilidad del Programa.**

Si el programador se enfrentara al problema de Programación de la Producción con un enfoque agresivo debería optar con la opción SR_RP, mientras que con un enfoque más conservador podría optar por la opción R_RP para cualquier conjunto.

Se puede concluir el análisis subrayando que en el caso estudiado la propuesta a proporcionado mejores resultados tanto en su Programa Predictivo como Reactivo que los empleados por la Empresa, y que su implementación parece factible tanto por los resultados alcanzados como porque ello no supone en ningún caso un incremento del número de trabajos cancelados, es más en algunos casos es similar o menor.

10.5 Conclusiones

En este capítulo se ha realizado la implementación de la propuesta planteada desde el capítulo 6 hasta el 9, mediante un software denominado plataforma IPSU-MAS, que ha sido adaptada a caso real de una Empresa productora de baldosas cerámicas seleccionada como piloto.

Para el adecuado funcionamiento de la propuesta en un caso real ha sido necesario, no sólo parametrizar la plataforma, sino incorporar algunos criterios y procesos que por no pertenecer al ámbito en el que se ha centrado el análisis de esta tesis no habían sido definidos. En concreto, han sido necesarios algunos criterios decisionales del ámbito del planificador de la producción. Por lo que se refiere, a parámetros propios del ámbito del programador de la producción se han establecido fundamentalmente los relacionados con los algoritmos Predictivos y Reactivos.

Por lo que se refiere a los resultados obtenidos, **se ha demostrado que con la propuesta realizada siempre se alcanza la Tasa de Productividad mejor que con el actual método empleado por la Empresa.** Tanto para el caso de Reprogramación completa como parcial, con

o sin incorporación de nuevos trabajos, con cualquier tipo de Margen Operativo y con consideraciones de turnos o trabajando 24 horas en todas las etapas el **Incremento de la Tasa de Productividad con respecto a las Empresa puede oscilar aproximadamente entre el 14,2% y 29,8%**. Así mismo, **el Rendimiento del Programa final frente al previsto sólo es mejor con el método empleado por la Empresa en un caso, y el número de trabajos cancelado por periodo es menor en la mayoría de los casos que se emplea IPSU-MAS.**

Emplear la plataforma IPSU-MAS mejora los procesos actuales de la Empresas tanto por la aportación de los procesos Predictivos como Reactivos, y en concreto que alrededor de un 75% de la mejora es debida a la primera y el 25% a la segunda para el caso de trabajar sin turnos y en el caso de trabajar con turnos la aportación de la Programación Reactiva llega hasta el 45%.

Dentro de los dos entornos sobre los que se ha contrastado la validez de la propuesta, con y sin turnos de trabajo, el más realista, es decir **con turnos de trabajo ha inclinado aún más la balanza a favor de la plataforma IPSU-MAS**, ya que ha dejado ver un mayor margen de mejora.

Para finalizar, cabe subrayar que como se ha indicado a lo largo del capítulo existen algunos aspectos que son mejorables, entre los que se destaca los que tienen que ver con la Toma de Decisiones sobre los trabajos a cancelar o a adelantar, o el análisis de riesgos y modificación del Margen Operativo. Cada uno de estos aspectos puede constituir líneas de trabajo futuras, tanto en el propio proceso a desarrollar, como en su coordinación entre los ámbitos de Planificación y Programación de la Producción, o como por la ampliación de la casuística que a su vez se producirá en el ámbito de la Programación de la Producción.

Con este capítulo finaliza la presentación de la propuesta desarrollada en esta tesis. El próximo capítulo será el último, y en el se realizará un resumen del trabajo realizado, de las conclusiones obtenidas, y de las líneas de trabajo futuro que han detectado.

10.6 Bibliografía

Giret-Boggino, A. S. 2005. ANEMONA: Una Metodología Multiagente para Sistemas Holónicos de Fabricación. Universidad Politécnica de Valencia.

Conclusiones y Trabajo Futuro

Capítulo

11

Tabla de contenido

11.	Capítulo 11	701
11.1	Introducción	701
11.2	Identificación de una Oportunidad	701
11.2.1	Identificación Preliminar del Problema	702
11.2.2	Análisis del Estado del Arte	703
11.2.3	Definición del Problema	705
11.3	Enfoque para el Desarrollo de la Propuesta	706
11.4	La Propuesta	707
11.4.1	Análisis, Diseño e Implementación	707
11.4.2	Prueba de Funcionamiento	709
11.5	Trabajo Futuro	709
11.6	Bibliografía	711

11.1 Introducción

Este último capítulo se plantea como una reflexión sobre todo el trabajo realizado en esta tesis. En él se exponen y se valoran de forma sintética los hitos más relevantes del proceso investigador que se ha planteado en este trabajo. También se identifican los aspectos que resultaría interesante intensificar en el futuro, tanto para complementar el trabajo realizado, como para constituir líneas de investigación a explotar en años sucesivos.

El capítulo se ha estructurado en cuatro apartados, además de este. A través de los tres primeros apartados se recorre el proceso investigador, y en el cuarto se explicitan las líneas de trabajo futuro. En el apartado 11.2 se comentan los aspectos destacados del periodo que abarca desde el momento en el que se empieza a considerar el Proceso de la Programación de la Producción en el entorno de la Industria Cerámica como ámbito de la investigación hasta que se define un problema concreto. En el apartado 11.3 se describe que ha supuesto para este trabajo, y que aporta en general, el enfoque seguido para desarrollar la propuesta, una vez identificado el problema. En el apartado 11.4 se identifican los hitos, y las aportaciones reseñables, de la propuesta de solución planteada en este trabajo. Finalmente, como cierre de la tesis, en el apartado 11.5 se plantean algunas líneas de investigación que se han considerado relevantes en un futuro inmediato.

11.2 Identificación de una Oportunidad

En este apartado se identifican los aspectos más destacables del trabajo realizado en la fase inicial del mismo, durante el periodo de tiempo dedicado a la definición de un problema concreto y poco analizado en la literatura que sirviera como referencia del proceso investigador. Quizás este periodo haya sido el de mayor duración de todos, y el que más tiempo se ha dedicado a reflexionar sin avances significativos.

El proceso recorrido para la identificación de una oportunidad investigadora se dividió, como este apartado, en tres etapas: En la primera de ellas se abordó, en líneas generales, la Industria Cerámica y su Proceso de Programación de la Producción hasta encontrar un nicho poco analizado que pudiera constituirse en el caso base que se pretendía. En la segunda etapa se analizó el estado del arte para establecer la viabilidad de la investigación. Y en la tercera etapa, y una vez se consideró válido el caso identificado de forma preliminar, se procedió a la definición del problema en su estado actual o “as is”.

11.2.1 Identificación Preliminar del Problema

En la etapa dedicada a la identificación preliminar del problema se realizó una aproximación a la Programación de la Producción mediante un análisis que se inició en el Sector Cerámico y finalizó con una visión general del proceso de fabricación del pavimento y revestimiento cerámico en una Empresa.

Una vez concluida la aproximación se realizaron las primeras aportaciones identificando comportamientos y situaciones características del proceso productivo de este tipo de industria.

Se observó como los plazos de fabricación son elevados (Lead Time) pero que éstos tienen que ser compatibilizados con un proceso de Gestión de Pedidos más dinámico que el de producción. Esta situación influye en la Generación de Planes Maestros que se protegen de los cambios en los pedidos mediante periodos de congelación en la Planificación lo que supone una pérdida de flexibilidad de los mismos.

También se destacó como el proceso fabril está sujeto a una fuerte incertidumbre con un componente doble. Por un lado, debido a las características del proceso de transformación de la materia prima se provocan desviaciones, fundamentalmente, respecto a los tonos y calibres de referencia lo que obliga a prolongar los tiempos de las operaciones de forma no determinada a priori. Y por otro lado, debido a la existencia de cambios de partida elevados y dependientes de la secuencia, que también pueden ser variables, se generan tiempos de preparación imposibles de acotar hasta que no se define la secuencia en cada recurso, o incluso hasta que el Programa se ejecuta.

Así mismo se identificó que el proceso productivo, desde el punto de vista del cálculo de los Programas Productivos, se compone de tres etapas: Prensado-Esmaltado, Hornos y Clasificación-Embalaje. Y como éste se puede clasificar como un Taller de Flujo Híbrido con Tiempos de cambio de Partida Dependiente de la Secuencia.

La aportación más destacada de esta etapa es la caracterización de como el Sistema de Planificación y Control de Operaciones de este tipo de industria ha reaccionado incluyendo tiempos ociosos y tamaños de lote mínimos cada vez mayores para combatir la incertidumbre existente. Y de que manera aunque se definen objetivos basados en la eficiencia de la utilización de los recursos la evolución actual (periodos congelados, grandes lotes, tiempos ociosos) no favorece en nada dichos objetivos. Aunque las empresas del sector son conscientes de que en estos momentos la competitividad requiere un incremento de la flexibilidad y la eficiencia en el empleo de los recursos, y por este motivo éstas están

invirtiendo en tecnología y en mejorar sus procesos, la priorización de los planes de actuación deja al descubierto un campo importante. La Programación de la Producción como herramienta para la Toma de Decisiones en un ámbito operativo podría ayudar a reducir los efectos de la Incertidumbre.

El proceso de Ayuda a la Toma de Decisiones se constituyó en el punto de partida de este trabajo. Y en este aspecto se identificó una doble opción de mejora. Por un lado, considerar el hecho de que la interacción de las actividades de Planificación y Programación de la Producción actuales no facilitan, en la mayoría de los casos, la coordinación entre ambos procesos lo que sin duda favorecería la flexibilidad, amortiguando el efecto de la Incertidumbre. Y por otro lado, la consideración del problema de Programación de la Producción no como una decisión única y válida para todo el periodo productivo considerado, sino como una sucesión de decisiones, cada una de las cuales se apoya en el estado actual del sistema, y que en cada momento actúa contrarrestando los efectos de la Incertidumbre, mejorando la eficiencia en términos de Tasa de Productividad y favoreciendo la Flexibilidad del taller.

Dos decisiones han resultado claves para el desarrollo de este trabajo. La primera de ellas hacer referencia a la decisión de continuar el trabajo manteniendo la complejidad y amplitud del problema planteado. La segunda de ellas es la de apostar desde un principio por una propuesta de solución basada en métodos poco analizados en el ámbito de la Programación de la Producción, como es el caso de los Sistemas Multiagente, y su combinación con otras técnicas más consolidadas.

11.2.2 Análisis del Estado del Arte

La revisión del estado del arte tuvo, como en todo proceso investigador un doble objetivo: Validar la originalidad de la propuesta y estudiar alternativas para abordar la solución (modelos, técnicas, etc.). En este caso se realizó una doble revisión. Por una parte, se analizaron las aportaciones que desde el ámbito de los Sistemas de Fabricación se centran en el ámbito Programación de la Producción sus modelos y sus métodos. Y por otra parte, se estudiaron las referencias que desde el ámbito de la Inteligencia Artificial, y en concreto de la Teoría de la Agencia, aportaban modelos y técnicas que han sido o pueden ser aplicables a la gestión de los sistemas de fabricación.

La aportación más destacada en relación a la revisión de la bibliografía se expresó en forma de conclusiones. Por lo que se refiere al análisis de los métodos y algoritmos para la Programación de la Producción procedentes del ámbito de los sistemas de fabricación las conclusiones más relevantes son:

- Existe una preocupación por los problemas con Incertidumbre, pero con una escasa presencia en la literatura.
- Los problemas dinámicos se han analizado desde un primer momento, sin embargo su aplicación ha resultado ser muy escasa frente a los casos estáticos.
- La Propuesta que mejor se ajusta a la forma de trabajar actualmente implantada en muchas de las empresas productoras de baldosas cerámicas es la de la Programación Predictivo-Reactiva de la Producción.
- Las referencias relevantes en el ámbito de la Programación Reactiva o Reprogramación con Incertidumbre proceden de una aplicación en el ámbito de la informática.
- Entre los métodos empleados son pocas las referencias que utilizan técnicas optimizadoras, decantándose la mayoría por algoritmos heurísticos o metaheurísticos. Entre estas, las heurísticas ad-hoc o las metaheurísticas con cierto mimetismo con la naturaleza o bio-inspiradas son las más empleadas.

Por lo que se refiere al análisis de los métodos y algoritmos para la Programación de la Producción procedentes del ámbito Sistemas Multiagente las conclusiones más relevantes son:

- Los Sistemas Multiagente han demostrado ser útiles para resolver el Problema de la Programación de la Producción, tanto desde un punto de vista cuantitativo del problema, como por lo que se refiere a la coordinación con otras áreas del Sistema de Planificación y Control de Operaciones.
- El número de referencias aplicado a la Programación de la Producción que se ha podido encontrar ha sido escaso.
- La aplicación de los Sistemas Multiagente no garantiza alcanzar buenas soluciones, es simplemente un entorno de trabajo que puede ser muy favorable en algunos problemas. Por lo que es necesario validar previamente el entorno de aplicación, así como conducir la misma en base a la teoría de la secuenciación.
- Cuando el problema incluye consideraciones como la Toma de Decisiones distribuida, el dinamismo del entorno, y por lo tanto la evolución del sistema, la teoría de Agentes proporciona una herramienta excelente para abordar el problema a través de los Sistemas Multiagente.
- Los autores han combinado los Sistemas Multiagente o basado en Agentes con técnicas procedentes de la investigación operativa

- Se destaca la existencia mínima de propuestas que combinen los Algoritmo Bio-Inspirados y los Sistemas Multiagente.
- Parece interesante profundizar en el problema de Programación de Producción, con metodologías Metaheurísticas bio-inspiradas basada en Sistemas Multiagente.

Además de estas aportaciones concretas la revisión de la literatura publicada ha resultado de interés para tomar como referencia diversos enfoques, y técnicas entre las que se destaca la publicación del profesor Zhong, (Zhong, W. C., Liu, J., Xue, M. Z., & Jiao, L. C. 2004), a partir de la cual se desarrolla parte de la propuesta de solución.

11.2.3 Definición del Problema

El análisis del estado del arte dejó las puertas abierta para continuar el proceso investigador en base al análisis preliminar del caso expuesto. En esta etapa se ordenaron y concretaron los datos existentes del problema, tal y como, este se contempla en el momento de iniciar la investigación (“as-is”).

En esta etapa se trabajó en tres aspectos:

- Se establecieron las partes de la que se compone el problema, que como ya se identificó preliminarmente era amplio y complejo. Se estableció un problema de Gestión del Programa Productivo y un Problema de Cálculo de los Programas Predictivos y Reactivos.
- Se definió el problema de Gestión del Programa Productivo y se confirmó que este se origina por la ausencia de metodología y procesos para su ejecución. El modelado del problema se realizó de forma literaria y se subrayó la gran diferencia entre el estado actual o “as-is” y el deseado o “to-be”.
- Se definió el problema de cálculo del Programa Predictivo y del Programa Reactivo de la Producción. Se empleó un lenguaje verbal y de modelado matemático para precisarlo. Así mismo, se consideró la posibilidad de emplear un enfoque Fuzzy, y se aportaron los correspondientes modelos fuzzy de ambos procesos.

El problema que se propuso implicaba la necesidad de realizar un seguimiento continuo del Estado del Programa, Valorar diversas situaciones, y Tomar Decisiones sobre la posibilidad de realizar ciertos cambios, y todo ello sin perder de vista el objetivo asociado a la Eficiencia del

Proceso. Se trata por lo tanto de un proceso complejo, donde se relacionan enfoques cuantitativos con otros cualitativos, y difícil de resolver con éxito.

Una vez, concretado el problema, se concluyó con que es casi imposible encontrar casos en los que ambos se hayan combinado en el mismo problema el enfoque Predictivo y Reactivo en la manera descrita. Por lo tanto, el problema identificado es novedoso tanto por la doble combinación, en el ámbito de la Toma de Decisiones (gestión y cálculo) y en el ámbito de los tipos de Problemas de cálculo de la Programación de la Producción detectados, y además se enmarca en un entorno en el que no se conocen planteamientos similares: La industria cerámica.

11.3 Enfoque para el Desarrollo de la Propuesta

Una vez planteado el problema se inició el proceso de generación de alternativas para la resolución del mismo. Después de un amplio periodo de trabajo en este sentido se detectó la necesidad de estructurar los resultados que se estaban obteniendo, ya que se detectaron ciertas dificultades para construir propuestas consistentes.

En este momento de cierta desorientación personal se introdujo un capítulo dedicado a definir un enfoque que se adecuara al tipo de propuesta con el que se trabajaba mentalmente pero que al no estar escrita complicaba en gran medida su desarrollo. En este aspecto el desarrollo de un enfoque para la elaboración de la propuesta ha supuesto una de las acciones más relevantes y que más han contribuido a la mejora del proceso investigador.

Como consecuencia de la necesidad comentada se identificó un enfoque que parece muy adecuado al caso planteado. Bajo la denominación M.A.T. se combinaron formando un solo conjunto, una Metodología, unos elementos Arquitectónicos y Tecnologías que se concretaron y adaptaron para estructurar y facilitar el desarrollo de una propuesta de solución al problema identificado.

El enfoque que se realizó se basa en una metodología de 5 pasos, apoyada en cada uno de ellos, en múltiples elementos arquitectónicos y una amplia gama de herramientas necesarias para llevar a cabo cada uno de los mismos.

11.4 La Propuesta

La propuesta se basa en el desarrollo del enfoque comentado, que en este apartado se resumen en dos escenarios: su desarrollo y su prueba de contraste. El desarrollo de la propuesta se alcanzó mediante el despliegue de los 4 primeros pasos de la metodología, siendo el último paso el que se corresponde con la prueba de funcionamiento.

11.4.1 Análisis, Diseño e Implementación

El Análisis, Diseño e Implementación de la propuesta constituye el núcleo del trabajo investigador, y como es natural es en este periodo donde más aportaciones se realizan.

Durante el análisis se trató de definir cual debe ser el estado final del problema, es decir el modelo “to-be”, mediante el análisis de los requerimientos del proceso de Gestión del Programa Productivo. En este proceso se aporta una exhaustiva definición del caso seleccionado aplicando el Modelo CIMOSA. Para la obtención del Modelo de Requerimientos final se empleó tanto el conocimiento sobre la situación actual, como el conocimiento de las prácticas de las empresas del sector, que permitían definir un horizonte alcanzable. El modelo obtenido supuso un hito para el desarrollo de este trabajo, y una de las pocas aportaciones conocidas de CIMOSA al proceso de Programación de la Producción. En él se aportan, entre otros elementos, la definición de los Procesos de Negocio, las Capacidades y la Información básicos del problema expuesto.

Por la complejidad y amplitud el diseño de la solución se planteó en tres etapas: Diseño de la Plataforma software IPSU-MAS, diseño de algoritmos de cálculo del Programa Predictivo de la Producción y diseño de algoritmos de cálculo del Programa Reactivo de la Producción.

El diseño de la Plataforma IPSU-MAS sirve, además de para alcanzar esquema de la solución que se propone, para poner de manifiesto la forma de enfocar la solución a un problema basado en un Sistema Multiagente. En su diseño han cobrado una enorme relevancia los aspectos “sociales” de los elementos para integración de los Agentes. Para el diseño de la Plataforma se empleó la metodología INGENIAS que permite relacionar de forma ordenada el entorno, los recursos, los objetivos, la definición de los Agentes y su interacción. La referencia de todo este proceso han sido los requisitos definidos con CIMOSA. Una aportación adicional ha sido la forma de integrar CIMOSA e INGENIAS entre sí, así como dentro de la metodología propia.

El problema cálculo de la Programación de la Producción con un enfoque Predictivo se planteó dentro del Proceso de Negocio de Programación Predictiva definido con CIMOSA y en el entorno del Taller de Flujo Híbrido con Tiempos de Cambio de Partida Dependientes de la Secuencia. En esta etapa se definió un nuevo Algoritmo hibridado entre un Algoritmo Genético y un Sistemas Multiagente. Con el objetivo de diferenciar la aportación de los Sistemas Multiagente, se generaron dos algoritmos diferenciados, SMAGA-1 y SMAGA-2. El algoritmo SMAGA-1 basa su diseño en la tecnología Multiagente pero no incorpora el conjunto de características propias de los Agentes en “sociedad”, y que por lo tanto, no aprovechan una condición que les es propia, “el trabajo en equipo”. Sin embargo, SMAGA-2 si incorpora características que le permiten sacar un mayor partido al trabajo conjunto en la sociedad. Para ambos algoritmo se realizó un proceso de ajuste paramétrico destacando como con SMAGA-1 se alcanzó un nuevo valor mínimo del makespan en 106 ejemplares, y con SMAGA-2 en 41 ejemplares de un total de 240 del juego de datos empleado. En el posterior análisis comparativo se comprobó como los algoritmos SMAGA-1 y SMAGA-2, junto con un tercer algoritmo, se encuentran entre los más competitivos para este tipo de problemas.

El problema cálculo de la Programación de la Producción con un enfoque Reactivo se planteó dentro del Proceso de Negocio de Programación Reactiva definido con CIMOSA y definió para el mismo tipo de taller que el Predictivo. Esta etapa contempló como objetivo la selección de métodos de reprogramación parcial y completa de Programas Productivos entre los más empleados en la escasa literatura existente, y el análisis del comportamiento de los mismos en base al variaciones de diversos factores relativos al entorno (tipo de programa en curso, tipo de evento, y configuración del taller), así como otros factores relativos a los propios algoritmos seleccionados. Entre los algoritmos empleados se incluyen aportaciones propias como la metaheurística SMAGA-2 y la heurística SSDS. El trabajo de análisis comparativo permitió establecer y comprender relaciones interesantes entre los factores estudiados. Se puede considerar que los resultados obtenidos en la experimentación son relevantes, así como la heurística SSDS propuesta.

Finalmente la implementación no fue más que la materialización del esfuerzo de análisis y diseño en un código ejecutable que permite automatizar los procesos previamente definidos. Esta tarea supuso un gran esfuerzo y mucho tiempo de dedicación que concluyó con una aplicación capaz de proponer Programas Productivos y Reactivos mediante la simulación de eventos, y que podría ser transformada en un prototipo comercial con un poco de trabajo adicional.

11.4.2 Prueba de Funcionamiento

Una prueba de funcionamiento de conjunto de todos los procesos, algoritmos y métodos implementados que permitió contrastar la eficacia y la eficiencia de la propuesta constituyó el último paso de la metodología. Fue necesario un ajuste de la Plataforma IPSU-MAS para adaptar el funcionamiento de la misma al caso de una Empresa real. Se desarrollaron algunos de los complementos que aún no se habían implementado y que son fundamentales para completar los procesos de Ayuda a la Toma de Decisión propuestos. También se establecieron los valores de los factores de los algoritmos Predictivos y Reactivos.

Los resultados obtenidos, demostraron que con la propuesta realizada siempre se alcanzan Tasas de Productividad mejores que con el método actual empleado por la Empresa Seleccionada para el contraste. Los tres índices de referencia (Incremento de la Tasa de Productividad, Rendimiento y número de trabajos cancelados) fueron favorables a la propuesta. El Incremento de la Tasa de Productividad con respecto a las Empresa osciló aproximadamente entre el 14,2% y 29,8%. El Rendimiento del Programa final frente al previsto también fue mejor con la propuesta en todos los casos menos en uno, y el número de trabajos cancelado por periodo fue menor en la mayoría de los casos en que se empleó la Plataforma IPSU-MAS.

11.5 Trabajo Futuro

El trabajo que queda pendiente de desarrollar es muy amplio. A continuación se exponen algunas de las líneas de trabajo que se consideran más interesantes:

En cuanto al problema propuesto se puede avanzar en dos sentidos:

- Por lo que se refiere al ámbito de procesos que se desea abordar, sería interesante dar un paso más, y afrontar como un proceso integrado al conjunto de actividades que se desarrollan desde la definición del Plan Maestro hasta el establecimiento del Programa Productivo.
 - En esta línea de trabajo sería de gran interés establecer un meta-modelo que posteriormente se pudiera concretar para diferentes modelos productivos.
 - Considerar dentro de este trabajo el desarrollo de diferentes propuestas relativas a la Toma de Decisiones Coordinadas. Se trata de dotar al sistema de un autentico método de coordinación entre el Planificador y el Programador.

- Expandir la propuesta al caso de una Cadena de Suministro/Distribución Colaborativa. Aprovechando el conocimiento adquirido se podría expandir el trabajo.
- En relación a las condiciones que determinan los aspectos más cuantitativos del problema sería interesante:
 - Que el tamaño de lote pueda ajustarse o definirse, según el tipo de sistema productivo, en el momento de establecer el Programa Productivo.
 - Que los trabajos puedan estar siendo operados en 2 o más recursos simultáneamente. Este aspecto, al menos en el caso de la Industria Cerámica, aproximaría el problema a un escenario más real.

En relación a los modelos, métodos y algoritmos empleados se sugiere seguir trabajando en la línea planteada. Una vez más el trabajo futuro que se propone se comenta en 3 bloques: Plataforma, Algoritmos Predictivos, y Algoritmos Reactivos.

La Plataformas IPSU-MAS no es más que un prototipo con una funcionalidad limitada. En este sentido la Plataforma se puede seguir mejorando, tanto mediante su transformación a un software comercial, como en la ampliación de sus funciones, sobre todo en lo referente a las actividades del Planificador. Otra función que se debe mejorar en la Plataforma es la relacionada con la capacidad de aprendizaje. Por lo que se refiere a las metodologías empleadas para su análisis, diseño e implementación, es decir CIMOSA e INGENIAS, se ha observado que podría ser útil mejorar la integración entre ambas, y complementarlas con una Ontología que glose el ámbito terminológico que se aborda.

En la línea relacionada con los aspectos cuantitativos del trabajo realizado existe un amplio campo para investigar tanto con el enfoque Predictivo como Reactivo. Un aspecto que sería importante replantearse, y que es común a ambos tipos de Programa, es el relativo a la ampliación de la investigación variando los indicadores de eficiencia que han definido las funciones objetivo de cada problema.

Los resultados alcanzados con los algoritmos Predictivos han resultado ser bastante buenos en términos de eficiencia, sin embargo se considera que tanto el algoritmo propuesto, SMAGA, como los factores analizados se pueden reconsiderar y con ellos mejorar aún más la eficiencia del mismo. Se debe entender que en este sentido es de especial interés intentar buscar algoritmos auto configurables, y en los que la "personalidad" de los Agentes se puede diferenciar aún más, entre otras cosas por el empleo de heurísticas diferentes, para iniciar soluciones o aplicadas a los operadores de cruce, mutación o aprendizaje. Sin duda, para

seguir trabajando en esta línea será necesario resolver el problema que se ha planteado con los elevados tiempos de ejecución de los algoritmos.

Los resultados obtenidos con los algoritmos Reactivos han sido también buenos en cuanto a que han facilitado el entendimiento del proceso. Una vez comprendida la dinámica del proceso se cree interesante trabajar en métodos más ajustados al tipo de sistema productivo cerámico, como demuestra el éxito de la heurística SSDS propuesta, así como abordar el caso de otros sistemas productivos similares. En este campo, y debido a la influencia que ha demostrado tener, parece interesante investigar sobre el proceso de degradación de los Programas Productivos en curso y vincular dichos resultados a las políticas de reprogramación.

Finalmente, comentar la posibilidad de generalizar el enfoque empleado (metodología, Arquitectura y Herramientas) para el desarrollo de la propuesta con el fin de abordar soluciones similares en la que se combinen aspectos cuantitativos y cualitativos y se soporten en un componente software.

11.6 Bibliografía

Zhong, W. C., Liu, J., Xue, M. Z., & Jiao, L. C. 2004. *A multiagent genetic algorithm for global numerical optimization. Ieee Transactions on Systems Man and Cybernetics Part B-Cybernetics*, 34(2): 1128-1141.

**Definición de los Elementos
Arquitectónicos de CIMOSA
Implementados para Modelado de
Requerimientos del Problema
Propuesto**

Anexo



1. Dominio (Domain):

a. **DM-Programación de la Producción**

- **Identificador:** DM-Programación de la Producción.
- **Nombre:** Programación de la Producción.
- **Descripción:** Es responsable de generar un programa inicial o predictivo en el cual se secuencian y temporizan los diferentes trabajos incluidos en un periodo determinado, indicando el recurso sobre el cual van a realizarse las diferentes operaciones, así como la fecha de inicio y fin de la misma. Así mismo es responsable de reaccionar ante determinados eventos y modificar el programa base en función de los mismos, en lo que se denomina programación reactiva.
- **Objetivos del dominio:**
 - Obtener un programa inicial o predictivo de la producción para un periodo determinado.
 - Obtener un programa reactivo de la producción en función de los eventos percibidos.
- **Restricciones del dominio:**
 - No existirá la posibilidad de que el mismo trabajo esté siendo procesado a la vez en diferentes operaciones.
 - No existirá la posibilidad de fraccionamiento de los trabajos en varios recursos. En caso de reprogramación, Si un trabajo en curso no se puede finalizar por un fallo del recurso sobre el que se estaba realizando, el trabajo se volverá a generar desde el principio.
 - En el momento en que un lote se empiece a procesar en un recurso no se podrá interrumpir la operación hasta no haber terminado completamente.
 - Existen fechas más tardías de finalización de los trabajos que se suponen conocidas al inicio del programa de producción.
 - En caso de reprogramación, los trabajos del programa de producción base o predictivo en ejecución (los que no han acabado y no están en cola de espera) deben seguir ejecutándose en el mismo recurso y condiciones en las que estaban antes de la reprogramación, si el recurso está disponible.
 - En función del Plan Maestro de Producción se definen todos los trabajos a realizar en cada periodo (programa predictivo), es decir éstos son conocidos antes de realizar el programa. En caso de reprogramación los trabajos a Programar son obtenidos a partir del Programa de Producción en curso. Se considera que no existen trabajos cancelados (esta circunstancia es casi inexistente y cuando se da el trabajo se realiza y se almacena), y que no

existen nuevos trabajos (el planificador siempre puede forzar un evento de reinicio de la programación).

Aclaraciones sobre las restricciones:

- Entre cada una de las etapa del sistema existe un almacén de capacidad ilimitada.
 - En caso de Programación Predictiva no existen restricciones de materias primas puesto que este problema ya se supone resuelto al nivel de planificación de producción. Y en caso de Programación Reactiva es mejor no generar necesidades de materia prima inesperada en lo que se ha denominado Ventana de Avance.
 - Desde el punto de vista de la secuenciación no existen restricciones de mano de obra ya se supone resuelto al nivel de planificación de producción.
 - Los tiempos de proceso y cambio de partida se suponen conocidos con anterioridad. Aunque no con exactitud. Presentan incertidumbre.
 - En la secuenciación se contempla la posibilidad de que algunos recursos no se encuentren disponibles, existiendo turnos de trabajo, paradas por mantenimiento o averías conocidos con anterioridad.
- **Dominio Cimosa: Sí.**
 - **Procesos del Dominio: DP-Gestión del Programa Productivo.**
 - **Dominios relacionado:**
 - DM-Planificación de la producción (establece el plan maestro y condiciona la planificación de necesidades).
 - DM-Diseño de procesos/productos (define las operaciones para cada producto).
 - DM-Control de las operaciones (realiza el seguimiento de las operaciones).
 - **Vistas de objetos:**
 - OV-No disponibilidad por mantenimiento máquina.
 - OV-Programa producción en curso.
 - OV-Programa producción sin validar.
 - OV-Programa producción validado.
 - OV-Ocupación programada máquina.
 - OV-Capacidad disponible almacén de productos.
 - OV-Tiempos de cambio de partida.

- OV-Tiempos de proceso.
- OV-Trabajos pendientes.
- OV-Trabajos Seleccionados.
- OV-Operaciones trabajos.
- OV-Máquinas factibles por trabajo/operación.
- OV-Parámetros del programa.
- OV-Inconsistencias.
- OV-Reglas de Inferencia.
- OV-Eventos Recibidos.
- **Eventos:**
 - Recibidos en este dominio:
 - EV-Nuevo periodo.
 - EV-Reinicio programa.
 - EV-Desplazamiento.
 - Generados en este dominio:
 - EV-Inconsistencia de datos.
 - EV-Reducción de Trabajos.

2. Eventos (Event):

a. EV-Nuevo periodo.

- **Identificador:** EV-Nuevo periodo.
- **Nombre:** Inicio nuevo periodo.
- **Descripción:** Activa el proceso de programación predictiva de la producción con el objeto de generar un nuevo programa base.
- **Procedencia:** Externa. Reloj.
- **Lanzadores:** DM-Programación de la Producción.
- **Object Views:** Ninguno.
- **Condiciones:** Ninguna.
- **Tiempo de aparición:** Es un evento programado o temporizado que se activará al inicio de cada periodo de programación.

b. EV-Reinicio programa.

- **Identificador:** EV-Reinicio programa.
- **Nombre:** Reinicio programa.
- **Descripción:** Activa el proceso de programación predictiva de la producción con el objeto de generar un nuevo programa base.
- **Procedencia:** Externa.
- **Lanzadores:** DM-Planificación de la producción.
- **Object Views:** Ninguno.
- **Condiciones:** Ninguna.
- **Tiempo de aparición:** No sujeto a temporización. Es una excepción.

c. EV-Desplazamiento.

- **Identificador:** EV-Desplazamiento.
- **Nombre:** Desplazamiento
- **Descripción:** Evento asociado a una disfunción en el taller. Implica la detección de un desplazamiento a izquierdas o derechas del programa en curso. Activa el proceso de programación reactiva de la producción con el objeto de reconstruir el programa base.
- **Procedencia:** Externa.
- **Lanzadores:** DM-Control de operaciones.
- **Object Views:** Ninguno.
- **Condiciones:** Ninguna.
- **Tiempo de aparición:** No sujeto a temporización. Es una excepción.

d. EV-Inconsistencia de datos

- **Identificador:** EV-Inconsistencia de datos.
- **Nombre:** Inconsistencia de datos.

- **Descripción:** Evento asociado a una ausencia o incoherencia de algún dato necesario para la programación o reprogramación de la producción. Activa desde el BP-Programación Predictiva o desde el BP- Programación Reactiva. Implica la cancelación del proceso de programación o reprogramación y una revisión de los datos inconsistentes. Va dirigido al DM-Planificación de la Producción.
- **Procedencia:** Interna. EA-Revisión y carga de parámetros y datos.
- **Lanzadores:** DP-Programación de la Producción.
- **Object Views:** OV-Inconsistencias.
- **Condiciones:** Controladas por programa.
- **Tiempo de aparición:** No sujeto a temporización. Es una excepción.

e. **EV-Reducción de Trabajos**

- **Identificador:** EV-Reducción de Trabajos.
- **Nombre:** Reducción de Trabajos.
- **Descripción:** Evento asociado a la imposibilidad de respetar las restricciones del modelo para realizar el programa de la producción. Activa desde el BP-Programación Predictiva o desde el BP- Programación Reactiva. Implica una revisión de los datos inconsistentes. Va dirigido al DM-Planificación de la Producción.
- **Procedencia:** Interna.
 - EA-Ajuste.
- **Lanzadores:** DP-Programación de la Producción.
- **Object Views:** Ninguno.
- **Condiciones:** Controladas por programa.
- **Tiempo de aparición:** No sujeto a temporización. Es una excepción.

3. **Procesos de Dominio (Domain Process):**

a. **DP- Gestión del Programa Productivo**

- **Identificador:** DP-Gestión del Programa Productivo.
- **Nombre:** Gestión del Programa Productivo.
- **Descripción:** Es responsable de generar un programa base o preventivo en el cual se secuencian y temporizan los diferentes trabajos incluidos en un periodo determinado,

indicando el recurso sobre el cual van a realizarse las diferentes operaciones, así como la fecha de inicio y fin de la misma. Así mismo es responsable de reaccionar ante determinados eventos y modificar el programa base en función de los mismos, en lo que se denomina programación reactiva.

- **Objetivos:**

- Obtener un programa base o predictivo de la producción para un periodo determinado.
- Obtener un programa reactivo de la producción en función de los eventos recibidos.

- **Restricciones:**

- No existirá la posibilidad de que el mismo trabajo esté siendo procesado a la vez en diferentes operaciones.
- No existirá la posibilidad de fraccionamiento de los trabajos en varios recursos. En caso de reprogramación, Si un trabajo en curso no se puede finalizar por un fallo del recurso sobre el que se estaba realizando, el trabajo se volverá a generar desde el principio.
- En el momento en que un lote se empiece a procesar en un recurso no se podrá interrumpir la operación hasta no haber terminado completamente.
- Existen fechas más tardías de finalización de los trabajos que se suponen conocidas al inicio del programa de producción.
- En caso de reprogramación, los trabajos del programa de producción base o predictivo pendientes de realizarse en un recurso disponible y con una fecha de inicio prevista menor a una ventana temporal V respecto de la fecha de reprogramación no podrán ser modificados en lo que se refiere al recurso y al avance de la fecha de inicio.
- A partir del Plan Maestro de Producción se definen todos los trabajos a realizar en cada periodo (programa predictivo), es decir éstos son conocidos antes de realizar el programa. En caso de reprogramación los trabajos a Programar son obtenidos a partir del Programa de Producción en curso. Se considera que no existen trabajos cancelados (esta circunstancia es casi inexistente y cuando se da el trabajo se realiza y se almacena), y que no existen nuevos trabajos (el planificador siempre puede forzar un evento de reinicio de la programación).

Aclaraciones sobre las restricciones:

- Entre cada etapa del sistema existe un almacén de capacidad ilimitada.
- No existen restricciones de materias primas puesto que este problema ya se supone resuelto al nivel de planificación de producción.

- Desde el punto de vista de la secuenciación no existen restricciones de mano de obra ya se supone resuelto al nivel de planificación de producción.
- Los tiempos de proceso y cambio de partida se suponen conocidos con anterioridad.
- En la secuenciación se contempla la posibilidad de que algunos recursos no se encuentren disponibles, existiendo turnos de trabajo, paradas por mantenimiento o averías conocidos con anterioridad.
- **Reglas:** BP-Programación Predictiva: Si el programa ajustado es aceptado o rechazado.
- **Eventos de lanzamiento:**
 - Recibidos en este dominio:
 - EV-Nuevo periodo.
 - EV-Reinicio programa.
 - EV-Desplazamiento.
- **Comportamiento del Proceso:** Ver diagrama de actividades del proceso.
- **Elementos comprendidos:**
 - BP-Programación Predictiva.
 - BP-Programación Reactiva.

4. Procesos de Negocio (Business Process):

a. **BP-Programación Predictiva**

- **Identificador:** BP-Programación Predictiva.
- **Nombre:** Programación Predictiva.
- **Descripción:** Es responsable de generar un programa base o preventivo en el cual se secuencian y temporizan los diferentes trabajos incluidos en un periodo determinado (según el Plan Maestro), indicando el recurso sobre el cual van a realizarse las diferentes operaciones, así como la fecha de inicio y fin de la misma.
- **Objetivos:** Obtener un programa base o predictivo, es decir, asignar, secuenciar y temporizar todos los trabajos pendientes para un periodo de tiempo en condiciones deterministas. Debe maximizar el uso de los recursos cumpliendo las fechas más tardías de finalización de los trabajos.
- **Restricciones:**

- No existirá la posibilidad de que el mismo trabajo esté siendo procesado a la vez en diferentes operaciones.
- No existirá la posibilidad de fraccionamiento de los trabajos en varios recursos.
- En el momento en que un lote se empieza a procesar en un recurso no se podrá interrumpir la operación hasta no haber terminado completamente.
- Existen fechas más tardías de finalización de los trabajos que se suponen conocidas al inicio del programa de producción.

Aclaraciones sobre las restricciones:

- Entre cada etapa del sistema existe un almacén de capacidad ilimitada.
 - No existen restricciones de materias primas puesto que este problema ya se supone resuelto al nivel de planificación de producción.
 - Desde el punto de vista de la secuenciación no existen restricciones de mano de obra ya se supone resuelto al nivel de planificación de producción.
 - Los tiempos de proceso y cambio de partida se suponen conocidos con anterioridad.
 - A partir del Plan Maestro de Producción se definen todos los trabajos a realizar en cada periodo (programa predictivo), es decir éstos son conocidos antes de realizar el programa.
 - En la secuenciación se contempla la posibilidad de que algunos recursos no se encuentren disponibles, existiendo turnos de trabajo, paradas por mantenimiento o averías conocidos con anterioridad.
- **Reglas:** BP-Programación Predictiva: Si el programa ajustado es aceptado o rechazado.
 - **Comportamiento del Proceso:** Ver diagrama de actividades.
 - **Estados de finalización:**
 - Cancelación de programación por inconsistencia de datos.
 - Programación rechazada.
 - Programación validada.
 - **Elementos comprendidos:**
 - EA-Revisión y carga de datos.
 - EA-Cálculo del programa predictivo.
 - EA-Ajustes.
 - EA-Validación/Rechazo.

b. BP-Programación Reactiva.

- **Identificador:** BP-Programación Reactiva.
 - **Nombre:** Programación Reactiva.
 - **Descripción:** Es responsable de reaccionar ante determinados eventos y modificar el programa base en función de los mismos, en lo que se denomina programación reactiva. Para ello debe identificar la perturbación, actualizar el estado del sistema y volver a generar un programa a partir del anterior.
 - **Objetivos:** Reaccionar eficazmente contra imprevistos que puedan afectar a la programación de la producción en el taller. Reprograma la producción reparando el programa base teniendo en cuenta las nuevas condiciones. Debe maximizar la robustez del nuevo programa cumpliendo las fechas más tardías de finalización de los trabajos.
 - **Restricciones:**
 - No existirá la posibilidad de que el mismo trabajo esté siendo procesado a la vez en diferentes operaciones.
 - No existirá la posibilidad de fraccionamiento de los trabajo en varios recursos. Si un trabajo en curso no se puede finalizar por un fallo del recurso sobre el que se estaba realizando, el trabajo se volverá a generar desde el principio.
 - En el momento en que un lote se empiece a procesar en un recurso no se podrá interrumpir la operación hasta no haber terminado completamente.
 - Existen fechas más tardías de finalización de los trabajos que se suponen conocidas al inicio del programa de producción.
 - Los trabajos del programa de producción base o predictivo pendientes de realizarse en un recurso disponible y con una fecha de inicio prevista menor a una ventana temporal **V** respecto de la fecha de reprogramación no podrán ser modificados en lo que se refiere al recurso y al avance de la fecha de inicio.
 - Los trabajos a Programar son obtenidos a partir del Programa de Producción en curso. Se considera que no existen trabajos cancelados (esta circunstancia es casi inexistente y cuando se da el trabajo se realiza y se almacena), y que no existen nuevos trabajos (el planificador siempre puede forzar un evento de reinicio de la programación).
- Aclaraciones sobre las restricciones:**
- Entre cada etapa del sistema existe un almacén de capacidad ilimitada.
 - No existen restricciones de materias primas puesto que este problema ya se supone resuelto al nivel de planificación de producción.

- Desde el punto de vista de la secuenciación no existen restricciones de mano de obra ya se supone resuelto al nivel de planificación de producción.
- Los tiempos de proceso y cambio de partida se suponen conocidos y actualizados con anterioridad.
- En la secuenciación se contempla la posibilidad de que algunos recursos no se encuentren disponibles, existiendo turnos de trabajo, paradas por mantenimiento o averías conocidos y actualizados con anterioridad.
- **Reglas:** BP-Programación Reactiva: Si el programa ajustado es aceptado o rechazado.
- **Comportamiento del Proceso:** Ver diagrama de actividades.
- **Estados de finalización:**
 - Cancelación de programación por inconsistencia de datos.
 - Programación validada.
- **Elementos comprendidos:**
 - EA-Identificación y valoración.
 - EA-Establecimiento de la estrategia de reprogramación
 - EA-Revisión y carga de datos.
 - EA-Cálculo del programa reactivo.
 - EA-Ajustes /Control de la estrategia de reprogramación.
 - EA-Validar/Rechazar.

5. Actividades de Empresa (Enterprise Activities):

Se ha modificado la estructura propuesta CIMOSA para incluir el estado final de la actividad como parte de la definición de la misma.

a. EA-1

- **Identificador:** EA-1.
- **Nombre:** Revisión y carga de datos.
- **Descripción:** Actividad asociada a la obtención y transformación de los datos necesarios.
- **Objetivos:**

- Obtención de datos.
- Transformación de datos.
- **Restricciones:** Conjunto de Trabajos Seleccionados.
- **Estado Final:** “Datos consistentes” o “Datos no consistentes”. En este último caso, genera un evento de Inconsistencia.
- **Reglas:** Si los datos son correctos se procede al cálculo del programa, en caso contrario, se genera un evento de inconsistencia y se cancela la programación.
- **Capacidades requeridas:** CS-Adquisición, transformación y validación de datos.

b. EA-2

- **Identificador:** EA-2.
- **Nombre:** Cálculo del programa predictivo.
- **Descripción:** Actividad asociada a la generación del programa a partir de los datos obtenidos en la actividad anterior.
- **Objetivos:** Obtener un programa de la producción que facilite la buena utilización de los recursos, los optimice, respetando la fecha máxima de finalización.
- **Restricciones:** Ausencia de inconsistencias de datos.
- **Estado Final:** Produce un estado de “Programa Calculado” o un evento de no factibilidad.
- **Reglas:**
 - Una vez calculado el programa se decide si se considera aceptable o si es necesario modificar y ajustar los parámetros de nuevo.
 - Si no se obtiene un programa factible aún, habiendo ajustado los parámetros se envía un programa nulo a la tarea de ajuste, en caso de obtener un programa factible se envía el programa obtenido.
- **Capacidades requeridas:** CS-Procedimientos de programación de la producción.

c. EA-3

- **Identificador:** EA-3.
- **Nombre:** Ajuste.

- **Descripción:** Actividad asociada a la adaptación fina del programa de producción incorporando los conocimientos de un experto humano. Permite ajustar el programa aprovechando los recursos de la mejor forma posible e incorporando o restringiendo trabajos de la forma más adecuada.
- **Objetivos:** Corregir todas las deficiencias referentes a la productividad del programa.
- **Restricciones:** Existencia de un Programa de Producción sin Validar o Existencia de una no Factibilidad Continuada.
- **Estado Final:** Produce un estado de “Programa Aceptado” o “Programa no Aceptado”, en el segundo caso implica un re-cálculo.
- **Reglas:**
 - Si el programa que se acaba de calcular es el mejor de todos los obtenidos hasta ese momento se guarda.
 - Si el programa recién calculado, una vez valorado se considera que puede ser sustituido por otro calculado anteriormente se cambia. El objetivo es volver a un estado anterior si una vez examinadas las posibilidades de un programa se ve que no da más de sí y no mejoro lo anterior.
 - Una vez valorado el programa se puede considerar que la capacidad considerada es muy superior a la ocupación teórica, o que ésta es muy inferior o que está equilibrada.
- **Capacidades requeridas:** CS-Ajuste fino de programas.

d. EA-4

- **Identificador:** EA-4.
- **Nombre:** Validación/Rechazo.
- **Descripción:** Sirve para dar por bueno un programa de producción y que éste se convierta en un nuevo programa activo o rechazar el programa si se cree conveniente.
- **Objetivos:** Poner en producción un programa en estado de producción o descartar un programa no deseable.
- **Restricciones:** Existencia de un Programa de Producción sin Validar.
- **Estado Final:** Termina con un estado de “Programa Validado” o “Programa Rechazado”.

- **Reglas:** Aceptación o rechazo del programa propuesto.
- **Capacidades requeridas:** Ninguna.

e. EA-5

- **Identificador:** EA-5.
- **Nombre:** Identificación y valoración.
- **Descripción:** Distingue las perturbaciones y las clasifica en diferentes tipos de eventos. Así mismo valora el impacto que tiene sobre el programa de producción en curso y consulta la base de datos del conocimiento para determinar la acción correctora a realizar. Al mismo tiempo actualiza los datos que afectan al cálculo del programa de producción.
- **Objetivos:**
 - Determinar acciones correctoras.
 - Recuperar datos actualizados.
- **Restricciones:** Existencia de un evento conocido.
- **Estado Final:** Se termina en un estado de “Programa Valorado”.
- **Reglas:** Ninguna.
- **Capacidades requeridas:** CS-Inferir Acciones

f. EA-6

- **Identificador:** EA-6.
- **Nombre:** Establecimiento de la Estrategia de Reprogramación.
- **Descripción:** Dirigir el proceso de gestión de la programación mediante el ajuste de los parámetros de control de la misma.
- **Objetivos:** Fijar los parámetros relativos al margen operativo, y ajustar por lo tanto la capacidad considerada, y el parámetro K relacionado con la función objetivo de la programación reactiva.
- **Restricciones:** Recomendación sobre la estrategia en forma de parámetros del programa.
- **Estado Final:** Se alcanza el estado de “Estrategia Definida”.

- **Reglas:** Si se considera que hay que reducir los trabajos se debe solicitar al planificador una lista priorizada de los mismo, para decidir siguiendo dicho orden cual se eliminar. Cabe la posibilidad de no seguir estrictamente el orden, en todo caso la lista definitiva de trabajos eliminados se debe aprobar por el planificador.
- **Capacidades requeridas:** Ninguna

g. EA-7

- **Identificador:** EA-7.
- **Nombre:** Ajuste (Control de la Estrategia de Reprogramación).
- **Descripción:** Actividad asociada a la adaptación fina del programa de producción incorporando los conocimientos de un experto humano. EL objetivo es valorar si el programa obtenido es el adecuado o no lo es. En caso de no ser el programa adecuado se procede a ajustar la estrategia de nuevo.
- **Objetivos:** Corregir todas las deficiencias cualitativas del programa.
- **Restricciones:** Definición de la estrategia de reprogramación en forma de parámetros del programa.
- **Estado Final:** Produce un estado de “Programa Aceptado”, “Programa no Aceptado”, “Programa Trivial por Inconsistencia de Datos”, o “Trabajos Seleccionados”.
- **Reglas:**
 - Si se parte de un programa factible, y que no modifica su conjunto de trabajos entonces se mantiene la asignación y secuenciación de las órdenes y sólo se actualizan los datos relativos a las fechas. Esto se puede producir, por ejemplo, con un desplazamiento poco significativo a derechas.
 - Si una vez calculado el programa reactivo, éste es correcto, se procederá a guardar el programa, en cambio si el programa se considera mejorable, se pasará de nuevo al ajuste de parámetros.
- **Capacidades requeridas:** CS-Ajuste fino de programas.

h. EA-8

- **Identificador:** EA-8.
- **Nombre:** Revisión y carga de datos.

- **Descripción:** Actividad asociada a la obtención y transformación de los datos necesarios.
- **Objetivos:**
 - Obtención de datos.
 - Transformación de datos.
- **Restricciones:** Conjunto de Trabajos Seleccionados.
- **Estado Final:** Se alcanza el estado de “Datos consistentes” o “Datos no consistentes”. En este último caso, genera un evento de Inconsistencia.
- **Reglas:** Si los datos son correctos se utilizarán en el resto de actividades, en caso contrario, se genera un evento de inconsistencia.
- **Capacidades requeridas:** CS-Adquisición, transformación y validación de datos.

i. EA-9

- **Identificador:** EA-9.
- **Nombre:** Cálculo del programa reactivo.
- **Descripción:** Actividad asociada a la modificación del programa en curso en función de los eventos detectados y una vez actualizados los datos en la actividad anterior.
- **Objetivos:** Obtener un programa de la producción reactiva que balancee la buena utilización de los recursos, los optimice, respetando la fecha máxima de finalización, y el impacto de los cambios en el programa nuevo con respecto al anterior.
- **Restricciones:** Ausencia de inconsistencias de datos.
- **Estado Final:** Finaliza con un estado de “Programa Validado” o “Programa Rechazado”.
- **Reglas:**
 - Si el cálculo del programa es factible se procede a la realización de ajustes, por otra parte si no podemos realizar el cálculo, se genera un evento de no factibilidad.
 - Si se obtiene un programa factible se debe tomar una decisión entre ajustar parámetros de los algoritmos y re-calcular o aceptar el programa resultante.
- **Capacidades requeridas:** CS-Procedimientos de reprogramación de la producción.

j. **EA-10**

- **Identificador:** EA-10.
- **Nombre:** Validación/Rechazo.
- **Descripción:** Sirve para dar por bueno un programa de producción y que éste se convierta puede valorar, o que se rechaza y se tenga que ajustar parámetros y volver a calcular.
- **Objetivos:** Dar por válido o rechazar un programa en función exclusivamente de la calidad de los algoritmos utilizados.
- **Restricciones:** Existencia de un Programa de Producción sin Validar.
- **Estado Final:**
- **Reglas:** Aceptación o rechazo del programa propuesto.
- **Capacidades requeridas:** Ninguna.

6. **Vistas de Objetos (OBJECT VIEW):**

a. **OV-Programa de Producción en Curso**

- **Identificador:** OV-Programa de Producción en Curso.
- **Nombre:** Programa de Producción en Curso.
- **Descripción:** Contiene el programa de la producción en curso. Es decir los trabajos con especificación de que recursos realizan cada operación y en que fecha se inicia y finaliza, diferenciando los cambios de partida de las operaciones.
- **Objeto del que depende:** EO-Programación de la Producción.
- **Propiedades:** Básicamente debe contener:
 - Identificador de Trabajo.
 - Identificador de operación.
 - Identificador de recurso asignado a operación.
 - Fecha de Inicio del Cambio de partida.
 - Fecha de fin de cambio de partida.
 - Fecha de inicio operación (se asume idéntica a la de fin de cambio de partida).
 - Fecha de Fin de operación.

b. OV-Programa de Producción Validado

- **Identificador:** OV-Programa de Producción Validado.
- **Nombre:** Programa de Producción Validado.
- **Descripción:** Contiene programas de la producción factibles para un determinado cálculo. Ser un programa válido sólo implica que puede ser puesto en ejecución en cualquier instante, pero no lo vaya a ser. Debe contener los trabajos con especificación de que recursos realizan cada operación y en que fecha se inicia y finaliza, diferenciando los cambios de partida de las operaciones.
- **Objeto del que depende:** EO-Programación de la Producción.
- **Propiedades:** Básicamente debe contener:
 - Identificador de Trabajo.
 - Identificador de operación.
 - Identificador de recurso asignado a operación.
 - Fecha de Inicio del Cambio de partida.
 - Fecha de fin de cambio de partida.
 - Fecha de inicio operación (se asume idéntica a la de fin de cambio de partida).
 - Fecha de Fin de operación.

c. OV-Programa de Producción Sin Validar

- **Identificador:** OV-Programa de Producción sin Validar.
- **Nombre:** Programa de Producción sin Validar.
- **Descripción:** Contiene programas de la producción factibles para un determinado cálculo. Ser un programa sin validar implica que está en proceso de mejora y que no puede ser puesto en ejecución. Debe contener los trabajos con especificación de que recursos realizan cada operación y en que fecha se inicia y finaliza, diferenciando los cambios de partida de las operaciones.
- **Objeto del que depende:** EO-Programación de la Producción.
- **Propiedades:** Básicamente debe contener:
 - Identificador de Trabajo.
 - Identificador de operación.

- Identificador de recurso asignado a operación.
- Fecha de Inicio del Cambio de partida.
- Fecha de fin de cambio de partida.
- Fecha de inicio operación (se asume idéntica a la de fin de cambio de partida).
- Fecha de Fin de operación.

d. OV-Capacidad Disponible de Almacén Producto

- **Identificador:** OV-Capacidad Disponible de Almacén Producto.
- **Nombre:** Capacidad Disponible de Almacén Producto.
- **Descripción:** Contiene la capacidad en metros cuadrados que está disponible en cada almacén de producto (final e intermedio).
- **Objeto del que depende:** EO-Inventario, EO-Almacenes.
- **Propiedades:** Básicamente debe contener:
 - Identificador de almacén.
 - Capacidad disponible.

e. OV-Tiempos de Cambio de Partida

- **Identificador:** OV-Tiempos de Cambio de Partida.
- **Nombre:** Tiempos de Cambio de Partida.
- **Descripción:** Contiene la información relativa a los tiempos de cambio de partida para cada producto en función del producto que le preceda y de la máquina en la que opere.
- **Objeto del que depende:** EO-Especificaciones productos-productos máquinas.
- **Propiedades:** Básicamente debe contener:
 - Identificador de producto predecesor.
 - Identificador de producto sucesor.
 - Identificador de máquina.
 - Tiempo cambio previsto.

f. OV-Tiempos de Proceso

- **Identificador:** OV-Tiempos de Proceso.
- **Nombre:** Tiempos de Proceso.
- **Descripción:** Contiene la información relativa a los tiempos de proceso para cada producto en función de la máquina en la que opere.
- **Objeto del que depende:** EO-Especificaciones productos-máquinas.
- **Propiedades:** Básicamente debe contener:
 - Identificador de producto.
 - Identificador de máquina.
 - Tiempo proceso previsto.

g. OV-Operaciones Trabajos

- **Identificador:** OV-Operaciones Trabajos.
- **Nombre:** Operaciones Trabajos.
- **Descripción:** Contiene la información relativa a las operaciones para cada producto.
- **Objeto del que depende:** EO-Especificaciones productos-máquinas.
- **Propiedades:** Básicamente debe contener:
 - Identificador de producto.
 - Identificador de operación.
 - Orden de precedencia.

h. OV-Máquinas Factibles por Trabajo/Operación

- **Identificador:** OV-Máquinas Factibles por Trabajo/Operación.
- **Nombre:** Máquinas Factibles por Trabajo/Operación.
- **Descripción:** Contiene la información relativa a las máquinas que pueden realizar una determinada operación para cada producto.
- **Objeto del que depende:** EO-Especificaciones productos-máquinas.
- **Propiedades:** Básicamente debe contener:
 - Identificador de producto.

- Identificador de operación.
- Identificador de máquina.

i. OV-Trabajos Pendientes

- **Identificador:** OV-Trabajos Pendientes.
- **Nombre:** Trabajos Pendientes.
- **Descripción:** Informa sobre el conjunto de trabajos pendientes de programar para un periodo dado.
- **Objeto del que depende:** EO-Plan Maestro.
- **Propiedades:** Básicamente debe contener:
 - Identificador de producto.

j. OV-Trabajos Seleccionados

- **Identificador:** OV-Trabajos Seleccionados.
- **Nombre:** Trabajos Seleccionados.
- **Descripción:** Informa sobre el conjunto de trabajos seleccionados en el programa en curso.
- **Objeto del que depende:** OV- Trabajos pendientes.
- **Propiedades:** Básicamente debe contener:
 - Identificador de producto.

k. OV-No Disponibilidad por Mantenimiento de Máquina

- **Identificador:** OV-No Disponibilidad por Mantenimiento de Máquina.
- **Nombre:** No Disponibilidad por Mantenimiento de Máquina.
- **Descripción:** Proporciona información sobre los periodos de no disponibilidad de las máquinas por estar previsto un mantenimiento preventivo.
- **Objeto del que depende:** EO-Programa de Mantenimiento.
- **Propiedades:** Básicamente debe contener:

- Identificador de máquina.
- Fecha de inicio del mantenimiento.
- Fecha de fin del mantenimiento.

I. OV-Parámetros del Programa

- **Identificador:** OV-Parámetros del Programa.
- **Nombre:** Parámetros del Programa.
- **Descripción:** Contiene el conjunto de parámetros necesarios para calcular un programa o reajustarlo desde el punto de vista de la gestión del proceso de programación.
- **Objeto del que depende:** EO-Parámetros del Programa.
- **Propiedades:** Básicamente debe contener:
 - Margen Operativo.
 - Valor del parámetro K de la función objetivo del modelo de reprogramación.
 - Tipo de heurística a aplicar.
 - Nº de trabajos deseable en el programa.
 - Lista de trabajos priorizados.

m. OV-Inconsistencias

- **Identificador:** OV-Inconsistencias.
- **Nombre:** Inconsistencias.
- **Descripción:** Contiene una lista con tipos de datos que se han detectado inconsistentes o ausentes en la actividad de revisión y carga de datos y parámetros.
- **Objeto del que depende:** EO-Inconsistencia de datos.
- **Propiedades:** Básicamente debe contener:
 - Identificador del programa
 - Tipo de dato (descripción)
 - Tipo de inconsistencia (descripción)

n. OV-Reglas de Inferencia

- **Identificador:** OV-Reglas de Inferencia.
- **Nombre:** Reglas de Inferencia.
- **Descripción:** Contiene el grado de éxito alcanzado con un determinada algoritmo en función del tipo de evento, operación en la que se ha detectado y tipo de impacto.
- **Objeto del que depende:** EO-Tipos de Impactos, EO-Tipos de Eventos, EO-Tipos de Procedimiento Reprogramación, EO-Operaciones.
- **Propiedades:** Básicamente debe contener:
 - Identificador evento.
 - Identificador del tipo de impacto.
 - Identificador del tipo de procedimiento aplicado.
 - Identificador de la operación.
 - Grado de éxito (numérico).

o. OV-Eventos Recibidos

- **Identificador:** OV-Eventos Recibidos.
- **Nombre:** Eventos Recibidos.
- **Descripción:** Contiene el tipo de evento recibido durante la ejecución de un programa.
- **Objeto del que depende:** EO-Tipos de Eventos.
- **Propiedades:** Básicamente debe contener:
 - Identificador de evento.
 - Descripción.

7. Objetos de Empresa (ENTERPRISE OBJECT):

a. EO-Programas de Producción

- **Identificador:** EO-Programas de Producción.
- **Nombre:** Programas de Producción.
- **Descripción:** Identifica los programas con su periodo y estado.
- **Puede estar vacío:** Si

- **Es parte de:** Ninguno del DM-Programación de la Producción.
- **Propiedades:** Básicamente debe contener:
 - Identificador del programa.
 - Fecha de inicio del periodo.
 - Fecha de fin del periodo.
 - Estado (Validado/No Validado)

b. EO-Almacenes

- **Identificador:** EO-Almacenes.
- **Nombre:** Almacenes.
- **Descripción:** Define los almacenes existentes.
- **Puede estar vacío:** No
- **Es parte de:** Ninguno del DM-Programación de la Producción.
- **Propiedades:** Básicamente debe contener:
 - Identificador del almacén.
 - Descripción

c. EO-Productos

- **Identificador:** EO-Productos.
- **Nombre:** Productos.
- **Descripción:** Identifica los productos del catálogo.
- **Puede estar vacío:** No
- **Es parte de:** Ninguno del DM-Programación de la Producción.
- **Propiedades:** Básicamente debe contener:
 - Identificador del producto.
 - Descripción.
 - Valido (S/N)
 - Familia.
 - Formato

- Tono
- Calibre

d. EO-Máquinas

- **Identificador:** EO-Máquinas.
- **Nombre:** Máquinas.
- **Descripción:** Identifica las máquinas del taller.
- **Puede estar vacío:** No
- **Es parte de:** Ninguno del DM-Programación de la Producción.
- **Propiedades:** Básicamente debe contener:
 - Identificador de máquina.
 - Descripción.

e. EO-Parámetros de Programa

- **Identificador:** EO-Parámetros de Programa.
- **Nombre:** Parámetros de Programa.
- **Descripción:** Contiene el conjunto de parámetros necesarios para calcular un programa o reajustarlo.
- **Puede estar vacío:** Si
- **Es parte de:** Ninguno del DM-Programación de la Producción.
- **Propiedades:** Básicamente debe contener:
 - Fecha de inicio del periodo
 - Fecha del fin del periodo
 - Fecha de fin de horizonte de programación.

f. EO-Programación Producción

- **Identificador:** EO-Programación Producción.
- **Nombre:** Programación Producción.

- **Descripción:** Contiene el programa de la producción en curso. Es decir los trabajos con especificación de que recursos realizan cada operación y en que fecha se inicia y finaliza, diferenciando los cambios de partida de las operaciones.
- **Puede estar vacío:** Si
- **Es parte de:** EO-Programas de Producción, EO-Productos, EO-Máquinas.
- **Propiedades:** Básicamente debe contener:
 - Identificador de programa
 - Identificador de trabajo.
 - Identificador de operación.
 - Identificador de recurso asignado a operación.
 - Fecha de Inicio del cambio de partida.
 - Fecha de fin de cambio de partida.
 - Fecha de inicio operación (se asume idéntica el de fin de cambio de partida).
 - Fecha de fin de operación
 -

g. EO-Inventario

- **Identificador:** EO-Inventario.
- **Nombre:** Inventarios.
- **Descripción:** Contiene el programa de la producción en curso. Es decir los trabajos con especificación de que recursos realizan cada operación y en que fecha se inicia y finaliza, diferenciando los cambios de partida de las operaciones.
- **Puede estar vacío:** Si
- **Es parte de:** EO-Almacenes, EO-Productos.
- **Propiedades:** Básicamente debe contener:
 - Identificador de almacén.
 - Identificador de producto.
 - Cantidad.

h. EO-Operaciones

- **Identificador:** EO-Operaciones.

- **Nombre:** Operaciones.
- **Descripción:** Identifica las operaciones existentes en el taller.
- **Puede estar vacío:** No
- **Es parte de:** Ninguno del DM-Programación de la Producción.
- **Propiedades:** Básicamente debe contener:
 - Identificador de operación.
 - Descripción.

i. **EO-Plan Maestro**

- **Identificador:** EO-Plan Maestro.
- **Nombre:** Plan Maestro.
- **Descripción:** Informa sobre los diferentes periodos previstos que conforman el plan maestro y los productos incluidos en cada uno. Contiene la cantidad y la fecha de finalización prevista.
- **Puede estar vacío:** Si
- **Es parte de:** Ninguno del DM-Programación de la Producción.
- **Propiedades:** Básicamente debe contener:
 - Identificador de Periodo.
 - Fecha de inicio del periodo.
 - Fecha de fin del periodo.
 - Validado (s/n)

j. **EO-Periodos del Plan Maestro**

- **Identificador:** EO-Periodos del Plan Maestro.
- **Nombre:** Periodos del Plan Maestro.
- **Descripción:** Informa sobre los diferentes periodos previstos que conforman el plan maestro y los productos incluidos en cada uno. Contiene la cantidad y la fecha de finalización prevista.
- **Puede estar vacío:** Si

- **Es parte de:** EO-Productos, EO-Plan Maestro.
- **Propiedades:** Básicamente debe contener:
 - Identificador de periodo.
 - Identificador del producto.
 - Cantidad (metros cuadrados).
 - Fecha de finalización.

k. EO-Inconsistencias

- **Identificador:** EO-Inconsistencias.
- **Nombre:** Inconsistencias.
- **Descripción:** Contiene una lista con tipos de datos que se han detectado inconsistentes o ausentes en la actividad de revisión y carga de datos y parámetros.
- **Puede estar vacío:** Si
- **Es parte de:** Ninguno del DM-Programación de la Producción.
- **Propiedades:** Básicamente debe contener:
 - Identificador del programa
 - Tipo de dato (descripción)
 - Tipo de inconsistencia (descripción)

l. EO-Parámetros del Programa

- **Identificador:** EO-Parámetros del Programa.
- **Nombre:** Parámetros del Programa.
- **Descripción:** Contiene el conjunto de parámetros necesarios para calcular un programa o reajustarlo.
- **Puede estar vacío:** No.
- **Es parte de:** ninguno.
- **Propiedades:** Básicamente debe contener:
 - Margen Operativo.
 - Valor del parámetro K de la función objetivo del modelo de reprogramación.

- Tipo de heurística a aplicar.
- Nº de trabajos deseable en el programa.
- Lista de trabajos priorizados.

m. EO-Especificaciones Productos-Productos Máquinas

- **Identificador:** EO-Especificaciones Productos-Productos Máquinas.
- **Nombre:** Especificaciones Productos-Productos Máquinas.
- **Descripción:** Identifica cualquier propiedad que pueda estar vinculada a la relación de precedencia/seguidor entre dos productos en función de una máquina. Se utiliza fundamentalmente para indicar los tiempos de cambio de partida.
- **Puede estar vacío:** No
- **Es parte de:** EO-Productos, EO-Maquinas.
- **Propiedades:** Básicamente debe contener:
 - Identificador de producto.
 - Identificador de producto.
 - Identificador de máquina.
 - Tiempo de cambio de partida.

n. EO-Especificaciones Productos-Máquinas

- **Identificador:** EO-Especificaciones Productos-Máquinas.
- **Nombre:** Especificaciones Productos-Máquinas.
- **Descripción:** Identifica cualquier propiedad que pueda estar vinculada a la relación entre un producto y una máquina. Se utiliza fundamentalmente para indicar los tiempos de proceso, las operaciones que se deben realizar sobre un trabajo y en que orden.
- **Puede estar vacío:** No
- **Es parte de:** EO-Productos, EO-Maquinas.
- **Propiedades:** Básicamente debe contener:
 - Identificador de producto.
 - Identificador de máquina.

- Orden de operación.
- Tiempo de proceso.

o. EO-Programa Mantenimiento

- **Identificador:** EO-Programa Mantenimiento.
- **Nombre:** Programa Mantenimiento.
- **Descripción:** Dispone de la información relativa a la fecha de inicio y fin de cada parada de una máquina por motivos de mantenimiento.
- **Puede estar vacío:** Si
- **Es parte de:** EO-Máquinas.
- **Propiedades:** Básicamente debe contener:
 - Identificador de máquina.
 - Fecha de inicio de la parada.
 - Fecha de fin de la parada.

p. EO-Tipos de Impactos

- **Identificador:** EO-Tipos de Impactos.
- **Nombre:** Tipos de Impactos.
- **Descripción:** Contiene información sobre los tipos de impactos que existen.
- **Puede estar vacío:** No.
- **Es parte de:** Ninguno del DM-Programación de la Producción.
- **Propiedades:** Básicamente debe contener:
 - Identificador de impacto.
 - Descripción.

q. EO-Tipos de Eventos

- **Identificador:** EO-Tipos de Eventos.
- **Nombre:** Tipos de Eventos.

- **Descripción:** Contiene información sobre los tipos de eventos que existen (modificación tiempo cambio partida, modificación tiempo proceso, avería máquina).
- **Puede estar vacío:** No.
- **Es parte de:** Ninguno del DM-Programación de la Producción.
- **Propiedades:** Básicamente debe contener:
 - Identificador de evento.
 - Descripción.

r. EO-Tipos de Procedimientos de Resolución

- **Identificador:** EO-Tipos de Procedimientos de Resolución.
- **Nombre:** Tipos de Procedimientos de Resolución.
- **Descripción:** Contiene información sobre los tipos de algoritmo que se pueden usar para resolver el problema de la reprogramación.
- **Puede estar vacío:** No.
- **Es parte de:** Ninguno del DM-Programación de la Producción.
- **Propiedades:** Básicamente debe contener:
 - Identificador de algoritmo.
 - Descripción.

s. EO- Reglas de Inferencia

- **Identificador:** EO- Reglas de Inferencia.
- **Nombre:** Reglas de Inferencia.
- **Descripción:** Contiene el grado de éxito alcanzado con un determinada algoritmo en función del tipo de evento, operación en la que se ha detectado y tipo de impacto.
- **Puede estar vacío:** No.
- **Es parte de:** EO-Tipos de Impactos, EO-Tipos de Eventos, EO-Tipos de Procedimiento Reprogramación, EO-Operaciones.
- **Propiedades:** Básicamente debe contener:
 - Identificador evento.

- Identificador del tipo de impacto.
- Identificador del tipo de procedimiento aplicado.
- Identificador de la operación.
- Grado de éxito (numérico).

8. Conjunto de Capacidades (Capability Set):

a. CS-Adquisición, transformación y validación de datos

- **Identificador:** CS-Adquisición, transformación y validación de datos.
- **Nombre:** Adquisición, transformación y validación de datos.
- **Descripción:** Se trata de la habilidad de realizar automáticamente la conexión física con las fuentes de datos y extraer la información que sea necesaria en cada ocasión en función del algoritmo de cálculo que se utilice. Al mismo tiempo que obtiene la información debe ser capaz de realizar la interpretación y transformación de los formatos entre origen y destino. En caso de no poder realizar su cometido debe ser capaz de identificar el problema y generar una alarma.

b. CS-Procedimientos de Programación de la Producción

- **Identificador:** CS-Procedimientos de Programación de la Producción.
- **Nombre:** CS-Procedimientos de Programación de la Producción.
- **Descripción:** Es la habilidad de crear un nuevo programa de producción eficiente según criterios de uso de los recursos. Se necesita la aplicación de un procedimiento/algoritmo para obtenerlo.

c. CS-Procedimientos de Reprogramación de la Producción

- **Identificador:** CS-Procedimientos de Reprogramación de la Producción.
- **Nombre:** CS-Procedimientos de Reprogramación de la Producción.
- **Descripción:** Es la habilidad de modificar el programa de producción en curso de forma eficiente balanceando los criterios de uso eficiente de los recursos y minimización de los cambios en el programa propuesto frente al programa en curso. Se necesita la aplicación de un procedimiento/algoritmo para obtenerlo.

d. CS-Ajuste Fino de Programas

- **Identificador:** CS-Ajuste Fino de Programas.
- **Nombre:** CS- Experiencia Ajuste Fino de Programas.
- **Descripción:** Es la habilidad de identificar posibles mejoras en un programa de producción en base a conocimiento y/o experiencias previas. En definitiva se trata de poder incorporar modificaciones sobre las restricciones difíciles de indicar de forma cuantitativa e identificar posibles mejoras del programa propuesto.

e. CS-Capacidad de Inferir Acciones

- **Identificador:** CS-Capacidad de Inferir Acciones.
- **Nombre:** CS- Capacidad de Inferir Acciones.
- **Descripción:** Consiste en la capacidad de poder identificar a reglas de posibles acciones a tomar en relación con el proceso de gestión del programa.

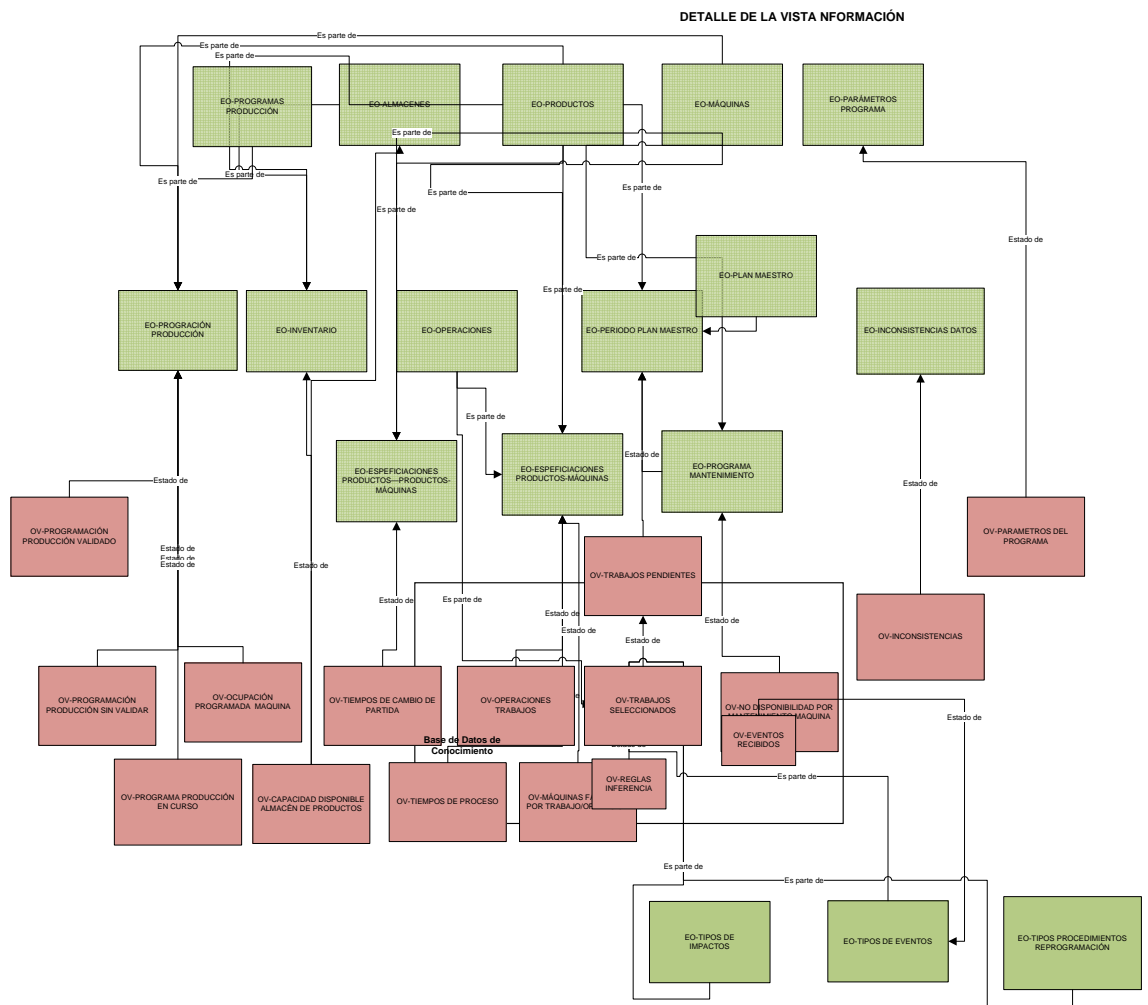


Figura I-1. Vista de Información del Modelo de Requerimientos (elaboración propia).

DETALLE DE LA VISTA INFORMACIÓN RECURSOS EN RELACIÓN EA

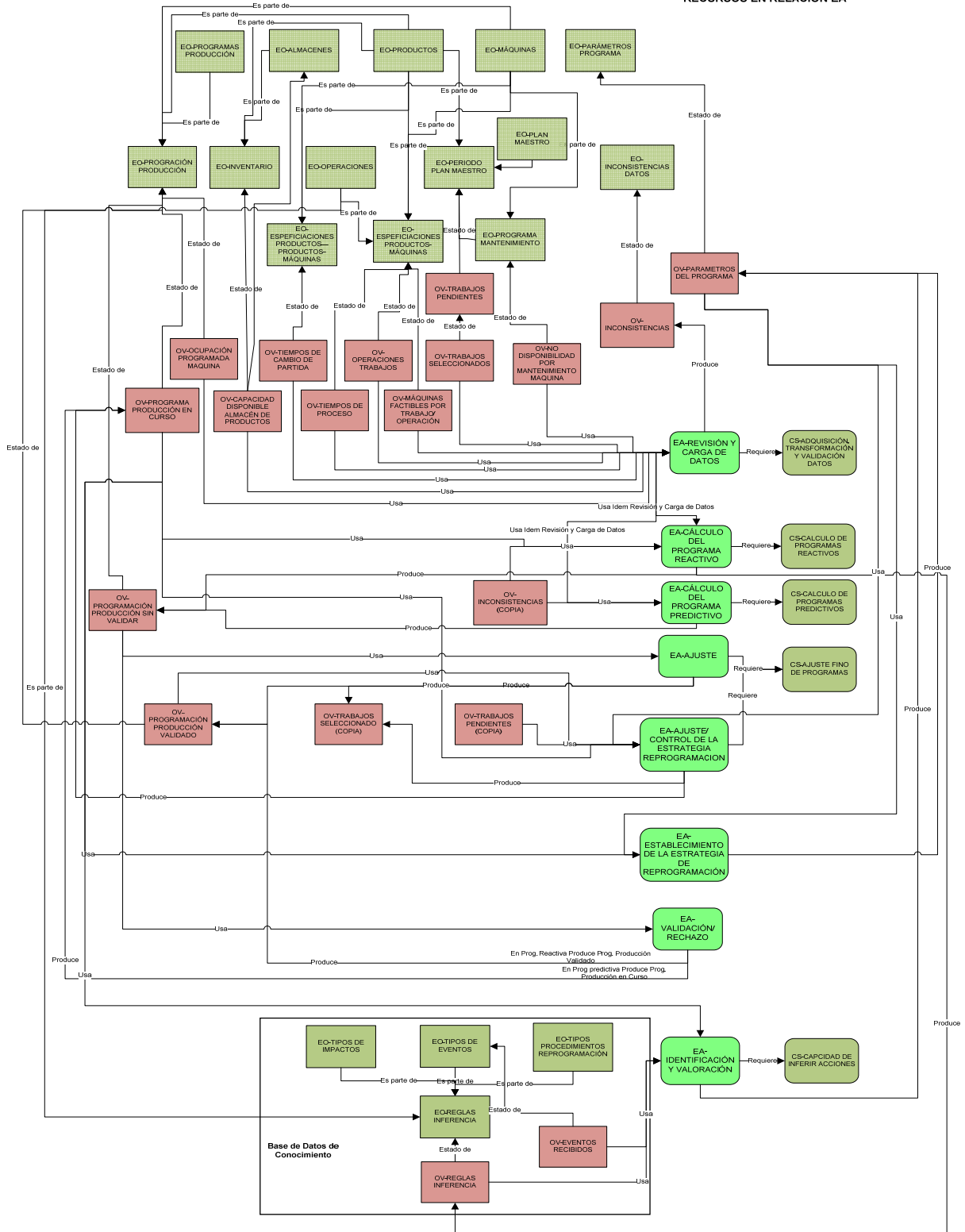


Figura I-2. Relación entre vistas del Modelo de Requerimientos (elaboración propia).

**Límites Conocidos y Nuevos Límites
Alcanzados por SMAGA-1 y SMAGA-2
para los Bancos de Datos Empleados
en la Experimentación**

Anexo



A-II.1 Límites de los conjuntos Taillard para el Problema de Taller de Flujo

Problema	n-m	CI-CS	CI-CS	CI-CS	CI-CS	CI-CS
Taillard 001-005	20-5	1278	1359	1081	1293	1235
Taillard 006-010	20-5	1195	1234	1206	1230	1108
Taillard 011-015	20-10	1582	1659	1496	1377	1419
Taillard 016-020	20-10	1397	1484	1538	1593	1591
Taillard 021-025	20-20	2297	2099	2326	2223	2291
Taillard 026-030	20-20	2226	2273	2200	2237	2178
Taillard 031-035	50-5	2724	2834	2621	2751	2863
Taillard 036-040	50-5	2829	2725	2683	2552	2782
Taillard 041-045	50-10	2991	2867	2839	3063	2976
Taillard 046-050	50-10	3006	3093	3037	2897	3065
Taillard 051-055	50-20	3771- 3847	3668- 3704	3591- 3640	3635- 3719	3553- 3610
Taillard 056-060	50-20	3679	3672- 3704	3627- 3691	3645- 3741	3696- 3756
Taillard 061-065	100-5	5493	5268	5175	5014	5250
Taillard 066-070	100-5	5135	5246	5094	5448	5322
Taillard 071-075	100-10	5770	5349	5676	5781	5467
Taillard 076-080	100-10	5303	5595	5617	5871	5845
Taillard 081-085	100-20	6106- 6202	6183	6252- 6271	6254- 6269	6262- 6314
Taillard 086-090	100-20	6302- 6364	6184- 6268	6315- 6401	6204- 6275	6404- 6434
Taillard 091-095	200-10	10862	10480	10922	10889	10524
Taillard 096-100	200-10	10329	10854	10730	10438	10675
Taillard 101-105	200-20	11152- 11181	11143- 11203	11281	11275	11259
Taillard 106-110	200-20	11176	11337- 11360	11301- 11334	11145- 11192	11284- 11288
Taillard 111-115	500-20	26040- 26059	26500- 26520	26371	26456	26334
Taillard 116-120	500-20	26469- 26477	26389	26560	26005	26457

Tabla II-1. Cotas inferiores (CI) y superiores (CS) de los resultados obtenidos para las instancias de (Taillard, E. 1993)

Instancia	SDST10	SDST50	SDST100	SDST125	Instancia	SDST10	SDST50	SDST100	SDST125
ta001	1330	1567	1891	2065	ta006	1269	1510	1824	1979
ta002	1401	1580	1881	2040	ta007	1294	1531	1855	2002
ta003	1161	1446	1758	1933	ta008	1282	1554	1894	2060
ta004	1370	1644	1973	2137	ta009	1313	1585	1879	2005
ta005	1303	1526	1813	1979	ta010	1178	1426	1732	1876
ta011	1677	2009	2444	2656	ta016	1487	1816	2231	2445
ta012	1751	2065	2458	2661	ta017	1573	1858	2282	2485
ta013	1588	1897	2303	2515	ta018	1630	1962	2381	2586
ta014	1465	1794	2212	2415	ta019	1676	1985	2376	2588
ta015	1510	1842	2282	2502	ta020	1688	2013	2443	2655
ta021	2391	2754	3244	3498	ta026	2321	2686	3168	3442
ta022	2193	2565	3047	3290	ta027	2360	2712	3191	3452
ta023	2414	2748	3207	3475	ta028	2296	2668	3165	3431
ta024	2315	2658	3164	3437	ta029	2335	2701	3192	3456
ta025	2386	2760	3242	3514	ta030	2267	2635	3111	3378
ta031	2814	3250	3893	4226	ta036	2945	3429	4073	4405
ta032	2946	3429	4056	4349	ta037	2848	3338	3999	4327
ta033	2734	3245	3900	4212	ta038	2809	3306	3966	4294
ta034	2883	3391	4020	4356	ta039	2673	3174	3808	4145
ta035	2952	3400	4014	4342	ta040	2867	3350	4022	4341
ta041	3210	3923	4812	5275	ta046	3196	3926	4816	5262
ta042	3080	3807	4714	5177	ta047	3285	3986	4898	5340
ta043	3060	3796	4705	5193	ta048	3222	3950	4849	5317
ta044	3227	3956	4830	5286	ta049	3093	3829	4723	5194
ta045	3200	3939	4812	5236	ta050	3272	3983	4880	5334
ta051	4108	4980	6074	6643	ta056	3930	4791	5920	6494
ta052	3942	4812	5910	6489	ta057	3966	4832	5958	6548
ta053	3895	4781	5908	6502	ta058	3964	4831	5939	6519
ta054	3973	4866	5997	6587	ta059	3996	4864	5948	6539
ta055	3867	4769	5927	6495	ta060	4008	4891	6026	6596
ta061	5647	6542	7714	8339	ta066	5312	6270	7468	8082
ta062	5465	6389	7610	8230	ta067	5459	6390	7611	8267
ta063	5406	6333	7539	8168	ta068	5316	6199	7424	7993
ta064	5213	6182	7421	8005	ta069	5641	6576	7773	8393
ta065	5466	6417	7620	8231	ta070	5537	6492	7735	8290
ta071	6084	7450	9201	10070	ta076	5607	6964	8689	9529
ta072	5683	7033	8794	9631	ta077	5884	7126	8858	9696
ta073	5931	7262	9004	9808	ta078	5958	7290	9028	9891
ta074	6182	7549	9276	10168	ta079	6177	7452	9133	10004
ta075	5842	7240	9002	9852	ta080	6081	7364	9114	10013
ta081	6744	8437	10578	11694	ta086	6867	8548	10679	11740
ta082	6701	8387	10535	11679	ta087	6779	8482	10645	11784
ta083	6770	8422	10552	11701	ta088	6954	8662	10794	11883
ta084	6734	8389	10479	11634	ta089	6808	8473	10612	11731

ta085	6785	8471	10539	11675	ta090	6870	8519	10651	11753
ta091	11354	14005	17307	18930	ta096	10974	13653	17022	18659
ta092	11226	13902	17210	18876	ta097	11423	14115	17428	19118
ta093	11521	14087	17386	19059	ta098	11362	14018	17407	19058
ta094	11294	13873	17206	18934	ta099	11098	13857	17194	18819
ta095	11207	13849	17244	18906	ta100	11284	13894	17263	18793
ta101	12168	15450	19618	21765	ta106	12220	15514	19677	21829
ta102	12278	15644	19816	21973	ta107	12329	15669	19888	22055
ta103	12337	15689	19881	21975	ta108	12362	15645	19826	21902
ta104	12300	15627	19810	21984	ta109	12228	15544	19757	21821
ta105	12196	15470	19589	21773	ta110	12330	15694	19813	21975
ta111	28491	36729	46716	52021	ta116	28756	37006	47074	52269
ta112	28940	37113	47291	52380	ta117	28516	36674	46889	51917
ta113	28695	36854	46964	52110	ta118	28884	36942	47183	52185
ta114	28719	36904	47021	52194	ta119	28398	36575	46636	51746
ta115	28596	36793	46994	51932	ta120	28737	36843	46902	52118

Tabla II-2. Mejores resultados conocidos para el makespan utilizando las instancias de (Vallada, E., Ruiz, R., & Maroto, C. 2003)

A-II.2 Límites de los conjuntos Taillard para el Problema de Taller de Flujo Híbrido con Tiempos de Cambio de Partida dependientes de la Secuencia.

A-II.2.1 Talleres entre una y tres Máquinas por Etapa (Problemas P13): Límites Conocidos.

Instancia	n	m	Cmax	Instancia	n	m	Cmax
ta001_SSD10_P13	20	5	1001	ta002_SSD10_P13	20	5	1029
ta003_SSD10_P13	20	5	1163	ta004_SSD10_P13	20	5	1178
ta005_SSD10_P13	20	5	1343	ta006_SSD10_P13	20	5	1314
ta007_SSD10_P13	20	5	1099	ta008_SSD10_P13	20	5	1074
ta009_SSD10_P13	20	5	1153	ta010_SSD10_P13	20	5	1038
ta011_SSD10_P13	20	10	1482	ta012_SSD10_P13	20	10	1439
ta013_SSD10_P13	20	10	1316	ta014_SSD10_P13	20	10	1326
ta015_SSD10_P13	20	10	1389	ta016_SSD10_P13	20	10	1347
ta017_SSD10_P13	20	10	1435	ta018_SSD10_P13	20	10	1426
ta019_SSD10_P13	20	10	1334	ta020_SSD10_P13	20	10	1316
ta021_SSD10_P13	20	20	1468	ta022_SSD10_P13	20	20	1897
ta023_SSD10_P13	20	20	1761	ta024_SSD10_P13	20	20	1890
ta025_SSD10_P13	20	20	1402	ta026_SSD10_P13	20	20	1849
ta027_SSD10_P13	20	20	1765	ta028_SSD10_P13	20	20	1886
ta029_SSD10_P13	20	20	1889	ta030_SSD10_P13	20	20	1724
ta031_SSD10_P13	50	5	2760	ta032_SSD10_P13	50	5	2593
ta033_SSD10_P13	50	5	2716	ta034_SSD10_P13	50	5	2726
ta035_SSD10_P13	50	5	3075	ta036_SSD10_P13	50	5	2772
ta037_SSD10_P13	50	5	2518	ta038_SSD10_P13	50	5	2692
ta039_SSD10_P13	50	5	2758	ta040_SSD10_P13	50	5	2710
ta041_SSD10_P13	50	10	2786	ta042_SSD10_P13	50	10	3002
ta043_SSD10_P13	50	10	3045	ta044_SSD10_P13	50	10	2992
ta045_SSD10_P13	50	10	2932	ta046_SSD10_P13	50	10	2924
ta047_SSD10_P13	50	10	3003	ta048_SSD10_P13	50	10	3002
ta049_SSD10_P13	50	10	1254	ta050_SSD10_P13	50	10	2998
ta051_SSD10_P13	50	20	3188	ta052_SSD10_P13	50	20	3419
ta053_SSD10_P13	50	20	3435	ta054_SSD10_P13	50	20	3201
ta055_SSD10_P13	50	20	3360	ta056_SSD10_P13	50	20	3383
ta057_SSD10_P13	50	20	3587	ta058_SSD10_P13	50	20	3421
ta059_SSD10_P13	50	20	3378	ta060_SSD10_P13	50	20	3444

ta061_SSD10_P13	100	5	2133	ta062_SSD10_P13	100	5	5285
ta063_SSD10_P13	100	5	2079	ta064_SSD10_P13	100	5	1839
ta065_SSD10_P13	100	5	5263	ta066_SSD10_P13	100	5	5323
ta067_SSD10_P13	100	5	5289	ta068_SSD10_P13	100	5	5062
ta069_SSD10_P13	100	5	5645	ta070_SSD10_P13	100	5	4983
ta071_SSD10_P13	100	10	2427	ta072_SSD10_P13	100	10	5741
ta073_SSD10_P13	100	10	5759	ta074_SSD10_P13	100	10	5749
ta075_SSD10_P13	100	10	2145	ta076_SSD10_P13	100	10	5842
ta077_SSD10_P13	100	10	5682	ta078_SSD10_P13	100	10	5728
ta079_SSD10_P13	100	10	5149	ta080_SSD10_P13	100	10	5718
ta081_SSD10_P13	100	20	5830	ta082_SSD10_P13	100	20	6501
ta083_SSD10_P13	100	20	6226	ta084_SSD10_P13	100	20	5811
ta085_SSD10_P13	100	20	6336	ta086_SSD10_P13	100	20	6206
ta087_SSD10_P13	100	20	5624	ta088_SSD10_P13	100	20	5498
ta089_SSD10_P13	100	20	6167	ta090_SSD10_P13	100	20	6040
ta091_SSD10_P13	200	10	11037	ta092_SSD10_P13	200	10	11337
ta093_SSD10_P13	200	10	10934	ta094_SSD10_P13	200	10	10900
ta095_SSD10_P13	200	10	10864	ta096_SSD10_P13	200	10	10519
ta097_SSD10_P13	200	10	10702	ta098_SSD10_P13	200	10	11069
ta099_SSD10_P13	200	10	10317	ta100_SSD10_P13	200	10	10876
ta101_SSD10_P13	200	20	11355	ta102_SSD10_P13	200	20	11612
ta103_SSD10_P13	200	20	11231	ta104_SSD10_P13	200	20	11511
ta105_SSD10_P13	200	20	10821	ta106_SSD10_P13	200	20	11535
ta107_SSD10_P13	200	20	11772	ta108_SSD10_P13	200	20	11301
ta109_SSD10_P13	200	20	11737	ta110_SSD10_P13	200	20	11621

Tabla II-3. Mejores resultados conocidos para el makespan del grupo de problemas SDD10_P13 propuestos por (Vallada, E. et al. 2003).

Instancia	n	m	Cmax	Instancia	n	m	Cmax
ta001_SSD50_P13	20	5	1065	ta002_SSD50_P13	20	5	1197
ta003_SSD50_P13	20	5	1298	ta004_SSD50_P13	20	5	1247
ta005_SSD50_P13	20	5	1423	ta006_SSD50_P13	20	5	1392
ta007_SSD50_P13	20	5	1220	ta008_SSD50_P13	20	5	1146
ta009_SSD50_P13	20	5	1318	ta010_SSD50_P13	20	5	1177
ta011_SSD50_P13	20	10	1752	ta012_SSD50_P13	20	10	1567
ta013_SSD50_P13	20	10	1570	ta014_SSD50_P13	20	10	1508
ta015_SSD50_P13	20	10	1628	ta016_SSD50_P13	20	10	1515
ta017_SSD50_P13	20	10	1680	ta018_SSD50_P13	20	10	1654
ta019_SSD50_P13	20	10	1647	ta020_SSD50_P13	20	10	1477
ta021_SSD50_P13	20	20	1691	ta022_SSD50_P13	20	20	2199

ta023_SSD50_P13	20	20	2030	ta024_SSD50_P13	20	20	2097
ta025_SSD50_P13	20	20	1615	ta026_SSD50_P13	20	20	2115
ta027_SSD50_P13	20	20	2082	ta028_SSD50_P13	20	20	2138
ta029_SSD50_P13	20	20	2253	ta030_SSD50_P13	20	20	1992
ta031_SSD50_P13	50	5	3093	ta032_SSD50_P13	50	5	2887
ta033_SSD50_P13	50	5	3237	ta034_SSD50_P13	50	5	3049
ta035_SSD50_P13	50	5	3386	ta036_SSD50_P13	50	5	3316
ta037_SSD50_P13	50	5	2697	ta038_SSD50_P13	50	5	2977
ta039_SSD50_P13	50	5	3004	ta040_SSD50_P13	50	5	3026
ta041_SSD50_P13	50	10	3092	ta042_SSD50_P13	50	10	3570
ta043_SSD50_P13	50	10	3645	ta044_SSD50_P13	50	10	3636
ta045_SSD50_P13	50	10	3482	ta046_SSD50_P13	50	10	3469
ta047_SSD50_P13	50	10	3450	ta048_SSD50_P13	50	10	3535
ta049_SSD50_P13	50	10	1667	ta050_SSD50_P13	50	10	3474
ta051_SSD50_P13	50	20	3747	ta052_SSD50_P13	50	20	4030
ta053_SSD50_P13	50	20	4153	ta054_SSD50_P13	50	20	3895
ta055_SSD50_P13	50	20	4060	ta056_SSD50_P13	50	20	4092
ta057_SSD50_P13	50	20	4307	ta058_SSD50_P13	50	20	4101
ta059_SSD50_P13	50	20	4111	ta060_SSD50_P13	50	20	4164
ta061_SSD50_P13	100	5	2966	ta062_SSD50_P13	100	5	5468
ta063_SSD50_P13	100	5	2782	ta064_SSD50_P13	100	5	2330
ta065_SSD50_P13	100	5	5807	ta066_SSD50_P13	100	5	6163
ta067_SSD50_P13	100	5	5861	ta068_SSD50_P13	100	5	5590
ta069_SSD50_P13	100	5	6427	ta070_SSD50_P13	100	5	5478
ta071_SSD50_P13	100	10	3312	ta072_SSD50_P13	100	10	6934
ta073_SSD50_P13	100	10	6363	ta074_SSD50_P13	100	10	6452
ta075_SSD50_P13	100	10	3029	ta076_SSD50_P13	100	10	6902
ta077_SSD50_P13	100	10	6583	ta078_SSD50_P13	100	10	6323
ta079_SSD50_P13	100	10	5371	ta080_SSD50_P13	100	10	6512
ta081_SSD50_P13	100	20	6917	ta082_SSD50_P13	100	20	7483
ta083_SSD50_P13	100	20	7651	ta084_SSD50_P13	100	20	6842
ta085_SSD50_P13	100	20	7636	ta086_SSD50_P13	100	20	7527
ta087_SSD50_P13	100	20	6534	ta088_SSD50_P13	100	20	6133
ta089_SSD50_P13	100	20	7525	ta090_SSD50_P13	100	20	7297
ta091_SSD50_P13	200	10	12113	ta092_SSD50_P13	200	10	13109
ta093_SSD50_P13	200	10	12689	ta094_SSD50_P13	200	10	12991
ta095_SSD50_P13	200	10	11956	ta096_SSD50_P13	200	10	11476
ta097_SSD50_P13	200	10	12135	ta098_SSD50_P13	200	10	13349
ta099_SSD50_P13	200	10	11376	ta100_SSD50_P13	200	10	12225
ta101_SSD50_P13	200	20	13428	ta102_SSD50_P13	200	20	14618
ta103_SSD50_P13	200	20	13330	ta104_SSD50_P13	200	20	14142

ta105_SSD50_P13	200	20	12376	ta106_SSD50_P13	200	20	13426
ta107_SSD50_P13	200	20	13956	ta108_SSD50_P13	200	20	13582
ta109_SSD50_P13	200	20	14664	ta110_SSD50_P13	200	20	14547

Tabla II-4. Mejores resultados conocidos para el makespan del grupo de problemas SDD50_P13 propuestos por (Vallada, E. et al. 2003)

Instancia	n	m	Cmax	Instancia	n	m	Cmax
ta001_SSD100_P13	20	5	1171	ta002_SSD100_P13	20	5	1410
ta003_SSD100_P13	20	5	1502	ta004_SSD100_P13	20	5	1340
ta005_SSD100_P13	20	5	1574	ta006_SSD100_P13	20	5	1557
ta007_SSD100_P13	20	5	1424	ta008_SSD100_P13	20	5	1232
ta009_SSD100_P13	20	5	1519	ta010_SSD100_P13	20	5	1335
ta011_SSD100_P13	20	10	2066	ta012_SSD100_P13	20	10	1818
ta013_SSD100_P13	20	10	1891	ta014_SSD100_P13	20	10	1704
ta015_SSD100_P13	20	10	1987	ta016_SSD100_P13	20	10	1804
ta017_SSD100_P13	20	10	2000	ta018_SSD100_P13	20	10	1967
ta019_SSD100_P13	20	10	1906	ta020_SSD100_P13	20	10	1677
ta021_SSD100_P13	20	20	1960	ta022_SSD100_P13	20	20	2571
ta023_SSD100_P13	20	20	2363	ta024_SSD100_P13	20	20	2471
ta025_SSD100_P13	20	20	1883	ta026_SSD100_P13	20	20	2486
ta027_SSD100_P13	20	20	2472	ta028_SSD100_P13	20	20	2490
ta029_SSD100_P13	20	20	2692	ta030_SSD100_P13	20	20	2353
ta031_SSD100_P13	50	5	3488	ta032_SSD100_P13	50	5	3274
ta033_SSD100_P13	50	5	3887	ta034_SSD100_P13	50	5	3420
ta035_SSD100_P13	50	5	3949	ta036_SSD100_P13	50	5	3925
ta037_SSD100_P13	50	5	2838	ta038_SSD100_P13	50	5	3410
ta039_SSD100_P13	50	5	3411	ta040_SSD100_P13	50	5	3480
ta041_SSD100_P13	50	10	3509	ta042_SSD100_P13	50	10	4334
ta043_SSD100_P13	50	10	4512	ta044_SSD100_P13	50	10	4400
ta045_SSD100_P13	50	10	4180	ta046_SSD100_P13	50	10	4224
ta047_SSD100_P13	50	10	4018	ta048_SSD100_P13	50	10	4248
ta049_SSD100_P13	50	10	2187	ta050_SSD100_P13	50	10	4062
ta051_SSD100_P13	50	20	4470	ta052_SSD100_P13	50	20	4910
ta053_SSD100_P13	50	20	5040	ta054_SSD100_P13	50	20	4726
ta055_SSD100_P13	50	20	4919	ta056_SSD100_P13	50	20	5122
ta057_SSD100_P13	50	20	5297	ta058_SSD100_P13	50	20	5118
ta059_SSD100_P13	50	20	5002	ta060_SSD100_P13	50	20	5049
ta061_SSD100_P13	100	5	3744	ta062_SSD100_P13	100	5	5745
ta063_SSD100_P13	100	5	3710	ta064_SSD100_P13	100	5	2819
ta065_SSD100_P13	100	5	6604	ta066_SSD100_P13	100	5	7071
ta067_SSD100_P13	100	5	6463	ta068_SSD100_P13	100	5	6314

ta069_SSD100_P13	100	5	7417	ta070_SSD100_P13	100	5	6219
ta071_SSD100_P13	100	10	4521	ta072_SSD100_P13	100	10	8518
ta073_SSD100_P13	100	10	7463	ta074_SSD100_P13	100	10	7482
ta075_SSD100_P13	100	10	3840	ta076_SSD100_P13	100	10	8331
ta077_SSD100_P13	100	10	7812	ta078_SSD100_P13	100	10	7075
ta079_SSD100_P13	100	10	5704	ta080_SSD100_P13	100	10	7567
ta081_SSD100_P13	100	20	8373	ta082_SSD100_P13	100	20	9200
ta083_SSD100_P13	100	20	9530	ta084_SSD100_P13	100	20	8375
ta085_SSD100_P13	100	20	9305	ta086_SSD100_P13	100	20	9291
ta087_SSD100_P13	100	20	7706	ta088_SSD100_P13	100	20	7005
ta089_SSD100_P13	100	20	9192	ta090_SSD100_P13	100	20	8973
ta091_SSD100_P13	200	10	13326	ta092_SSD100_P13	200	10	15702
ta093_SSD100_P13	200	10	15038	ta094_SSD100_P13	200	10	15869
ta095_SSD100_P13	200	10	13322	ta096_SSD100_P13	200	10	12697
ta097_SSD100_P13	200	10	14024	ta098_SSD100_P13	200	10	16111
ta099_SSD100_P13	200	10	12668	ta100_SSD100_P13	200	10	14058
ta101_SSD100_P13	200	20	15866	ta102_SSD100_P13	200	20	18262
ta103_SSD100_P13	200	20	16106	ta104_SSD100_P13	200	20	17385
ta105_SSD100_P13	200	20	14527	ta106_SSD100_P13	200	20	15863
ta107_SSD100_P13	200	20	16586	ta108_SSD100_P13	200	20	16452
ta109_SSD100_P13	200	20	18231	ta110_SSD100_P13	200	20	17866

Tabla II-5. Mejores resultados conocidos para el makespan del grupo de problemas SSD100_P13 propuestos por (Vallada, E. et al. 2003)

Instancia	n	m	Cmax	Instancia	n	m	Cmax
ta001_SSD125_P13	20	5	1203	ta002_SSD125_P13	20	5	1497
ta003_SSD125_P13	20	5	1604	ta004_SSD125_P13	20	5	1396
ta005_SSD125_P13	20	5	1657	ta006_SSD125_P13	20	5	1635
ta007_SSD125_P13	20	5	1490	ta008_SSD125_P13	20	5	1294
ta009_SSD125_P13	20	5	1630	ta010_SSD125_P13	20	5	1409
ta011_SSD125_P13	20	10	2252	ta012_SSD125_P13	20	10	1983
ta013_SSD125_P13	20	10	2017	ta014_SSD125_P13	20	10	1790
ta015_SSD125_P13	20	10	2145	ta016_SSD125_P13	20	10	1945
ta017_SSD125_P13	20	10	2101	ta018_SSD125_P13	20	10	2128
ta019_SSD125_P13	20	10	2085	ta020_SSD125_P13	20	10	1795
ta021_SSD125_P13	20	20	2137	ta022_SSD125_P13	20	20	2750
ta023_SSD125_P13	20	20	2560	ta024_SSD125_P13	20	20	2652
ta025_SSD125_P13	20	20	2040	ta026_SSD125_P13	20	20	2668
ta027_SSD125_P13	20	20	2628	ta028_SSD125_P13	20	20	2699
ta029_SSD125_P13	20	20	2953	ta030_SSD125_P13	20	20	2521

ta031_SSD125_P13	50	5	3778	ta032_SSD125_P13	50	5	3447
ta033_SSD125_P13	50	5	4272	ta034_SSD125_P13	50	5	3693
ta035_SSD125_P13	50	5	4172	ta036_SSD125_P13	50	5	4312
ta037_SSD125_P13	50	5	2912	ta038_SSD125_P13	50	5	3647
ta039_SSD125_P13	50	5	3655	ta040_SSD125_P13	50	5	3722
ta041_SSD125_P13	50	10	3837	ta042_SSD125_P13	50	10	4612
ta043_SSD125_P13	50	10	5068	ta044_SSD125_P13	50	10	4784
ta045_SSD125_P13	50	10	4554	ta046_SSD125_P13	50	10	4489
ta047_SSD125_P13	50	10	4328	ta048_SSD125_P13	50	10	4627
ta049_SSD125_P13	50	10	2374	ta050_SSD125_P13	50	10	4354
ta051_SSD125_P13	50	20	4746	ta052_SSD125_P13	50	20	5379
ta053_SSD125_P13	50	20	5516	ta054_SSD125_P13	50	20	5248
ta055_SSD125_P13	50	20	5318	ta056_SSD125_P13	50	20	5534
ta057_SSD125_P13	50	20	5722	ta058_SSD125_P13	50	20	5512
ta059_SSD125_P13	50	20	5352	ta060_SSD125_P13	50	20	5594
ta061_SSD125_P13	100	5	4109	ta062_SSD125_P13	100	5	5915
ta063_SSD125_P13	100	5	4038	ta064_SSD125_P13	100	5	3066
ta065_SSD125_P13	100	5	6913	ta066_SSD125_P13	100	5	7632
ta067_SSD125_P13	100	5	6815	ta068_SSD125_P13	100	5	6736
ta069_SSD125_P13	100	5	7850	ta070_SSD125_P13	100	5	6628
ta071_SSD125_P13	100	10	4905	ta072_SSD125_P13	100	10	9418
ta073_SSD125_P13	100	10	7957	ta074_SSD125_P13	100	10	8138
ta075_SSD125_P13	100	10	4378	ta076_SSD125_P13	100	10	9318
ta077_SSD125_P13	100	10	8648	ta078_SSD125_P13	100	10	7445
ta079_SSD125_P13	100	10	5896	ta080_SSD125_P13	100	10	7977
ta081_SSD125_P13	100	20	9156	ta082_SSD125_P13	100	20	9958
ta083_SSD125_P13	100	20	10311	ta084_SSD125_P13	100	20	9081
ta085_SSD125_P13	100	20	10221	ta086_SSD125_P13	100	20	10336
ta087_SSD125_P13	100	20	8230	ta088_SSD125_P13	100	20	7544
ta089_SSD125_P13	100	20	10045	ta090_SSD125_P13	100	20	10005
ta091_SSD125_P13	200	10	14183	ta092_SSD125_P13	200	10	16779
ta093_SSD125_P13	200	10	16468	ta094_SSD125_P13	200	10	17119
ta095_SSD125_P13	200	10	14048	ta096_SSD125_P13	200	10	13419
ta097_SSD125_P13	200	10	15154	ta098_SSD125_P13	200	10	17596
ta099_SSD125_P13	200	10	13388	ta100_SSD125_P13	200	10	14897
ta101_SSD125_P13	200	20	16874	ta102_SSD125_P13	200	20	20018
ta103_SSD125_P13	200	20	17470	ta104_SSD125_P13	200	20	18983
ta105_SSD125_P13	200	20	15402	ta106_SSD125_P13	200	20	17064
ta107_SSD125_P13	200	20	17878	ta108_SSD125_P13	200	20	17653
ta109_SSD125_P13	200	20	20035	ta110_SSD125_P13	200	20	19981

Tabla II-6. Mejores resultados conocidos para el makespan del grupo de problemas SDD125_P13 propuestos por (Vallada, E. et al. 2003)

**A-II.2.2 Talleres entre una y tres Máquinas por Etapa (Problemas P13):
 Límites Nuevos Alcanzados con SMAGA-1 y SMAGA-2.**

En la columna Cmax_Ref se indica el mejor Cmax conocido para la instancia indicada. Los valores IPSO negativos, es decir cuyos Cmax son inferiores a los conocidos, se resaltan con color rojo. La negrilla se emplea para destacar el mejor resultado entre los que suponen una mejora.

Instancia	n	m	Cmax_Ref	IPSOVEP SMAGA-1	Cmax SMAGA-1	IPSOVEP SMAGA-2	Cmax SMAGA-2
ta002_SSD10_P13	20	5	1029	-0,0971	1028	0	1029
ta004_SSD10_P13	20	5	1178	0	1178	0	1178
ta006_SSD10_P13	20	5	1314	0	1314	0	1314
ta008_SSD10_P13	20	5	1074	-0,6517	1067	-0,5586	1068
ta010_SSD10_P13	20	5	1038	-0,3853	1034	-0,3853	1034
ta012_SSD10_P13	20	10	1439	0	1439	0	1439
ta014_SSD10_P13	20	10	1326	-0,0754	1325	0,07541	1327
ta016_SSD10_P13	20	10	1347	-0,2227	1344	-0,0742	1346
ta018_SSD10_P13	20	10	1426	-0,6311	1417	-0,6311	1417
ta020_SSD10_P13	20	10	1316	0	1316	-0,0759	1315
ta022_SSD10_P13	20	20	1897	-0,4217	1889	0	1897
ta024_SSD10_P13	20	20	1890	-0,6878	1877	-0,7407	1876
ta026_SSD10_P13	20	20	1849	-0,2163	1845	-0,054	1848
ta028_SSD10_P13	20	20	1886	-1,1134	1865	-0,5302	1876
ta030_SSD10_P13	20	20	1724	-0,986	1707	-0,696	1712
ta032_SSD10_P13	50	5	2593	0,03856	2594	0,15426	2597
ta034_SSD10_P13	50	5	2726	-0,2934	2718	-0,1834	2721
ta036_SSD10_P13	50	5	2772	0,03607	2773	0,03607	2773
ta038_SSD10_P13	50	5	2692	0	2692	0,22288	2698
ta040_SSD10_P13	50	5	2710	-0,2583	2703	-0,369	2700
ta042_SSD10_P13	50	10	3002	-0,0333	3001	0,23317	3009
ta044_SSD10_P13	50	10	2992	-1,0026	2962	-0,7018	2971
ta046_SSD10_P13	50	10	2924	-0,4445	2911	0,37619	2935
ta048_SSD10_P13	50	10	3002	0,03331	3003	0,23317	3009
ta050_SSD10_P13	50	10	2998	-0,0333	2997	-0,1334	2994
ta052_SSD10_P13	50	20	3419	-1,4039	3371	-1,2576	3376
ta054_SSD10_P13	50	20	3201	0,71852	3224	0,37488	3213
ta056_SSD10_P13	50	20	3383	-0,1477	3378	0,56163	3402

ta058_SSD10_P13	50	20	3421	0,58462	3441	0,02923	3422
ta060_SSD10_P13	50	20	3444	0,26132	3453	0,92915	3476

Tabla II-7. Mejores resultados alcanzados para el makespan del grupo de problemas SDD10_P13 propuestos por (Vallada, E. et al. 2003).

Instancia	n	m	Cmax_Ref	IPSOVEP SMAGA-1	Cmax SMAGA-1	IPSOVEP SMAGA-2	Cmax SMAGA-2
ta002_SSD50_P13	20	5	1197	0	1197	0	1197
ta004_SSD50_P13	20	5	1247	-0,1603	1245	0,08019	1248
ta006_SSD50_P13	20	5	1392	0	1392	0,35919	1397
ta008_SSD50_P13	20	5	1146	0	1146	0,17452	1148
ta010_SSD50_P13	20	5	1177	-0,1699	1175	0	1177
ta012_SSD50_P13	20	10	1567	0	1567	0	1567
ta014_SSD50_P13	20	10	1508	0	1508	0	1508
ta016_SSD50_P13	20	10	1515	0	1515	0	1515
ta018_SSD50_P13	20	10	1654	0	1654	0	1654
ta020_SSD50_P13	20	10	1477	0	1477	0	1477
ta022_SSD50_P13	20	20	2199	-0,2623	2193	0,7276	2215
ta024_SSD50_P13	20	20	2097	-0,4221	2088	0	2097
ta026_SSD50_P13	20	20	2115	0	2115	0	2115
ta028_SSD50_P13	20	20	2138	-0,0467	2137	0,09354	2140
ta030_SSD50_P13	20	20	1992	-0,7282	1977	-0,1506	1989
ta032_SSD50_P13	50	5	2887	-0,9625	2859	0,41565	2899
ta034_SSD50_P13	50	5	3049	-0,9511	3020	-0,2951	3040
ta036_SSD50_P13	50	5	3316	-0,5428	3298	-0,5126	3299
ta038_SSD50_P13	50	5	2977	-0,2687	2969	0,43668	2990
ta040_SSD50_P13	50	5	3026	0,23132	3033	0,16523	3031
ta042_SSD50_P13	50	10	3570	-1,2044	3527	-1,1764	3528
ta044_SSD50_P13	50	10	3636	-1,1276	3595	-0,22	3628
ta046_SSD50_P13	50	10	3469	-0,1729	3463	-0,1153	3465
ta048_SSD50_P13	50	10	3535	-0,2828	3525	-0,2545	3526
ta050_SSD50_P13	50	10	3474	-0,9786	3440	0	3474
ta052_SSD50_P13	50	20	4030	0,08019	4033	0,19851	4038
ta054_SSD50_P13	50	20	3895	-0,1283	3890	-0,154	3889
ta056_SSD50_P13	50	20	4092	0,12218	4097	0,65982	4119
ta058_SSD50_P13	50	20	4101	-0,1699	4094	0,43891	4119
ta060_SSD50_P13	50	20	4164	0,43227	4182	0,28818	4176

Tabla II-8. Mejores resultados alcanzados para el makespan del grupo de problemas SDD50_P13 propuestos por (Vallada, E. et al. 2003)

Instancia	n	m	Cmax_Ref	IPSOVEP SMAGA-1	Cmax SMAGA-1	IPSOVEP SMAGA-2	Cmax SMAGA-2
ta002_SSD100_P13	20	5	1410	0,14184	1412	0,78014	1421
ta004_SSD100_P13	20	5	1340	-0,1492	1338	1,04477	1354
ta006_SSD100_P13	20	5	1557	0	1557	0	1557
ta008_SSD100_P13	20	5	1232	0	1232	-0,1623	1230
ta010_SSD100_P13	20	5	1335	-0,2247	1332	-0,2247	1332
ta012_SSD100_P13	20	10	1818	0	1818	0	1818
ta014_SSD100_P13	20	10	1704	0	1704	0,23474	1708
ta016_SSD100_P13	20	10	1804	0	1804	1,05321	1823
ta018_SSD100_P13	20	10	1967	0	1967	0,45754	1976
ta020_SSD100_P13	20	10	1677	0	1677	0	1677
ta022_SSD100_P13	20	20	2571	0	2571	0	2571
ta024_SSD100_P13	20	20	2471	0	2471	0,24281	2477
ta026_SSD100_P13	20	20	2486	-0,2011	2481	-0,4827	2474
ta028_SSD100_P13	20	20	2490	0	2490	0	2490
ta030_SSD100_P13	20	20	2353	0	2353	0	2353
ta032_SSD100_P13	50	5	3274	-0,8246	3247	0,03054	3275
ta034_SSD100_P13	50	5	3420	0,76023	3446	-0,4093	3406
ta036_SSD100_P13	50	5	3925	-0,2292	3916	-1,2993	3874
ta038_SSD100_P13	50	5	3410	-0,4692	3394	-0,1173	3406
ta040_SSD100_P13	50	5	3480	-1,7816	3418	-0,1149	3476
ta042_SSD100_P13	50	10	4334	-2,6765	4218	-1,2459	4280
ta044_SSD100_P13	50	10	4400	-1,0909	4352	-1,5	4334
ta046_SSD100_P13	50	10	4224	-2,2253	4130	-1,9649	4141
ta048_SSD100_P13	50	10	4248	-2,8719	4126	-0,8239	4213
ta050_SSD100_P13	50	10	4062	-1,8956	3985	-0,123	4057
ta052_SSD100_P13	50	20	4910	-0,3258	4894	0,50916	4935
ta054_SSD100_P13	50	20	4726	0,06347	4729	-0,1692	4718
ta056_SSD100_P13	50	20	5122	-1,4837	5046	-0,3709	5103
ta058_SSD100_P13	50	20	5118	-1,3091	5051	-0,508	5092
ta060_SSD100_P13	50	20	5049	-0,1584	5041	-0,0594	5046

Tabla II-9. Mejores resultados alcanzados para el makespan del grupo de problemas SDD100_P13 propuestos por (Vallada, E. et al. 2003)

Instancia	n	m	Cmax_Ref	IPSOVEP	Cmax	IPSOVEP	Cmax
				SMAGA-1	SMAGA-1	SMAGA-2	SMAGA-2
ta002_SSD125_P13	20	5	1497	0	1497	1,4696	1519
ta004_SSD125_P13	20	5	1396	-0,7163	1386	-0,5014	1389
ta006_SSD125_P13	20	5	1635	0,97859	1651	0,97859	1651
ta008_SSD125_P13	20	5	1294	-1,0046	1281	-1,0046	1281
ta010_SSD125_P13	20	5	1409	0	1409	0	1409
ta012_SSD125_P13	20	10	1983	0	1983	0,05042	1984
ta014_SSD125_P13	20	10	1790	0	1790	0	1790
ta016_SSD125_P13	20	10	1945	0,92544	1963	0	1945
ta018_SSD125_P13	20	10	2128	0,23496	2133	0,28195	2134
ta020_SSD125_P13	20	10	1795	0	1795	0	1795
ta022_SSD125_P13	20	20	2750	0	2750	0	2750
ta024_SSD125_P13	20	20	2652	0	2652	0,5279	2666
ta026_SSD125_P13	20	20	2668	-0,1124	2665	0,22488	2674
ta028_SSD125_P13	20	20	2699	0	2699	0	2699
ta030_SSD125_P13	20	20	2521	0,43633	2532	0	2521
ta032_SSD125_P13	50	5	3447	-0,058	3445	1,33449	3493
ta034_SSD125_P13	50	5	3693	-2,3016	3608	-0,5415	3673
ta036_SSD125_P13	50	5	4312	-1,1827	4261	-0,8116	4277
ta038_SSD125_P13	50	5	3647	-0,4935	3629	-0,9871	3611
ta040_SSD125_P13	50	5	3722	-1,7463	3657	-0,7791	3693
ta042_SSD125_P13	50	10	4612	-2,9921	4474	-0,4119	4593
ta044_SSD125_P13	50	10	4784	-0,9824	4737	-1,8185	4697
ta046_SSD125_P13	50	10	4489	-0,9578	4446	-0,3564	4473
ta048_SSD125_P13	50	10	4627	-2,5718	4508	-0,6267	4598
ta050_SSD125_P13	50	10	4354	-1,4469	4291	-0,9186	4314
ta052_SSD125_P13	50	20	5379	-0,9295	5329	-0,7808	5337
ta054_SSD125_P13	50	20	5248	-0,381	5228	-0,3239	5231
ta056_SSD125_P13	50	20	5534	-0,9215	5483	-0,8673	5486
ta058_SSD125_P13	50	20	5512	0,3447	5531	0,2177	5524
ta060_SSD125_P13	50	20	5594	-2,1451	5474	-1,3943	5516

Tabla II-10. Mejores resultados alcanzados para el makespan del grupo de problemas SDD125_P13 propuestos por (Vallada, E. et al. 2003)

A-II.2.3 Talleres con tres Máquinas por Etapa (Problemas P3): Límites Conocidos

Instancia	N	m	Cmax	Instancia	n	m	Cmax
ta001_SSD10_P3	20	5	341	ta002_SSD10_P3	20	5	351
ta003_SSD10_P3	20	5	336	ta004_SSD10_P3	20	5	352
ta005_SSD10_P3	20	5	323	ta006_SSD10_P3	20	5	348
ta007_SSD10_P3	20	5	349	ta008_SSD10_P3	20	5	332
ta009_SSD10_P3	20	5	301	ta010_SSD10_P3	20	5	322
ta011_SSD10_P3	20	10	557	ta012_SSD10_P3	20	10	541
ta013_SSD10_P3	20	10	511	ta014_SSD10_P3	20	10	461
ta015_SSD10_P3	20	10	513	ta016_SSD10_P3	20	10	509
ta017_SSD10_P3	20	10	542	ta018_SSD10_P3	20	10	513
ta019_SSD10_P3	20	10	508	ta020_SSD10_P3	20	10	533
ta021_SSD10_P3	20	20	933	ta022_SSD10_P3	20	20	786
ta023_SSD10_P3	20	20	863	ta024_SSD10_P3	20	20	782
ta025_SSD10_P3	20	20	838	ta026_SSD10_P3	20	20	859
ta027_SSD10_P3	20	20	888	ta028_SSD10_P3	20	20	869
ta029_SSD10_P3	20	20	871	ta030_SSD10_P3	20	20	878
ta031_SSD10_P3	50	5	678	ta032_SSD10_P3	50	5	678
ta033_SSD10_P3	50	5	673	ta034_SSD10_P3	50	5	680
ta035_SSD10_P3	50	5	661	ta036_SSD10_P3	50	5	700
ta037_SSD10_P3	50	5	642	ta038_SSD10_P3	50	5	656
ta039_SSD10_P3	50	5	737	ta040_SSD10_P3	50	5	684
ta041_SSD10_P3	50	10	875	ta042_SSD10_P3	50	10	867
ta043_SSD10_P3	50	10	939	ta044_SSD10_P3	50	10	837
ta045_SSD10_P3	50	10	941	ta046_SSD10_P3	50	10	940
ta047_SSD10_P3	50	10	942	ta048_SSD10_P3	50	10	947
ta049_SSD10_P3	50	10	899	ta050_SSD10_P3	50	10	926
ta051_SSD10_P3	50	20	1345	ta052_SSD10_P3	50	20	2168
ta053_SSD10_P3	50	20	1321	ta054_SSD10_P3	50	20	1324
ta055_SSD10_P3	50	20	1290	ta056_SSD10_P3	50	20	1325
ta057_SSD10_P3	50	20	1342	ta058_SSD10_P3	50	20	1284
ta059_SSD10_P3	50	20	1297	ta060_SSD10_P3	50	20	1316
ta061_SSD10_P3	100	5	1289	ta062_SSD10_P3	100	5	1297
ta063_SSD10_P3	100	5	1319	ta064_SSD10_P3	100	5	1263
ta065_SSD10_P3	100	5	1269	ta066_SSD10_P3	100	5	1260
ta067_SSD10_P3	100	5	1284	ta068_SSD10_P3	100	5	1236
ta069_SSD10_P3	100	5	1314	ta070_SSD10_P3	100	5	1252

ta071_SSD10_P3	100	10	1573	ta072_SSD10_P3	100	10	1532
ta073_SSD10_P3	100	10	1616	ta074_SSD10_P3	100	10	1555
ta075_SSD10_P3	100	10	1451	ta076_SSD10_P3	100	10	1554
ta077_SSD10_P3	100	10	1596	ta078_SSD10_P3	100	10	1534
ta079_SSD10_P3	100	10	1548	ta080_SSD10_P3	100	10	1519
ta081_SSD10_P3	100	20	2027	ta082_SSD10_P3	100	20	2044
ta083_SSD10_P3	100	20	2053	ta084_SSD10_P3	100	20	1977
ta085_SSD10_P3	100	20	1990	ta086_SSD10_P3	100	20	1992
ta087_SSD10_P3	100	20	2053	ta088_SSD10_P3	100	20	2032
ta089_SSD10_P3	100	20	2077	ta090_SSD10_P3	100	20	2072
ta091_SSD10_P3	200	10	2734	ta092_SSD10_P3	200	10	2749
ta093_SSD10_P3	200	10	2770	ta094_SSD10_P3	200	10	2790
ta095_SSD10_P3	200	10	2774	ta096_SSD10_P3	200	10	2772
ta097_SSD10_P3	200	10	2781	ta098_SSD10_P3	200	10	2791
ta099_SSD10_P3	200	10	2724	ta100_SSD10_P3	200	10	2850
ta101_SSD10_P3	200	20	3483	ta102_SSD10_P3	200	20	3361
ta103_SSD10_P3	200	20	3460	ta104_SSD10_P3	200	20	3410
ta105_SSD10_P3	200	20	3459	ta106_SSD10_P3	200	20	3461
ta107_SSD10_P3	200	20	3481	ta108_SSD10_P3	200	20	3442
ta109_SSD10_P3	200	20	3408	ta110_SSD10_P3	200	20	3457

Tabla II-11. Mejores resultados conocidos para el makespan del grupo de problemas SDD10_P3 propuestos por (Vallada, E. et al. 2003)

Instancia	N	m	Cmax	Instancia	n	m	Cmax
ta001_SSD50_P3	20	5	446	ta002_SSD50_P3	20	5	459
ta003_SSD50_P3	20	5	440	ta004_SSD50_P3	20	5	461
ta005_SSD50_P3	20	5	431	ta006_SSD50_P3	20	5	440
ta007_SSD50_P3	20	5	470	ta008_SSD50_P3	20	5	429
ta009_SSD50_P3	20	5	434	ta010_SSD50_P3	20	5	412
ta011_SSD50_P3	20	10	673	ta012_SSD50_P3	20	10	652
ta013_SSD50_P3	20	10	640	ta014_SSD50_P3	20	10	580
ta015_SSD50_P3	20	10	648	ta016_SSD50_P3	20	10	637
ta017_SSD50_P3	20	10	675	ta018_SSD50_P3	20	10	621
ta019_SSD50_P3	20	10	665	ta020_SSD50_P3	20	10	646
ta021_SSD50_P3	20	20	1070	ta022_SSD50_P3	20	20	928
ta023_SSD50_P3	20	20	1018	ta024_SSD50_P3	20	20	926
ta025_SSD50_P3	20	20	1004	ta026_SSD50_P3	20	20	999
ta027_SSD50_P3	20	20	1042	ta028_SSD50_P3	20	20	1020
ta029_SSD50_P3	20	20	1020	ta030_SSD50_P3	20	20	1012
ta031_SSD50_P3	50	5	982	ta032_SSD50_P3	50	5	955
ta033_SSD50_P3	50	5	944	ta034_SSD50_P3	50	5	977

ta035_SSD50_P3	50	5	991	ta036_SSD50_P3	50	5	951
ta037_SSD50_P3	50	5	945	ta038_SSD50_P3	50	5	932
ta039_SSD50_P3	50	5	1006	ta040_SSD50_P3	50	5	965
ta041_SSD50_P3	50	10	1215	ta042_SSD50_P3	50	10	1215
ta043_SSD50_P3	50	10	1257	ta044_SSD50_P3	50	10	1174
ta045_SSD50_P3	50	10	1235	ta046_SSD50_P3	50	10	1288
ta047_SSD50_P3	50	10	1353	ta048_SSD50_P3	50	10	1284
ta049_SSD50_P3	50	10	1243	ta050_SSD50_P3	50	10	1261
ta051_SSD50_P3	50	20	1746	ta052_SSD50_P3	50	20	1658
ta053_SSD50_P3	50	20	1715	ta054_SSD50_P3	50	20	1718
ta055_SSD50_P3	50	20	1729	ta056_SSD50_P3	50	20	1713
ta057_SSD50_P3	50	20	1747	ta058_SSD50_P3	50	20	1678
ta059_SSD50_P3	50	20	1714	ta060_SSD50_P3	50	20	1704
ta061_SSD50_P3	100	5	1926	ta062_SSD50_P3	100	5	1903
ta063_SSD50_P3	100	5	1953	ta064_SSD50_P3	100	5	1891
ta065_SSD50_P3	100	5	1838	ta066_SSD50_P3	100	5	1860
ta067_SSD50_P3	100	5	1888	ta068_SSD50_P3	100	5	1856
ta069_SSD50_P3	100	5	1923	ta070_SSD50_P3	100	5	1790
ta071_SSD50_P3	100	10	2336	ta072_SSD50_P3	100	10	2184
ta073_SSD50_P3	100	10	2278	ta074_SSD50_P3	100	10	2214
ta075_SSD50_P3	100	10	2182	ta076_SSD50_P3	100	10	2253
ta077_SSD50_P3	100	10	2272	ta078_SSD50_P3	100	10	2205
ta079_SSD50_P3	100	10	2262	ta080_SSD50_P3	100	10	2222
ta081_SSD50_P3	100	20	2812	ta082_SSD50_P3	100	20	2784
ta083_SSD50_P3	100	20	2832	ta084_SSD50_P3	100	20	2736
ta085_SSD50_P3	100	20	2772	ta086_SSD50_P3	100	20	2789
ta087_SSD50_P3	100	20	2810	ta088_SSD50_P3	100	20	2845
ta089_SSD50_P3	100	20	2916	ta090_SSD50_P3	100	20	2853
ta091_SSD50_P3	200	10	4135	ta092_SSD50_P3	200	10	4148
ta093_SSD50_P3	200	10	4165	ta094_SSD50_P3	200	10	4164
ta095_SSD50_P3	200	10	4180	ta096_SSD50_P3	200	10	4078
ta097_SSD50_P3	200	10	4170	ta098_SSD50_P3	200	10	4135
ta099_SSD50_P3	200	10	4103	ta100_SSD50_P3	200	10	4209
ta101_SSD50_P3	200	20	4918	ta102_SSD50_P3	200	20	4854
ta103_SSD50_P3	200	20	4887	ta104_SSD50_P3	200	20	4895
ta105_SSD50_P3	200	20	4879	ta106_SSD50_P3	200	20	4937
ta107_SSD50_P3	200	20	4944	ta108_SSD50_P3	200	20	4926
ta109_SSD50_P3	200	20	4926	ta110_SSD50_P3	200	20	4874

Tabla II-12. Mejores resultados conocidos para el makespan del grupo de problemas SSD50_P3 propuestos por (Vallada, E. et al. 2003)

Instancia	N	m	Cmax	Instancia	n	m	Cmax
ta001_SSD100_P3	20	5	543	ta002_SSD100_P3	20	5	564
ta003_SSD100_P3	20	5	539	ta004_SSD100_P3	20	5	571
ta005_SSD100_P3	20	5	569	ta006_SSD100_P3	20	5	538
ta007_SSD100_P3	20	5	603	ta008_SSD100_P3	20	5	531
ta009_SSD100_P3	20	5	567	ta010_SSD100_P3	20	5	509
ta011_SSD100_P3	20	10	836	ta012_SSD100_P3	20	10	798
ta013_SSD100_P3	20	10	793	ta014_SSD100_P3	20	10	717
ta015_SSD100_P3	20	10	819	ta016_SSD100_P3	20	10	769
ta017_SSD100_P3	20	10	861	ta018_SSD100_P3	20	10	760
ta019_SSD100_P3	20	10	794	ta020_SSD100_P3	20	10	794
ta021_SSD100_P3	20	20	1239	ta022_SSD100_P3	20	20	1113
ta023_SSD100_P3	20	20	1207	ta024_SSD100_P3	20	20	1112
ta025_SSD100_P3	20	20	1181	ta026_SSD100_P3	20	20	1184
ta027_SSD100_P3	20	20	1234	ta028_SSD100_P3	20	20	1204
ta029_SSD100_P3	20	20	1200	ta030_SSD100_P3	20	20	1191
ta031_SSD100_P3	50	5	1296	ta032_SSD100_P3	50	5	1256
ta033_SSD100_P3	50	5	1282	ta034_SSD100_P3	50	5	1275
ta035_SSD100_P3	50	5	1283	ta036_SSD100_P3	50	5	1248
ta037_SSD100_P3	50	5	1254	ta038_SSD100_P3	50	5	1248
ta039_SSD100_P3	50	5	1322	ta040_SSD100_P3	50	5	1248
ta041_SSD100_P3	50	10	1552	ta042_SSD100_P3	50	10	1555
ta043_SSD100_P3	50	10	1616	ta044_SSD100_P3	50	10	1570
ta045_SSD100_P3	50	10	1636	ta046_SSD100_P3	50	10	1658
ta047_SSD100_P3	50	10	1685	ta048_SSD100_P3	50	10	1614
ta049_SSD100_P3	50	10	1639	ta050_SSD100_P3	50	10	1612
ta051_SSD100_P3	50	20	2236	ta052_SSD100_P3	50	20	2106
ta053_SSD100_P3	50	20	2170	ta054_SSD100_P3	50	20	2145
ta055_SSD100_P3	50	20	2166	ta056_SSD100_P3	50	20	2172
ta057_SSD100_P3	50	20	2251	ta058_SSD100_P3	50	20	2100
ta059_SSD100_P3	50	20	2158	ta060_SSD100_P3	50	20	2159
ta061_SSD100_P3	100	5	2526	ta062_SSD100_P3	100	5	2518
ta063_SSD100_P3	100	5	2529	ta064_SSD100_P3	100	5	2531
ta065_SSD100_P3	100	5	2459	ta066_SSD100_P3	100	5	2494
ta067_SSD100_P3	100	5	2545	ta068_SSD100_P3	100	5	2388
ta069_SSD100_P3	100	5	2492	ta070_SSD100_P3	100	5	2499
ta071_SSD100_P3	100	10	3027	ta072_SSD100_P3	100	10	2945
ta073_SSD100_P3	100	10	3002	ta074_SSD100_P3	100	10	2981
ta075_SSD100_P3	100	10	2883	ta076_SSD100_P3	100	10	2993
ta077_SSD100_P3	100	10	3024	ta078_SSD100_P3	100	10	3005

ta079_SSD100_P3	100	10	3008	ta080_SSD100_P3	100	10	2965
ta081_SSD100_P3	100	20	3697	ta082_SSD100_P3	100	20	3706
ta083_SSD100_P3	100	20	3737	ta084_SSD100_P3	100	20	3660
ta085_SSD100_P3	100	20	3615	ta086_SSD100_P3	100	20	3666
ta087_SSD100_P3	100	20	3702	ta088_SSD100_P3	100	20	3703
ta089_SSD100_P3	100	20	3757	ta090_SSD100_P3	100	20	3701
ta091_SSD100_P3	200	10	5552	ta092_SSD100_P3	200	10	5599
ta093_SSD100_P3	200	10	5583	ta094_SSD100_P3	200	10	5507
ta095_SSD100_P3	200	10	5730	ta096_SSD100_P3	200	10	5466
ta097_SSD100_P3	200	10	5609	ta098_SSD100_P3	200	10	5526
ta099_SSD100_P3	200	10	5550	ta100_SSD100_P3	200	10	5580
ta101_SSD100_P3	200	20	6522	ta102_SSD100_P3	200	20	6465
ta103_SSD100_P3	200	20	6492	ta104_SSD100_P3	200	20	6504
ta105_SSD100_P3	200	20	6596	ta106_SSD100_P3	200	20	6473
ta107_SSD100_P3	200	20	6609	ta108_SSD100_P3	200	20	6526
ta109_SSD100_P3	200	20	6589	ta110_SSD100_P3	200	20	6531

Tabla II-13. Mejores resultados conocidos para el makespan del grupo de problemas SDD100_P3 propuestos por (Vallada, E. et al. 2003)

Instancia	N	m	Cmax	Instancia	n	m	Cmax
ta001_SSD125_P3	20	5	600	ta002_SSD125_P3	20	5	609
ta003_SSD125_P3	20	5	617	ta004_SSD125_P3	20	5	610
ta005_SSD125_P3	20	5	605	ta006_SSD125_P3	20	5	588
ta007_SSD125_P3	20	5	647	ta008_SSD125_P3	20	5	586
ta009_SSD125_P3	20	5	592	ta010_SSD125_P3	20	5	555
ta011_SSD125_P3	20	10	900	ta012_SSD125_P3	20	10	858
ta013_SSD125_P3	20	10	880	ta014_SSD125_P3	20	10	782
ta015_SSD125_P3	20	10	866	ta016_SSD125_P3	20	10	827
ta017_SSD125_P3	20	10	911	ta018_SSD125_P3	20	10	831
ta019_SSD125_P3	20	10	880	ta020_SSD125_P3	20	10	861
ta021_SSD125_P3	20	20	1309	ta022_SSD125_P3	20	20	1206
ta023_SSD125_P3	20	20	1320	ta024_SSD125_P3	20	20	1209
ta025_SSD125_P3	20	20	1314	ta026_SSD125_P3	20	20	1264
ta027_SSD125_P3	20	20	1345	ta028_SSD125_P3	20	20	1293
ta029_SSD125_P3	20	20	1310	ta030_SSD125_P3	20	20	1269
ta031_SSD125_P3	50	5	1426	ta032_SSD125_P3	50	5	1362
ta033_SSD125_P3	50	5	1370	ta034_SSD125_P3	50	5	1368
ta035_SSD125_P3	50	5	1409	ta036_SSD125_P3	50	5	1372
ta037_SSD125_P3	50	5	1399	ta038_SSD125_P3	50	5	1387

ta039_SSD125_P3	50	5	1443	ta040_SSD125_P3	50	5	1370
ta041_SSD125_P3	50	10	1830	ta042_SSD125_P3	50	10	1732
ta043_SSD125_P3	50	10	1732	ta044_SSD125_P3	50	10	1732
ta045_SSD125_P3	50	10	1772	ta046_SSD125_P3	50	10	1801
ta047_SSD125_P3	50	10	1858	ta048_SSD125_P3	50	10	1778
ta049_SSD125_P3	50	10	1832	ta050_SSD125_P3	50	10	1760
ta051_SSD125_P3	50	20	2424	ta052_SSD125_P3	50	20	2331
ta053_SSD125_P3	50	20	2344	ta054_SSD125_P3	50	20	2352
ta055_SSD125_P3	50	20	2469	ta056_SSD125_P3	50	20	2373
ta057_SSD125_P3	50	20	2445	ta058_SSD125_P3	50	20	2307
ta059_SSD125_P3	50	20	2398	ta060_SSD125_P3	50	20	2370
ta061_SSD125_P3	100	5	2824	ta062_SSD125_P3	100	5	2778
ta063_SSD125_P3	100	5	2781	ta064_SSD125_P3	100	5	2780
ta065_SSD125_P3	100	5	2802	ta066_SSD125_P3	100	5	2721
ta067_SSD125_P3	100	5	2845	ta068_SSD125_P3	100	5	2694
ta069_SSD125_P3	100	5	2885	ta070_SSD125_P3	100	5	2649
ta071_SSD125_P3	100	10	3318	ta072_SSD125_P3	100	10	3198
ta073_SSD125_P3	100	10	3325	ta074_SSD125_P3	100	10	3324
ta075_SSD125_P3	100	10	3235	ta076_SSD125_P3	100	10	3307
ta077_SSD125_P3	100	10	3292	ta078_SSD125_P3	100	10	3266
ta079_SSD125_P3	100	10	3387	ta080_SSD125_P3	100	10	3322
ta081_SSD125_P3	100	20	4102	ta082_SSD125_P3	100	20	4137
ta083_SSD125_P3	100	20	4154	ta084_SSD125_P3	100	20	4081
ta085_SSD125_P3	100	20	4064	ta086_SSD125_P3	100	20	4035
ta087_SSD125_P3	100	20	4098	ta088_SSD125_P3	100	20	4106
ta089_SSD125_P3	100	20	4190	ta090_SSD125_P3	100	20	4115
ta091_SSD125_P3	200	10	6300	ta092_SSD125_P3	200	10	6266
ta093_SSD125_P3	200	10	6310	ta094_SSD125_P3	200	10	6356
ta095_SSD125_P3	200	10	6291	ta096_SSD125_P3	200	10	6172
ta097_SSD125_P3	200	10	6395	ta098_SSD125_P3	200	10	6134
ta099_SSD125_P3	200	10	6247	ta100_SSD125_P3	200	10	6306
ta101_SSD125_P3	200	20	7243	ta102_SSD125_P3	200	20	7232
ta103_SSD125_P3	200	20	7265	ta104_SSD125_P3	200	20	7233
ta105_SSD125_P3	200	20	7197	ta106_SSD125_P3	200	20	7306
ta107_SSD125_P3	200	20	7380	ta108_SSD125_P3	200	20	7188
ta109_SSD125_P3	200	20	7267	ta110_SSD125_P3	200	20	7238

Tabla II-14. Mejores resultados conocidos para el makespan del grupo de problemas SSD125_P3 propuestos por (Vallada, E. et al. 2003)

A-II.2.4 Talleres entre una y tres Máquinas por Etapa (Problemas P13): Límites Nuevos Alcanzados con SMAGA-1 y SMAGA-2.

En la columna Cmax_Ref se indica el mejor Cmax conocido para la instancia indicada. Los valores IPSO negativos, es decir cuyos Cmax son inferiores a los conocidos, se resaltan con color rojo. La negrilla se emplea para destacar el mejor resultado entre los que suponen una mejora.

Instancia	n	m	Cmax_Ref	IPSOVEP	Cmax	IPSOVEP	Cmax
				SMAGA-1	SMAGA-1	SMAGA-2	SMAGA-2
ta002_SSD10_P3	20	5	351	0,5698	353	0,8547	354
ta004_SSD10_P3	20	5	352	0,85227	355	1,42045	357
ta006_SSD10_P3	20	5	348	0	348	0,28735	349
ta008_SSD10_P3	20	5	332	0	332	0,3012	333
ta010_SSD10_P3	20	5	322	-0,9316	319	0	322
ta012_SSD10_P3	20	10	541	0	541	-0,3696	539
ta014_SSD10_P3	20	10	461	1,30151	467	1,08459	466
ta016_SSD10_P3	20	10	509	0,19646	510	0	509
ta018_SSD10_P3	20	10	513	-0,1949	512	0,77972	517
ta020_SSD10_P3	20	10	533	0	533	0,93808	538
ta022_SSD10_P3	20	20	786	-0,2544	784	-0,2544	784
ta024_SSD10_P3	20	20	782	0	782	0,5115	786
ta026_SSD10_P3	20	20	859	0	859	0,93131	867
ta028_SSD10_P3	20	20	869	0	869	0	869
ta030_SSD10_P3	20	20	878	0,22779	880	0	878
ta032_SSD10_P3	50	5	678	-0,5899	674	1,32743	687
ta034_SSD10_P3	50	5	680	1,02941	687	2,5	697
ta036_SSD10_P3	50	5	700	-1	693	-0,7142	695
ta038_SSD10_P3	50	5	656	-1,2195	648	-1,6768	645
ta040_SSD10_P3	50	5	684	1,02339	691	0,58479	688
ta042_SSD10_P3	50	10	867	0,11534	868	3,34486	896
ta044_SSD10_P3	50	10	837	3,2258	864	5,85424	886
ta046_SSD10_P3	50	10	940	2,23404	961	2,02127	959
ta048_SSD10_P3	50	10	947	-0,4223	943	0,52798	952
ta050_SSD10_P3	50	10	926	0,43196	930	0,86393	934
ta052_SSD10_P3	50	20	2168	-39,437	1313	-39,391	1314
ta054_SSD10_P3	50	20	1324	-0,2265	1321	1,73716	1347
ta056_SSD10_P3	50	20	1325	0,22641	1328	-0,3773	1320
ta058_SSD10_P3	50	20	1284	-0,4672	1278	0,46728	1290

ta060_SSD10_P3	50	20	1316	0	1316	0,83586	1327
----------------	----	----	------	---	------	---------	------

Tabla II-15. Mejores resultados alcanzados para el makespan del grupo de problemas SDD10_P13 propuestos por (Vallada, E. et al. 2003).

Instancia	n	m	Cmax_Ref	IPSOVEP	Cmax	IPSOVEP	Cmax
				SMAGA-1	SMAGA-1	SMAGA-2	SMAGA-2
ta002_SSD50_P3	20	5	459	-0,8714	455	-0,2178	458
ta004_SSD50_P3	20	5	461	0,43383	463	-1,3071	455
ta006_SSD50_P3	20	5	440	0	440	2,72727	452
ta008_SSD50_P3	20	5	429	0,6993	432	1,36363	435
ta010_SSD50_P3	20	5	412	0,24271	413	1,21359	417
ta012_SSD50_P3	20	10	652	-0,4601	649	-0,7281	647
ta014_SSD50_P3	20	10	580	1,89655	591	3,44827	600
ta016_SSD50_P3	20	10	637	-0,1569	636	0,62794	641
ta018_SSD50_P3	20	10	621	-0,6441	617	-0,6441	617
ta020_SSD50_P3	20	10	646	0,77399	651	1,08359	653
ta022_SSD50_P3	20	20	928	0,10775	929	1,18534	939
ta024_SSD50_P3	20	20	926	0,86393	934	1,40388	939
ta026_SSD50_P3	20	20	999	0,5005	1004	0,7007	1006
ta028_SSD50_P3	20	20	1020	0,19607	1022	0,39215	1024
ta030_SSD50_P3	20	20	1012	0,69169	1019	0,49407	1017
ta032_SSD50_P3	50	5	955	-0,2094	953	0,10471	956
ta034_SSD50_P3	50	5	977	-0,7164	970	-0,4094	973
ta036_SSD50_P3	50	5	951	-1,2618	939	1,22824	963
ta038_SSD50_P3	50	5	932	-0,6437	926	-1,0515	922
ta040_SSD50_P3	50	5	965	-1,3471	952	-0,829	957
ta042_SSD50_P3	50	10	1215	-0,823	1205	-0,5181	1209
ta044_SSD50_P3	50	10	1174	2,04429	1198	-0,2469	1171
ta046_SSD50_P3	50	10	1288	-0,6211	1280	0,54347	1295
ta048_SSD50_P3	50	10	1284	-0,3115	1280	-0,6211	1276
ta050_SSD50_P3	50	10	1261	-0,1586	1259	-0,5551	1254
ta052_SSD50_P3	50	20	1658	-0,0603	1657	1,34813	1680
ta054_SSD50_P3	50	20	1718	-0,291	1713	0,23282	1722
ta056_SSD50_P3	50	20	1713	0,17513	1716	0	1713
ta058_SSD50_P3	50	20	1678	-0,5363	1669	-0,467	1670
ta060_SSD50_P3	50	20	1704	0,70422	1716	0,23837	1708

Tabla II-16. Mejores resultados alcanzados para el makespan del grupo de problemas SDD50_P13 propuestos por (Vallada, E. et al. 2003)

Instancia	n	m	Cmax_Ref	IPSOVEP	Cmax	IPSOVEP	Cmax
				SMAGA-1	SMAGA-1	SMAGA-2	SMAGA-2
ta002_SSD100_P3	20	5	564	-1,4184	556	0,53191	567
ta004_SSD100_P3	20	5	571	-1,0507	565	2,97723	588
ta006_SSD100_P3	20	5	538	-0,1858	537	-0,5576	535
ta008_SSD100_P3	20	5	531	0,94161	536	3,38983	549
ta010_SSD100_P3	20	5	509	-1,1787	503	-1,3752	502
ta012_SSD100_P3	20	10	798	0	798	-0,5012	794
ta014_SSD100_P3	20	10	717	0	717	1,3947	727
ta016_SSD100_P3	20	10	769	0,26007	771	0,26007	771
ta018_SSD100_P3	20	10	760	-0,3947	757	1,97368	775
ta020_SSD100_P3	20	10	794	0,62972	799	0,88161	801
ta022_SSD100_P3	20	20	1113	-0,2695	1110	-0,4492	1108
ta024_SSD100_P3	20	20	1112	0,08992	1113	0,26978	1115
ta026_SSD100_P3	20	20	1184	-1,3513	1168	-0,7601	1175
ta028_SSD100_P3	20	20	1204	-0,1661	1202	0,7475	1213
ta030_SSD100_P3	20	20	1191	-0,5037	1185	0,50377	1197
ta032_SSD100_P3	50	5	1256	-2,3885	1226	0,87579	1267
ta034_SSD100_P3	50	5	1275	-2,745	1240	-2,8235	1239
ta036_SSD100_P3	50	5	1248	-1,282	1232	-1,7628	1226
ta038_SSD100_P3	50	5	1248	-2,4038	1218	-1,2019	1233
ta040_SSD100_P3	50	5	1248	0,32051	1252	0,40064	1253
ta042_SSD100_P3	50	10	1555	0	1555	1,47909	1578
ta044_SSD100_P3	50	10	1570	-1,656	1544	0,63694	1580
ta046_SSD100_P3	50	10	1658	-0,8443	1644	0,72376	1670
ta048_SSD100_P3	50	10	1614	0,61957	1624	0,49566	1622
ta050_SSD100_P3	50	10	1612	-1,2406	1592	1,98511	1644
ta052_SSD100_P3	50	20	2106	-0,0474	2105	1,4245	2136
ta054_SSD100_P3	50	20	2145	0	2145	-0,4662	2135
ta056_SSD100_P3	50	20	2172	-0,8287	2154	-0,6906	2157
ta058_SSD100_P3	50	20	2100	0,04761	2101	-0,7142	2085
ta060_SSD100_P3	50	20	2159	0,18527	2163	-0,0926	2157

Tabla II-17. Mejores resultados alcanzados para el makespan del grupo de problemas SDD100_P13 propuestos por (Vallada, E. et al. 2003)

Instancia	n	m	Cmax_Ref	IPSOVEP		Cmax	
				SMAGA-1	SMAGA-1	SMAGA-2	SMAGA-2
ta002_SSD125_P3	20	5	609	0,82101	614	-1,642	599
ta004_SSD125_P3	20	5	610	2,29508	624	1,4754	619
ta006_SSD125_P3	20	5	588	-3,2312	569	0	588
ta008_SSD125_P3	20	5	586	1,02389	592	0,68259	590
ta010_SSD125_P3	20	5	555	-1,081	549	1,44144	563
ta012_SSD125_P3	20	10	858	1,3986	870	0,81585	865
ta014_SSD125_P3	20	10	782	1,15089	791	-3,4526	755
ta016_SSD125_P3	20	10	827	0,84643	834	2,17654	845
ta018_SSD125_P3	20	10	831	1,44404	843	1,92539	847
ta020_SSD125_P3	20	10	861	-1,3937	849	-0,5807	856
ta022_SSD125_P3	20	20	1206	0,24875	1209	-0,3316	1202
ta024_SSD125_P3	20	20	1209	-0,6617	1201	0,08271	1210
ta026_SSD125_P3	20	20	1264	0,39556	1269	-0,5537	1257
ta028_SSD125_P3	20	20	1293	-0,3093	1289	0,61871	1301
ta030_SSD125_P3	20	20	1269	-0,5516	1262	0,39401	1274
ta032_SSD125_P3	50	5	1362	-1,6886	1339	0	1362
ta034_SSD125_P3	50	5	1368	-0,3654	1363	2,33918	1400
ta036_SSD125_P3	50	5	1372	-3,0612	1330	0,87463	1384
ta038_SSD125_P3	50	5	1387	-2,3071	1355	-0,2162	1384
ta040_SSD125_P3	50	5	1370	1,75182	1394	-0,5839	1362
ta042_SSD125_P3	50	10	1732	-1,3856	1708	0,75057	1745
ta044_SSD125_P3	50	10	1732	-0,5773	1722	0,28868	1737
ta046_SSD125_P3	50	10	1801	0,16657	1804	0,83287	1816
ta048_SSD125_P3	50	10	1778	0,84364	1793	1,12485	1798
ta050_SSD125_P3	50	10	1760	0,79545	1774	1,47727	1786
ta052_SSD125_P3	50	20	2331	-1,4157	2298	-0,3003	2324
ta054_SSD125_P3	50	20	2352	1,06292	2377	1,23299	2381
ta056_SSD125_P3	50	20	2373	-0,4635	2362	0,04214	2374
ta058_SSD125_P3	50	20	2307	0,5635	2320	0,34677	2315
ta060_SSD125_P3	50	20	2370	0,63291	2385	0,08438	2372

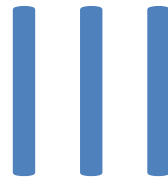
Tabla II-18. Mejores resultados alcanzados para el makespan del grupo de problemas SDD125_P3 propuestos por (Vallada, E. et al. 2003)

Taillard, E. 1993. Benchmarks for Basic Scheduling Problems. *European Journal of Operational Research*, 64(2): 278-285.

Vallada, E., Ruiz, R., and Maroto, C. *Synthetic and Real Benchmarks for Complex Flow-Shops Problems. Informe Técnico. 2003. Grupo de Investigación Operativa (GIO). Universidad Politécnica de Valencia (Spain).*

**Análisis Estadístico de los Resultados
Obtenidos en la Experimentación para
la Programación Predictiva de la
Producción**

Anexo



A-III.1 Ajuste de Parámetros SMAGA-1 para la Configuración P13

A-III.1.1 Bloque SSD10

A-III.1.1.1 Subconjunto SSD10_P13_20

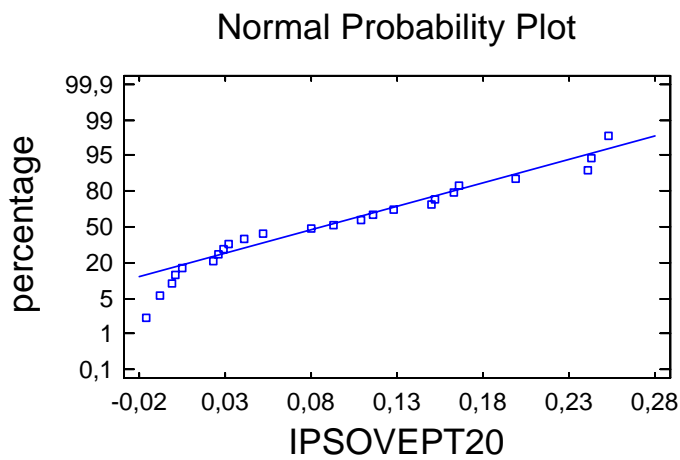


Figura II-1. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPSOVEPT del conjunto SSD10_P13_20 en relación a las frecuencias relativas acumuladas.

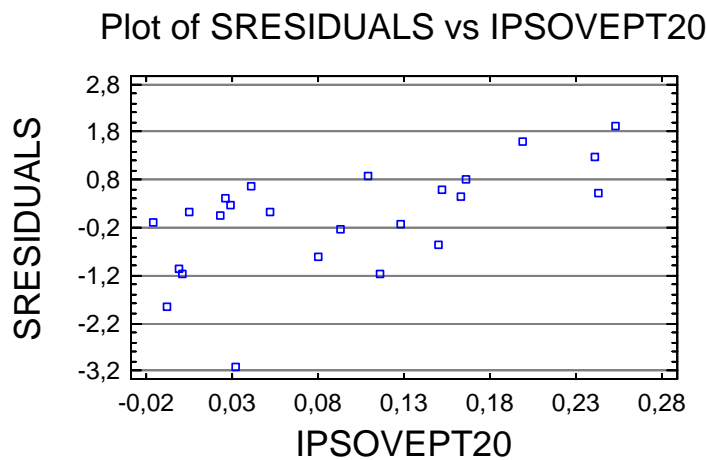


Figura II-2. Residuos estudentizados frente a la variable IPSOVEPT del conjunto experimental SSD10_P13_20.

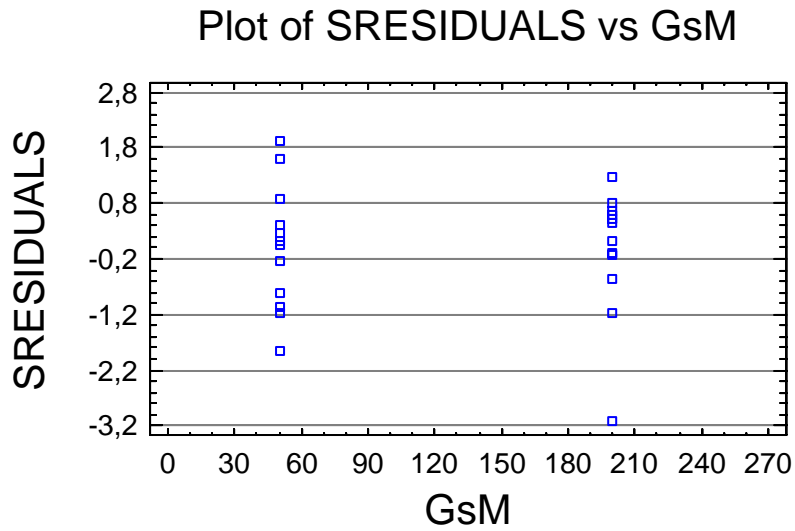


Figura II-3. Residuos estudentizados frente a los niveles del factor GsM.

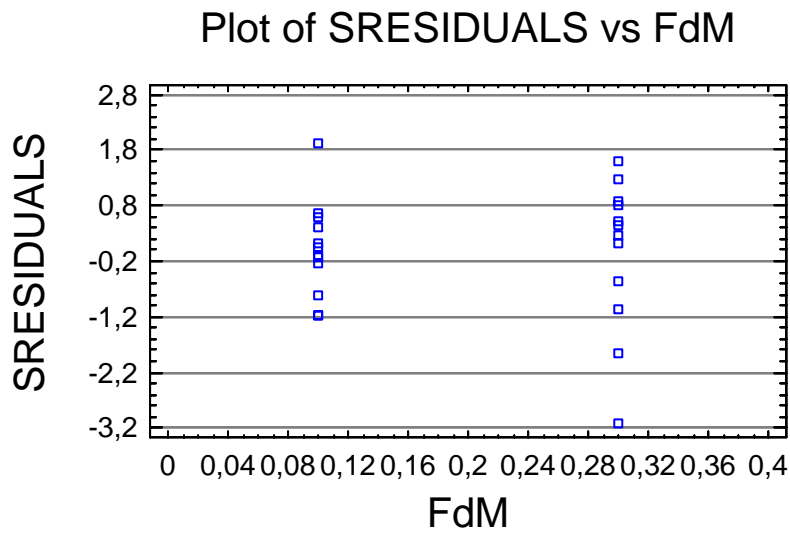


Figura II-4. Residuos estudentizados frente a los niveles del factor FdM.

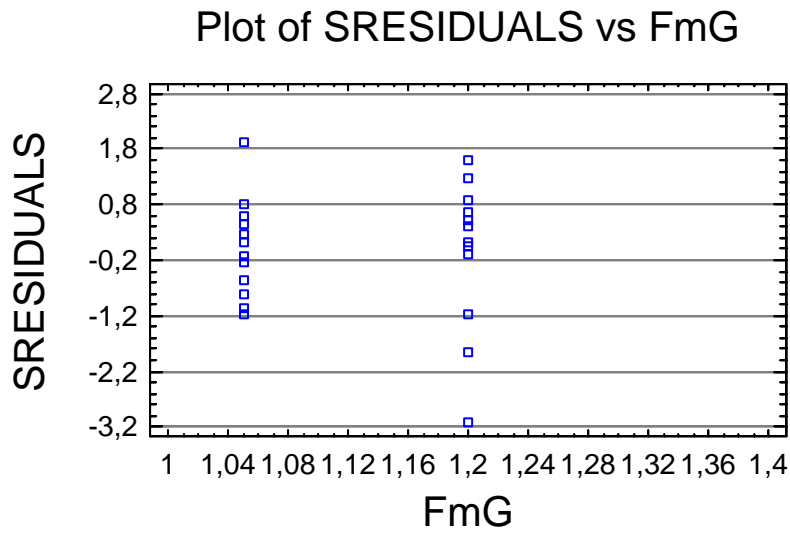


Figura II-5. Residuos estudentizados frente a los niveles del factor FmG.

Correlation Matrix for Estimated Effects

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
(1) average	1,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
(2) A:GsM	0,0000	1,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
(3) B:FdM	0,0000	0,0000	1,0000	0,0000	0,0000	0,0000	0,0000	0,0000
(4) C:FmG	0,0000	0,0000	0,0000	1,0000	0,0000	0,0000	0,0000	0,0000
(5) AB	0,0000	0,0000	0,0000	0,0000	1,0000	0,0000	0,0000	0,0000
(6) AC	0,0000	0,0000	0,0000	0,0000	0,0000	1,0000	0,0000	0,0000
(7) BC	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	1,0000	0,0000
(8) block	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	1,0000

Tabla II-1. Matriz de correlaciones para el conjunto experimental SSD10_P13_20.

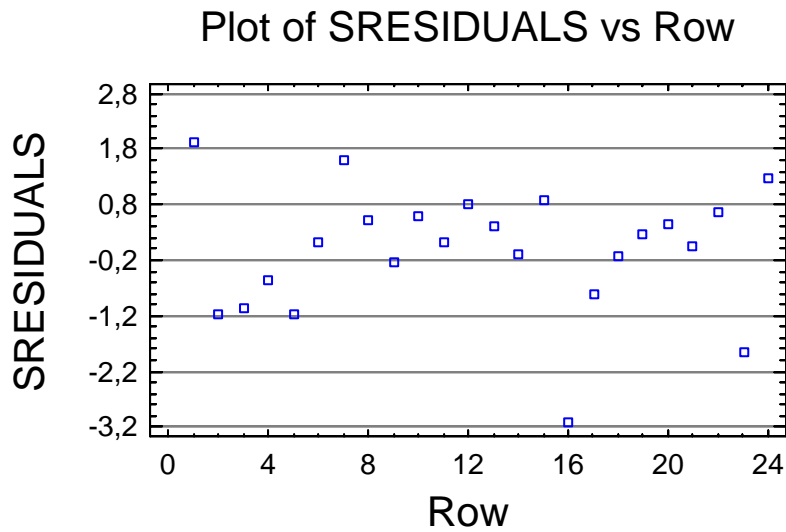


Figura II-6. Residuos estudentizados frente al orden de ejecución.

Analysis of Variance for IPSOVEPT20

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
A:GsM	0,0181661	1	0,0181661	4,26	0,0568
B:FdM	0,00597277	1	0,00597277	1,40	0,2550
C:GmF	0,00629737	1	0,00629737	1,48	0,2431
AB	0,0184553	1	0,0184553	4,33	0,0551
AC	0,00122302	1	0,00122302	0,29	0,6001
BC	0,0415501	1	0,0415501	9,74	0,0070
blocks	0,0131768	2	0,00658841	1,54	0,2454
Total error	0,0639671	15	0,00426447		
Total (corr.)	0,168809	23			

Tabla II-2. ANOVA correspondiente al conjunto experimental SSD10_P13_20.

Observando la Tabla II-2 vemos que disponemos de 24 datos y como vemos, tenemos 15 grados de libertad residuales. ANOVA realiza un contraste de hipótesis sobre las medias y cuando el p-value es menor que un α determinado (la probabilidad de rechazar la hipótesis nula cuando es cierta) significa que realmente existen diferencias estadísticamente significativas entre las medias de los niveles o variantes para un factor o interacción. En este caso se establece un $\alpha=0,05$, lo que significa que sólo es significativa la combinación de factores BC, es decir FdM y GmF.

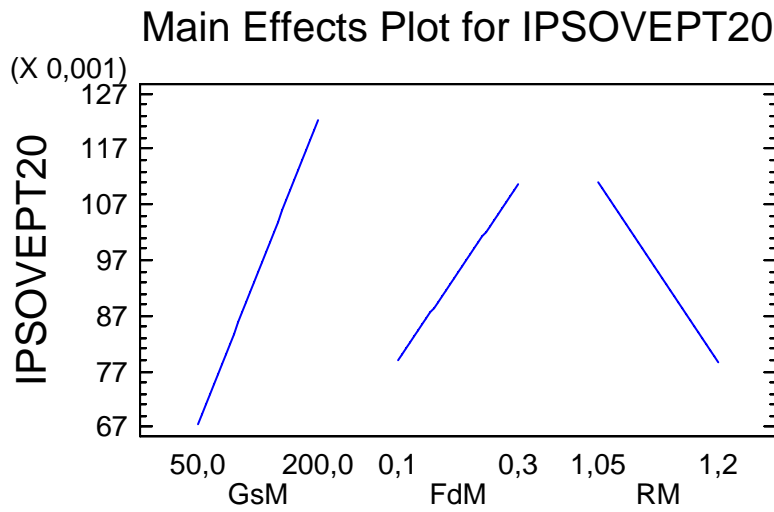


Figura II-7. Gráfico de efectos principales de los factores del conjunto experimental SSD10_P13_20.

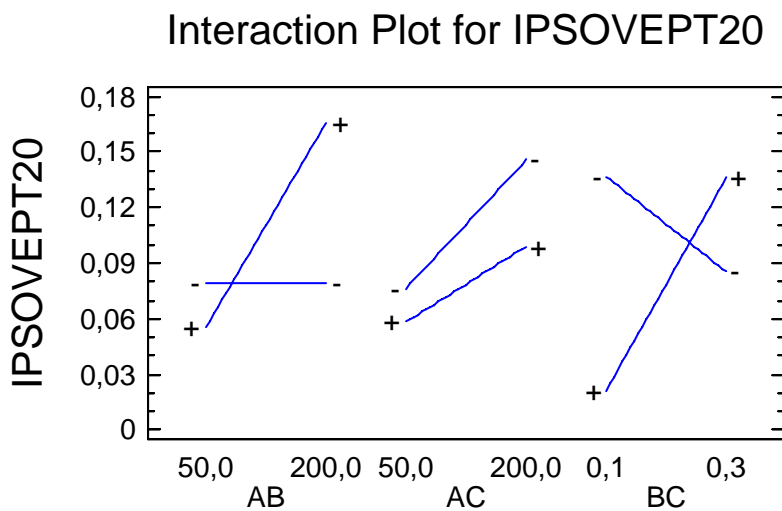


Figura II-8. Gráfico de interacción entre los factores del conjunto experimental SSD10_P13_20.

Analizando la Figura II-8 se deduce que el valor adecuado son para el factor FdM es 0.1, en combinación con el valor más alto del parámetro GmF, es decir, con 1.2. El valor del factor GsM, se puede establecer libremente. No obstante, el valor elegido para GsM será 50, ya que con este valor se obtienen mejores resultados teniendo en cuenta todas las interacciones,

aunque no se pueda afirmar que éstas sean significativas. Por lo tanto, los parámetros quedarían $G_sM=50$, $F_dM=0.1$ y $F_mG=1.2$.

A-III.1.1.2 Subconjunto SSD10_P13_50

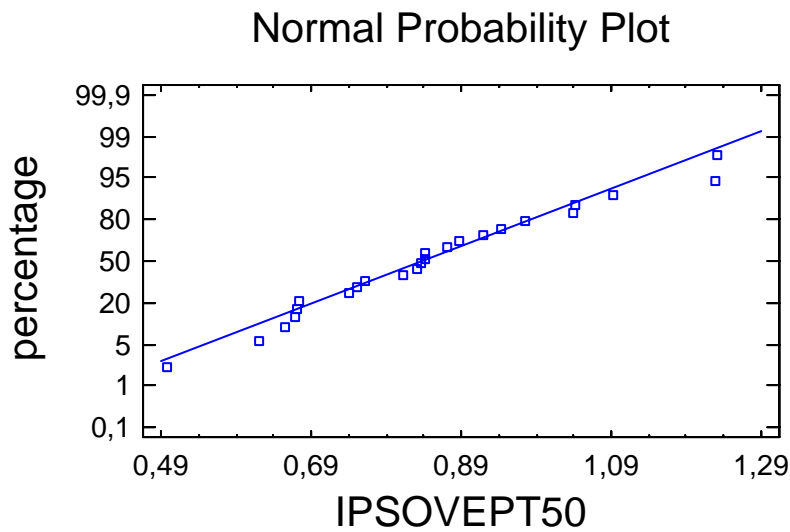


Figura II-9. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPSOVEPT del conjunto SSD10_P13_50 en relación a las frecuencias relativas acumuladas.

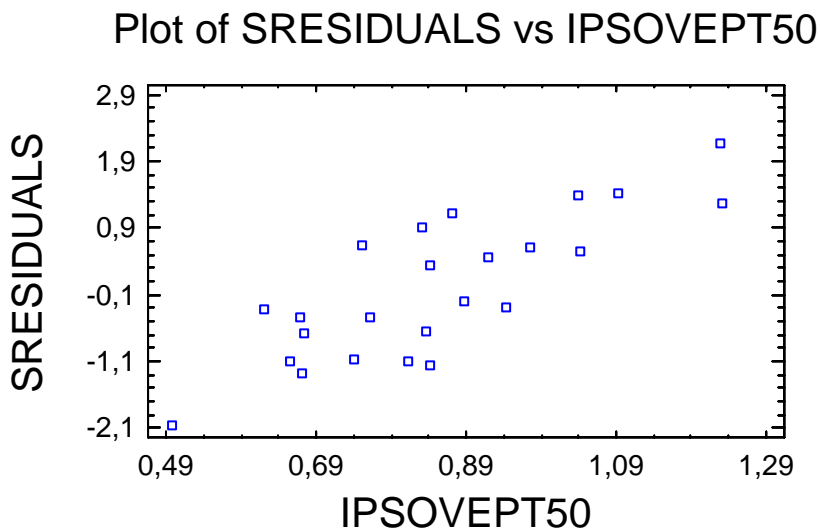


Figura II-10. Residuos estudentizados frente a la variable IPSOVEPT del conjunto experimental SSD10_P13_50.

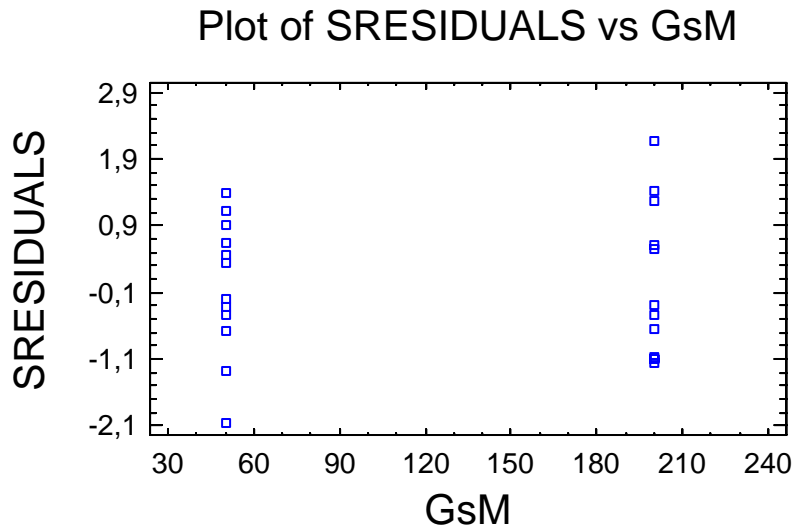


Figura II-11. Residuos estudentizados frente a los niveles del factor GsM.

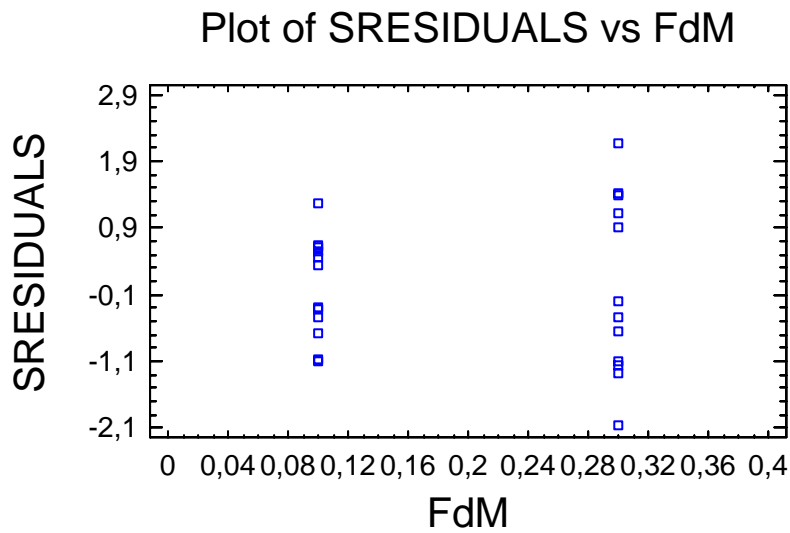


Figura II-12. Residuos estudentizados frente a los niveles del factor FdM.

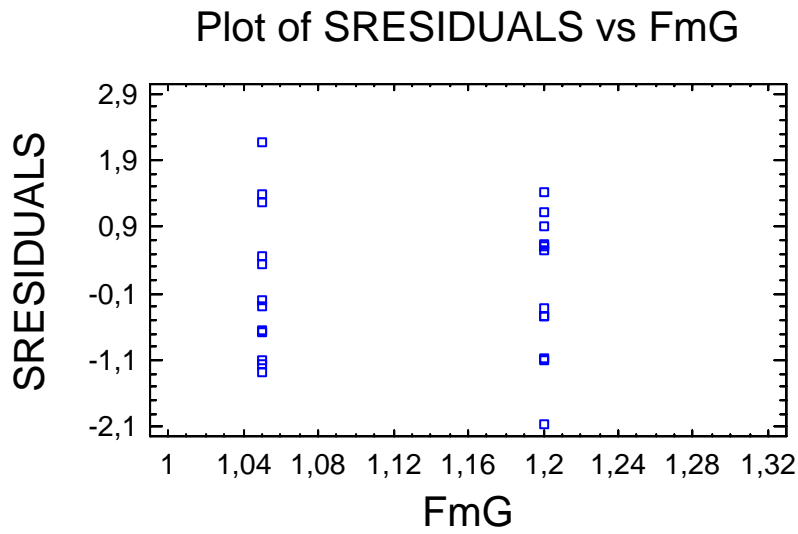


Figura II-13. Residuos estudentizados frente a los niveles del factor FmG.

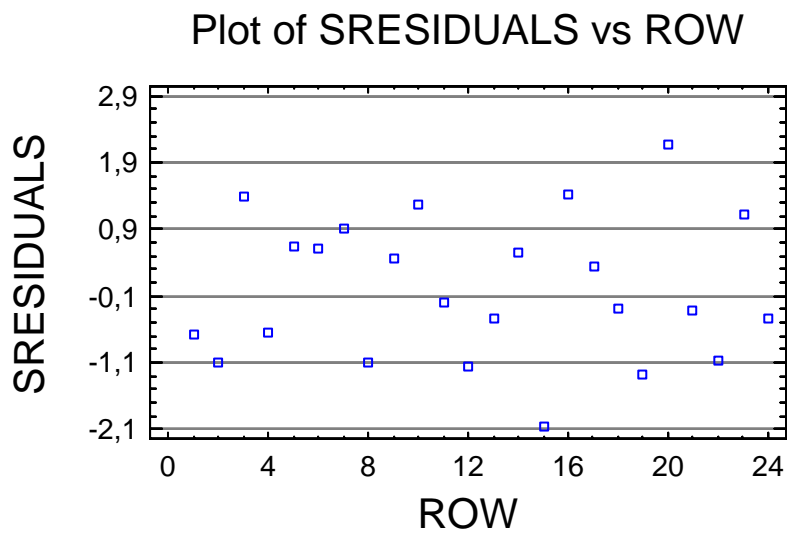


Figura II-14. Residuos estudentizados frente al orden de ejecución.

Analysis of Variance for IPSOVEPT50

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
A:GsM	0,148723	1	0,148723	4,35	0,0544
B:FdM	0,00000129865	1	0,00000129865	0,00	0,9952
C:FmG	0,0838872	1	0,0838872	2,46	0,1380
AB	0,0176728	1	0,0176728	0,52	0,4830
AC	0,00108121	1	0,00108121	0,03	0,8612
BC	0,00118466	1	0,00118466	0,03	0,8548
blocks	0,0261343	2	0,0130672	0,38	0,6886
Total error	0,512426	15	0,0341617		
Total (corr.)	0,79111	23			

Tabla II-3. ANOVA correspondiente al conjunto experimental SSD10_P13_50.

Observando la Tabla II-3 disponemos de 24 datos y como vemos, tenemos 15 grados de libertad residuales. El contraste de hipótesis sobre las medias para un $\alpha=0,05$ implica que no existen factores significativos ni para los efectos principales ni en ninguna otra combinación.

A-III.1.2 Bloque SSD50

A-III.1.2.1 Subconjunto SSD50_P13_20

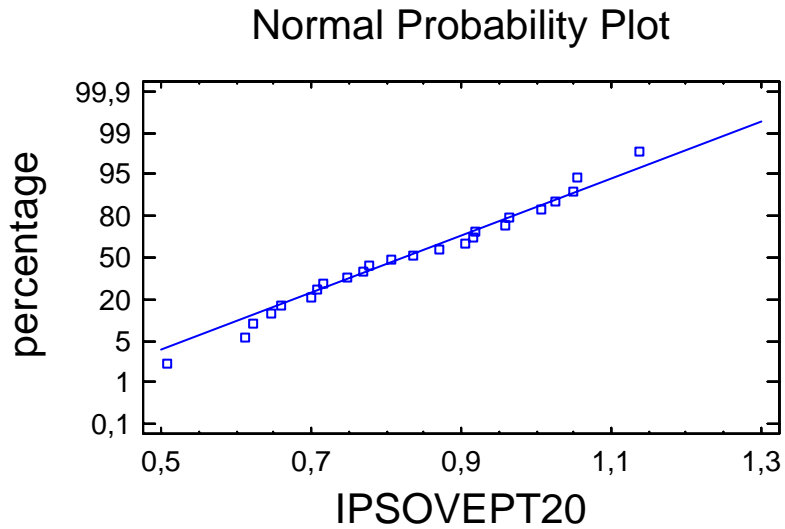


Figura II-15. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPSOVEPT del conjunto SSD50_P13_20 en relación a las frecuencias relativas acumuladas.

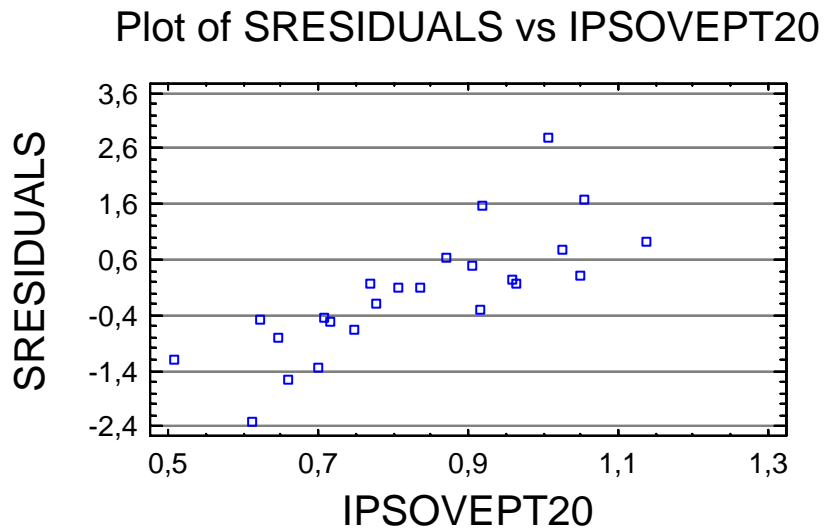


Figura II-16. Residuos estudentizados frente a la variable IPSOVEPT del conjunto experimental SSD50_P13_20.

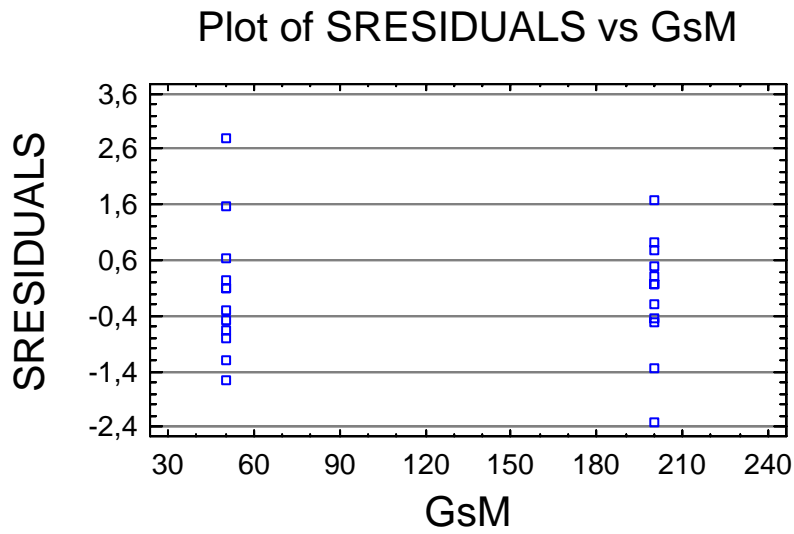


Figura II-17. Residuos estudentizados frente a los niveles del factor GsM.

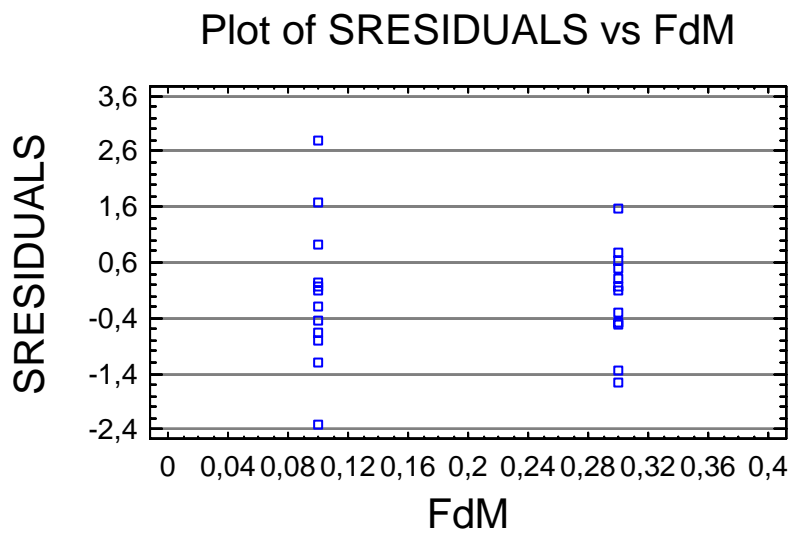


Figura II-18. Residuos estudentizados frente a los niveles del factor FdM.

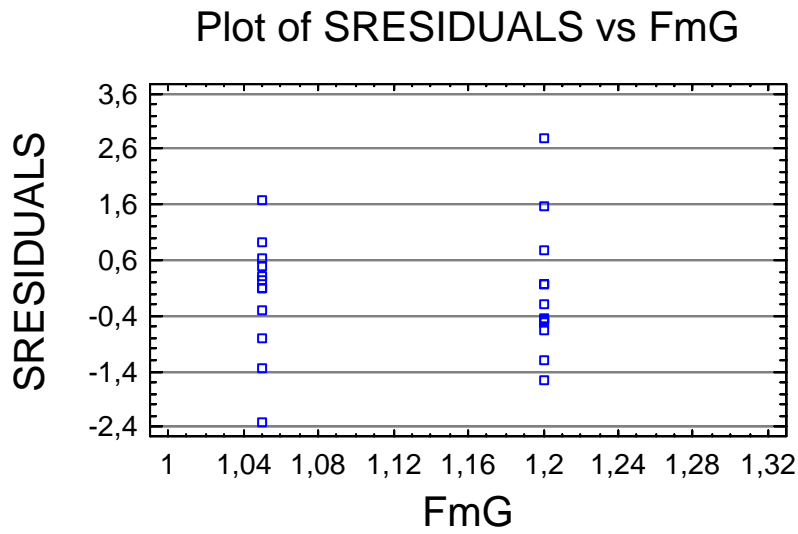


Figura II-19. Residuos estudentizados frente a los niveles del factor FmG.

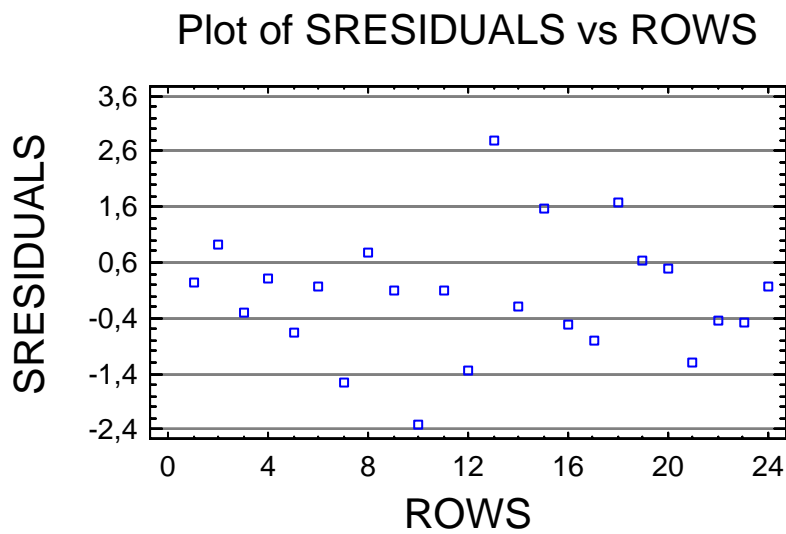


Figura II-20. Residuos estudentizados frente al orden de ejecución.

Analysis of Variance for IPSOVEPT20

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
A:GsM	0,0354851	1	0,0354851	1,28	0,2765
B:FdM	0,000155004	1	0,000155004	0,01	0,9415
C:FmG	0,0475201	1	0,0475201	1,71	0,2109
AB	0,00232669	1	0,00232669	0,08	0,7764
AC	0,00020854	1	0,00020854	0,01	0,9322
BC	0,000202533	1	0,000202533	0,01	0,9331
blocks	0,130621	2	0,0653104	2,35	0,1297
Total error	0,417342	15	0,0278228		
Total (corr.)	0,633861	23			

Tabla II-4. ANOVA correspondiente al conjunto experimental SSD50_P13_20.

Observando la Tabla II-4 disponemos de 24 datos y como vemos, tenemos 15 grados de libertad residuales. El contraste de hipótesis sobre las medias para un $\alpha=0,05$ implica que no existen factores significativos ni para los efectos principales ni en ninguna otra combinación.

A-III.1.2.2 Subconjunto SSD50_P13_50

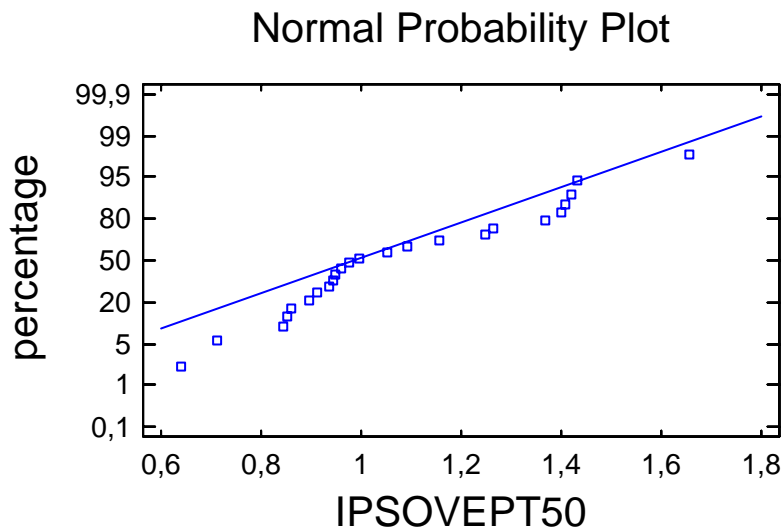


Figura II-21. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPSOVEPT del conjunto SSD50_P13_50 en relación a las frecuencias relativas acumuladas.

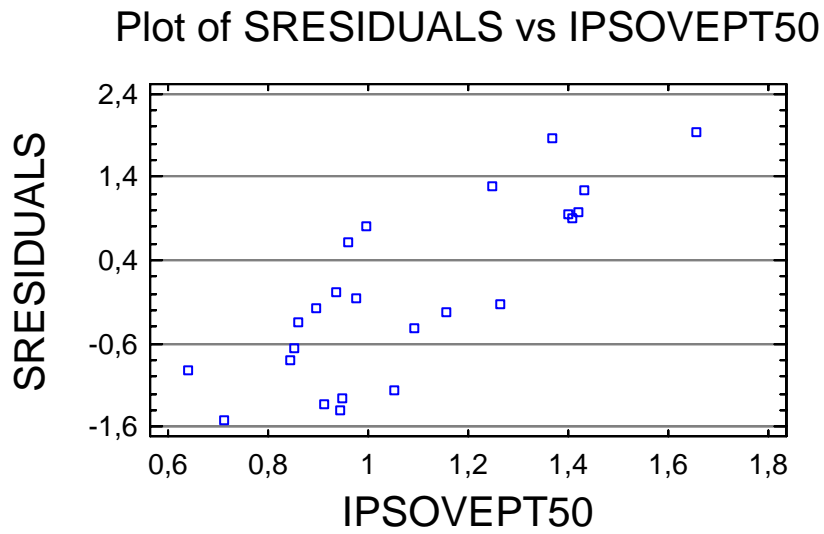


Figura II-22. Residuos estudentizados frente a la variable IPSOVEPT del conjunto experimental SSD50_P13_50.

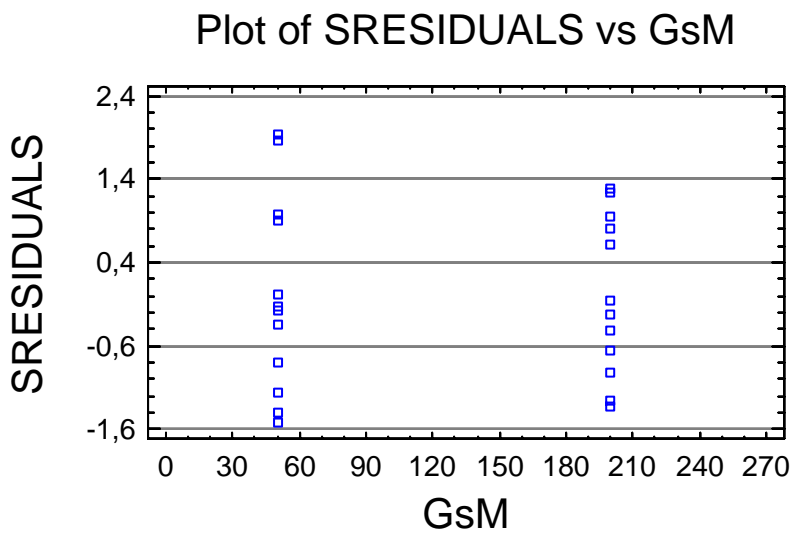


Figura II-23. Residuos estudentizados frente a los niveles del factor GsM.

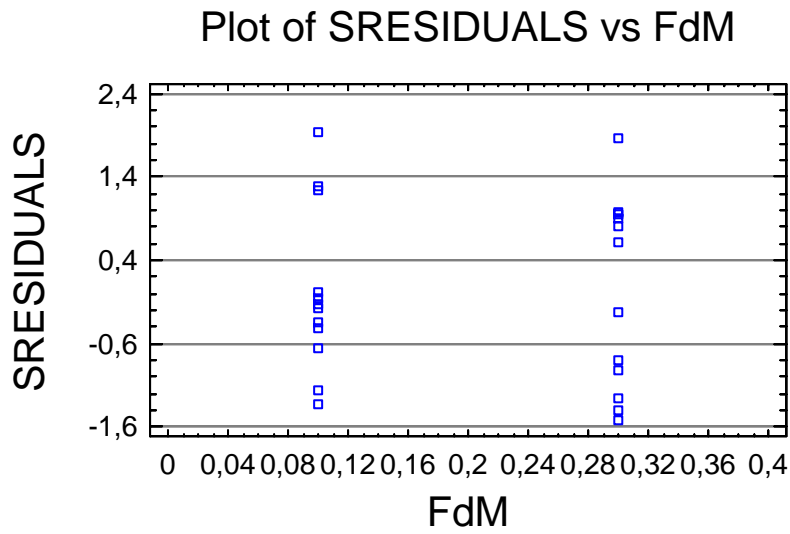


Figura II-24. Residuos estudentizados frente a los niveles del factor FdM.

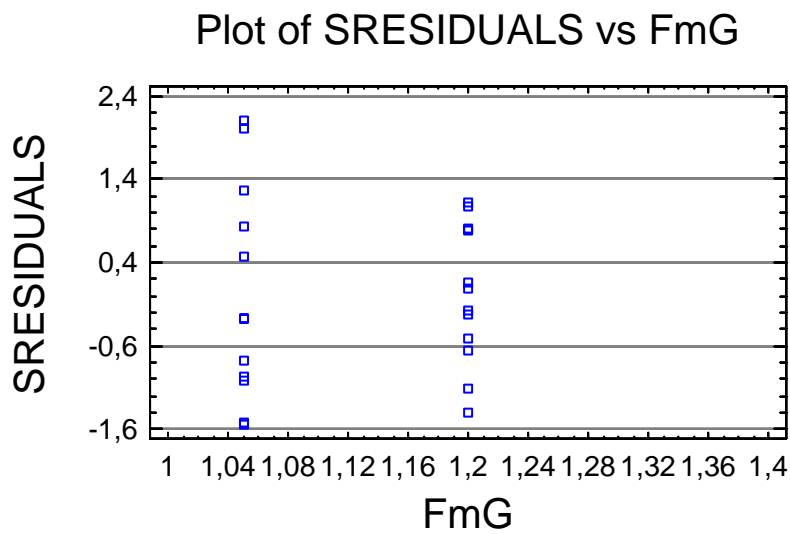


Figura II-25. Residuos estudentizados frente a los niveles del factor FmG.

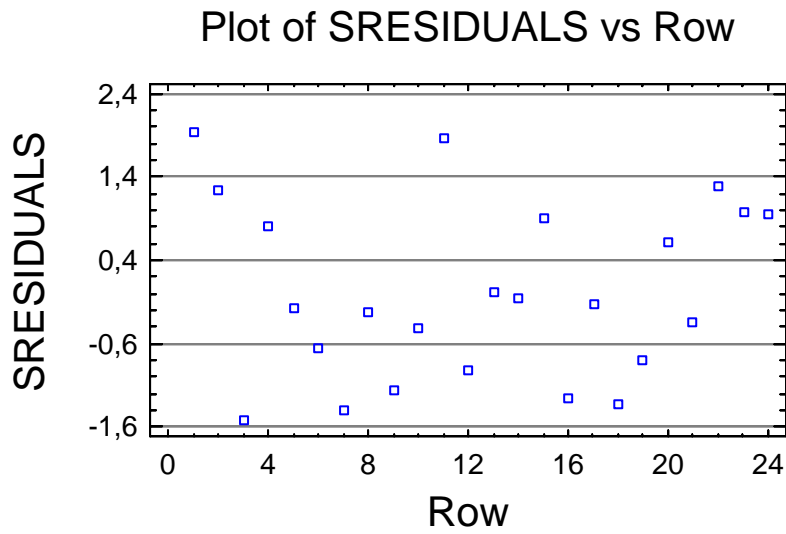


Figura II-26. Residuos estudentizados frente al orden de ejecución.

Analysis of Variance for IPSOVEPT50

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
A:GsM	0,0230193	1	0,0230193	0,35	0,5645
B:FdM	0,00598945	1	0,00598945	0,09	0,7679
C:FmG	0,000537878	1	0,000537878	0,01	0,9294
AB	0,00806356	1	0,00806356	0,12	0,7322
AC	0,039553	1	0,039553	0,60	0,4519
BC	0,481785	1	0,481785	7,27	0,0166
blocks	0,0146084	2	0,00730419	0,11	0,8964
Total error	0,994732	15	0,0663155		
Total (corr.)	1,56829	23			

Tabla II-5. ANOVA correspondiente al conjunto experimental SSD50_P13_50.

Observando la Tabla II-5 vemos que disponemos de 24 datos y, tenemos 15 grados de libertad residuales. El contraste de hipótesis sobre las medias para un $\alpha=0,05$ implica que sólo es significativa la combinación de factores BC, es decir FdM y GmF.

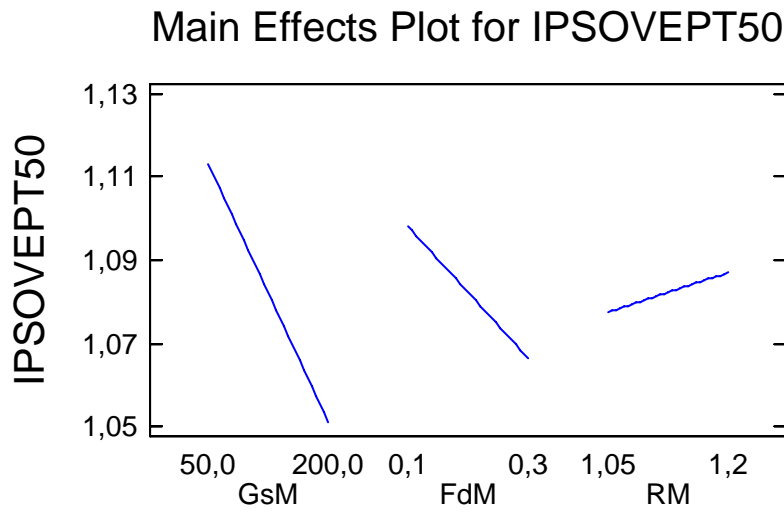


Figura II-27. Gráfico de efectos principales de los factores del conjunto experimental SSD50_P13_50.

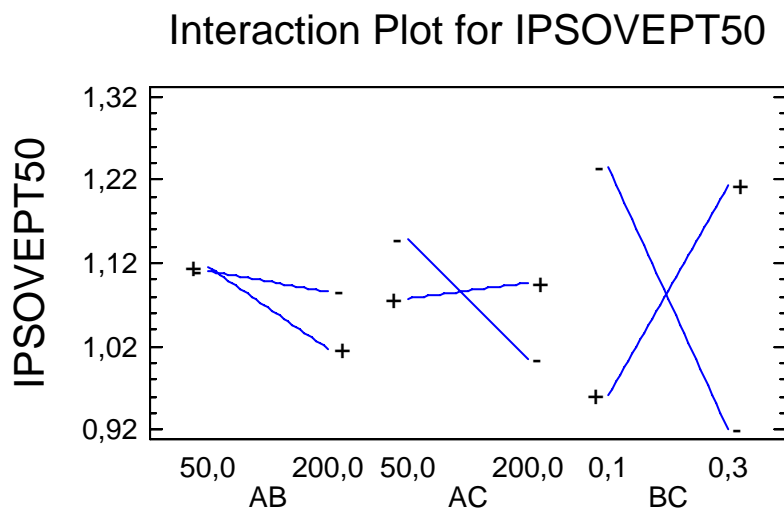


Figura II-28. Gráfico de interacción entre los factores del conjunto experimental SSD50_P13_50.

Analizando la Figura II-28 se deduce que el valor adecuado son para el factor FdM es 0.3, en combinación con el valor más bajo del parámetro GmF, es decir, con 1.05. El valor del factor GsM, se puede establecer libremente. No obstante, el valor elegido para GsM será 200, aunque no se pueda afirmar que éstas sean significativas. Por lo tanto, los parámetros quedarían GsM=200, FdM=0.3 y FmG=1.05.

A-III.1.3 Bloque SSD100

A-III.1.3.1 Subconjunto SSD100_P13_20

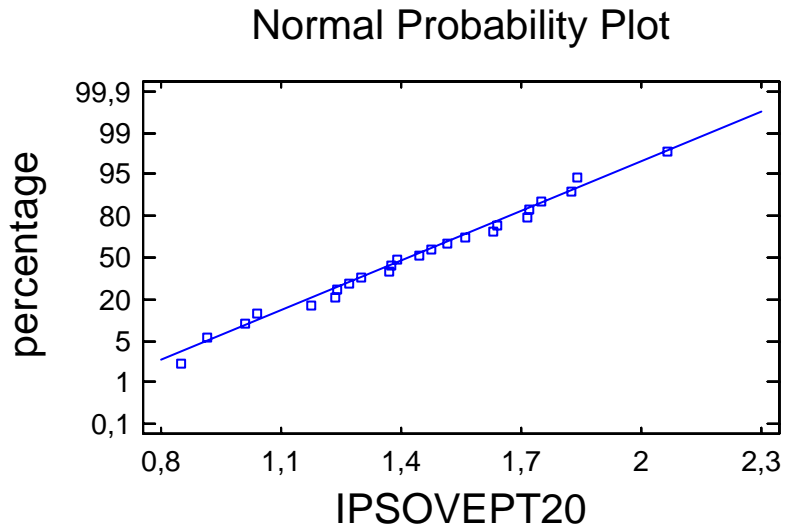


Figura II-29. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPSOVEPT del conjunto SSD100_P13_20 en relación a las frecuencias relativas acumuladas.

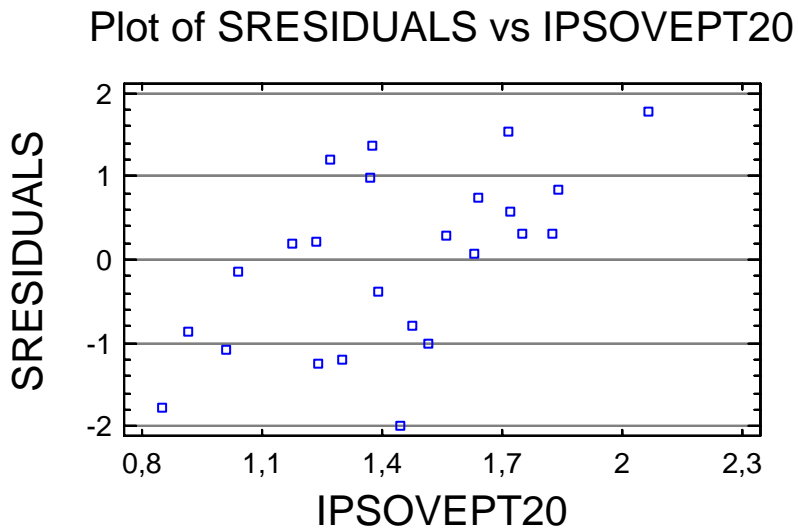


Figura II-30. Residuos estudentizados frente a la variable IPSOVEPT del conjunto experimental SSD100_P13_20.

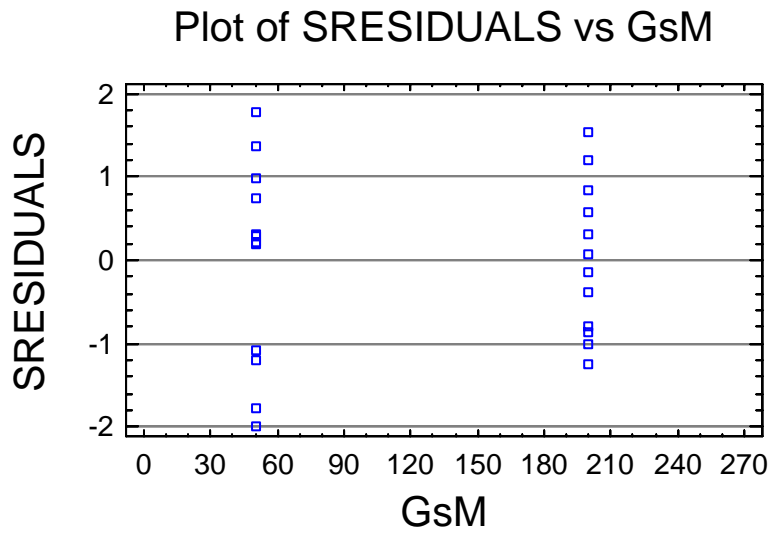


Figura II-31. Residuos estudentizados frente a los niveles del factor GsM.

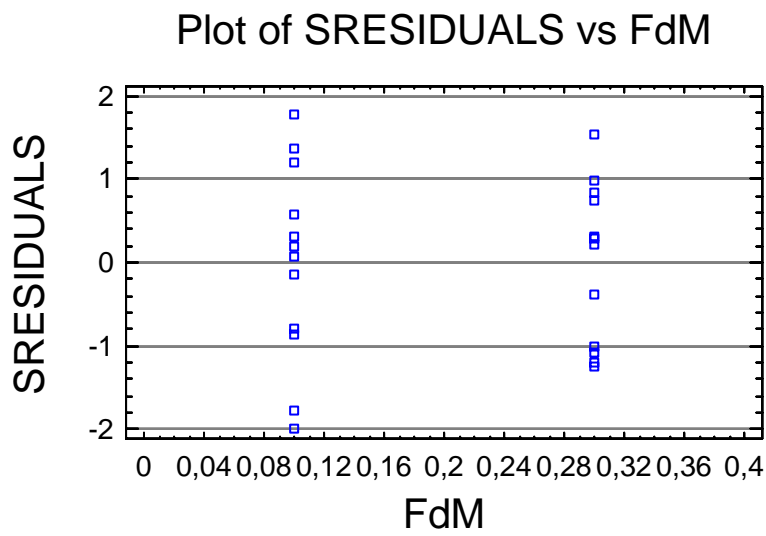


Figura II-32. Residuos estudentizados frente a los niveles del factor FdM.

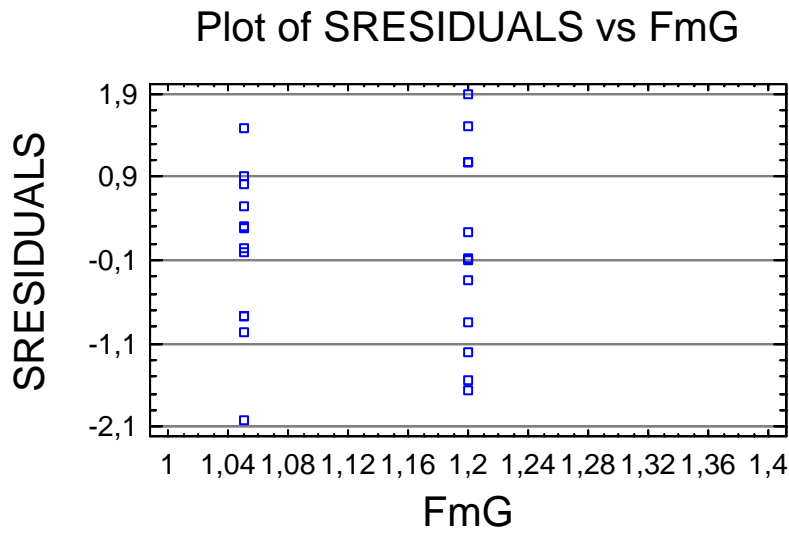


Figura II-33. Residuos estudentizados frente a los niveles del factor FmG.

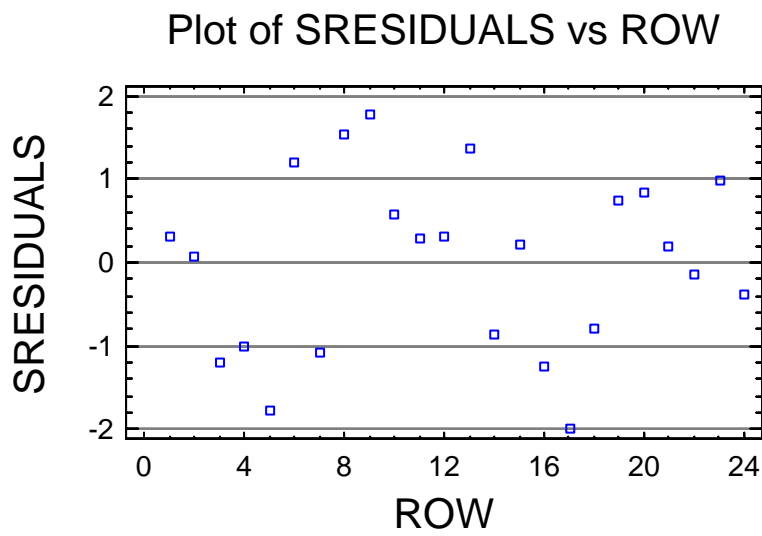


Figura II-34. Residuos estudentizados frente al orden de ejecución.

Analysis of Variance for IPSOVEPT20

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
A:GsM	0,0170934	1	0,0170934	0,37	0,5546
B:FdM	0,0253732	1	0,0253732	0,54	0,4728
C:FmG	1,11959	1	1,11959	23,93	0,0002
AB	0,169729	1	0,169729	3,63	0,0762
AC	0,00828279	1	0,00828279	0,18	0,6799
BC	0,146803	1	0,146803	3,14	0,0968
blocks	0,0353668	2	0,0176834	0,38	0,6916
Total error	0,701781	15	0,0467854		
Total (corr.)	2,22402	23			

Tabla II-6. ANOVA correspondiente al conjunto experimental SSD100_P13_20.

Observando la Tabla II-6 vemos que disponemos de 24 datos y, tenemos 15 grados de libertad residuales. El contraste de hipótesis sobre las medias para un $\alpha=0,05$ implica que sólo es significativo el efecto principal del factor C, es decir FmG.

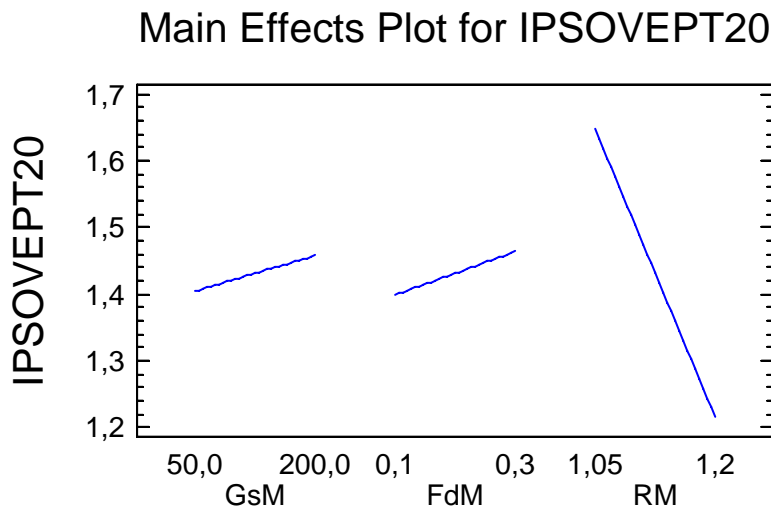


Figura II-35. Gráfico de efectos principales de los factores del conjunto experimental SSD100_P13_20.

Analizando la Figura II-35 se deduce que el valor adecuado son para el factor FmG es 1.2. El valor de los factores GsM y FdM, se pueden establecer libremente. No obstante, el valor elegido para GsM será 50 y para FdM será 0.1, aunque no se pueda afirmar que éstos sean significativos. Por lo tanto, los parámetros quedarían GsM=50, FdM=0.1 y FmG=1.2.

A-III.1.3.2 Subconjunto SSD100_P13_50

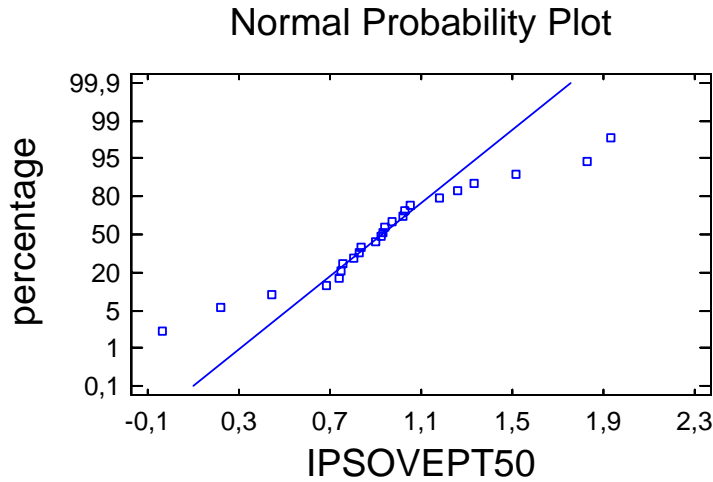


Figura II-36. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPSOVEPT del conjunto SSD100_P13_50 en relación a las frecuencias relativas acumuladas.

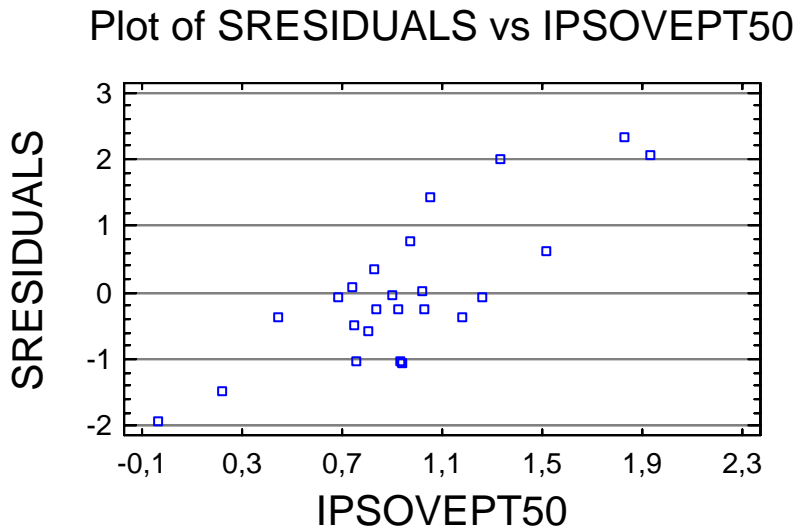


Figura II-37. Residuos estudentizados frente a la variable IPSOVEPT del conjunto experimental SSD100_P13_50.

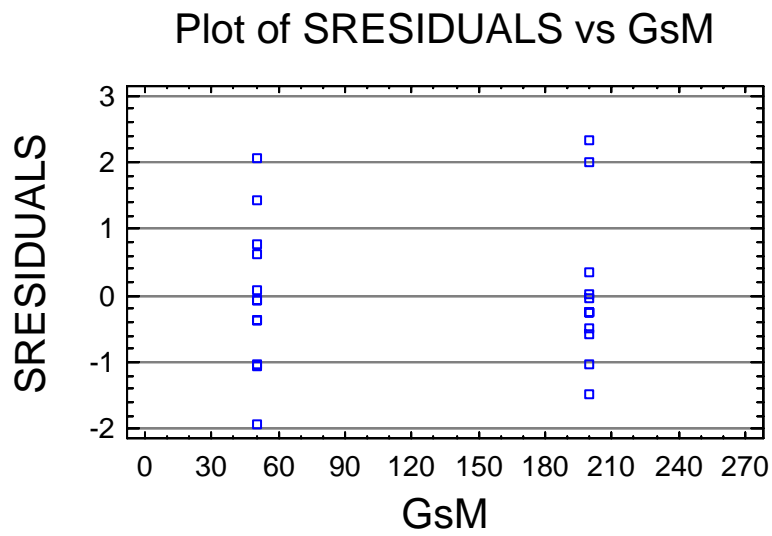


Figura II-38. Residuos estudentizados frente a los niveles del factor GsM.

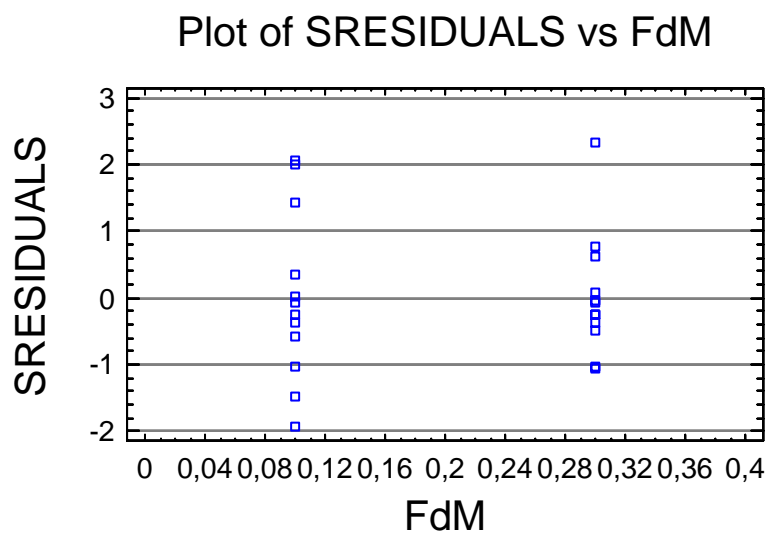


Figura II-39. Residuos estudentizados frente a los niveles del factor FdM.

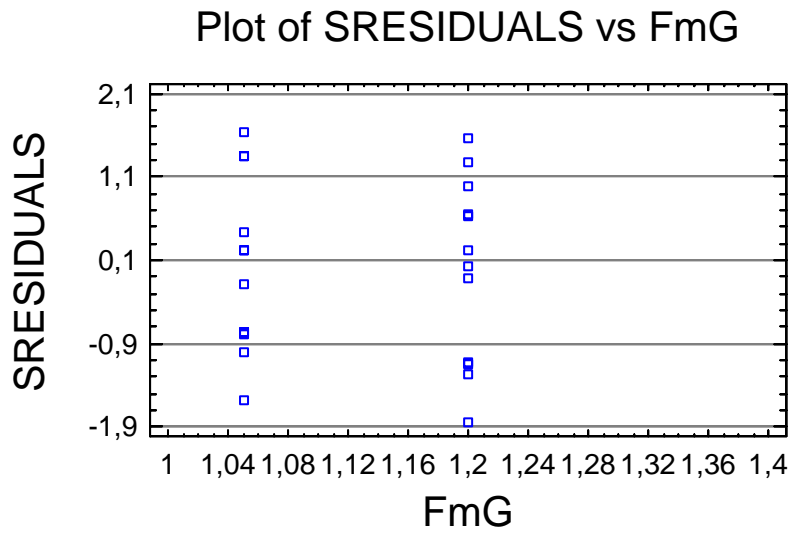


Figura II-40. Residuos estudentizados frente a los niveles del factor FmG.

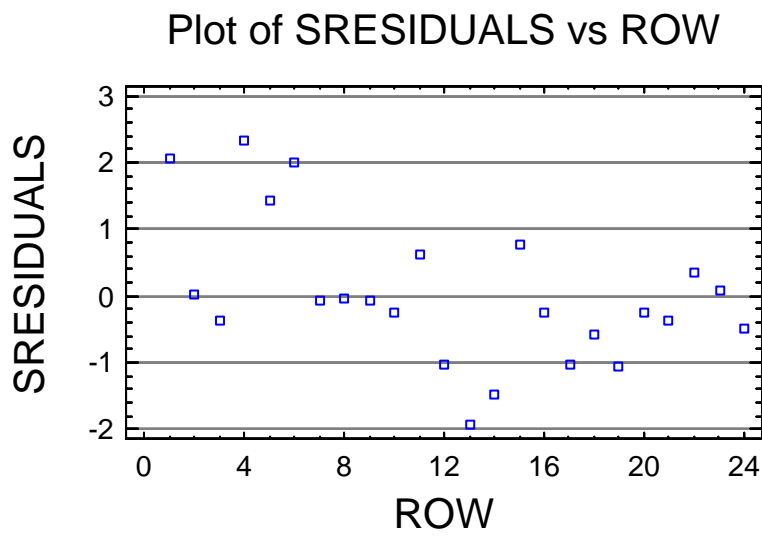


Figura II-41. Residuos estudentizados frente al orden de ejecución.

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
A:GsM	0,00615752	1	0,00615752	0,05	0,8237
B:FdM	0,0832285	1	0,0832285	0,69	0,4177
C:FmG	1,21633	1	1,21633	10,15	0,0061
AB	0,0116444	1	0,0116444	0,10	0,7595
AC	0,243384	1	0,243384	2,03	0,1746
BC	0,0167412	1	0,0167412	0,14	0,7138
blocks	0,99552	2	0,49776	4,15	0,0367
Total error	1,79761	15	0,119841		
Total (corr.)	4,37062	23			

Tabla II-7. ANOVA correspondiente al conjunto experimental SSD100_P13_50.

Observando la Tabla II-7 vemos que disponemos de 24 datos y, tenemos 15 grados de libertad residuales. El contraste de hipótesis sobre las medias para un $\alpha=0,05$ implica que sólo es significativo el efecto principal del factor C, es decir FmG.

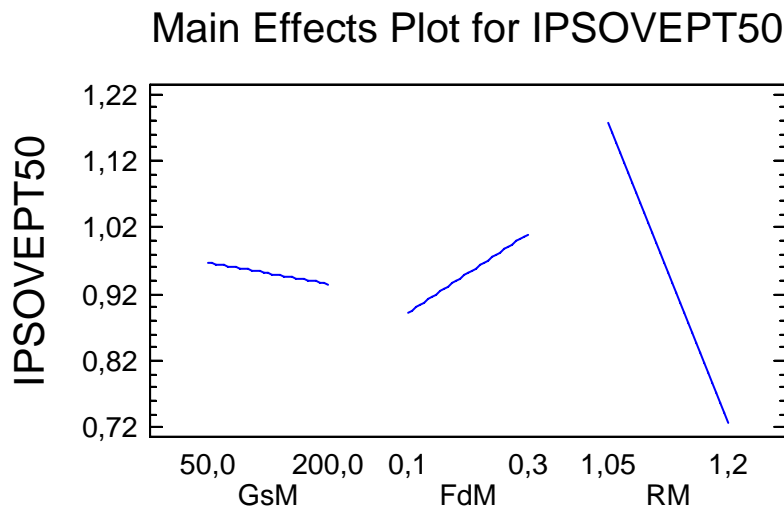


Figura II-42. Gráfico de efectos principales de los factores del conjunto experimental SSD100_P13_50.

Analizando la Figura II-42 se deduce que el valor adecuado son para el factor FmG es 1.2. El valor de los factores GsM y FdM, se pueden establecer libremente. No obstante, el valor elegido para GsM será 50 y para FdM será 0.1, aunque no se pueda afirmar que éstos sean significativos. Por lo tanto, los parámetros quedarían GsM=50, FdM=0.1 y FmG=1.2.

A-III.1.4 Bloque SSD125

A-III.1.4.1 Subconjunto SSD125_P13_20

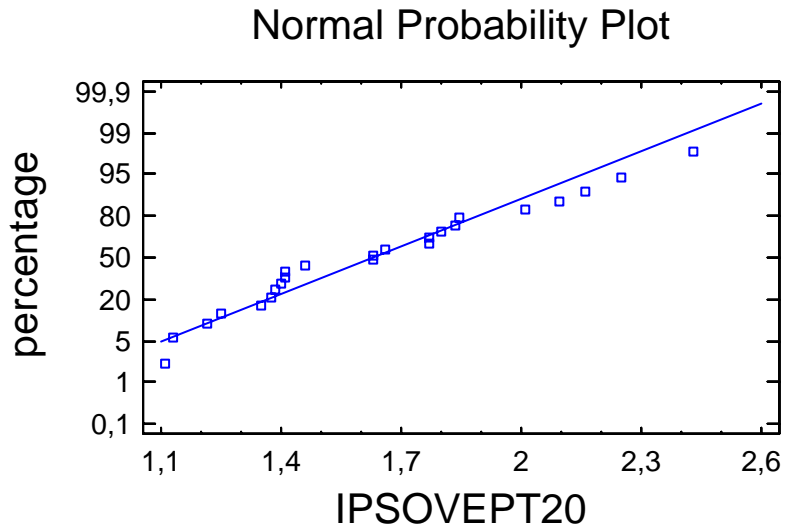


Figura II-43. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPSOVEPT del conjunto SSD125_P13_20 en relación a las frecuencias relativas acumuladas.

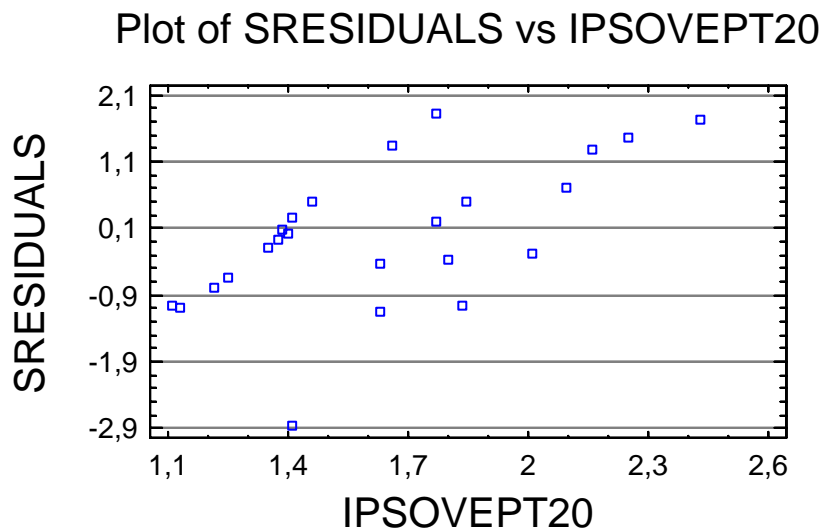


Figura II-44. Residuos estudentizados frente a la variable IPSOVEPT del conjunto experimental SSD125_P13_20.

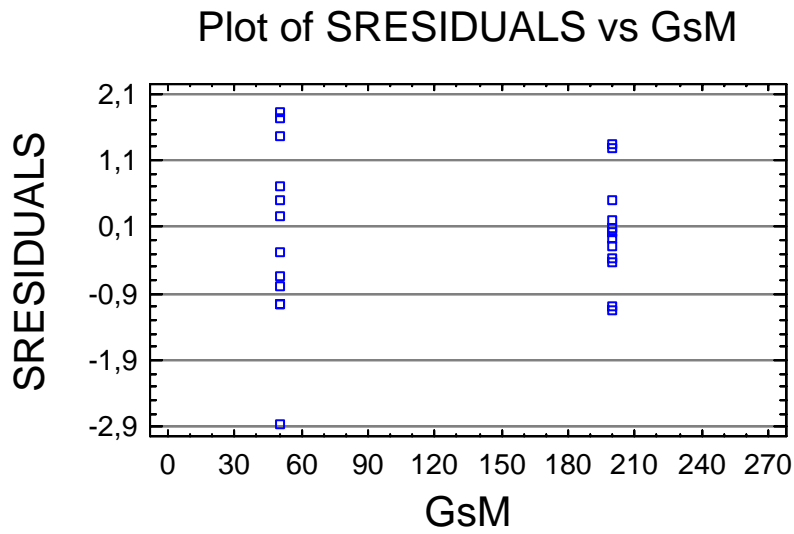


Figura II-45. Residuos estudentizados frente a los niveles del factor GsM.

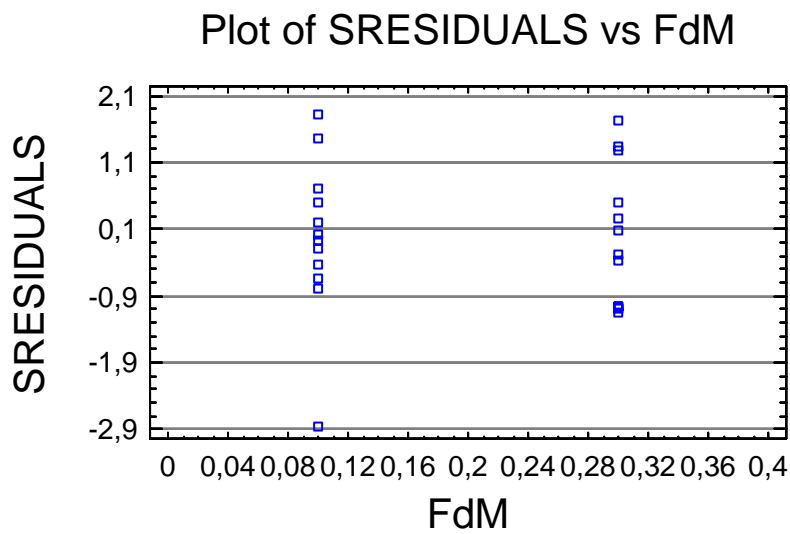


Figura II-46. Residuos estudentizados frente a los niveles del factor FdM.

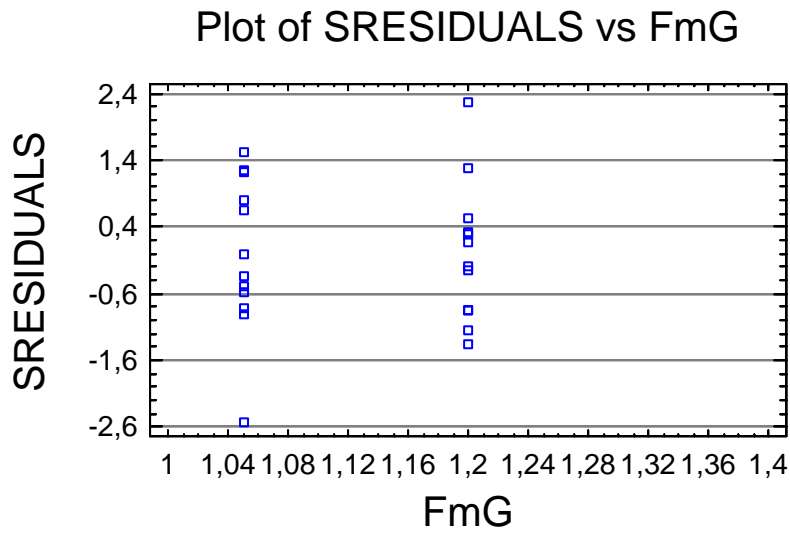


Figura II-47. Residuos estudentizados frente a los niveles del factor FmG.

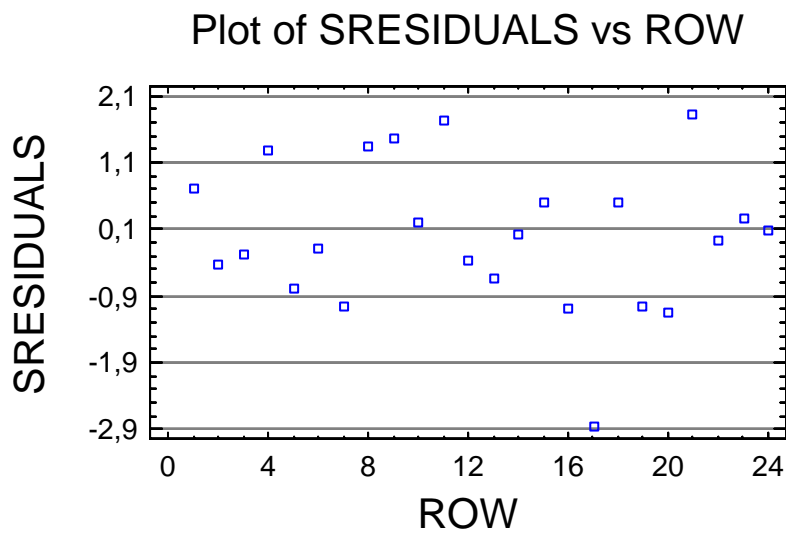


Figura II-48. Residuos estudentizados frente al orden de ejecución.

Analysis of Variance for IPSOVEPT20

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
A:GsM	0,0511366	1	0,0511366	0,68	0,4229
B:FdM	0,0181848	1	0,0181848	0,24	0,6303
C:FmG	1,68171	1	1,68171	22,33	0,0003
AB	0,000680595	1	0,000680595	0,01	0,9255
AC	0,0679722	1	0,0679722	0,90	0,3572
BC	0,0472016	1	0,0472016	0,63	0,4409
blocks	0,0456949	2	0,0228474	0,30	0,7428
Total error	1,12986	15	0,0753238		
Total (corr.)	3,04244	23			

Tabla II-8. ANOVA correspondiente al conjunto experimental SSD125_P13_20.

Observando la Tabla II-8 vemos que disponemos de 24 datos y, tenemos 15 grados de libertad residuales. El contraste de hipótesis sobre las medias para un $\alpha=0,05$ implica que sólo es significativo el efecto principal del factor C, es decir FmG.

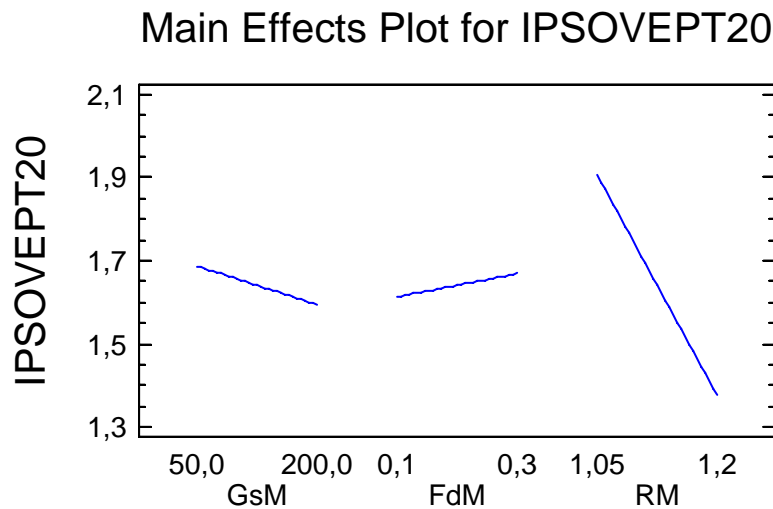


Figura II-49. Gráfico de efectos principales de los factores del conjunto experimental SSD100_P13_20.

Analizando la Figura II-49 se deduce que el valor adecuado son para el factor FmG es 1.2. El valor de los factores GsM y FdM, se pueden establecer libremente. No obstante, el valor elegido para GsM será 200 y para FdM será 0.1, aunque no se pueda afirmar que éstos sean significativos. Por lo tanto, los parámetros quedarían GsM=200, FdM=0.1 y FmG=1.2.

A-III.1.4.2 Subconjunto SSD125_P13_50

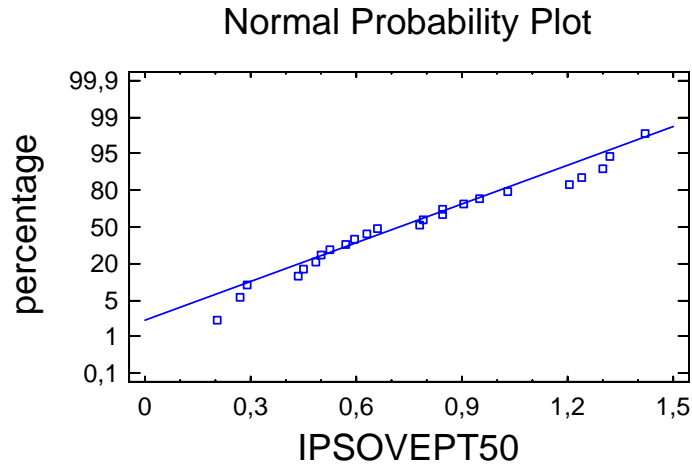


Figura II-50. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPISOVEPT del conjunto SSD125_P13_50 en relación a las frecuencias relativas acumuladas.

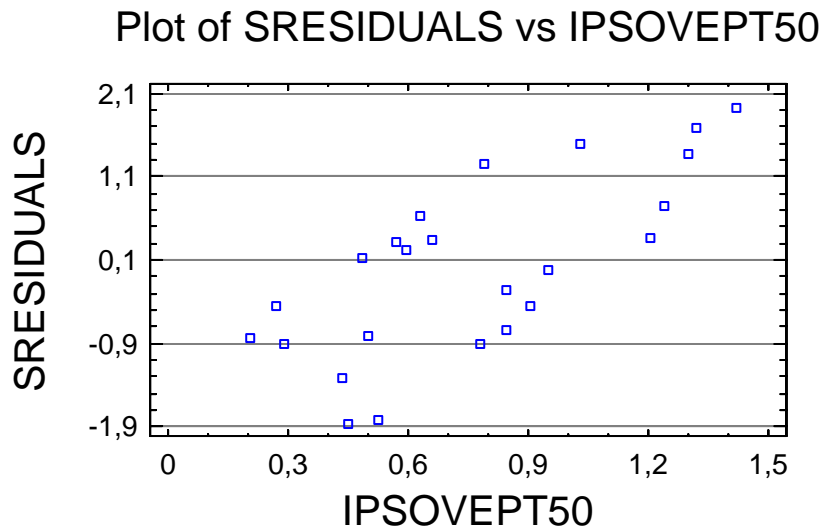


Figura II-51. Residuos estudentizados frente a la variable IPISOVEPT del conjunto experimental SSD125_P13_50.

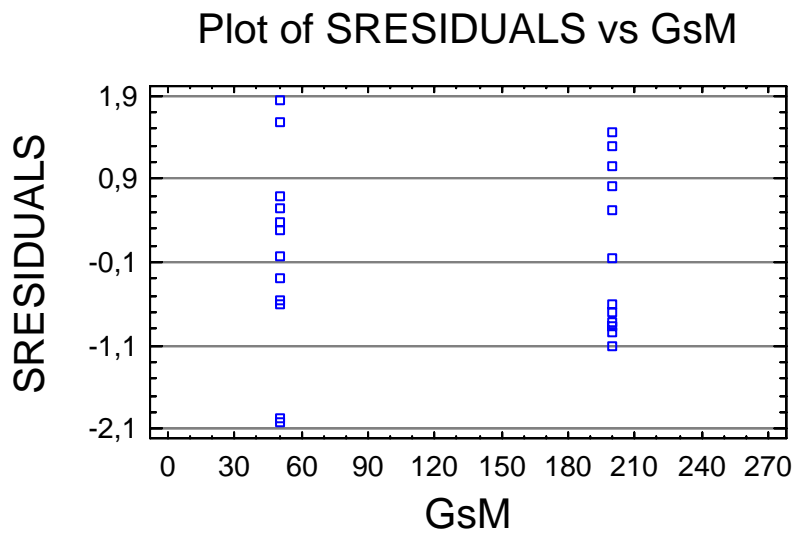


Figura II-52. Residuos estudentizados frente a los niveles del factor GsM.

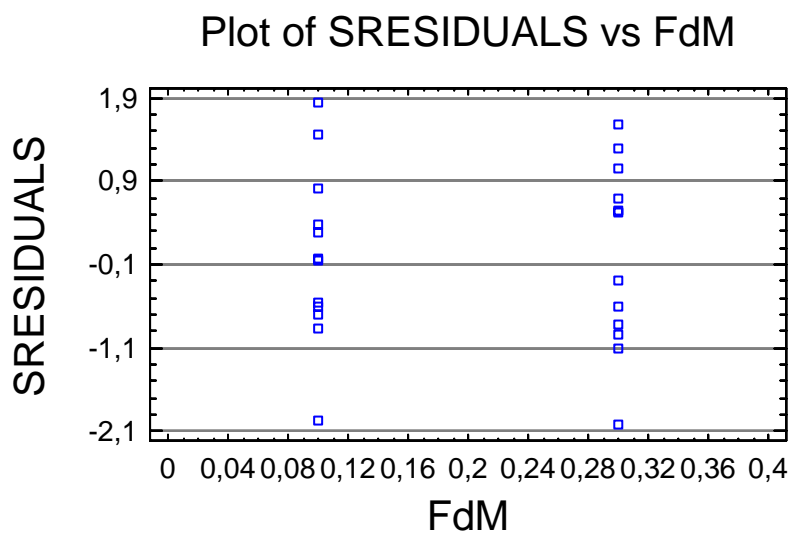


Figura II-53. Residuos estudentizados frente a los niveles del factor FdM.

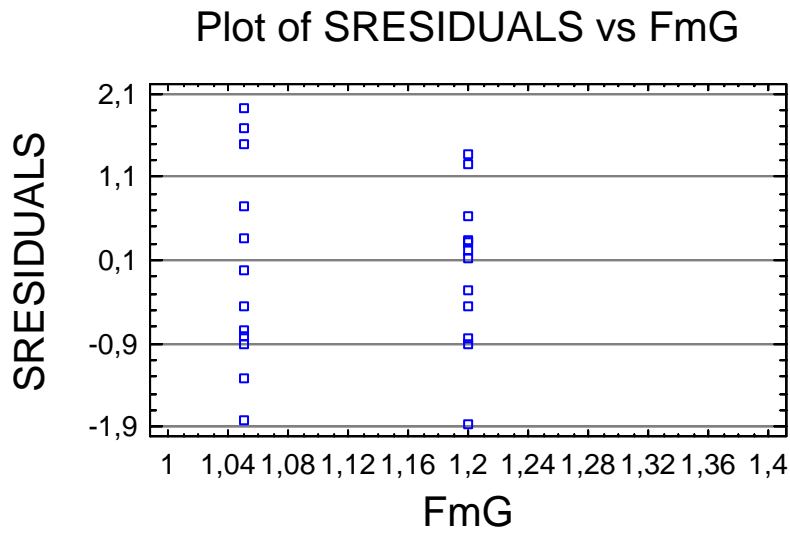


Figura II-54. Residuos estudentizados frente a los niveles del factor FmG.

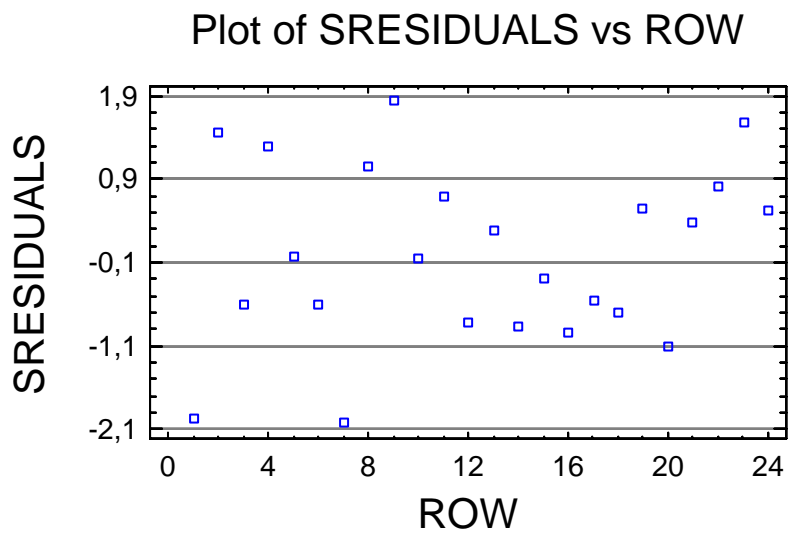


Figura II-55. Residuos estudentizados frente al orden de ejecución.

Analysis of Variance for IPSOVEPT50

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
A:GsM	0,263598	1	0,263598	2,65	0,1246
B:FdM	0,0475548	1	0,0475548	0,48	0,5001
C:FmG	0,686706	1	0,686706	6,90	0,0191
AB	0,159772	1	0,159772	1,60	0,2246
AC	0,00332736	1	0,00332736	0,03	0,8574
BC	0,184779	1	0,184779	1,86	0,1933
blocks	0,0288672	2	0,0144336	0,14	0,8663
Total error	1,49386	15	0,0995904		
Total (corr.)	2,86846	23			

Tabla II-9. ANOVA correspondiente al conjunto experimental SSD125_P13_50.

Observando la Tabla II-9 vemos que disponemos de 24 datos y, tenemos 15 grados de libertad residuales. El contraste de hipótesis sobre las medias para un $\alpha=0,05$ implica que sólo es significativo el efecto principal del factor C, es decir FmG.

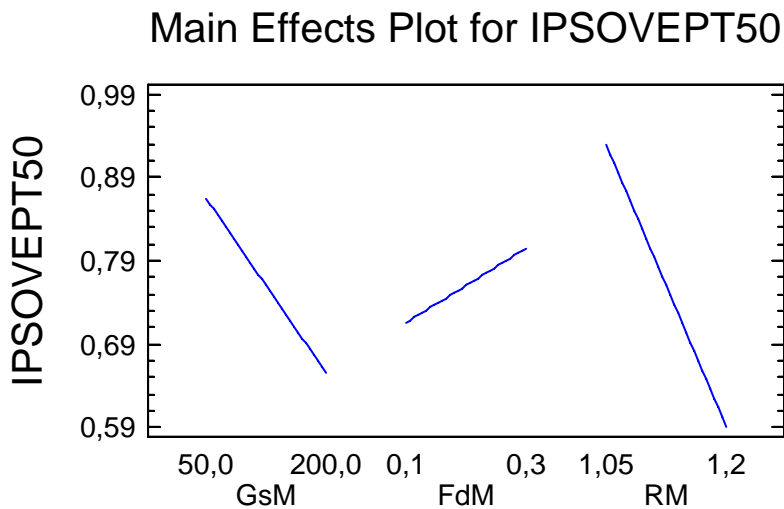


Figura II-56. Gráfico de efectos principales de los factores del conjunto experimental SSD125_P13_50.

Analizando la Figura II-56 se deduce que el valor adecuado son para el factor FmG es 1.2. El valor de los factores GsM y FdM, se pueden establecer libremente. No obstante, el valor elegido para GsM será 200 y para FdM será 0.1, aunque no se pueda afirmar que éstos sean significativos. Por lo tanto, los parámetros quedarían GsM=200, FdM=0.1 y FmG=1.2.

A-III.2 Ajuste de Parámetros SMAGA-1 para la Configuración P3

A-III.2.1 Bloque SSD10

A-III.2.1.1 Subconjunto SSD10_P3_20

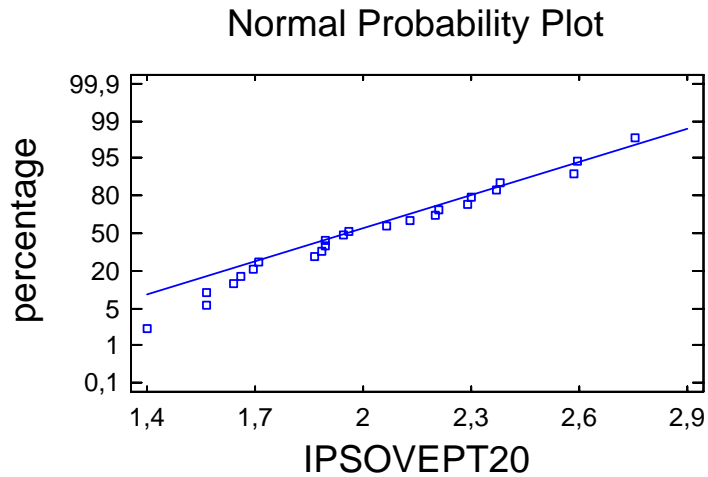


Figura II-57. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPISOVEPT del conjunto SSD10_P3_20 en relación a las frecuencias relativas acumuladas.

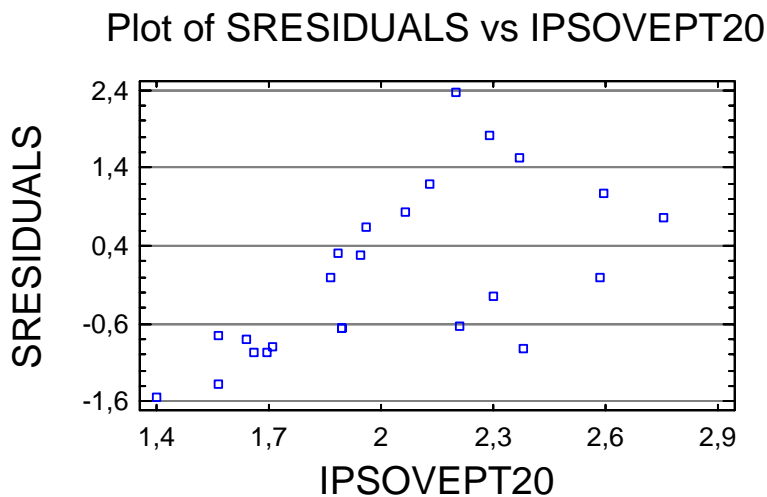


Figura II-58. Residuos estudentizados frente a la variable IPISOVEPT del conjunto experimental SSD10_P3_20.

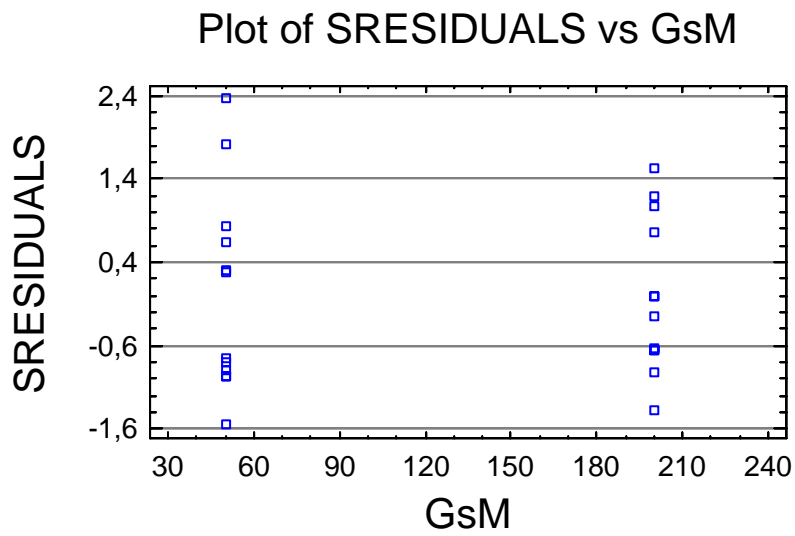


Figura II-59. Residuos estudentizados frente a los niveles del factor GsM.

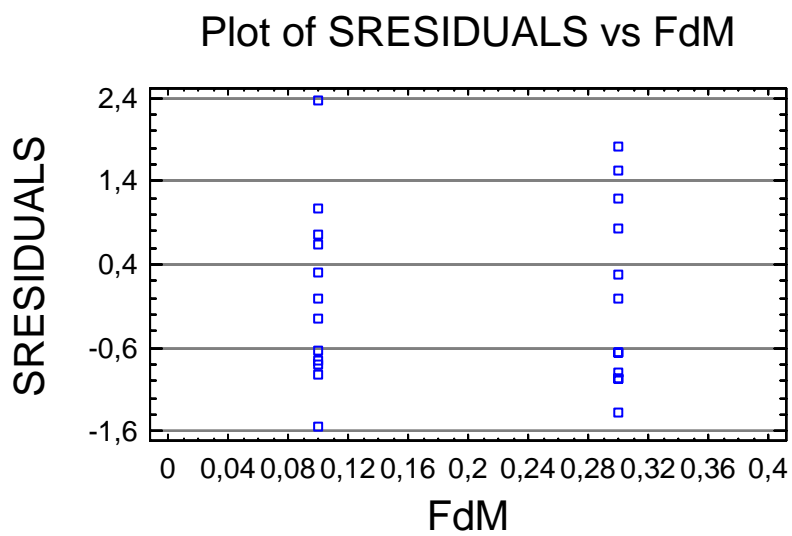


Figura II-60. Residuos estudentizados frente a los niveles del factor FdM.

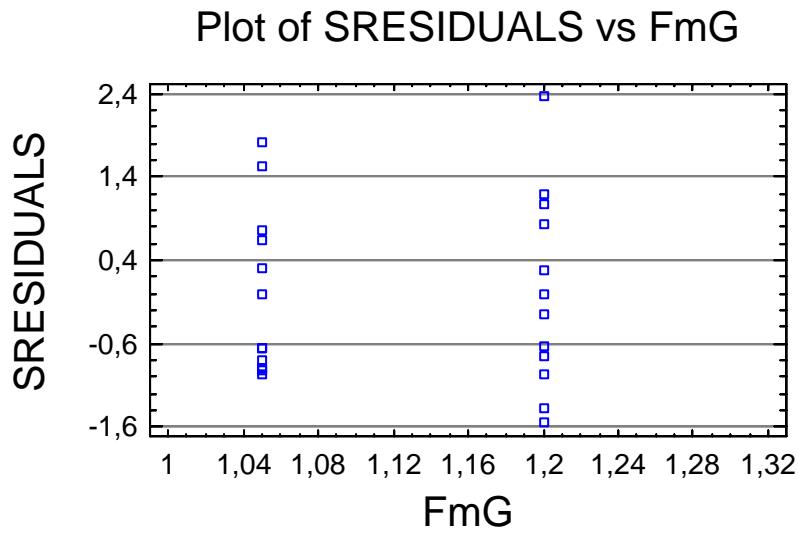


Figura II-61. Residuos estudentizados frente a los niveles del factor FmG.

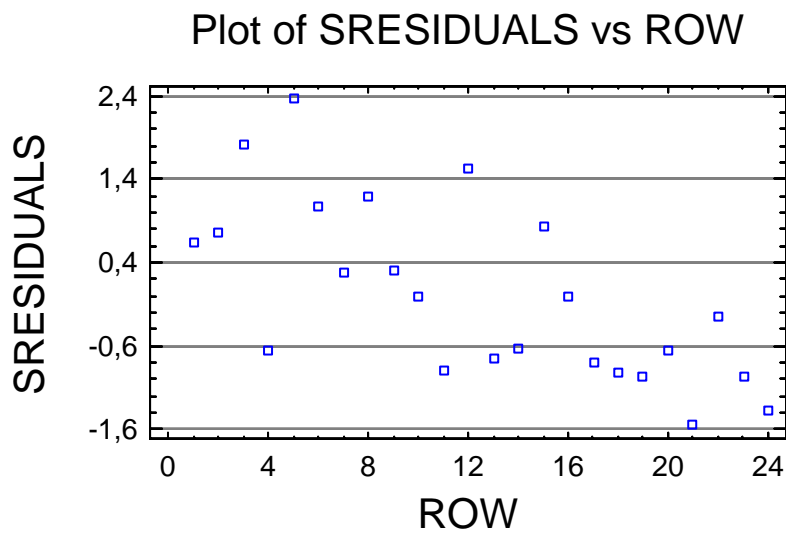


Figura II-62. Residuos estudentizados frente al orden de ejecución.

Analysis of Variance for IPSOVEPT20

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
A:GsM	0,852452	1	0,852452	12,12	0,0029
B:FdM	0,23617	1	0,23617	3,36	0,0845
C:FmG	0,101766	1	0,101766	1,45	0,2455
AB	0,610131	1	0,610131	8,67	0,0091
AC	0,0316892	1	0,0316892	0,45	0,5111
BC	0,0038395	1	0,0038395	0,05	0,8181
Total error	1,19569	17	0,0703348		
Total (corr.)	3,03174	23			

Tabla II-10. ANOVA correspondiente al conjunto experimental SSD10_P3_20.

Observando la Tabla II-10 vemos que disponemos de 24 datos y tenemos 17 grados de libertad residuales. El contraste de hipótesis sobre las medias para un $\alpha=0,05$ implica que es significativo el efecto principal del factor A, y el efecto combinado AB, es decir de los factores GsM y FdM.

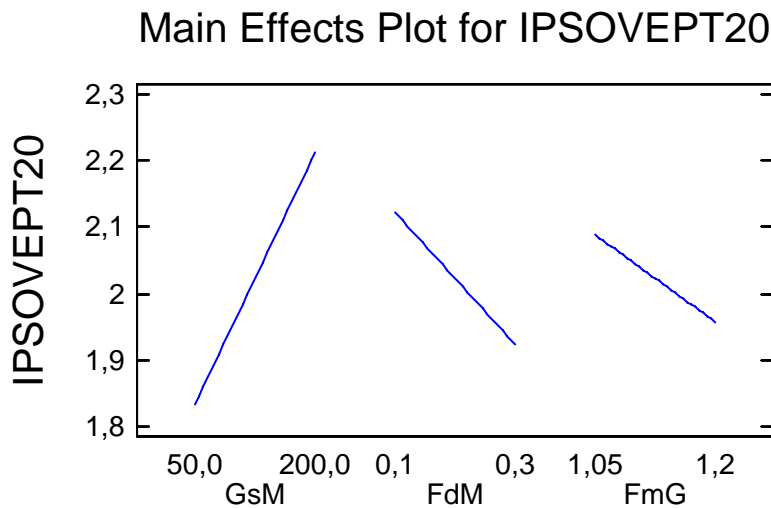


Figura II-63. Gráfico de efectos principales de los factores del conjunto experimental SSD10_P3_20.

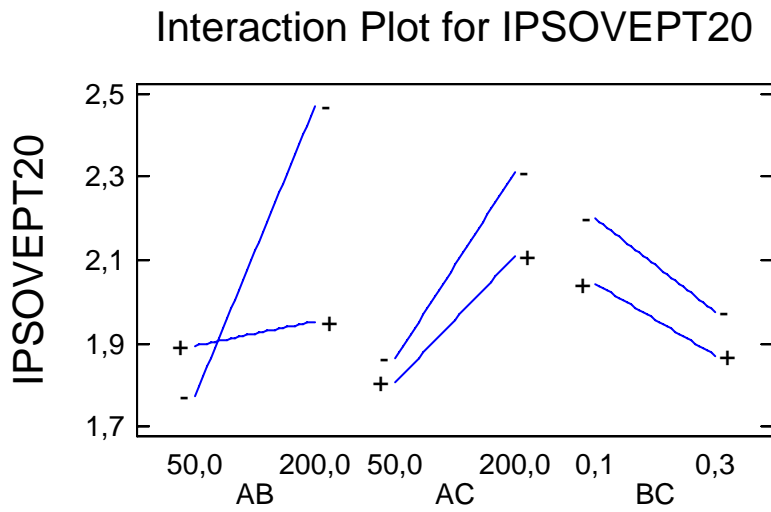


Figura II-64. Gráfico de interacción entre los factores del conjunto experimental SSD10_P3_20.

Analizando la Figura II-64 se deduce que el valor adecuado son para el factor GsM es 50, en combinación con el valor más bajo del parámetro FdM, es decir, con 0.1. El valor del factor FmG, se puede establecer libremente. No obstante, el valor elegido para FmG será 1.2. Por lo tanto, los parámetros quedarían GsM=50, FdM=0.1 y FmG=1.2.

A-III.2.1.2 Subconjunto SSD10_P3_50

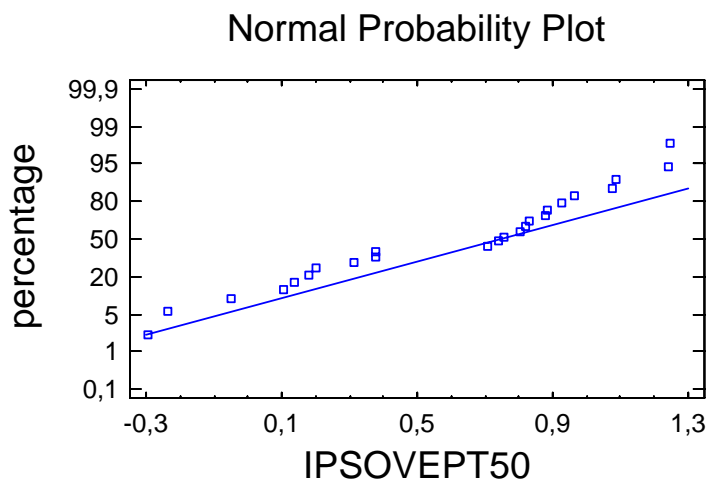


Figura II-65. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPSOVEPT del conjunto SSD10_P3_50 en relación a las frecuencias relativas acumuladas.

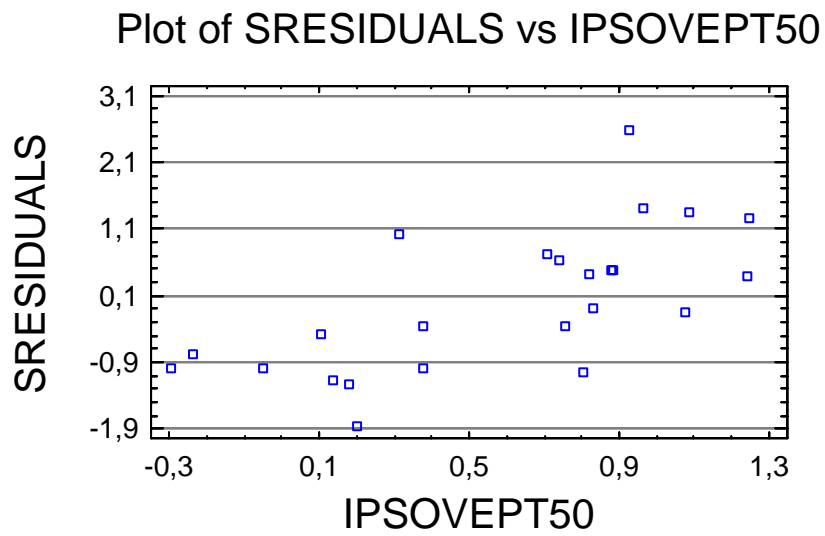


Figura II-66. Residuos estudentizados frente a la variable IPSOVEPT del conjunto experimental SSD10_P3_50.

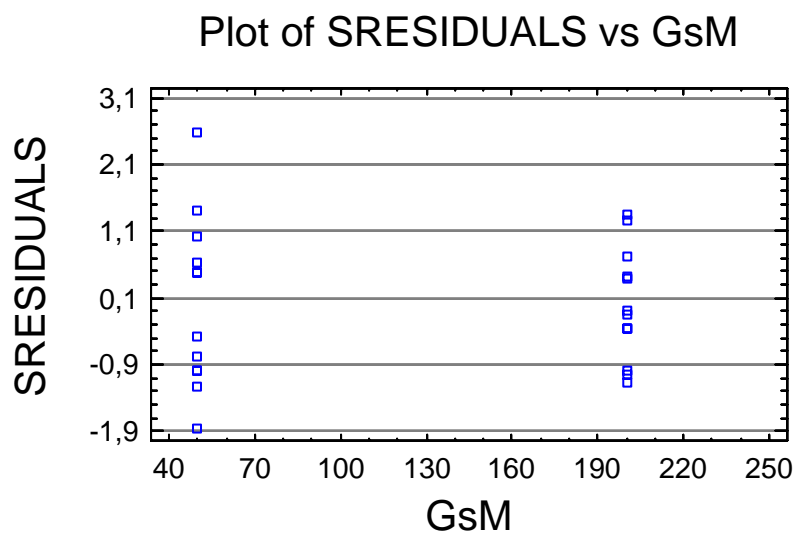


Figura II-67. Residuos estudentizados frente a los niveles del factor GsM.

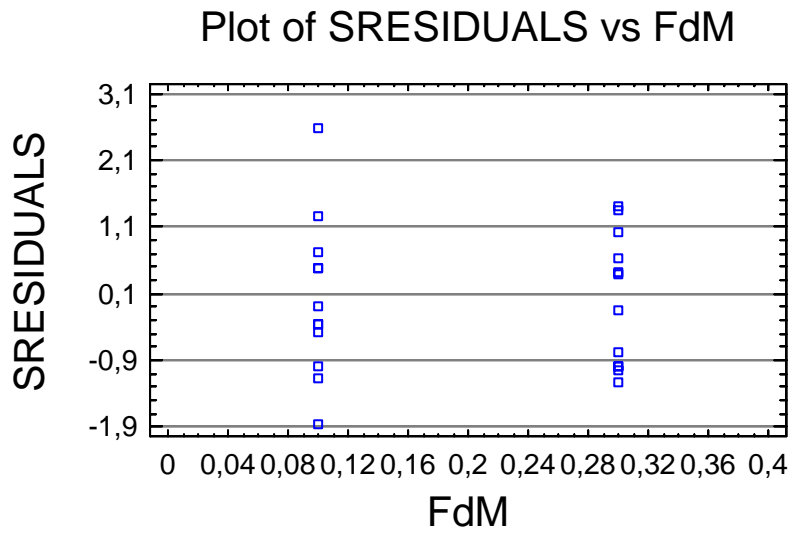


Figura II-68. Residuos estudentizados frente a los niveles del factor FdM.

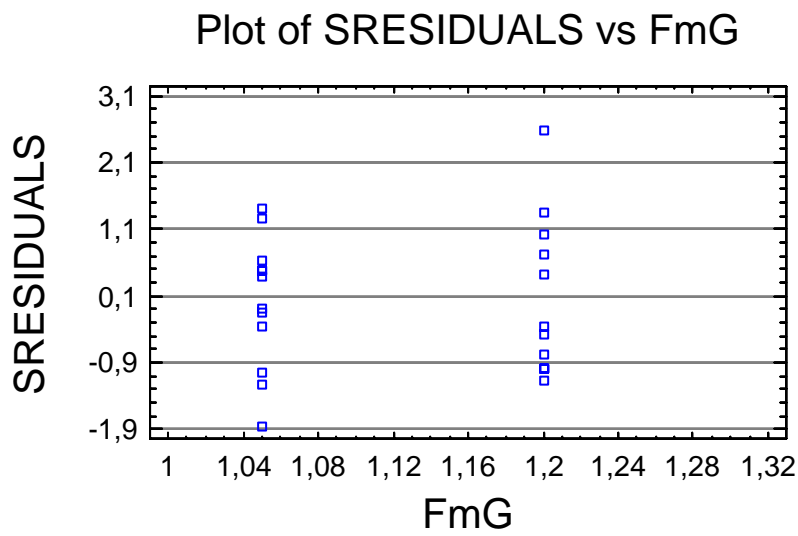


Figura II-69. Residuos estudentizados frente a los niveles del factor FmG.

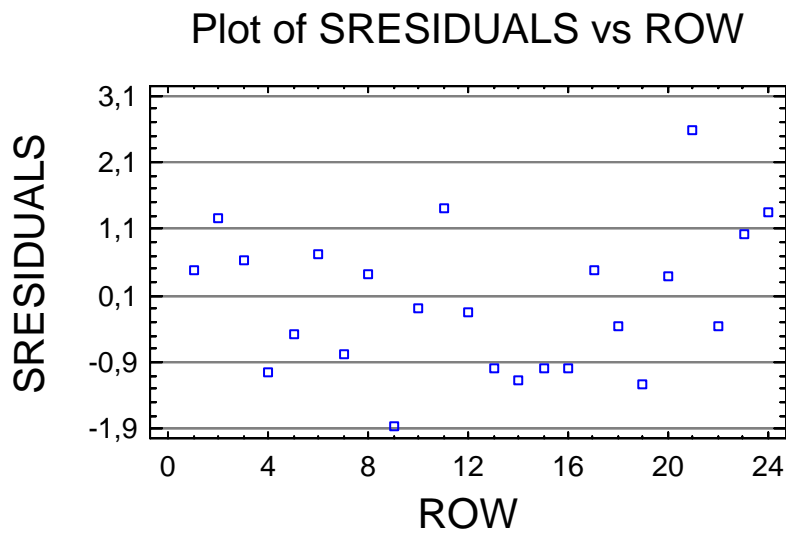


Figura II-70. Residuos estudentizados frente al orden de ejecución.

Analysis of Variance for IPSOVEPT50

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
A:GsM	0,973443	1	0,973443	7,46	0,0142
B:FdM	0,000178228	1	0,000178228	0,00	0,9709
C:FmG	1,27339	1	1,27339	9,76	0,0062
AB	0,288179	1	0,288179	2,21	0,1555
AC	0,0170192	1	0,0170192	0,13	0,7224
BC	0,00484283	1	0,00484283	0,04	0,8495
Total error	2,21794	17	0,130467		
Total (corr.)	4,775	23			

Tabla II-11. ANOVA correspondiente al conjunto experimental SSD10_P3_50.

Observando la Tabla II-11 vemos que disponemos de 24 datos y tenemos 17 grados de libertad residuales. El contraste de hipótesis sobre las medias para un $\alpha=0,05$ implica que son significativos los efectos principal del factor A y C, es decir los factores GsM y FmG.

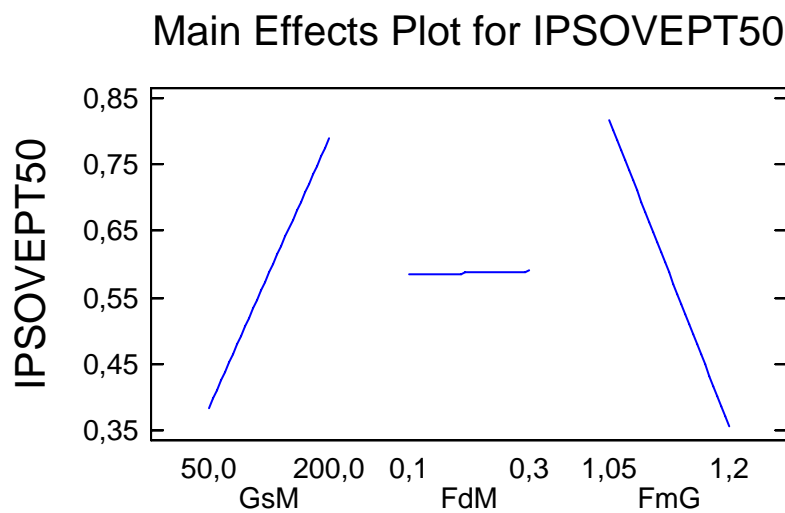


Figura II-71. Gráfico de efectos principales de los factores del conjunto experimental SSD10_P3_50.

Analizando la Figura II-71 se deduce que el valor adecuado son para el factor GsM es 50, y del factor FmG es 1.2. El valor del factor FdM, se puede establecer libremente. No obstante, el valor elegido para FdM será 0.1. Por lo tanto, los parámetros quedarían GsM=50, FdM=0.1 y FmG=1.2.

A-III.2.2 Bloque SSD50

A-III.2.2.1 Subconjunto SSD50_P3_50

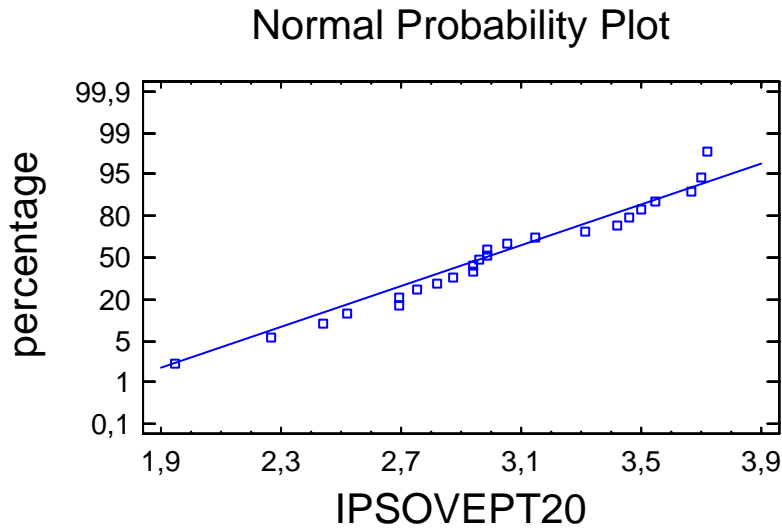


Figura II-72. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPSOVEPT del conjunto SSD50_P3_20 en relación a las frecuencias relativas acumuladas.

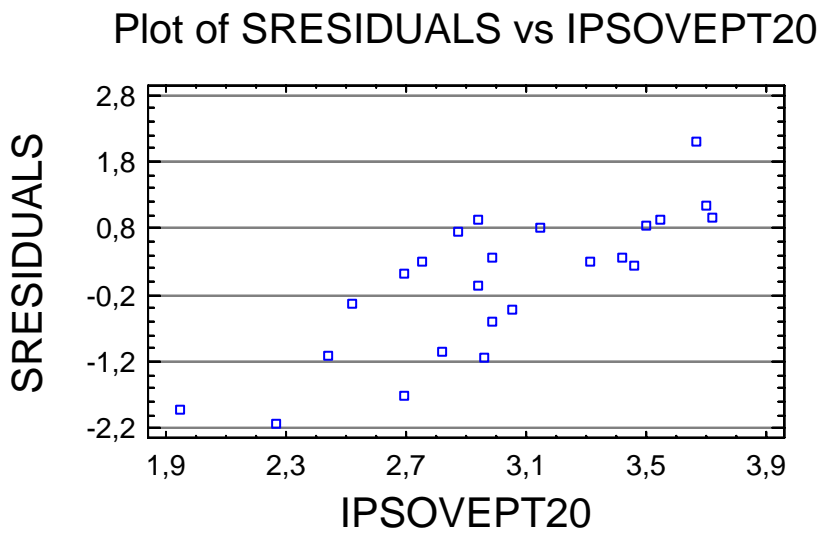


Figura II-73. Residuos estudentizados frente a la variable IPSOVEPT del conjunto experimental SSD50_P3_20.

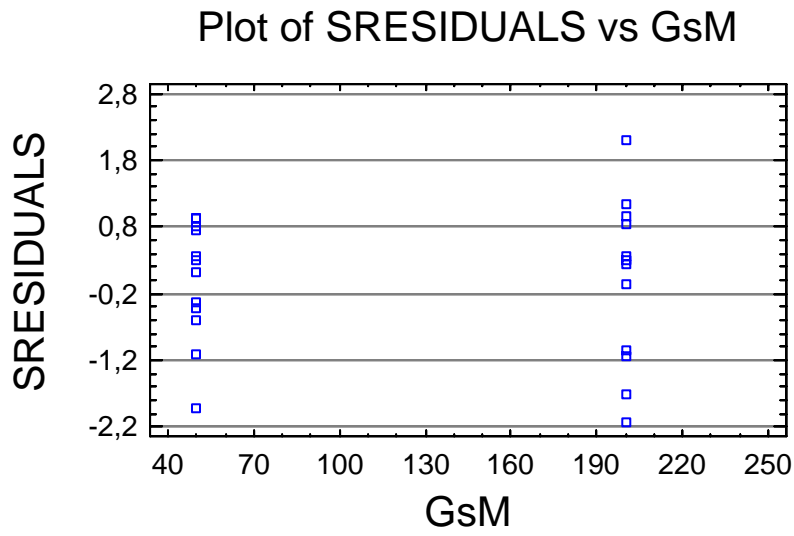


Figura II-74. Residuos estudentizados frente a los niveles del factor GsM.

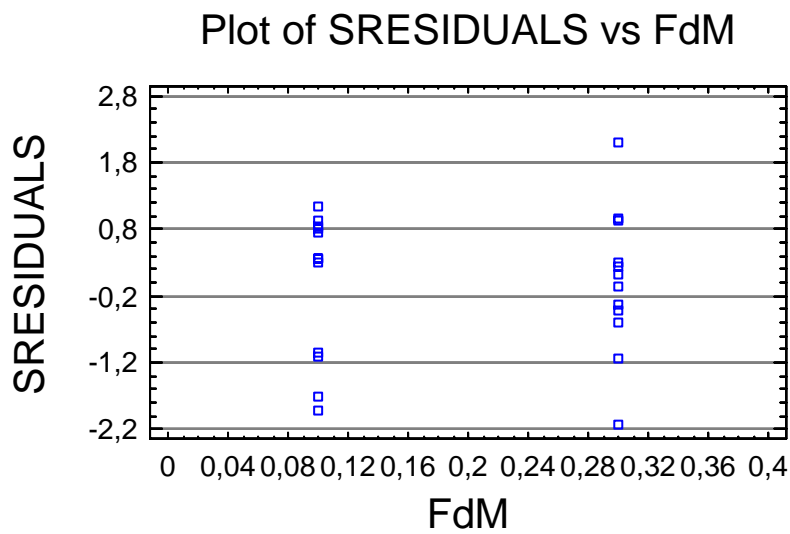


Figura II-75. Residuos estudentizados frente a los niveles del factor FdM.

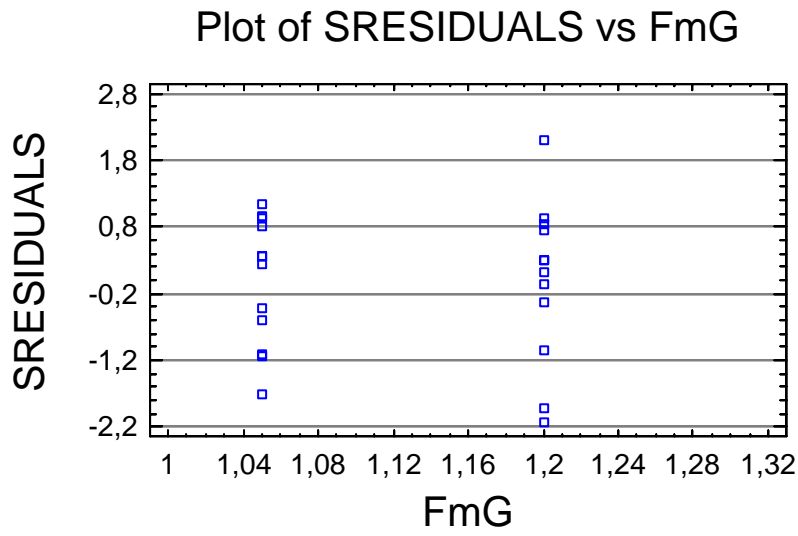


Figura II-76. Residuos estudentizados frente a los niveles del factor FmG.

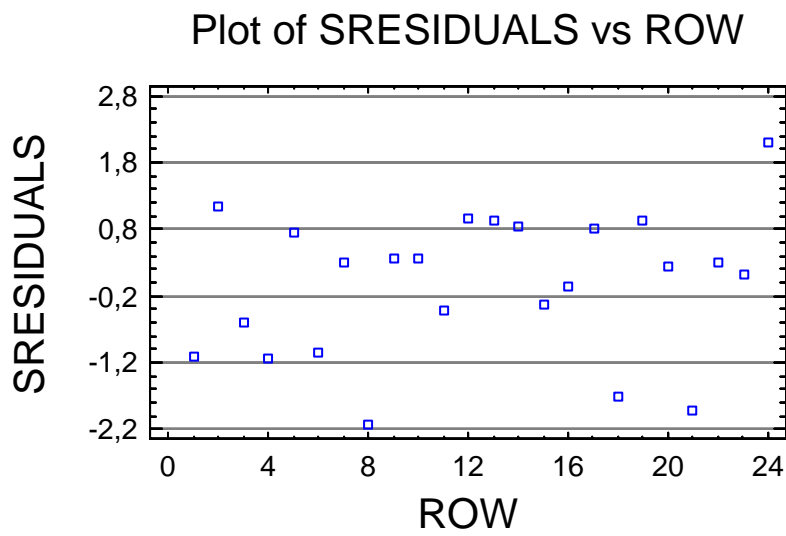


Figura II-77. Residuos estudentizados frente al orden de ejecución.

Analysis of Variance for IPSOVEPT20

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
A:GsM	0,873161	1	0,873161	4,68	0,0451
B:FdM	0,0266483	1	0,0266483	0,14	0,7102
C:FmG	0,632221	1	0,632221	3,39	0,0832
AB	0,114931	1	0,114931	0,62	0,4434
AC	0,040188	1	0,040188	0,22	0,6485
BC	0,148618	1	0,148618	0,80	0,3846
Total error	3,17207	17	0,186593		
Total (corr.)	5,00784	23			

Tabla II-12. ANOVA correspondiente al conjunto experimental SSD50_P3_20.

Observando la Tabla II-12 vemos que disponemos de 24 datos y tenemos 17 grados de libertad residuales. El contraste de hipótesis sobre las medias para un $\alpha=0,05$ implica que es significativo el efecto principal del factor A, es decir el factor GsM.

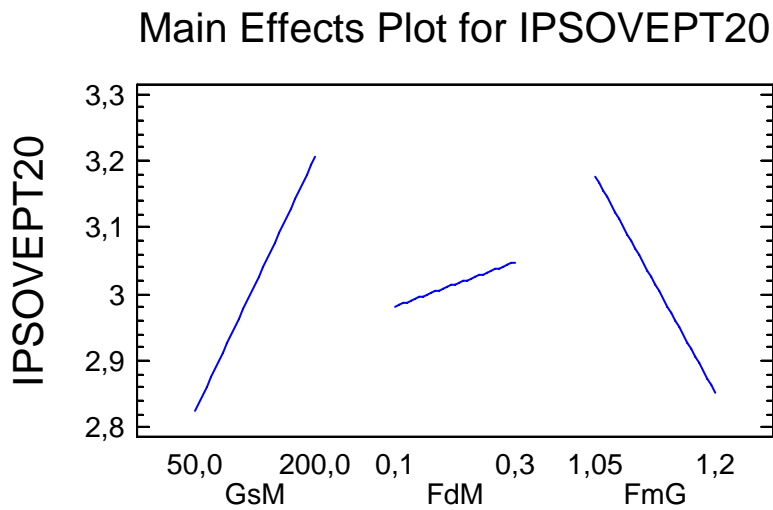


Figura II-78. Gráfico de efectos principales de los factores del conjunto experimental SSD50_P3_20.

Analizando la Figura II-78 se deduce que el valor adecuado son para el factor GsM es 50. El valor de los factores FdM y FmG se puede establecer libremente. No obstante, el valor elegido para FdM es 0.3 y para FmG es 1.2. Por lo tanto, los parámetros quedarían GsM=50, FdM=0.3 y FmG=1.2.

A-III.2.2.2 Subconjunto SSD50_P3_50

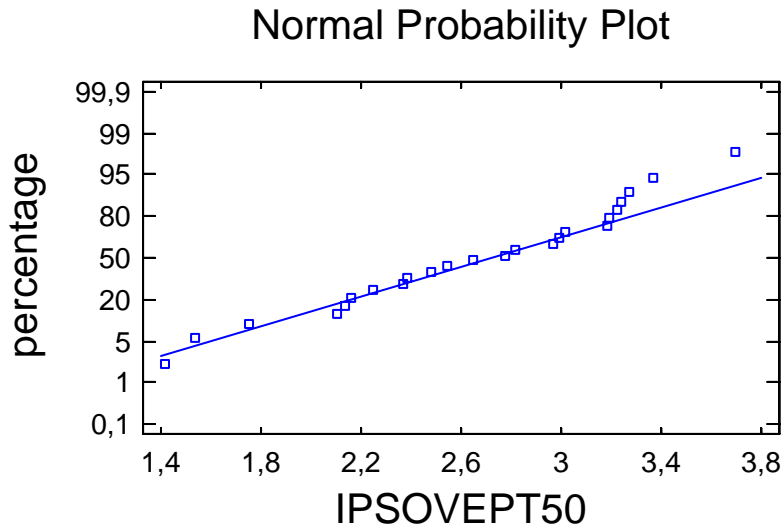


Figura II-79. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPISOVEPT del conjunto SSD50_P3_50 en relación a las frecuencias relativas acumuladas.

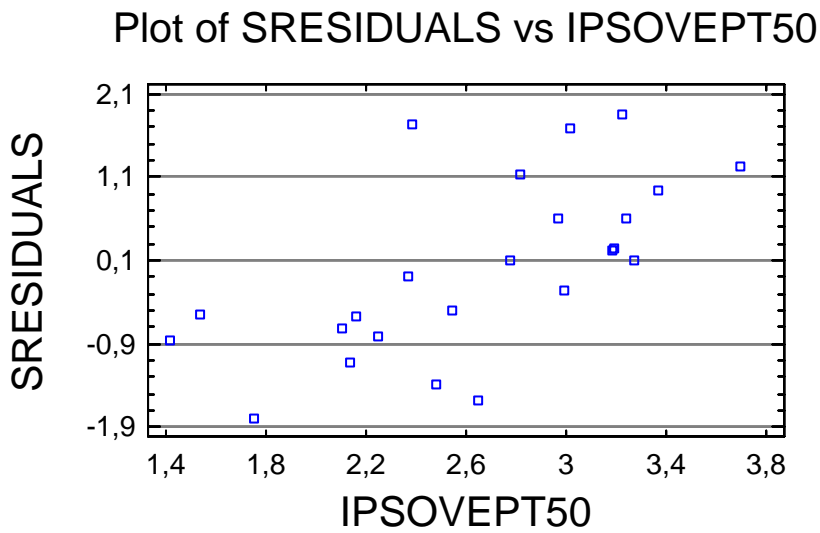


Figura II-80. Residuos estudentizados frente a la variable IPISOVEPT del conjunto experimental SSD50_P3_50.

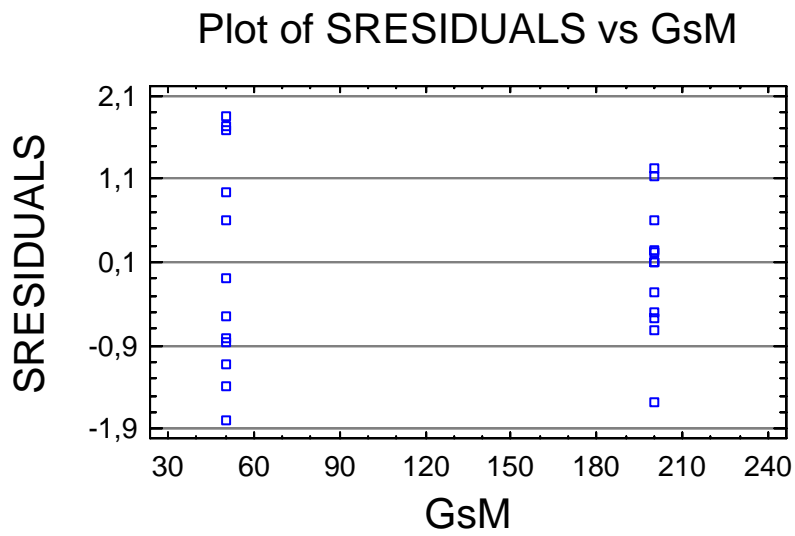


Figura II-81. Residuos estudentizados frente a los niveles del factor GsM.

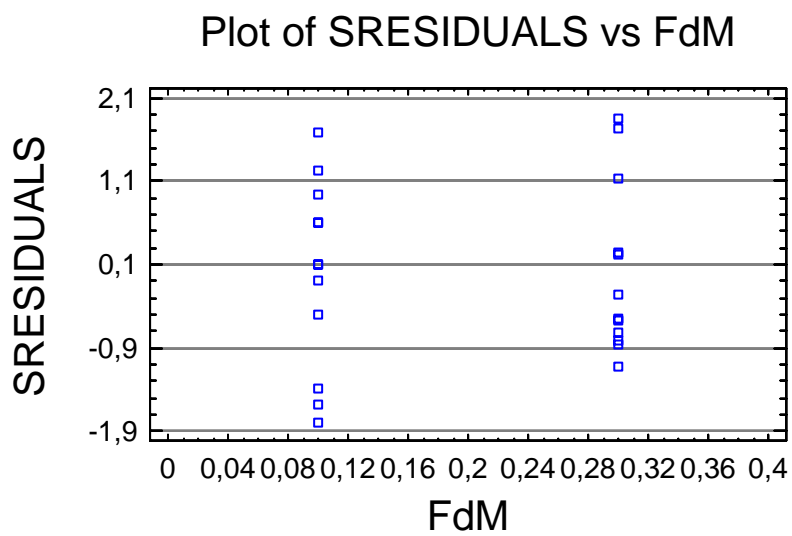


Figura II-82. Residuos estudentizados frente a los niveles del factor FdM.

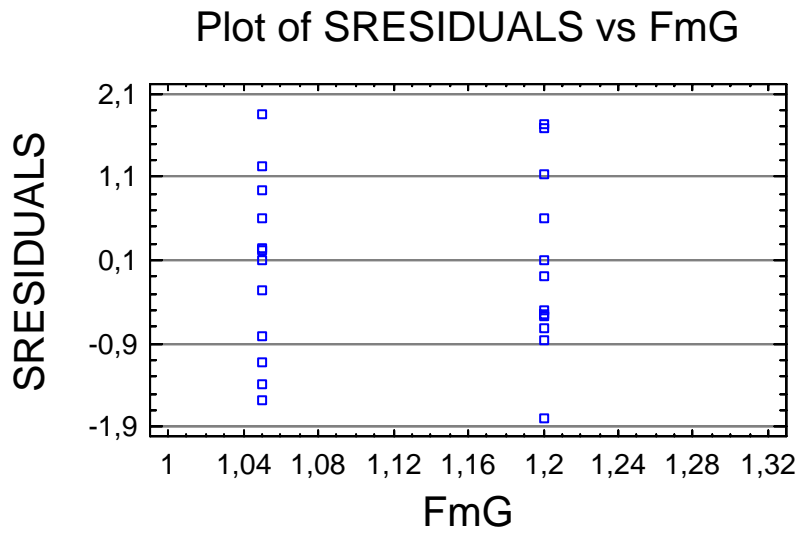


Figura II-83. Residuos estudentizados frente a los niveles del factor FmG.

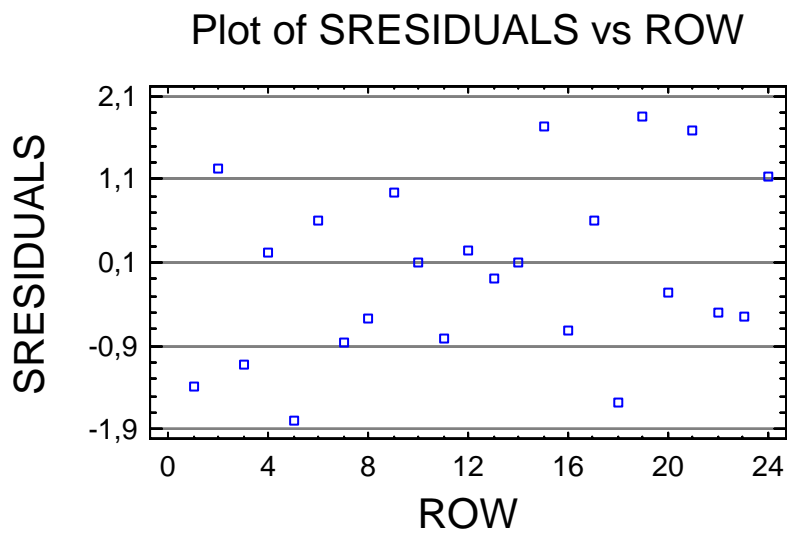


Figura II-84. Residuos estudentizados frente al orden de ejecución.

Analysis of Variance for IPSOVEPT50

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
A:GsM	1,11928	1	1,11928	5,40	0,0328
B:FdM	0,932839	1	0,932839	4,50	0,0489
C:FmG	2,55962	1	2,55962	12,34	0,0027
AB	0,138847	1	0,138847	0,67	0,4245
AC	0,0166177	1	0,0166177	0,08	0,7805
BC	0,0683413	1	0,0683413	0,33	0,5734
Total error	3,52544	17	0,207379		
Total (corr.)	8,36098	23			

Tabla II-13. ANOVA correspondiente al conjunto experimental SSD50_P3_50.

Observando la Tabla II-13 vemos que disponemos de 24 datos y tenemos 17 grados de libertad residuales. El contraste de hipótesis sobre las medias para un $\alpha=0,05$ implica que son significativos los efectos principal del factor A, B y C, es decir los factores GsM, FdM y FmG.

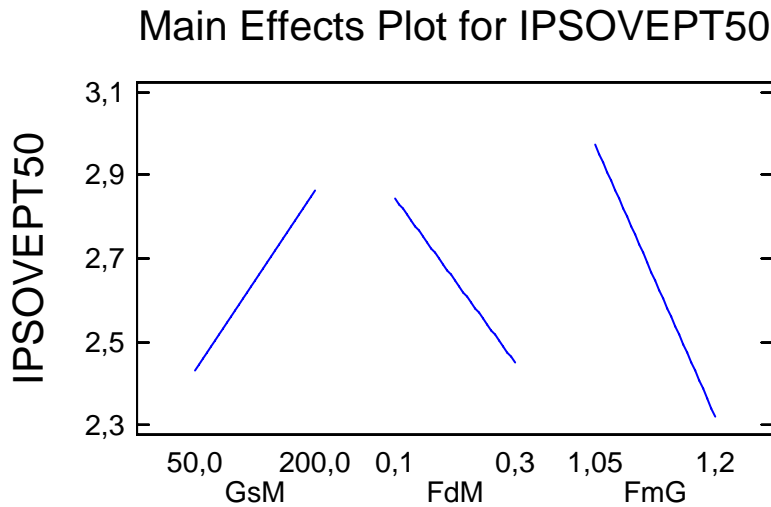


Figura II-85. Gráfico de efectos principales de los factores del conjunto experimental SSD50_P3_50.

Analizando la Figura II-85 se deduce que el valor adecuado son para el factor GsM es 50, y del factor FdM es 0.3 y el del factor FmG es 1.2. Por lo tanto, los parámetros quedarían GsM=50, FdM=0.3 y FmG=1.2.

A-III.2.3 Bloque SSD100

A-III.2.3.1 Subconjunto SSD100_P3_50

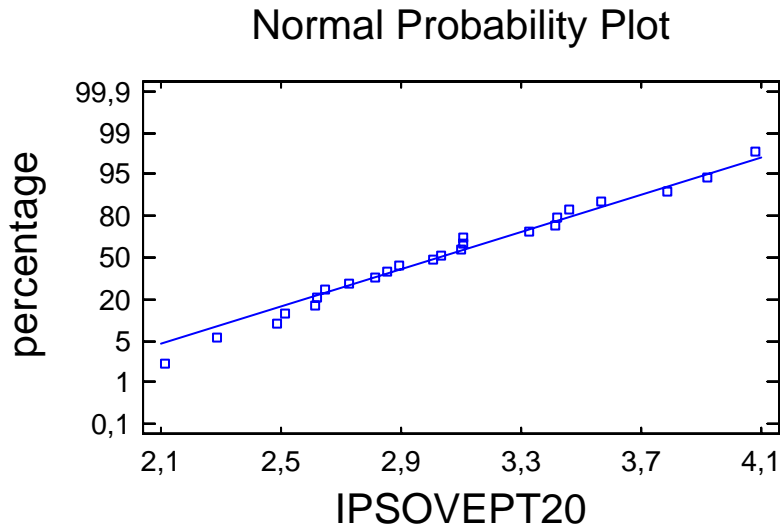


Figura II-86. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPSOVEPT del conjunto SSD100_P3_20 en relación a las frecuencias relativas acumuladas.

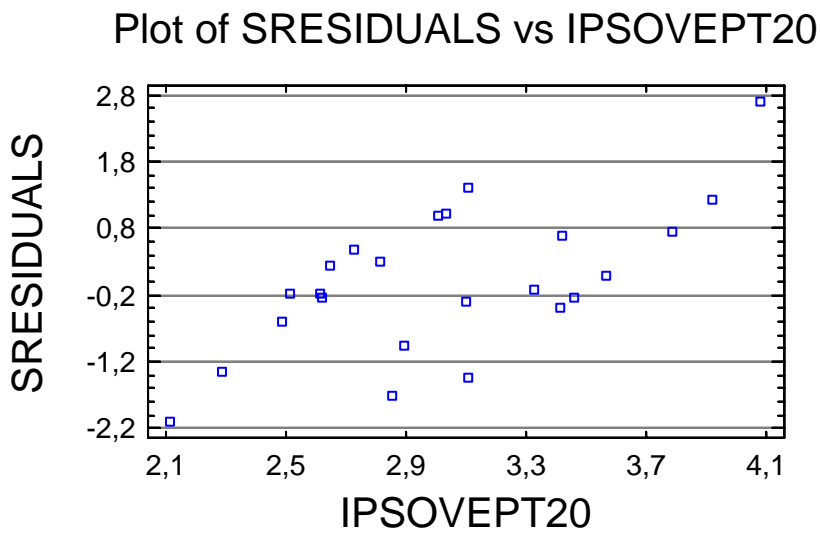


Figura II-87. Residuos estudentizados frente a la variable IPSOVEPT del conjunto experimental SSD100_P3_20.

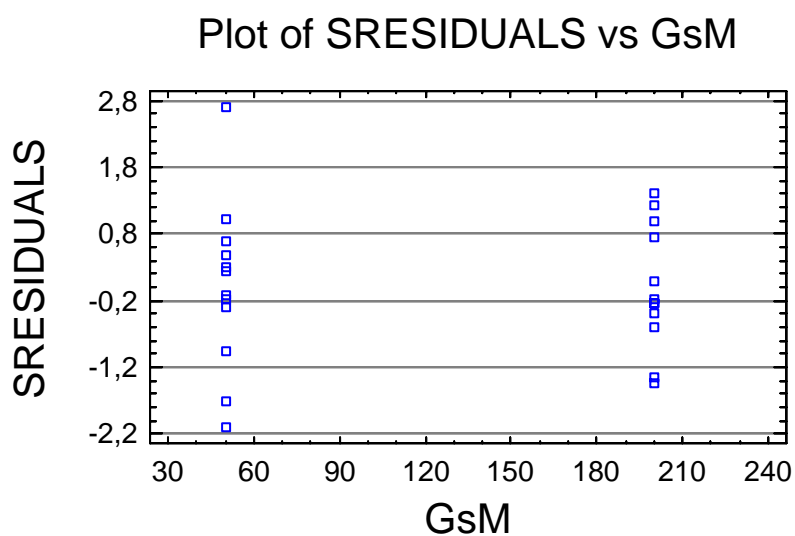


Figura II-88. Residuos estudentizados frente a los niveles del factor GsM.

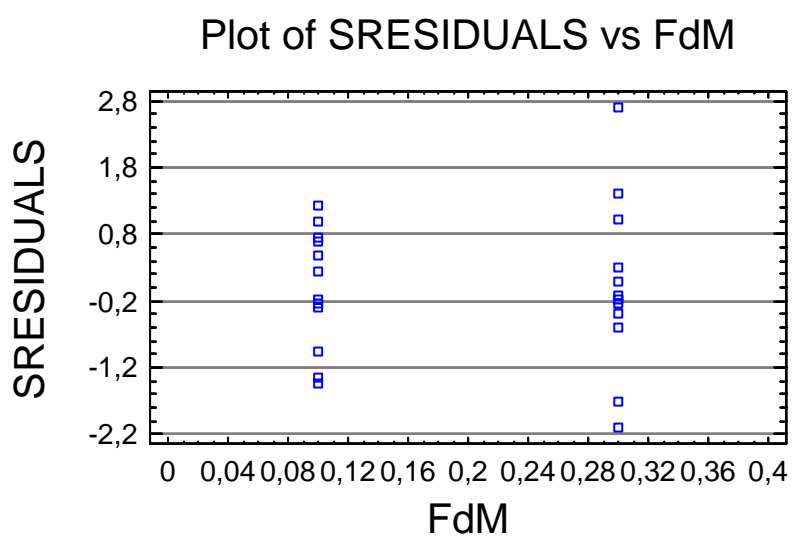


Figura II-89. Residuos estudentizados frente a los niveles del factor FdM.

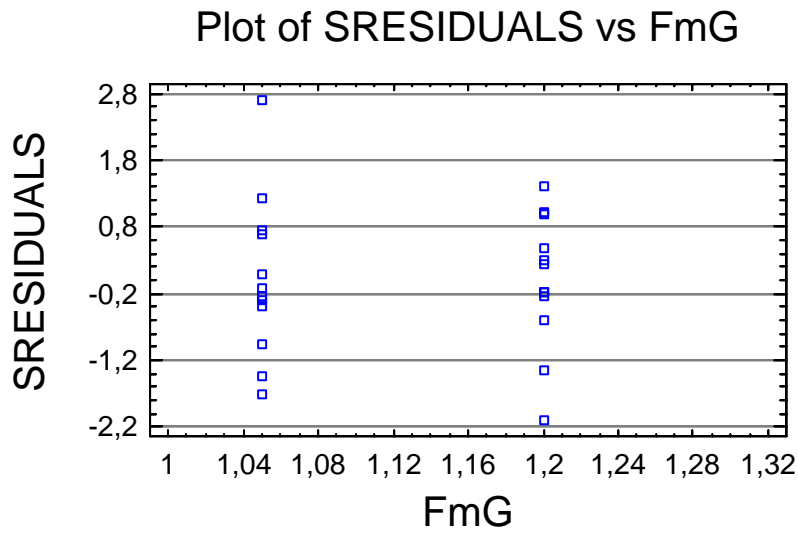


Figura II-90. Residuos estudentizados frente a los niveles del factor FmG.

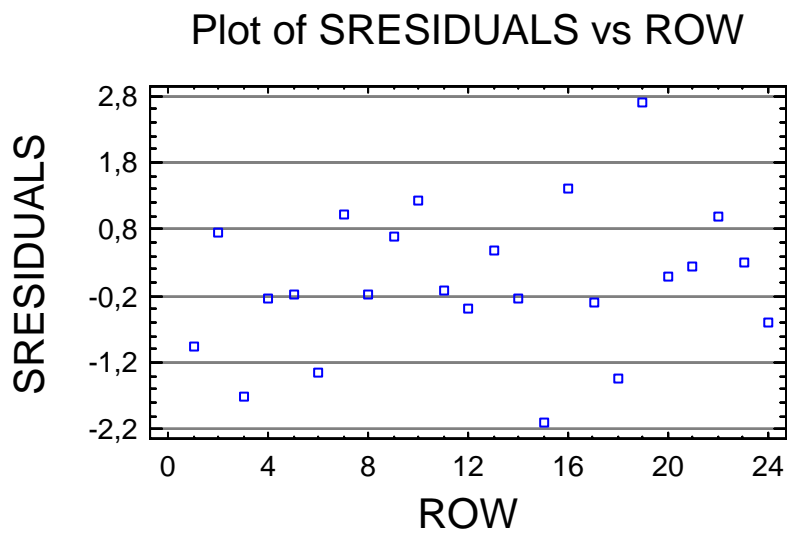


Figura II-91. Residuos estudentizados frente al orden de ejecución.

Analysis of Variance for IPSOVEPT20

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
A:GsM	0,142373	1	0,142373	1,03	0,3237
B:FdM	0,0306171	1	0,0306171	0,22	0,6434
C:FmG	3,32803	1	3,32803	24,15	0,0001
AB	0,042376	1	0,042376	0,31	0,5864
AC	0,0710052	1	0,0710052	0,52	0,4826
BC	0,000538061	1	0,000538061	0,00	0,9509
Total error	2,34285	17	0,137815		
Total (corr.)	5,95779	23			

Tabla II-14. ANOVA correspondiente al conjunto experimental SSD100_P3_20.

Observando la Tabla II-14 vemos que disponemos de 24 datos y tenemos 17 grados de libertad residuales. El contraste de hipótesis sobre las medias para un $\alpha=0,05$ implica que es significativo el efecto principal del factor C, es decir el factor FmG.

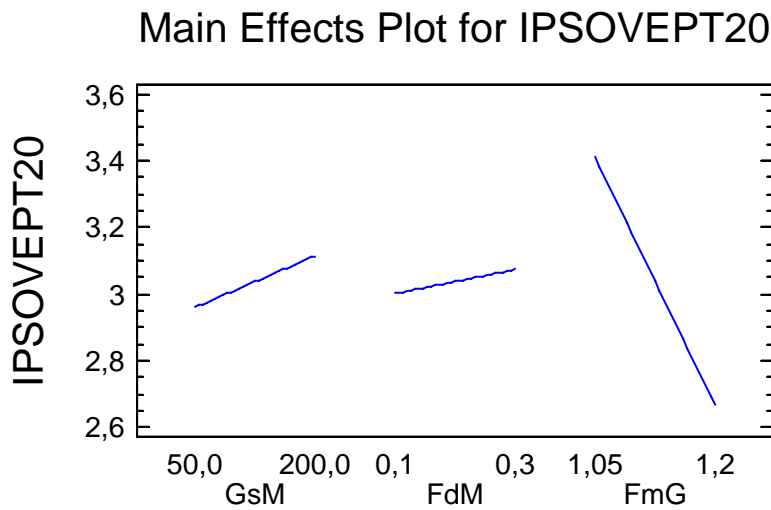


Figura II-92. Gráfico de efectos principales de los factores del conjunto experimental SSD100_P3_20.

Analizando la Figura II-92 se deduce que el valor adecuado son para el factor FmG es 1.2. El valor de los factores GsM y FdM se puede establecer libremente. No obstante, el valor elegido para GsM es 50 y para FdM es 0.3. Por lo tanto, los parámetros quedarían GsM=50, FdM=0.3 y FmG=1.2.

A-III.2.3.2 Subconjunto SSD100_P3_50

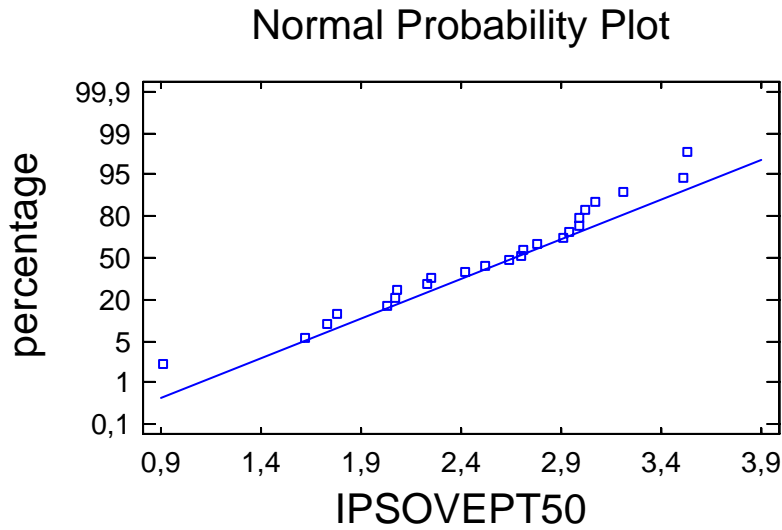


Figura II-93. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPISOVEPT del conjunto SSD100_P3_50 en relación a las frecuencias relativas acumuladas.

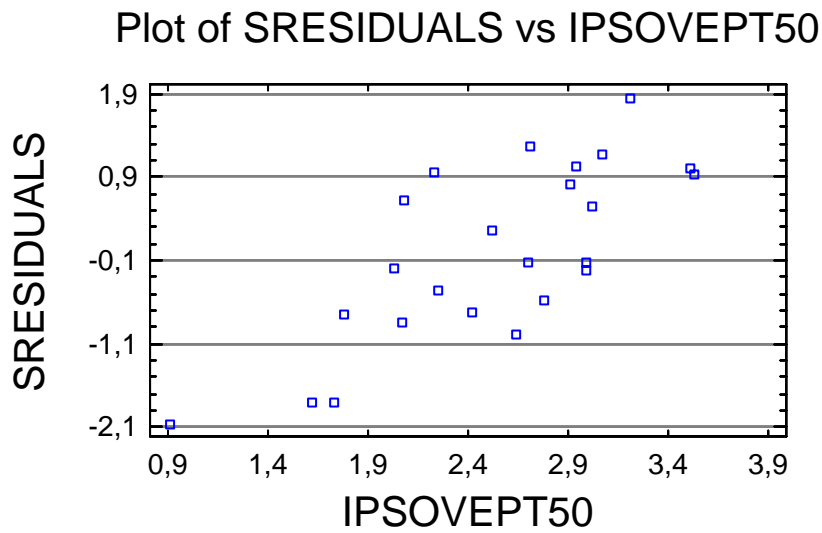


Figura II-94. Residuos estudentizados frente a la variable IPISOVEPT del conjunto experimental SSD100_P3_50.

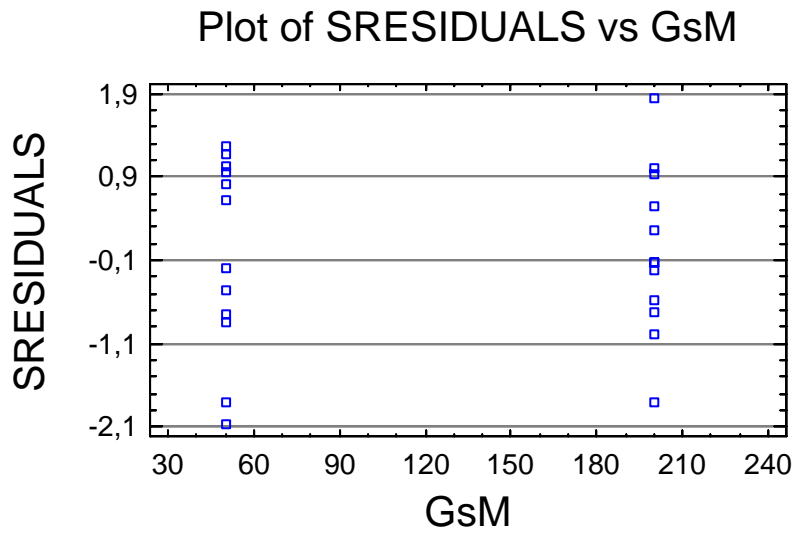


Figura II-95. Residuos estudentizados frente a los niveles del factor GsM.

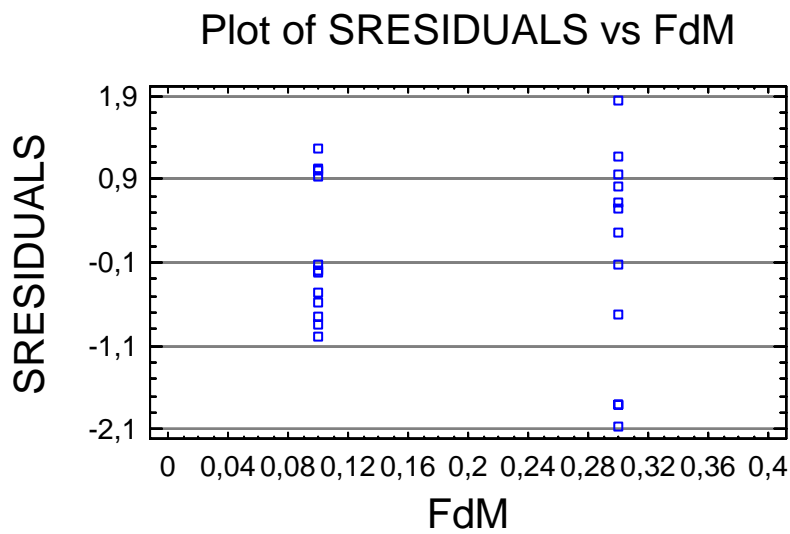


Figura II-96. Residuos estudentizados frente a los niveles del factor FdM.

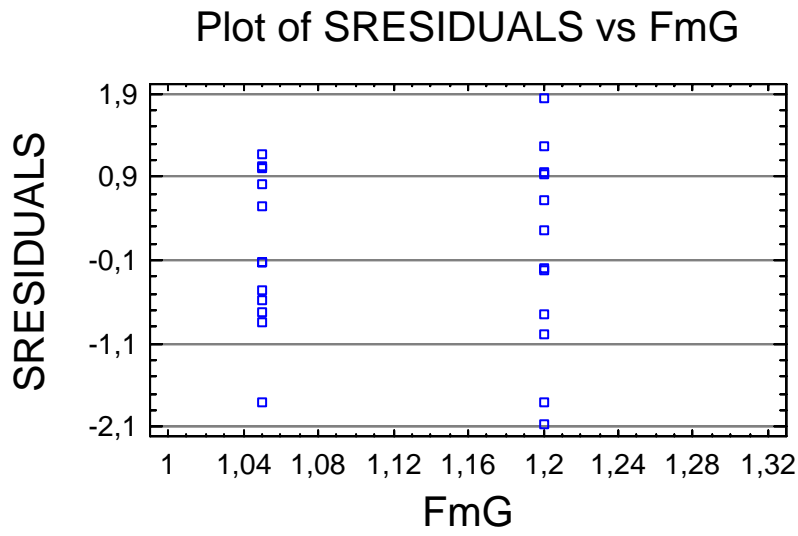


Figura II-97. Residuos estudentizados frente a los niveles del factor FmG.

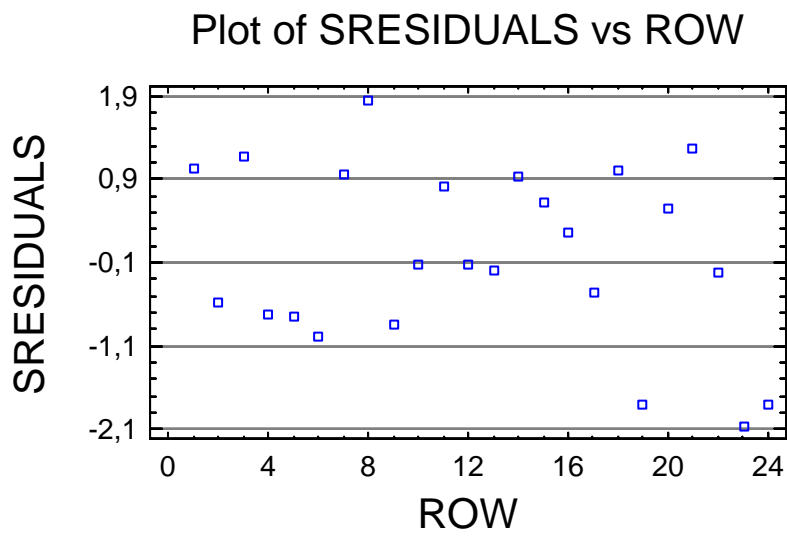


Figura II-98. Residuos estudentizados frente al orden de ejecución.

Analysis of Variance for IPSOVEPT50

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
A:GsM	2,17161	1	2,17161	7,14	0,0161
B:FdM	0,61867	1	0,61867	2,04	0,1718
C:FmG	0,719581	1	0,719581	2,37	0,1423
AB	0,182039	1	0,182039	0,60	0,4496
AC	0,226648	1	0,226648	0,75	0,3999
BC	0,24148	1	0,24148	0,79	0,3852
Total error	5,16769	17	0,303982		
Total (corr.)	9,32772	23			

Tabla II-15. ANOVA correspondiente al conjunto experimental SSD100_P3_50.

Observando la Tabla II-15 vemos que disponemos de 24 datos y tenemos 17 grados de libertad residuales. El contraste de hipótesis sobre las medias para un $\alpha=0,05$ implica que es significativo el efecto principal del factor A, es decir los factores GsM.

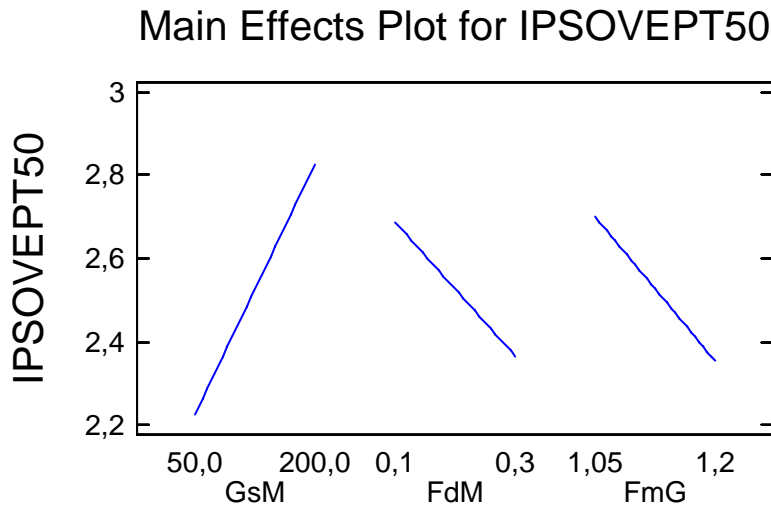


Figura II-99. Gráfico de efectos principales de los factores del conjunto experimental SSD100_P3_50.

Analizando la Figura II-99 se deduce que el valor adecuado son para el factor GsM es 50. Los factores B y C pueden tomar cualquier valor de los analizados, ya que no son significativos. No obstante, se ha seleccionado el valor 0.3 para el factor FdM y 1.2 para el factor FmG.. Por lo tanto, los parámetros quedarían GsM=50, FdM=0.3 y FmG=1.2.

A-III.2.4 Bloque SSD125

A-III.2.4.1 Subconjunto SSD125_P3_50

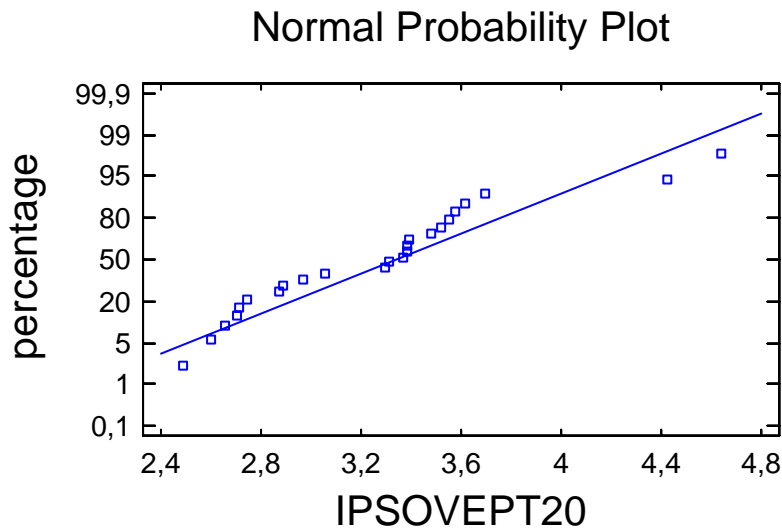


Figura II-100. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPSOVEPT del conjunto SSD125_P3_20 en relación a las frecuencias relativas acumuladas.

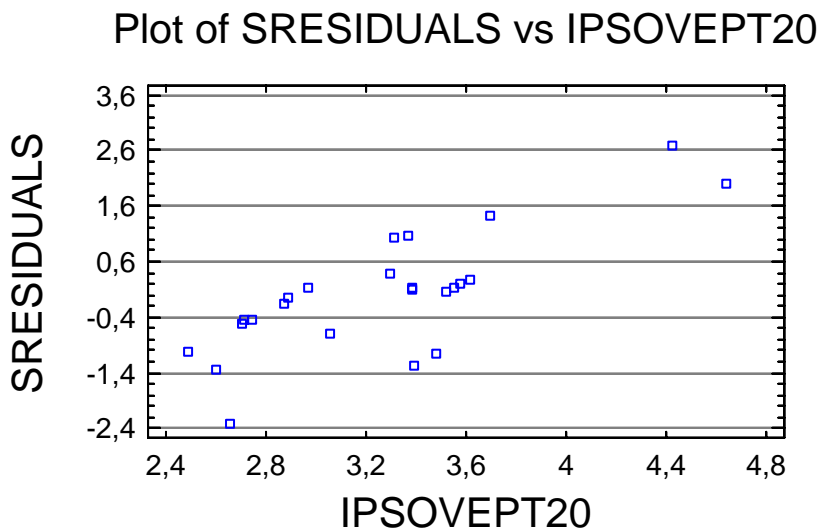


Figura II-101. Residuos estudentizados frente a la variable IPSOVEPT del conjunto experimental SSD125_P3_20.

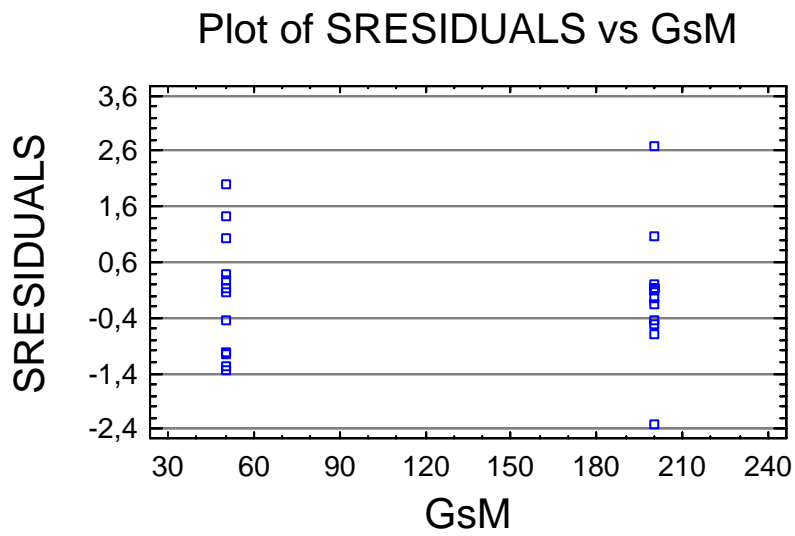


Figura II-102. Residuos estudentizados frente a los niveles del factor GsM.

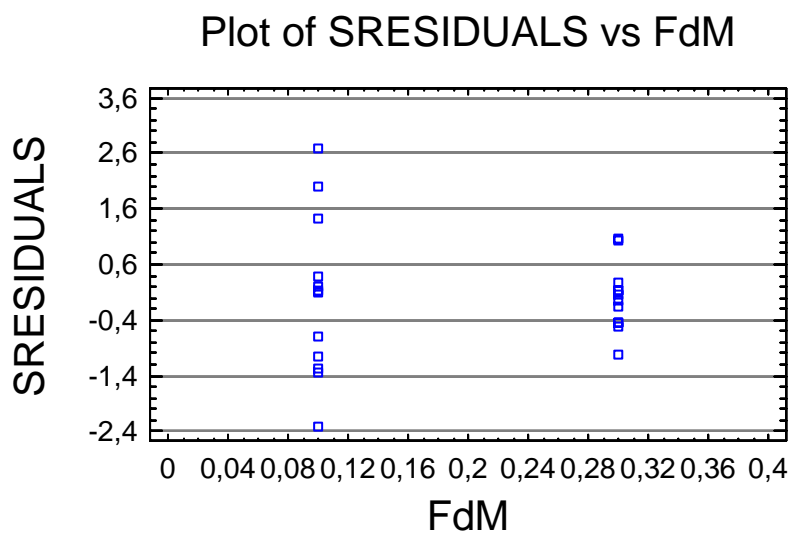


Figura II-103. Residuos estudentizados frente a los niveles del factor FdM.

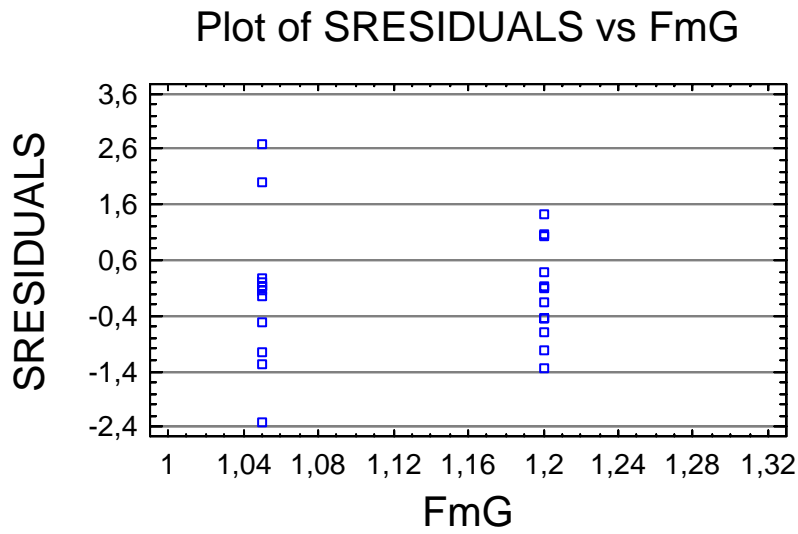


Figura II-104. Residuos estudentizados frente a los niveles del factor FmG.

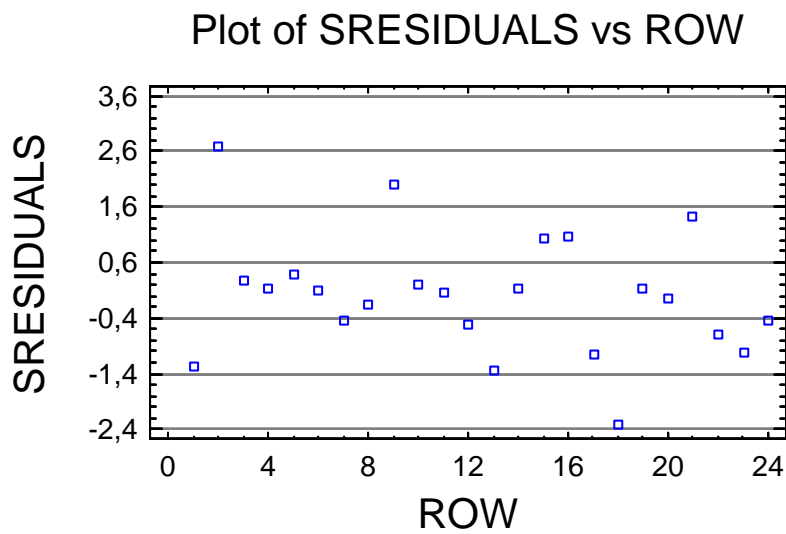


Figura II-105. Residuos estudentizados frente al orden de ejecución.

Analysis of Variance for IPSOVEPT20

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
A:GsM	0,213925	1	0,213925	0,92	0,3515
B:FdM	0,977548	1	0,977548	4,22	0,0577
C:FmG	0,849711	1	0,849711	3,67	0,0746
AB	0,0448374	1	0,0448374	0,19	0,6661
AC	0,569313	1	0,569313	2,46	0,1376
BC	0,04005	1	0,04005	0,17	0,6833
blocks	0,464722	2	0,232361	1,00	0,3897
Total error	3,471	15	0,2314		
Total (corr.)	6,6311	23			

Tabla II-16. ANOVA correspondiente al conjunto experimental SSD125_P3_20.

Observando la Tabla II-16 vemos que disponemos de 24 datos y tenemos 15 grados de libertad residuales. El contraste de hipótesis sobre las medias para un $\alpha=0,05$ implica que no es significativo ningún efecto.

El valor de los factores GsM, FdM y FmG se puede establecer libremente. No obstante, el valor elegido para GsM es 50, para FdM es 0.1 y para FmGes 1.2. Por lo tanto, los parámetros quedarían GsM=50, FdM=0.1 y FmG=1.2.

A-III.2.4.2 Subconjunto SSD125_P3_50

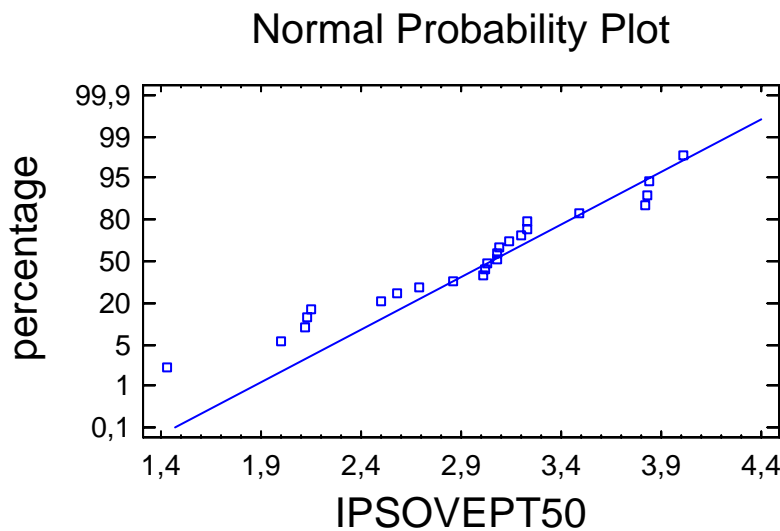


Figura II-106. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPSOVEPT del conjunto SSD125_P3_50 en relación a las frecuencias relativas acumuladas.

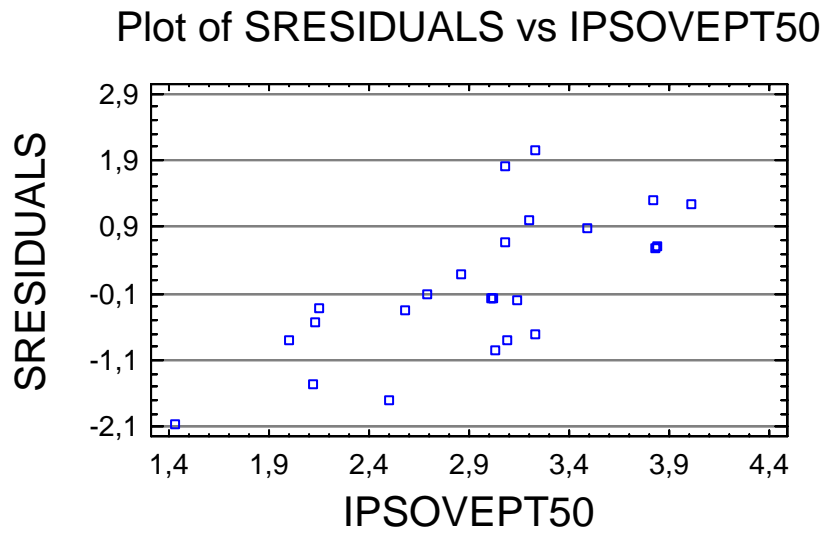


Figura II-107. Residuos estudentizados frente a la variable IPSOVEPT del conjunto experimental SSD125_P3_50.

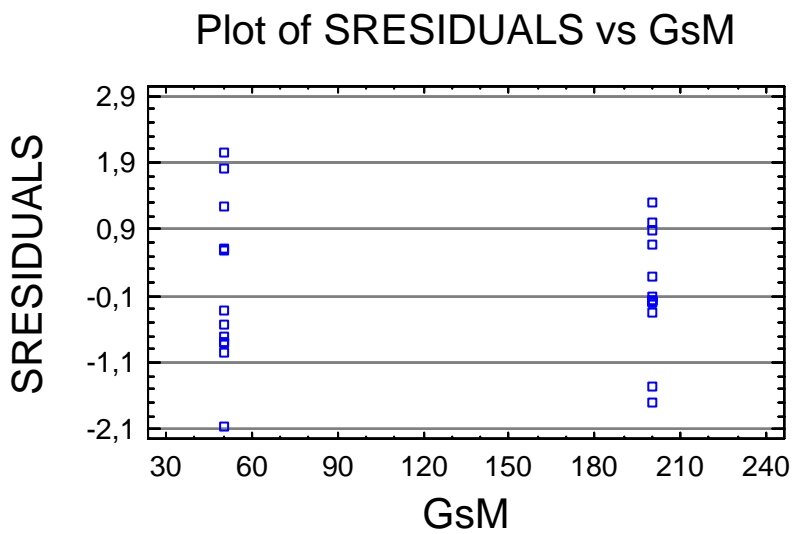


Figura II-108. Residuos estudentizados frente a los niveles del factor GsM.

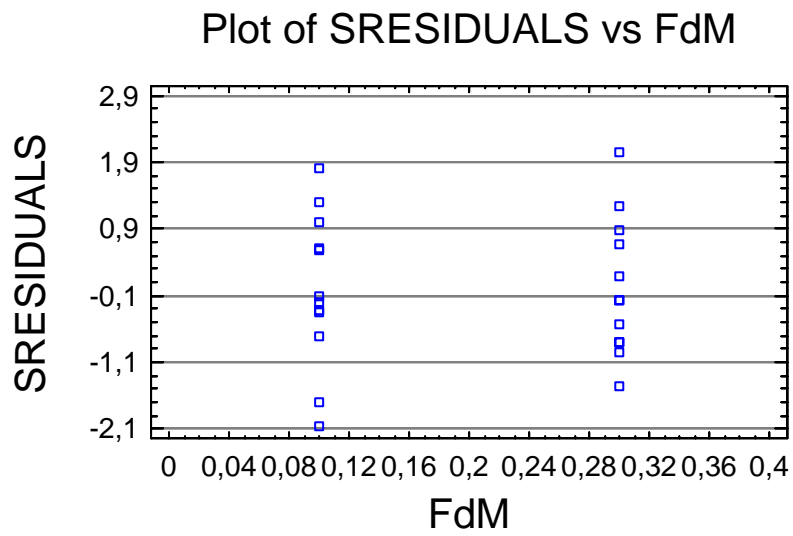


Figura II-109. Residuos estudentizados frente a los niveles del factor FdM.

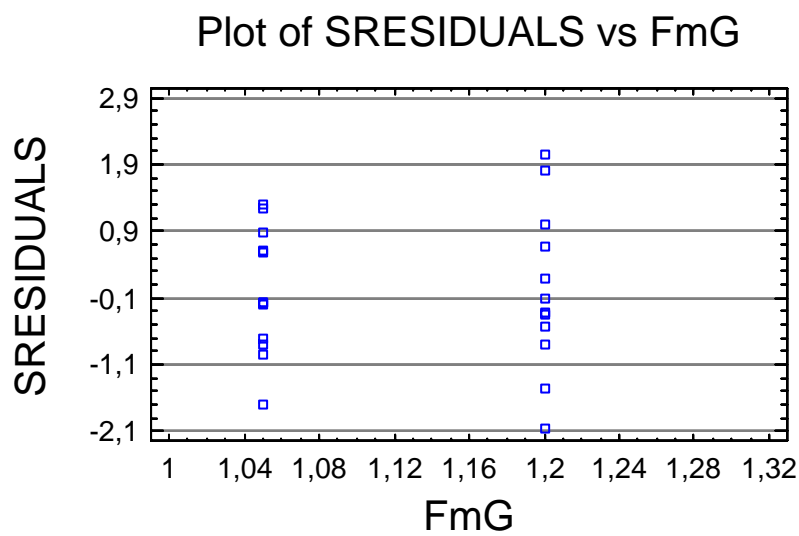


Figura II-110. Residuos estudentizados frente a los niveles del factor FmG.

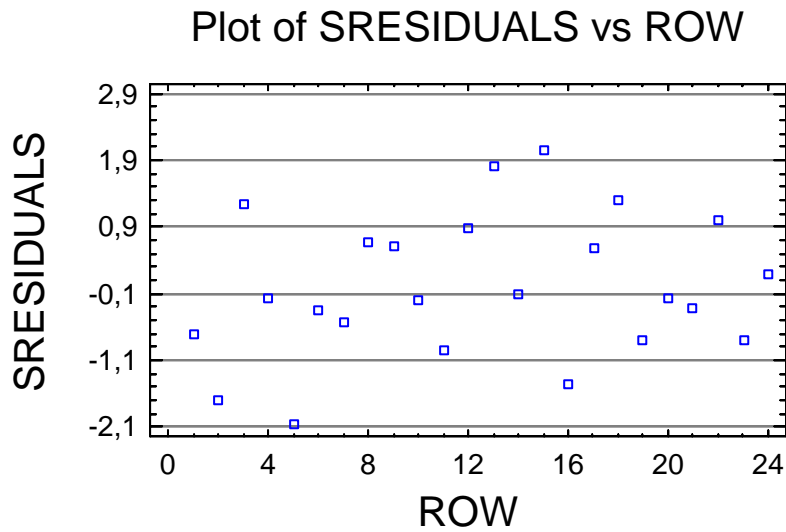


Figura II-111. Residuos estudentizados frente al orden de ejecución.

Analysis of Variance for IPSOVEPT50

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
A:GsM	0,0088181	1	0,0088181	0,03	0,8650
B:FgM	0,00781194	1	0,00781194	0,03	0,8729
C:FmG	3,72634	1	3,72634	12,58	0,0025
AB	0,00320773	1	0,00320773	0,01	0,9183
AC	0,868554	1	0,868554	2,93	0,1050
BC	0,042377	1	0,042377	0,14	0,7099
Total error	5,03547	17	0,296204		
Total (corr.)	9,69258	23			

Tabla II-17. ANOVA correspondiente al conjunto experimental SSD125_P3_50.

Observando la Tabla II-17 vemos que disponemos de 24 datos y tenemos 17 grados de libertad residuales. El contraste de hipótesis sobre las medias para un $\alpha=0,05$ implica que es significativo el efecto principal del factor C, es decir los factores FmG.

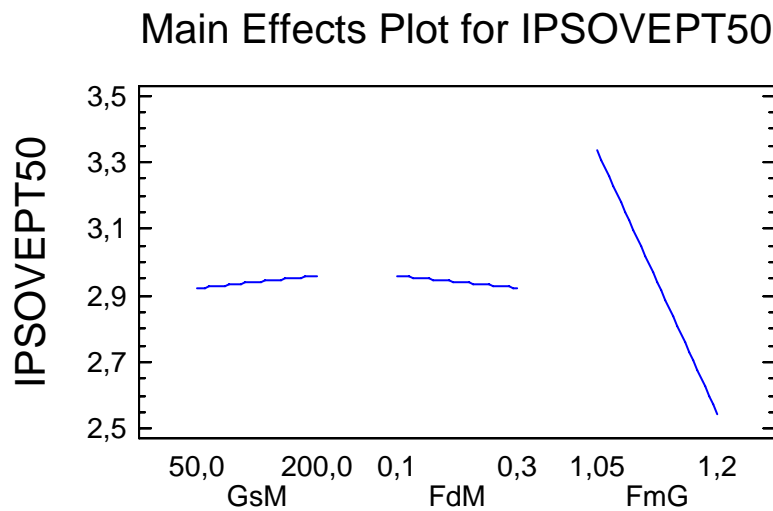


Figura II-112. Gráfico de efectos principales de los factores del conjunto experimental SSD125_P3_50.

Analizando la Figura II-112 se deduce que el valor adecuado son para el factor FmG es 1.2. Los factores B y C pueden tomar cualquier valor de los analizados, ya que no son significativos. No obstante, se ha seleccionado el valor 50 para el factor GsM y 0.3 para el factor FdM. Por lo tanto, los parámetros quedarían GsM=50, FdM=0.3 y FmG=1.2.

A-III.3 Ajuste de Parámetros SMAGA-2 para la Configuración P13

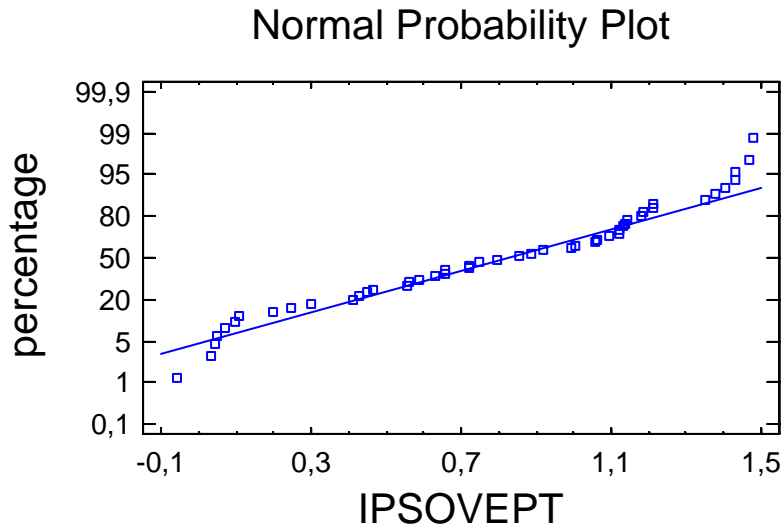


Figura II-113. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPSPVEPT de la configuración P13 en relación a las frecuencias relativas acumuladas.

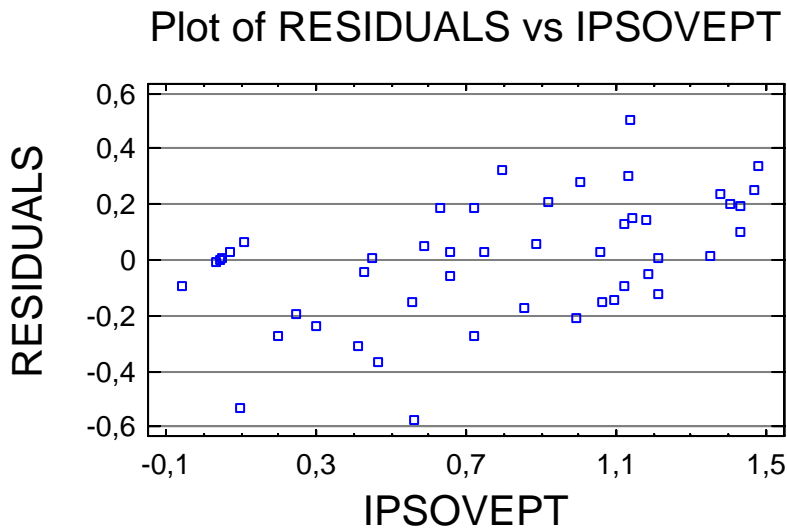


Figura II-114. Residuos frente a la variable IPSPVEPT de la configuración P13.

Residual Plot for IPSOVEPT

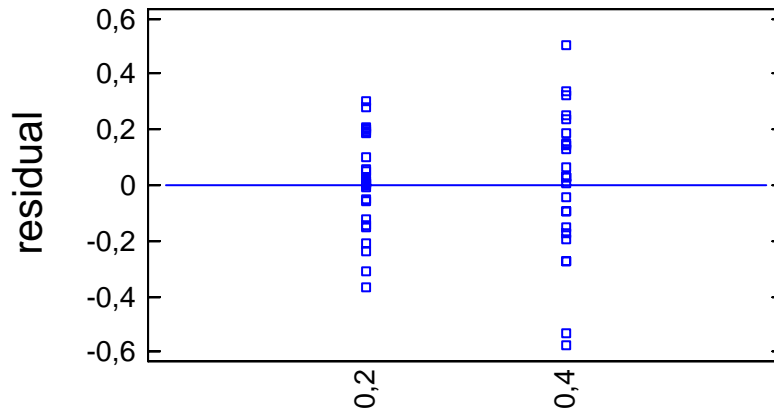


Figura II-115. Residuos frente a los niveles del factor RM.

Residual Plot for IPSOVEPT

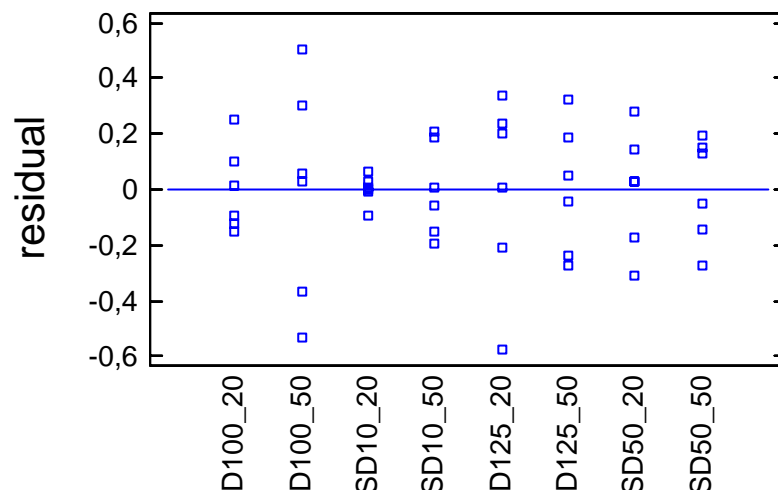


Figura II-116. Residuos frente a los niveles del factor Conjuntos.

Residual Plot for IPSOVEPT

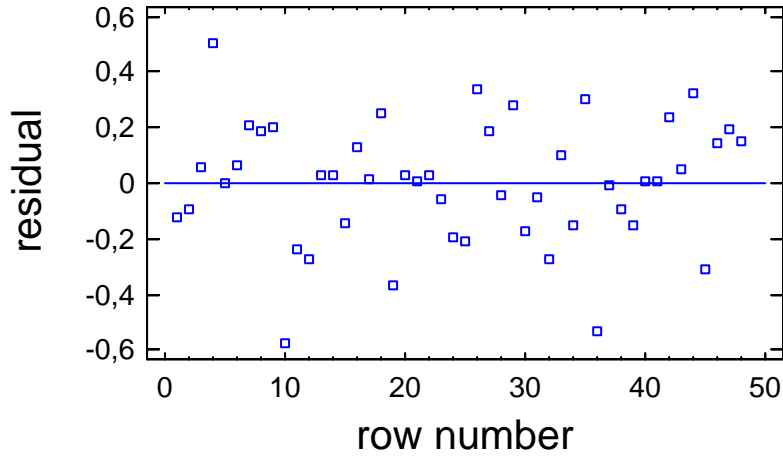


Figura II-117. Residuos frente al orden de ejecución.

Analysis of Variance for IPSOVEPT - Type III Sums of Squares

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
MAIN EFFECTS					
A:Conjunto	7,10645	7	1,01521	14,04	0,0000
B:RM	0,0749496	1	0,0749496	1,04	0,3162
INTERACTIONS					
AB	0,35273	7	0,05039	0,70	0,6742
RESIDUAL	2,31365	32	0,0723015		
TOTAL (CORRECTED)	9,84778	47			

All F-ratios are based on the residual mean square error.

Tabla II-18. ANOVA correspondiente IPSPPVEPT de la configuración P13.

Observando la Tabla II-18 vemos que disponemos de 48 datos y tenemos 32 grados de libertad residuales. El contraste de hipótesis sobre las medias para un $\alpha=0,05$ implica que sólo es significativo el efecto principal del factor A, es decir del Conjunto.

Means and 95,0 Percent LSD Intervals

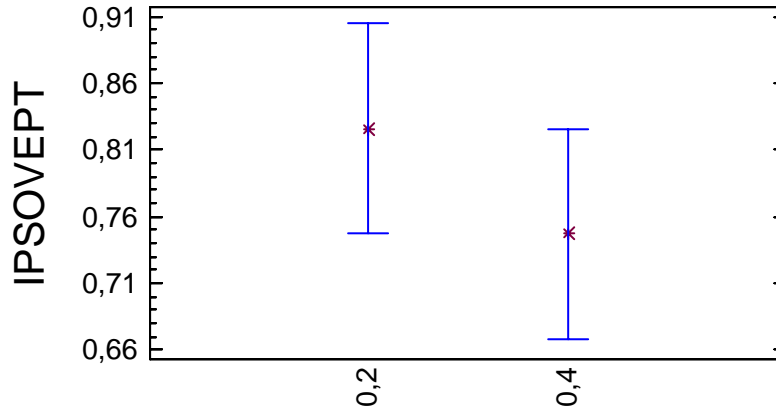


Figura II-118. Gráfico del efecto principal del factor RM de la configuración P13.

Means and 95,0 Percent LSD Intervals

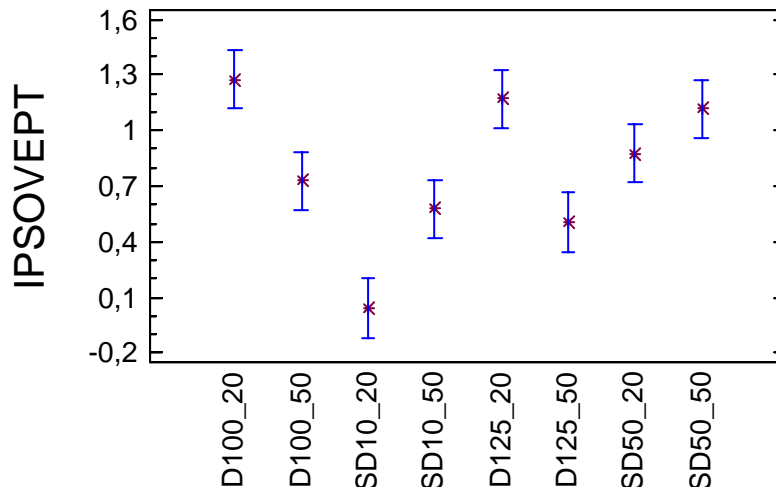


Figura II-119. Gráfico del efecto principal del factor Conjunto de la configuración P13.

Se han indicados los valores promedio del factor RM, aunque no es significativo, y del factor Conjunto.

A-III.4 Ajuste de Parámetros SMAGA-2 para la Configuración P3

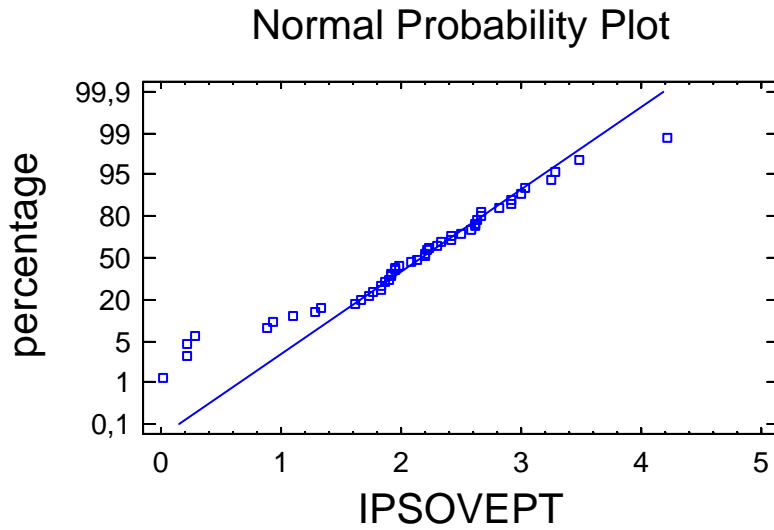


Figura II-120. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPSPPVEPT de la configuración P3 en relación a las frecuencias relativas acumuladas.

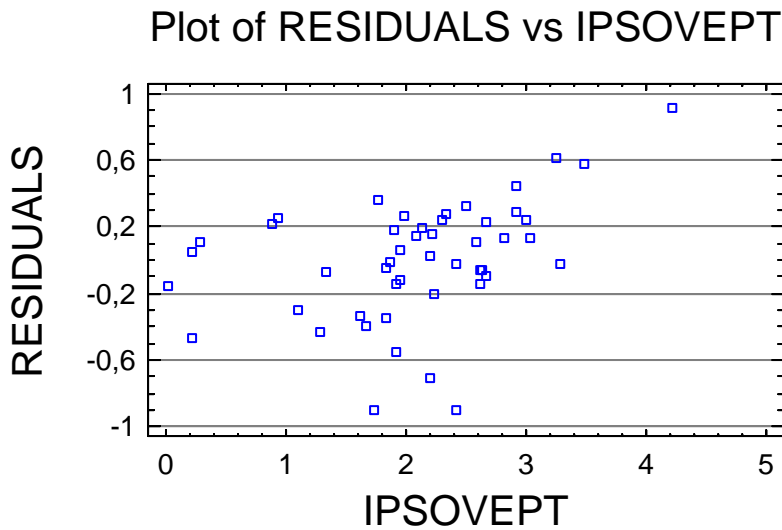


Figura II-121. Residuos frente a la variable IPSPPVEPT de la configuración P3.

Residual Plot for IPSOVEPT

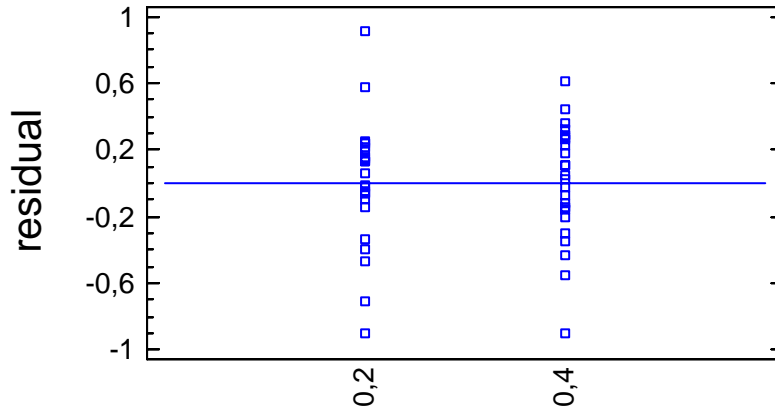


Figura II-122. Residuos frente a los niveles del factor RM.

Residual Plot for IPSOVEPT

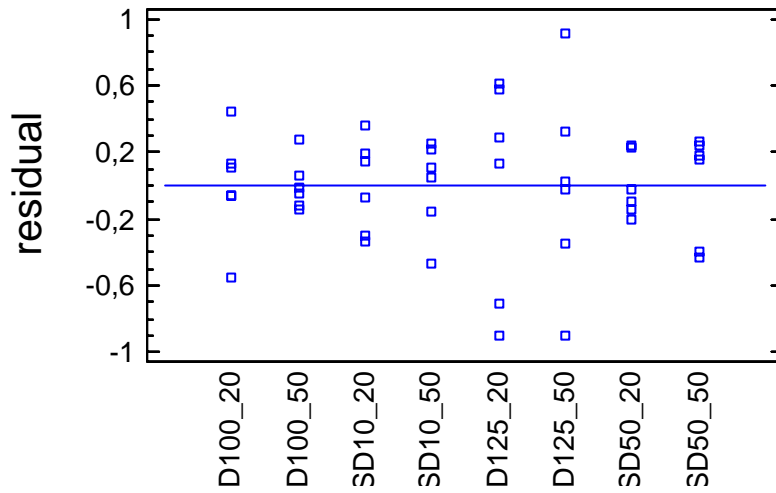


Figura II-123. Residuos frente a los niveles del factor Conjuntos.

Residual Plot for IPSOVEPT

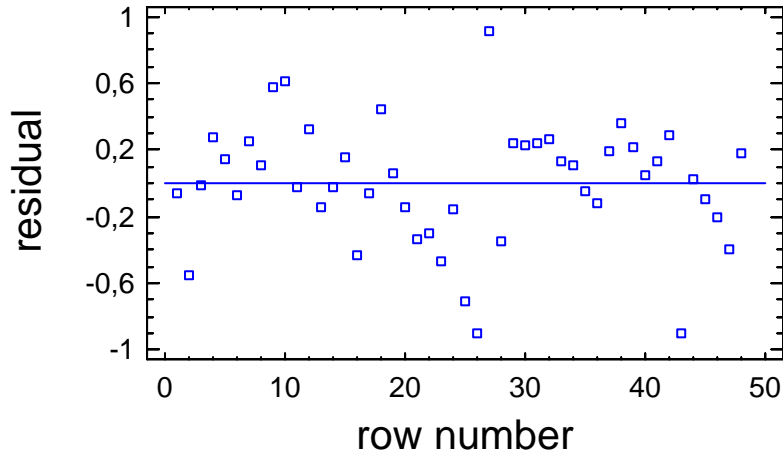


Figura II-124. Residuos frente al orden de ejecución.

Analysis of Variance for IPSOVEPT - Type III Sums of Squares

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
MAIN EFFECTS					
A:Conjunto	26,3796	7	3,76851	19,69	0,0000
B:RM	1,85508	1	1,85508	9,69	0,0039
INTERACTIONS					
AB	1,44438	7	0,20634	1,08	0,4000
RESIDUAL	6,1255	32	0,191422		
TOTAL (CORRECTED)	35,8045	47			

All F-ratios are based on the residual mean square error.

Tabla II-19. ANOVA correspondiente IPSPPVEPT de la configuración P3.

Observando la Tabla II-19 vemos que disponemos de 48 datos y tenemos 32 grados de libertad residuales. El contraste de hipótesis sobre las medias para un $\alpha=0,05$ implica que es significativo el efecto principal de los factor A y B, es decir del Conjunto y RM.

Means and 95,0 Percent LSD Intervals

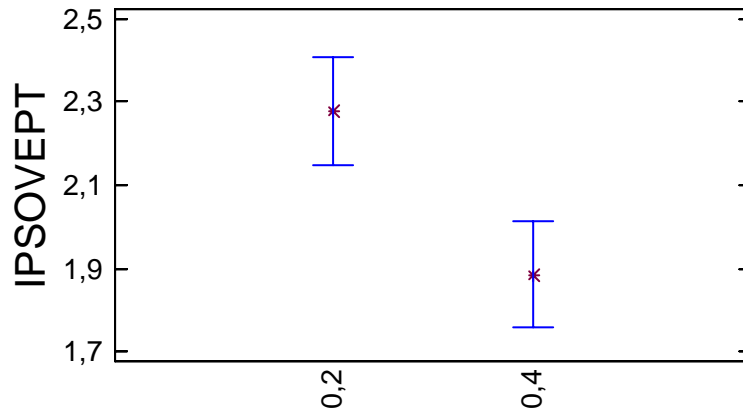


Figura II-125. Gráfico del efecto principal del factor RM de la configuración P3.

Means and 95,0 Percent LSD Intervals

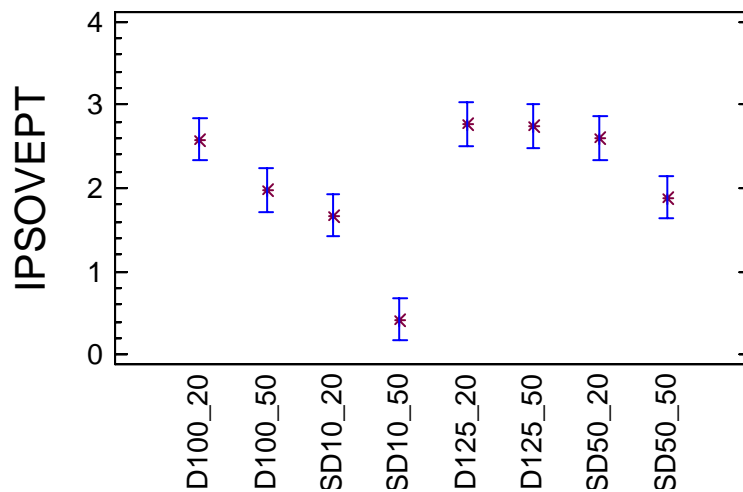


Figura II-126. Gráfico del efecto principal del factor Conjunto de la configuración P13.

Se han indicados los valores promedio del factor RM, y del factor Conjunto.

A-III.5 Análisis Comparativo para la Configuración P13

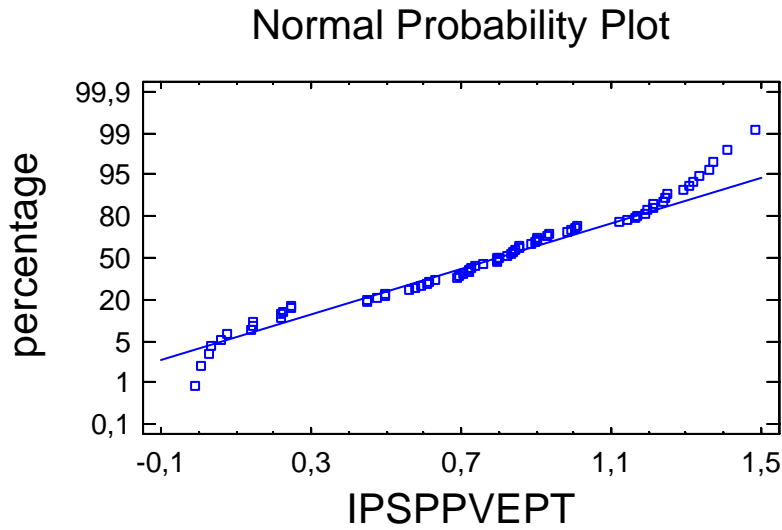


Figura II-127. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPSPPVEPT de la configuración P13 en relación a las frecuencias relativas acumuladas.

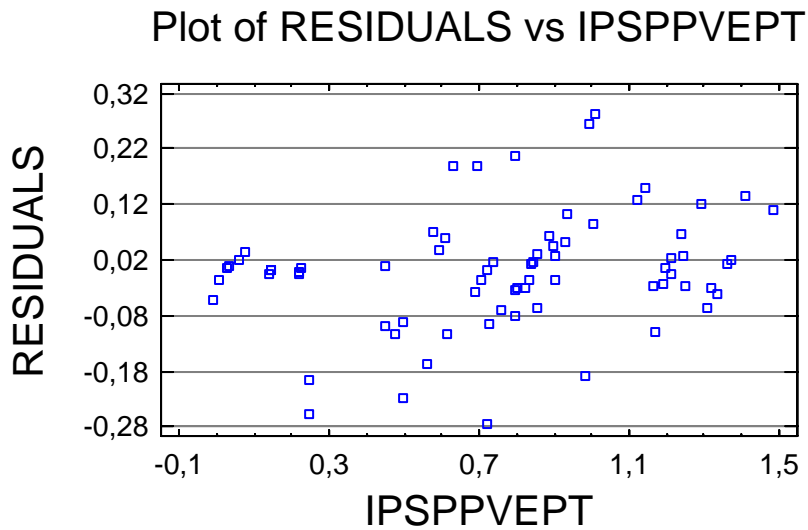


Figura II-128. Residuos frente a la variable IPSPPVEPT de la configuración P13.

Residual Plot for IPSPPVEPT

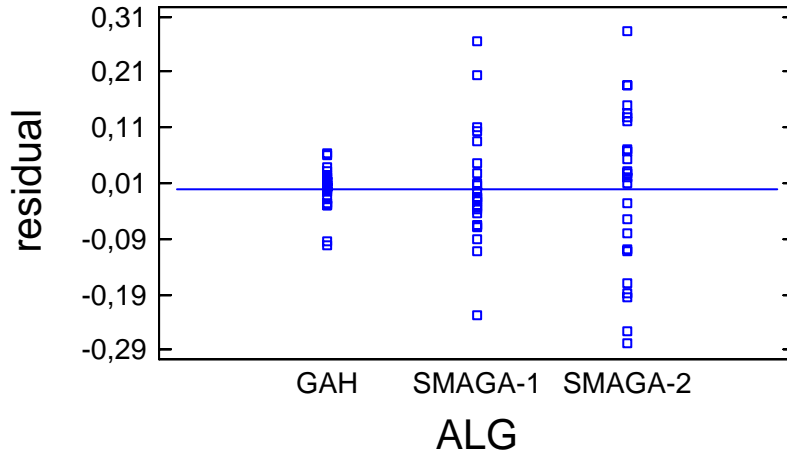


Figura II-129. Residuos frente a los niveles del factor ALG.

Residual Plot for IPSPPVEPT

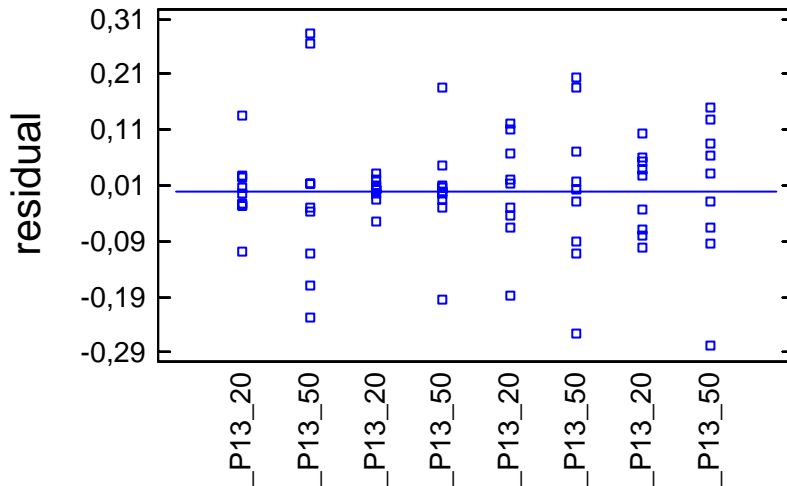


Figura II-130. Residuos frente a los niveles del factor Conjuntos.

Residual Plot for IPSPVEPT

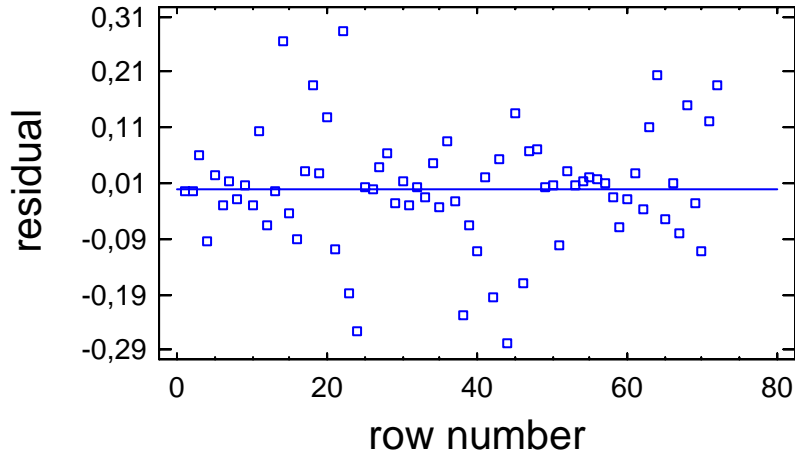


Figura II-131. Residuos frente al orden de ejecución.

Analysis of Variance for IPSPVEPT - Type III Sums of Squares

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
MAIN EFFECTS					
A:ALG	0,0976657	2	0,0488328	2,97	0,0607
B:Conjunto	9,88458	7	1,41208	85,92	0,0000
INTERACTIONS					
AB	0,948611	14	0,0677579	4,12	0,0001
RESIDUAL	0,788885	48	0,0164351		
TOTAL (CORRECTED)	11,7197	71			

Tabla II-20. ANOVA correspondiente IPSPVEPT de la configuración P13.

Observando la Tabla II-20 vemos que disponemos de 72 datos y tenemos 48 grados de libertad residuales. El contraste de hipótesis sobre las medias para un $\alpha=0,05$ implica que sólo es significativo el efecto principal del factor B, es decir del Conjunto, y la interacción de los efectos ALG y Conjunto.

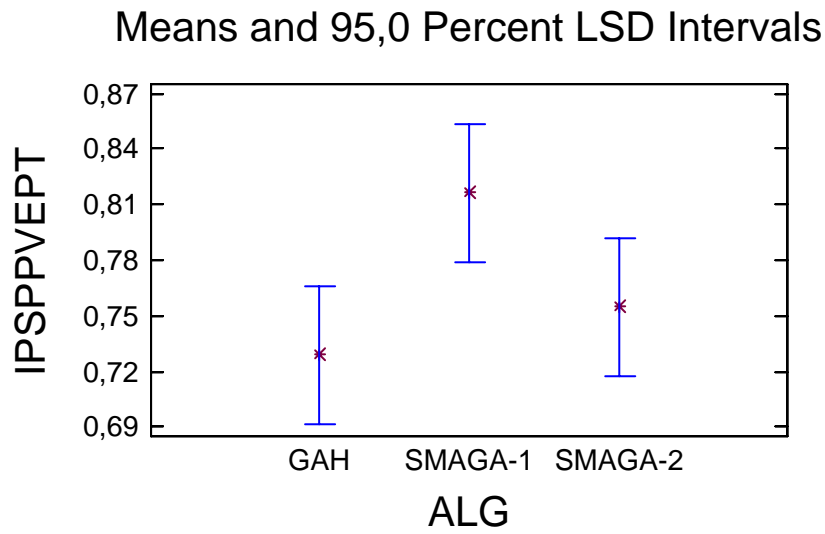


Figura II-132. Gráfico del efecto principal del factor ALG de la configuración P13.

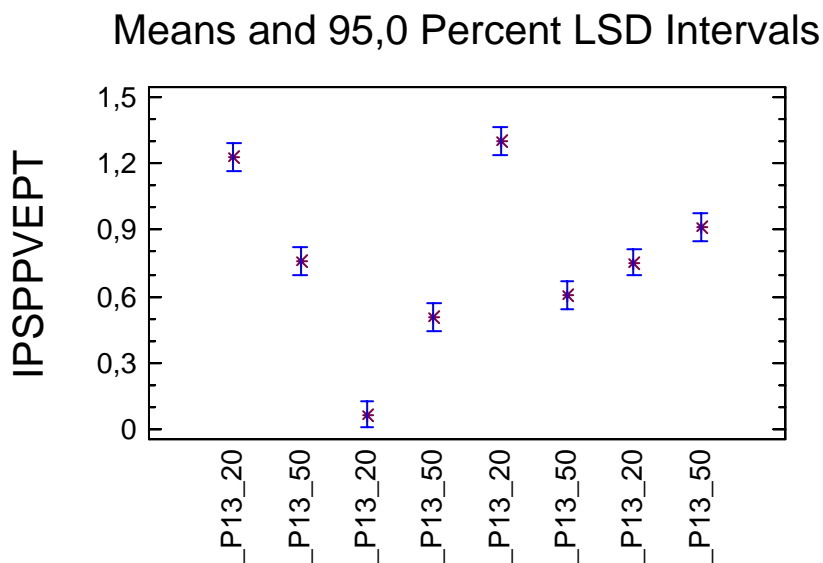


Figura II-133. Gráfico del efecto principal del factor Conjunto de la configuración P13.

Se han indicados los valores promedio del factor ALG, aunque no es significativo, y del factor Conjunto.

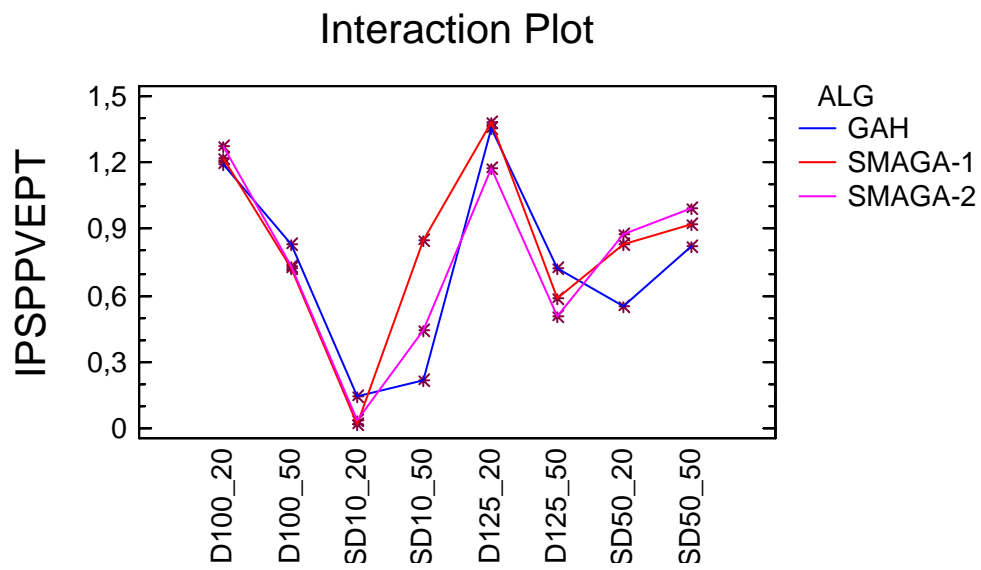


Figura II-134. Gráfico de la interacción de los factores ALG y Conjunto de la configuración P13.

En la Figura II-135 se muestra la interacción entre los tres algoritmos analizados. Se observa como efectivamente existen diferencias con respecto a los diferentes conjuntos.

A-III.6 Análisis Comparativo para la Configuración P3

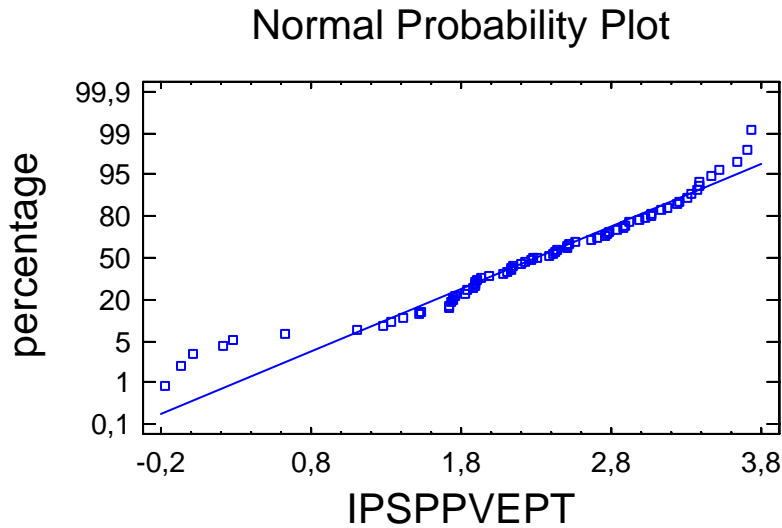


Figura II-135. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPSOVEPT de la configuración P3 en relación a las frecuencias relativas acumuladas.

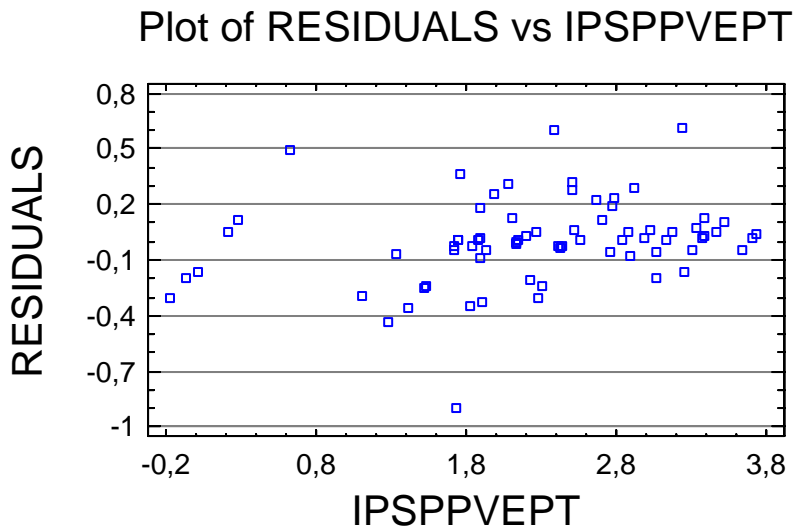


Figura II-136. Residuos frente a la variable IPSOVEPT de la configuración P3.

Residual Plot for IPSPPVEPT

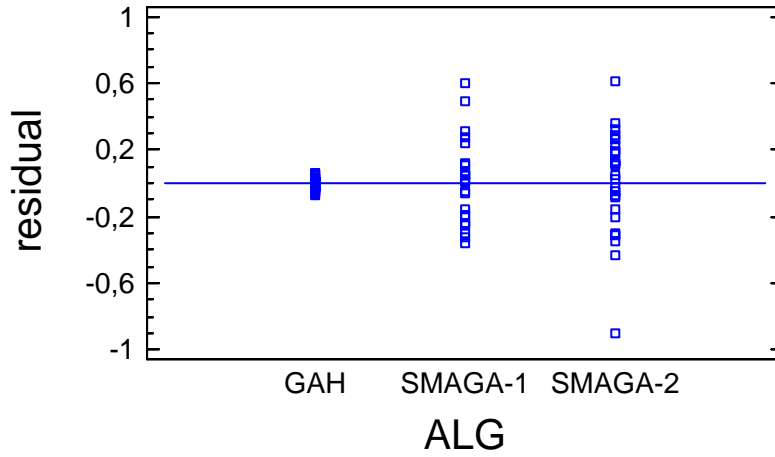


Figura II-137. Residuos frente a los niveles del factor ALG.

Residual Plot for IPSPPVEPT

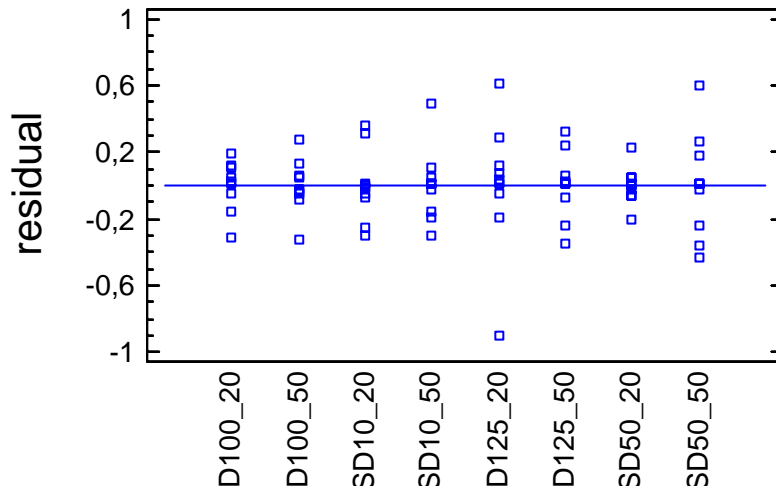


Figura II-138. Residuos frente a los niveles del factor Conjuntos.

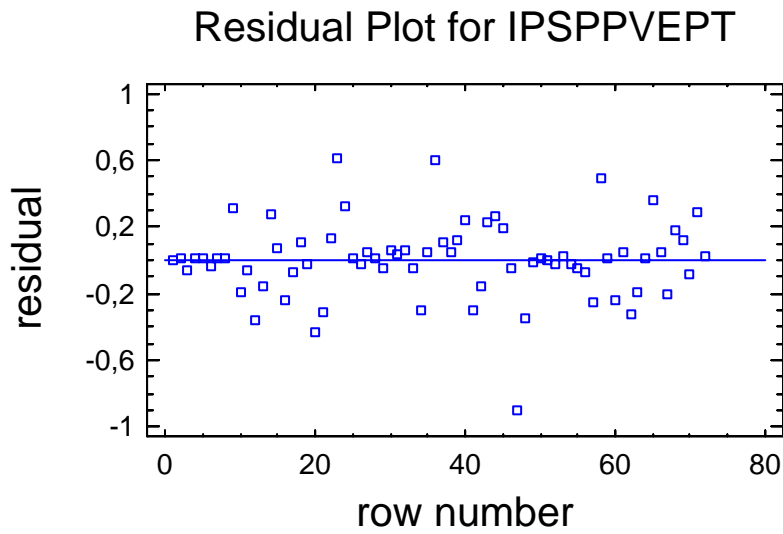


Figura II-139. Residuos frente al orden de ejecución.

Analysis of Variance for IPSPVEPT - Type III Sums of Squares

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
MAIN EFFECTS					
A:ALG	7,35266	2	3,67633	45,86	0,0000
B:Conjunto	44,6062	7	6,37231	79,48	0,0000
INTERACTIONS					
AB	3,53932	14	0,252809	3,15	0,0015
RESIDUAL	3,8482	48	0,0801709		
TOTAL (CORRECTED)	59,3464	71			

All F-ratios are based on the residual mean square error.

Tabla II-21. ANOVA correspondiente IPSOVEPT de la configuración P3.

Observando la Tabla II-21 vemos que disponemos de 72 datos y tenemos 48 grados de libertad residuales. El contraste de hipótesis sobre las medias para un $\alpha=0,05$ implica que son significativos los efectos principales del factor A y B, es decir de los factores ALG y Conjunto, así como la interacción entre ambos.

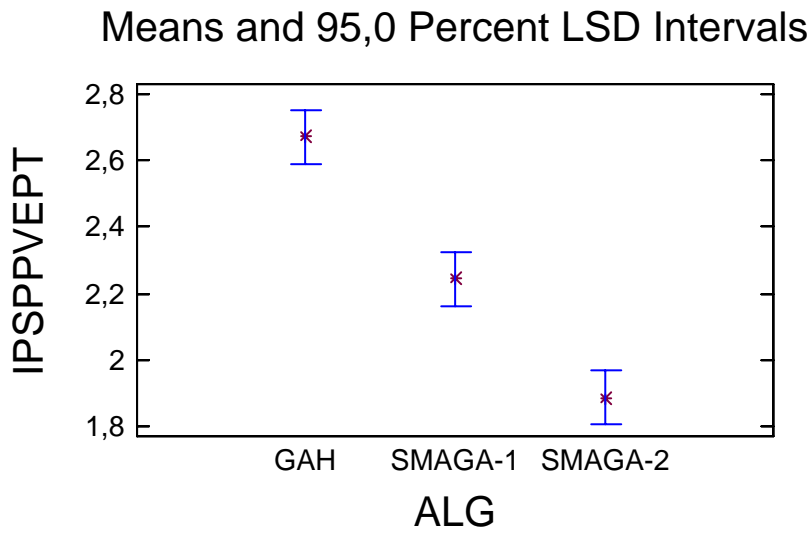


Figura II-140. Gráfico del efecto principal del factor ALG de la configuración P3.

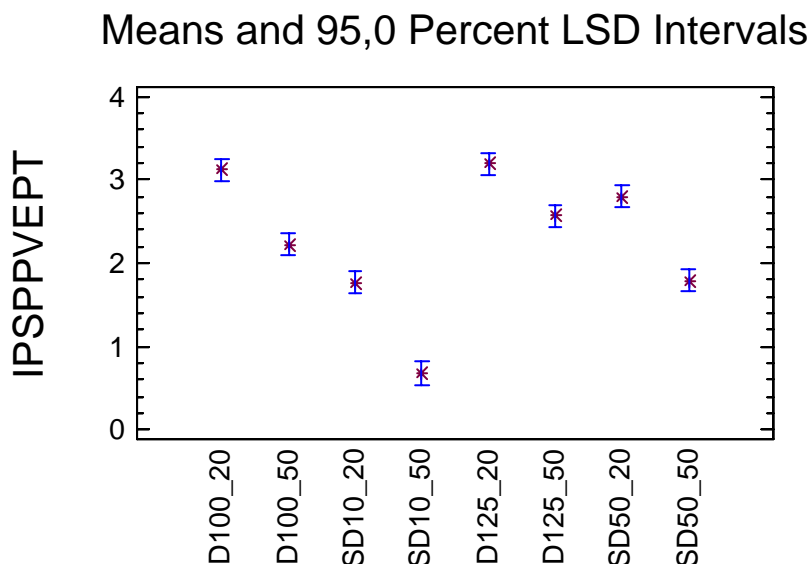


Figura II-141. Gráfico del efecto principal del factor Conjunto de la configuración P3.

Analizando la Figura II-140 se deduce que el valor adecuado son para el factor ALG es SMAGA-2. En este caso los valores de la Figura II-141 carecen de relevancia, ya que era conocido desde el inicio que el tipo de conjunto experimental sobre el que se actuaba era relevante. En todo caso este análisis lo confirma.

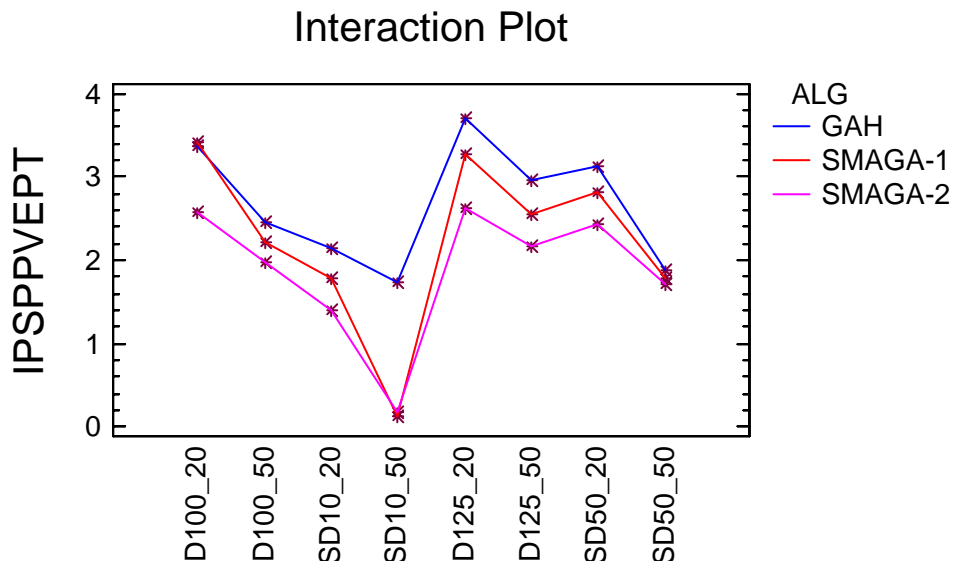


Figura II-142. Gráfico de la interacción de los factores ALG y Conjunto de la configuración P13.

En la Figura II-135 se muestra la interacción entre los tres algoritmos analizados. Se observa como efectivamente existen diferencias en alguno de los diferentes conjuntos analizados.

El test siguiente permite agrupar factores que presentan valores similares indica que existen 3 conjuntos diferentes, y que por lo tanto, son valores significativos.

Method: 95,0 percent LSD

ALG	Count	LS Mean	LS Sigma	Homogeneous Groups
-----	-------	---------	----------	--------------------

SMAGA-2	24	1,88761	0,0577967	X
SMAGA-1	24	2,24379	0,0577967	X
GAH	24	2,66935	0,0577967	X

Contrast	Difference	+/- Limits
----------	------------	------------

GAH - SMAGA-1	*0,425561	0,164343
GAH - SMAGA-2	*0,78174	0,164343
SMAGA-1 - SMAGA-2	*0,356178	0,164343

* denotes a statistically significant difference.

**Resultados y Análisis Estadístico de los
mismos Obtenidos en la
Experimentación para la Programación
Reactiva**

Anexo

IV

A-IV.1 Resultados y Análisis de los mismo en situaciones de Adelantos en las Operaciones

En este apartado se proporcionan los datos obtenidos durante la simulación de reprogramación de un programa predictivo sometido a eventos que suponen un adelanto en la fecha de finalización de las operaciones. Los resultados que se muestran son el promedio de adelantos grandes, medianos y pequeños correspondientes a los tipos de eventos BAT_AS_10, BAT_AS_50 y BAT_AS_80. En concreto los datos se dividen en dos apartados, en el primero se muestran los datos obtenidos partiendo de un algoritmo predictivo obtenido con SMAGA-2, y en el siguiente apartado los obtenidos con un algoritmo SMAGA-2 cuya secuencia ha sido modificada intercambiando 2 elementos de la misma. En ambos casos se muestran los datos obtenidos para la combinación de los 4 factores considerados (CEX, K, VA y ALG). De igual manera, en ambos casos, los algoritmos considerados son 2, SSDS y SMAGA-2, y ambos son sensibles al cambio de valor en el factor K y VA.

También se realiza el análisis estadístico sólo de los datos obtenidos con el programa predictivo establecido con algoritmo SMAGA-2. En concreto el análisis se divide en 4 bloques, en función de la configuración del taller y número de trabajos: Bloque P13_20, P13_50, P3_20 y P3_50.

En primer lugar se proporcionan los datos de los bloques P13_20 y P13_50, y el análisis estadístico correspondiente, y posteriormente se muestran los datos de los bloques P3_20 y P3_50 y se realiza en estudio estadístico correspondiente.

A-IV.1.1 Resultados para la Configuración P13 con Programación Predictiva basada en SMAGA-2

Conjunto Experimental	K=0 y VA=0,2		K=0,01 y VA=0,2		K=0,02 y VA=0,2	
	SSDS	SMAGA-2	SSDS	SMAGA-2	SSDS	SMAGA-2
SSD10_P13_20	-0,078	0,036	2,105	1,233	4,159	2,549
SSD50_P13_20	-0,057	0,405	1,852	1,785	3,788	3,184
SSD100_P13_20	-0,037	0,971	1,662	2,433	3,461	3,866
SSD125_P13_20	-0,057	1,241	1,536	2,669	3,199	3,867

Tabla IV-1. Valor de IPSPPVEPT a partir de SMAGA-2 con adelanto para 20 trabajos, y un valor de VA de 0,2.

Conjunto Experimental	K=0 y VA=0,4		K=0,01 y VA=0,4		K=0,02 y VA=0,4	
	SSDS	SMAGA-2	SSDS	SMAGA-2	SSDS	SMAGA-2
	SSD10_P13_20	-0,083	0,037	2,477	1,432	5,274
SSD50_P13_20	-0,060	0,436	2,137	2,223	4,498	3,466
SSD100_P13_20	-0,037	0,966	1,897	2,920	4,017	4,326
SSD125_P13_20	-0,213	1,107	1,903	2,994	3,850	4,713

Tabla IV-2. Valor de IPSPPVEPT a partir de SMAGA-2 con adelanto para 20 trabajos, y un valor de VA de 0,4.

Conjunto Experimental	K=0 y VA=0,1		K=0,005 y VA=0,1		K=0,01 y VA=0,1	
	SSDS	SMAGA-2	SSDS	SMAGA-2	SSDS	SMAGA-2
	SSD10_P13_50	-0,011	0,317	1,399	2,260	2,853
SSD50_P13_50	-0,140	1,396	1,114	3,358	2,300	5,316
SSD100_P13_50	-0,016	2,263	0,993	4,119	2,122	6,201
SSD125_P13_50	-0,117	2,272	0,971	4,296	1,955	6,538

Tabla IV-3. Valor de IPSPPVEPT a partir de SMAGA-2 con adelanto para 50 trabajos, y un valor de VA de 0,1.

Conjunto Experimental	K=0 y VA=0,1		K=0,005 y VA=0,1		K=0,01 y VA=0,1	
	SSDS	SMAGA-2	SSDS	SMAGA-2	SSDS	SMAGA-2
	SSD10_P13_50	-0,309	0,332	1,900	3,230	3,703
SSD50_P13_50	-0,863	1,356	1,573	4,236	3,210	7,187
SSD100_P13_50	-0,191	2,299	1,395	5,031	2,734	7,968
SSD125_P13_50	-0,021	2,366	1,341	5,351	2,806	8,351

Tabla IV-4. Valor de IPSPPVEPT a partir de SMAGA-2 con adelanto para 50 trabajos, y un valor de VA de 0,2.

A-IV.1.2 Resultados para la Configuración P13 con Programación Predictiva basada en SMAGA-2 Modificada

Conjunto Experimental	K=0 y VA=0,2		K=0,01 y VA=0,2		K=0,02 y VA=0,2	
	SSDS	SMAGA-2	SSDS	SMAGA-2	SSDS	SMAGA-2
	SSD10_P13_20	-0,116	-1,481	2,318	0,017	4,760
SSD50_P13_20	-0,085	-2,241	2,009	-0,759	4,088	0,637
SSD100_P13_20	-0,062	-3,167	1,681	-1,677	3,294	-0,043
SSD125_P13_20	-0,053	-3,627	1,565	-2,164	3,223	-0,486

Tabla IV-5. Valor de IPSPPVEPT a partir de SMAGA-2 Modificado con adelanto para 20 trabajos, y un valor de VA de 0,2.

Conjunto Experimental	K=0 y VA=0,4		K=0,01 y VA=0,4		K=0,02 y VA=0,4	
	SSDS	SMAGA-2	SSDS	SMAGA-2	SSDS	SMAGA-2
SSD10_P13_20	-0,117	-1,440	2,807	0,229	5,698	1,739
SSD50_P13_20	-0,083	-2,225	2,354	-0,485	4,685	1,234
SSD100_P13_20	-0,065	-3,380	1,944	-1,339	4,108	0,71026
SSD125_P13_20	-0,052	-3,375	1,987	-1,715	4,092	0,136

Tabla IV-6. Valor de IPSPPVEPT a partir de SMAGA-2 Modificado con adelanto para 20 trabajos, y un valor de VA de 0,4.

Conjunto Experimental	K=0 y VA=0,1		K=0,005 y VA=0,1		K=0,01 y VA=0,1	
	SSDS	SMAGA-2	SSDS	SMAGA-2	SSDS	SMAGA-2
SSD10_P13_50	-0,041	-3,067	1,441	-0,954	2,958	1,117
SSD50_P13_50	-0,032	-4,738	1,140	-2,440	2,293	-0,410
SSD100_P13_50	-0,021	-6,265	0,952	-4,078	1,946	-1,848
SSD125_P13_50	-0,019	-6,162	0,939	-4,016	1,942	-1,831

Tabla IV-7. Valor de IPSPPVEPT a partir de SMAGA-2 Modificado con adelanto para 50 trabajos, y un valor de VA de 0,1.

Conjunto Experimental	K=0 y VA=0,1		K=0,005 y VA=0,1		K=0,01 y VA=0,1	
	SSDS	SMAGA-2	SSDS	SMAGA-2	SSDS	SMAGA-2
SSD10_P13_50	-0,212	-3,121	2,082	-0,089	4,086	2,743
SSD50_P13_50	-0,115	-4,511	1,596	-1,536	3,253	1,582
SSD100_P13_50	-0,020	-6,400	1,355	-3,000	2,853	0,172
SSD125_P13_50	-0,113	-6,420	1,363	-2,922	2,725	0,432

Tabla IV-8. Valor de IPSPPVEPT a partir de SMAGA-2 Modificado con adelanto para 50 trabajos, y un valor de VA de 0,2.

A-IV.1.3 Análisis Estadístico para la Configuración P13 con Programación Predictiva basada en SMAGA-2

A-IV.1.3.1 Análisis del Bloque P13_20

En relación con la normalidad del experimento se ha obtenido un Chi-Square = 4,50004 con 4 d.f. y un P-Value = 0,342543, lo que descarta la posibilidad de que la variable analizada no proceda de una distribución normal. Así mismo se puede observar en la Figura IV-1 como la variable se distribuye muy próxima a la línea continua diagonal, lo que implica su proximidad a la distribución Normal.

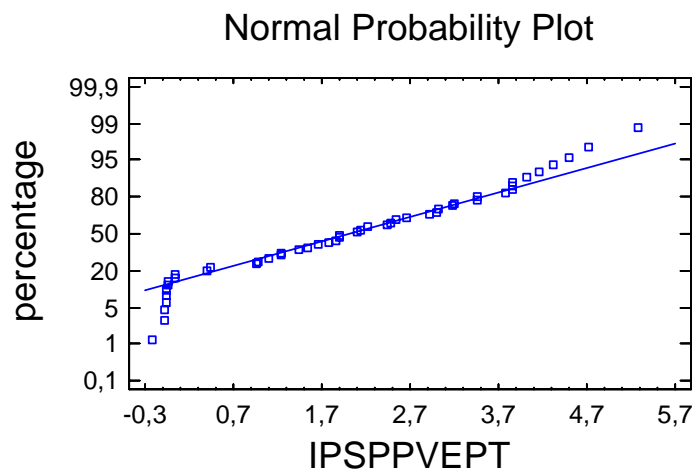


Figura IV-1. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPSPVEPT del bloque Adelanto-Predictiva SMAGA-2-P13_20 en relación a las frecuencias relativas acumuladas.

Por lo que respecta a la homogeneidad de las varianzas se ha realizado el test de Levene para la variable dependiente con respecto a las variables independientes obteniéndose los siguientes valores:

- IPSPVEPT respecto a CEX: test de Levene: 0,0904059 P-Value = 0,964946
- IPSPVEPT respecto a K: test de Levene: 0,56945 P-Value = 0,569861
- IPSPVEPT respecto a VA: test de Levene: 1,01089 P-Value = 0,319949
- IPSPVEPT respecto a ALG: test de Levene: 1,20258 P-Value = 0,278514

Por lo que se descarta que las varianzas no sean homogéneas.

Por lo que se refiere a la linealidad se empleará el “gráfico de residuos (X_t, \bar{u}_t) y (\hat{Y}_t, \bar{u}_t)”, que permite disponer los residuos, en el eje de ordenadas, frente a las variables exógenas (X_t), y endógena (\hat{Y}_t).

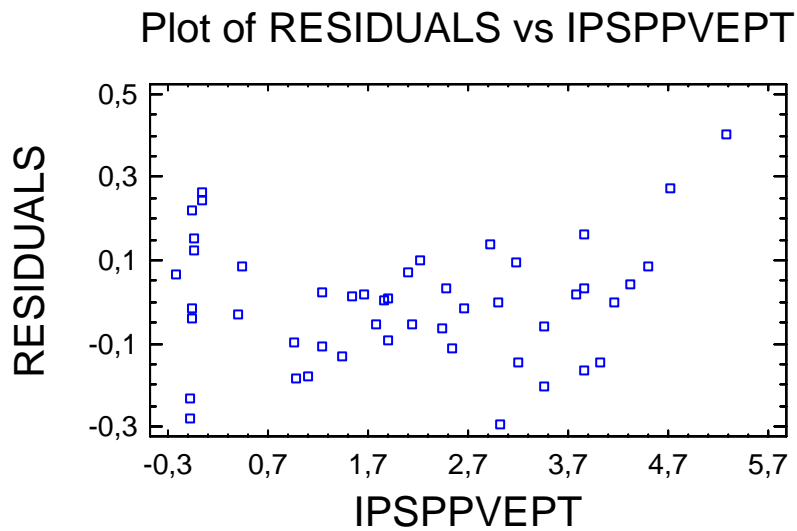


Figura IV-2. Gráfico de Residuos frente a los niveles de la Variable IPSPVEPT del bloque Adelanto-Predictiva SMAGA-2-P13_20.

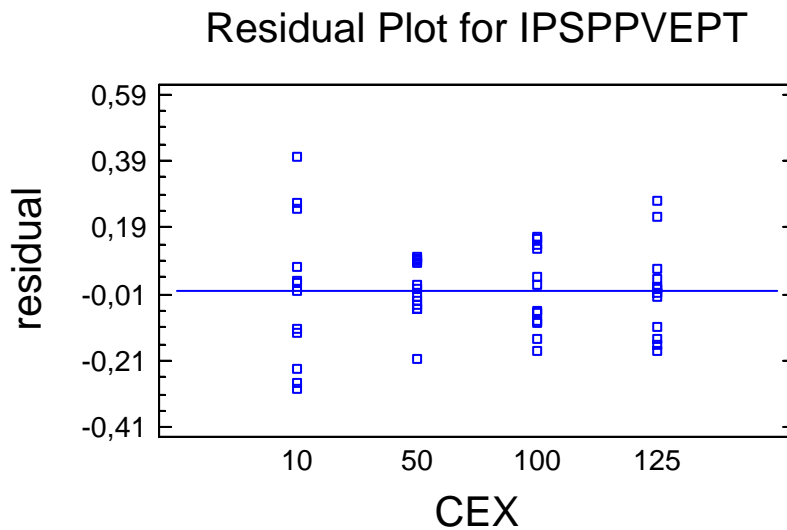


Figura IV-3. Gráfico de Residuos frente a los niveles del factor CEX del bloque Adelanto-Predictiva SMAGA-2-P13_20.

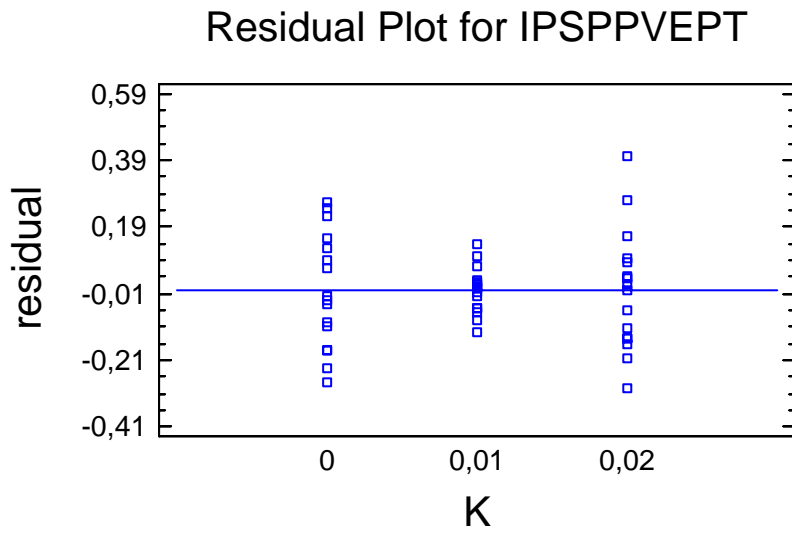


Figura IV-4. Gráfico de Residuos frente a los niveles del factor k del bloque Adelanto-Predictiva SMAGA-2-P13_20.

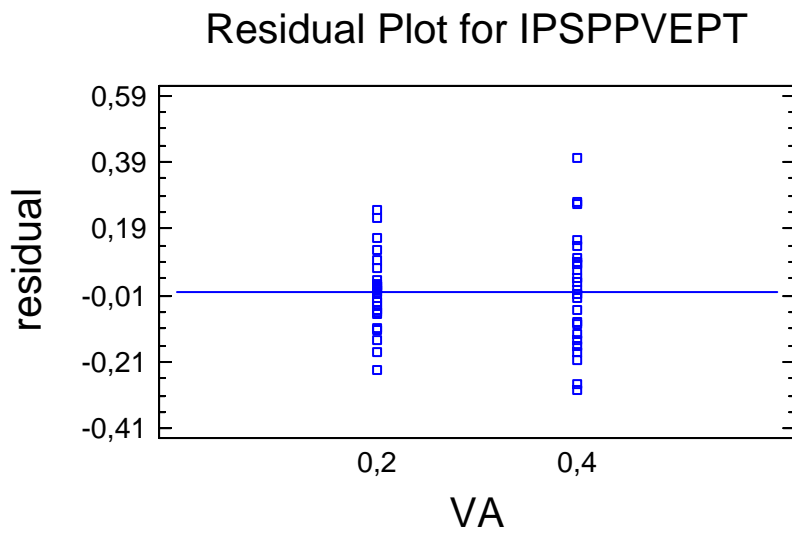


Figura IV-5. Gráfico de Residuos frente a los niveles del factor VA del bloque Adelanto-Predictiva SMAGA-2-P13_20.

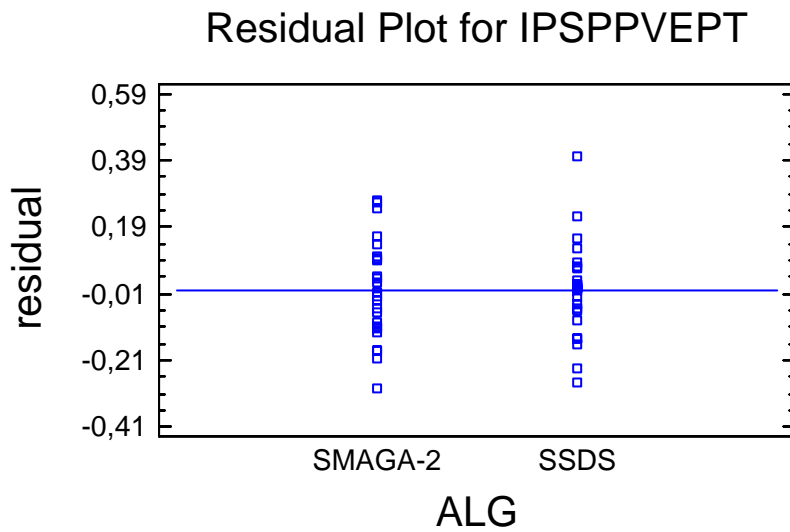


Figura IV-6. Gráfico de Residuos frente a los niveles del factor ALG del bloque Adelanto-Predictiva SMAGA-2-P13_20.

Como se observa en las figuras correspondientes a los residuos no se observa una falta de linealidad significativa en los mismos. Por lo que se descarta la ausencia de linealidad.

La correlación se detecta mediante el gráfico de residuos en relación al orden de ejecución de los tratamientos. Como se observa en la Figura IV-7 no se puede observar ninguna tendencia significativa en la distribución de los mismos por lo que se descarta la existencia de correlación.

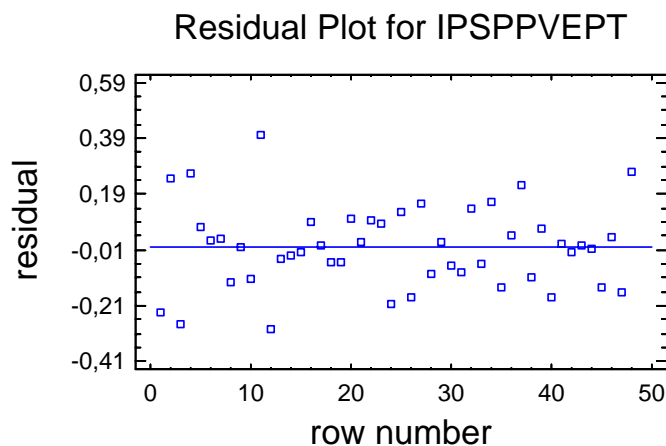


Figura IV-7. Residuos frente al orden de ejecución de los tratamientos del bloque Adelanto-Predictiva SMAGA-2-P13_20.

Una vez verificado el cumplimiento de las condiciones para el análisis ANOVA se obtiene y analiza la misma.

Analysis of Variance for IPSPVEPT - Type III Sums of Squares

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
MAIN EFFECTS					
A:CEX	1,24925	3	0,416416	9,42	0,0003
B:K	100,356	2	50,1782	1135,04	0,0000
C:VA	1,18115	1	1,18115	26,72	0,0000
D:ALG	0,458868	1	0,458868	10,38	0,0038
INTERACTIONS					
AB	0,191357	6	0,0318928	0,72	0,6366
AC	0,00970767	3	0,00323589	0,07	0,9738
AD	7,22712	3	2,40904	54,49	0,0000
BC	0,903048	2	0,451524	10,21	0,0007
BD	2,60698	2	1,30349	29,49	0,0000
CD	0,0110183	1	0,0110183	0,25	0,6224
RESIDUAL	1,01679	23	0,0442083		
TOTAL (CORRECTED)	115,212	47			

All F-ratios are based on the residual mean square error.

Tabla IV-9. ANOVA correspondiente IPSPVEPT del bloque Adelanto-Predictiva SMAGA-2-P13_20.

Como se puede ver en la Tabla IV-9 disponemos de 48 datos y tenemos 23 grados de libertad residuales. El contraste de hipótesis sobre la potencia (P-value) para un $\alpha=0,05$ implica que son significativos todos efectos principales de todos los factores, y además lo son las interacciones AD, BC y BD.

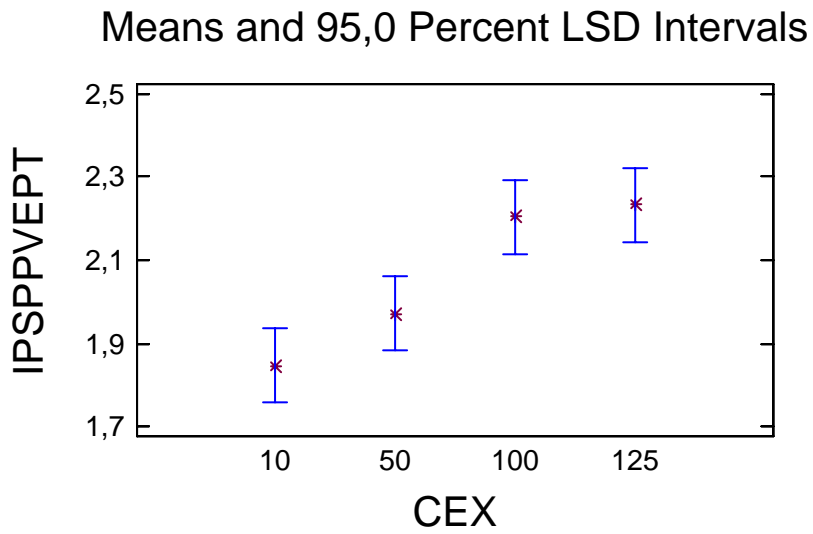


Figura IV-8. Gráfico del efecto principal del factor CEX del bloque Adelanto-Predictiva SMAGA-2-P13_20.

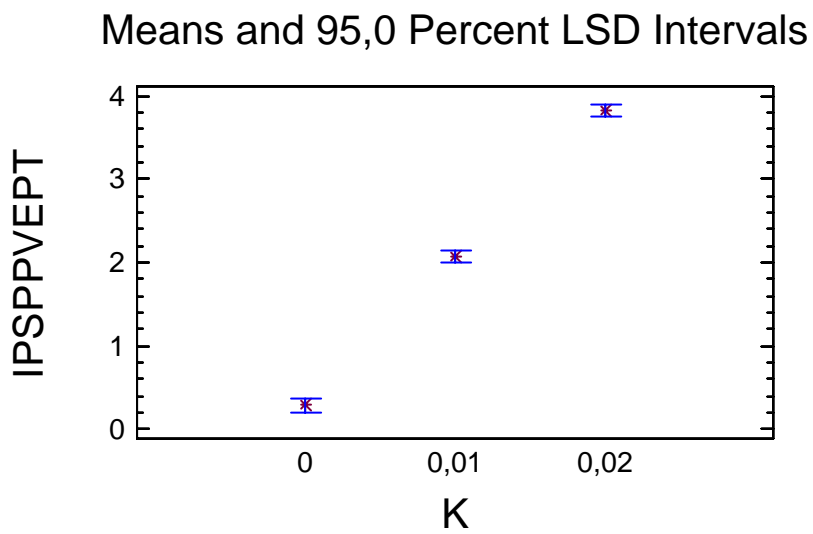


Figura IV-9. Gráfico del efecto principal del factor K del bloque Adelanto-Predictiva SMAGA-2-P13_20.

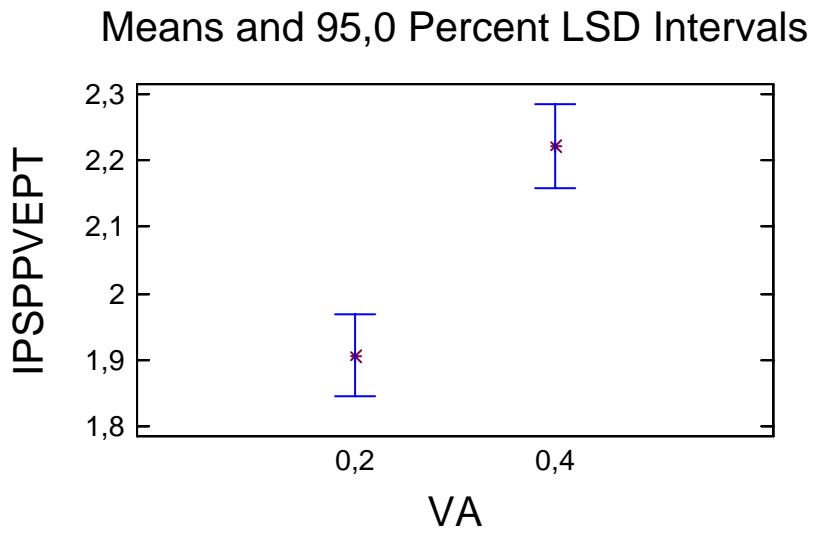


Figura IV-10. Gráfico del efecto principal del factor VA del bloque Adelanto-Predictiva SMAGA-2-P13_20.

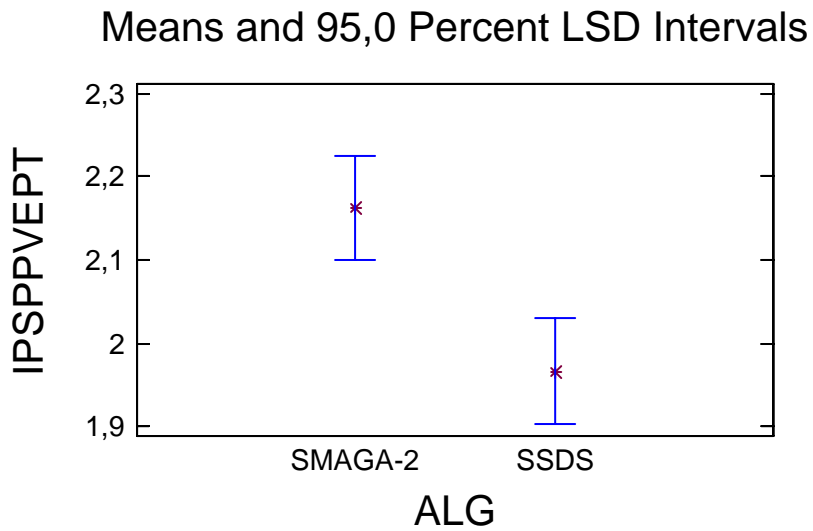


Figura IV-11. Gráfico del efecto principal del factor ALG del bloque Adelanto-Predictiva SMAGA-2-P13_20.

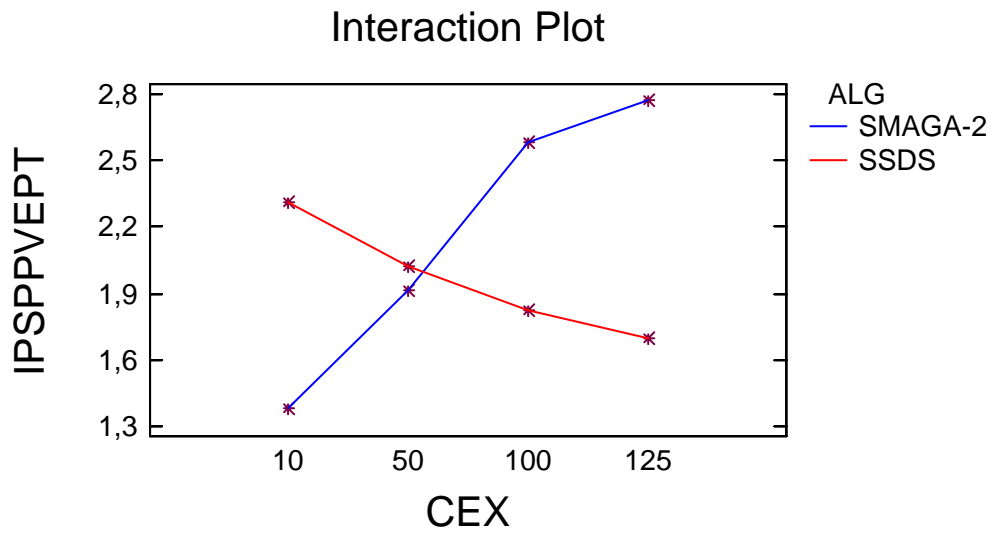


Figura IV-12. Gráfico del efecto de la interacción entre el factor CEX y el factor ALG del bloque Adelanto-Predictiva SMAGA-2-P13_20.

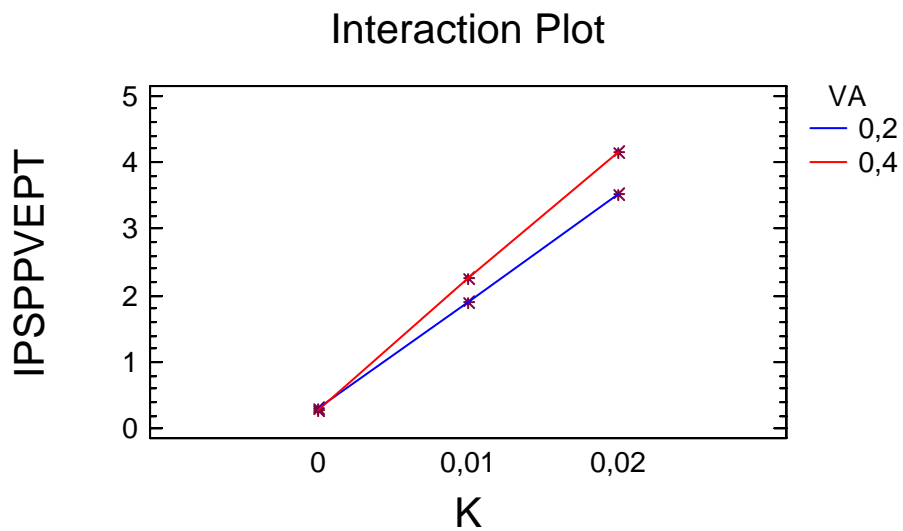


Figura IV-13. Gráfico del efecto de la interacción entre el factor K y el factor VA del bloque Adelanto-Predictiva SMAGA-2-P13_20.

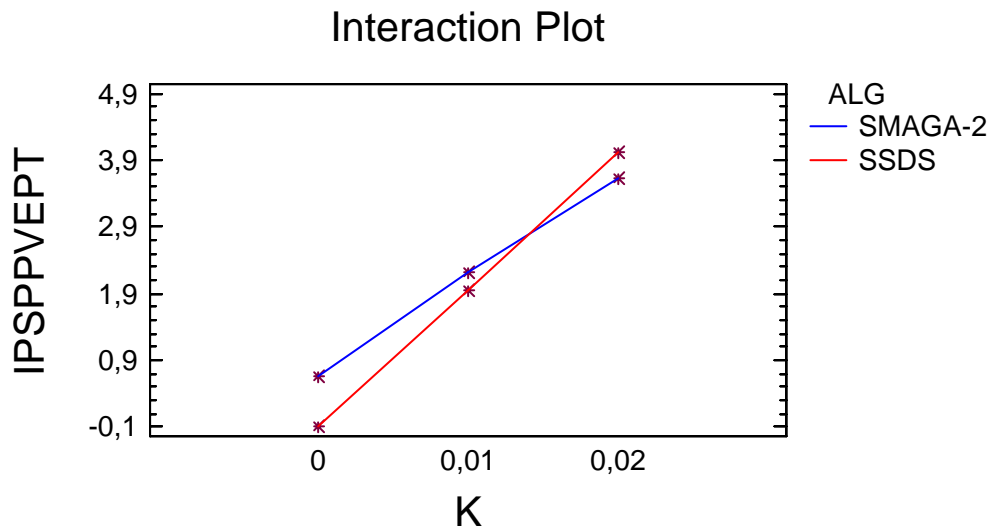


Figura IV-14. Gráfico del efecto de la interacción entre el factor K y el factor ALG del bloque Adelanto-Predictiva SMAGA-2-P13_20.

A-IV.1.3.2 Análisis del Bloque P13_50

En este bloque experimental se ha necesitado realizar una transformación de la variable dependiente debido a que se ha observado la existencia de alguna relación no lineal entre los factores. La transformación que se ha empleado ha sido el cálculo de logaritmo neperiano (LOG en la aplicación Statgraphics) de la variable a la cual se le ha sumado una unidad para evitar valores negativos que impidieran el cálculo del logaritmo. En todo caso, la significación de los factores y los test realizados son tan válidos como si se hubiesen realizado sin transformación.

En relación con la normalidad del experimento se ha obtenido un Chi-Square = 4,79168 con 4 d.f., y un P-Value = 0,309348, lo que descarta la posibilidad de que la variable analizada no proceda de una distribución normal. Así mismo se puede observa en la Figura IV-15 como la variable se distribuye muy próxima a la línea continua diagonal, lo que implica su proximidad a la distribución Normal.

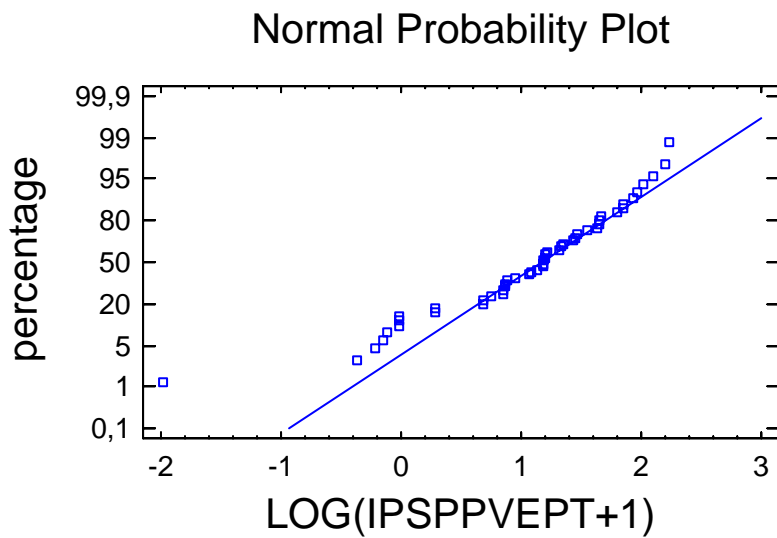


Figura IV-15. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPSPVEPT del bloque Adelanto-Predictiva SMAGA-2-P13_50 en relación a las frecuencias relativas acumuladas.

Por lo que respecta a la homogeneidad de las varianzas se ha realizado el test de Levene para la variable dependiente con respecto a las variables independientes obteniéndose los siguientes valores:

- IPSPVEPT respecto a CEX: test de Levene: 0,0904059 P-Value = 0,89672
- IPSPVEPT respecto a K: test de Levene: 0,56945 P-Value = 0,0587367
- IPSPVEPT respecto a VA: test de Levene: 1,01089 P-Value = 0,347143
- IPSPVEPT respecto a ALG: test de Levene: 1,20258 P-Value = 0,26133

Por lo que se descarta que las varianzas no sean homogéneas.

Por lo que se refiere a la linealidad se empleará el “gráfico de residuos (X_t, \bar{u}_t) y (\hat{Y}_t, \bar{u}_t)”, que permite disponer los residuos (\bar{u}_t), en el eje de ordenadas, frente a las variables exógenas (X_t), y endógena (\hat{Y}_t).

Plot of RESIDUALS vs LOG(IPSPVEPT+1)

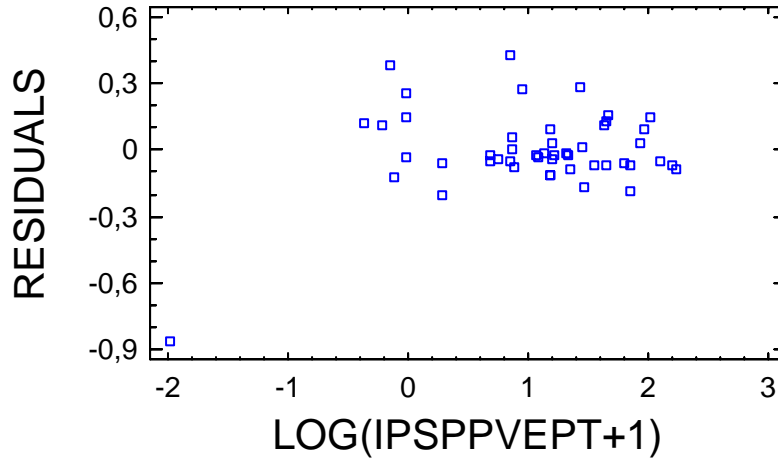


Figura IV-16. Gráfico de Residuos frente a los niveles de la Variable IPSPVEPT del bloque Adelanto-Predictiva SMAGA-2-P13_50.

Residual Plot for LOG(IPSPVEPT+1)

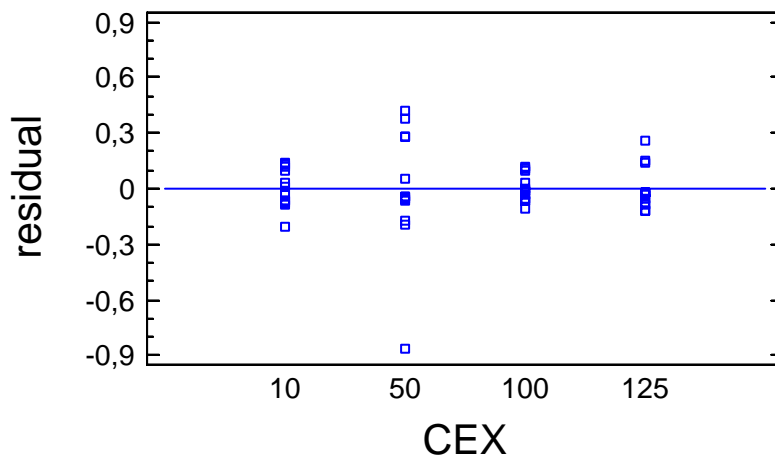


Figura IV-17. Gráfico de Residuos frente a los niveles del factor CEX del bloque Adelanto-Predictiva SMAGA-2-P13_50.

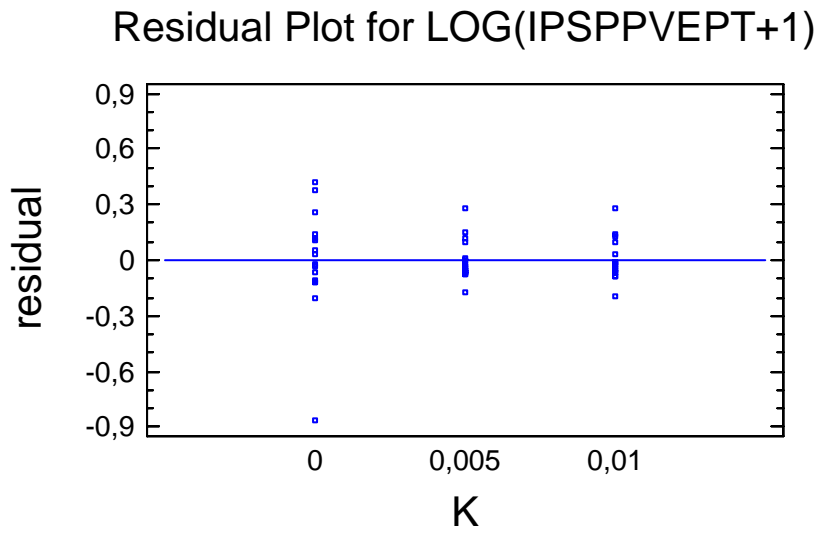


Figura IV-18. Gráfico de Residuos frente a los niveles del factor k del bloque Adelanto-Predictiva SMAGA-2-P13_50.

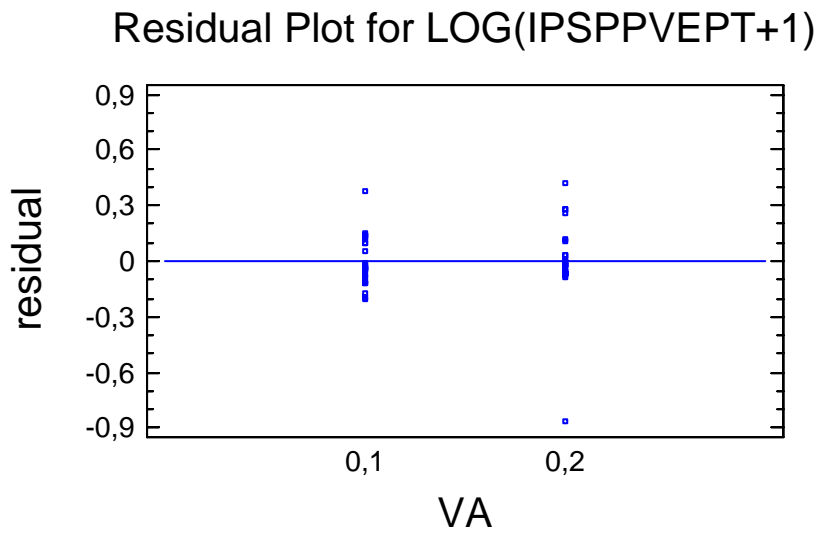


Figura IV-19. Gráfico de Residuos frente a los niveles del factor VA del bloque Adelanto-Predictiva SMAGA-2-P13_50.

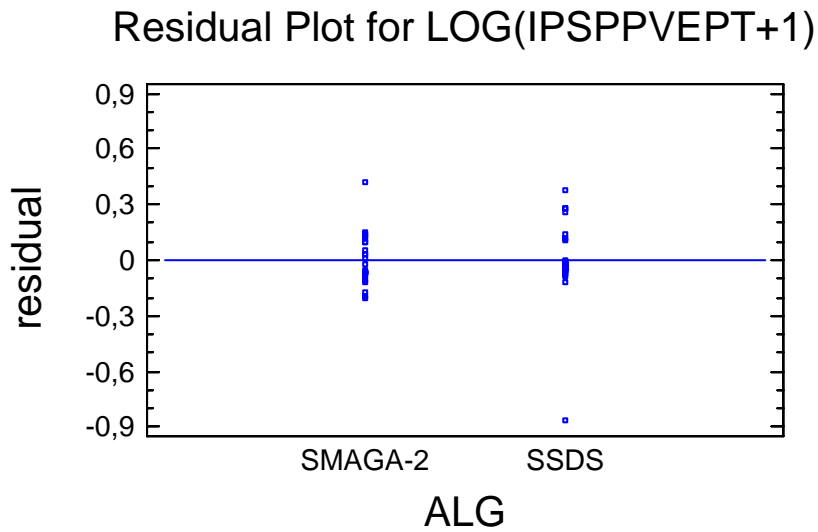


Figura IV-20. Gráfico de Residuos frente a los niveles del factor ALG del bloque Adelanto-Predictiva SMAGA-2-P13_50.

Como se observa en las figuras correspondientes a los residuos no se observa una falta de linealidad significativa en los mismos. Por lo que se descarta la ausencia de linealidad de la variable transformada.

La correlación se detecta mediante el gráfico de residuos en relación al orden de ejecución de los tratamientos. Como se observa en la Figura IV-21 no se puede observar ninguna tendencia significativa en la distribución de los mismos por lo que se descarta la existencia de correlación.

Residual Plot for LOG(IPSPVEPT+1)

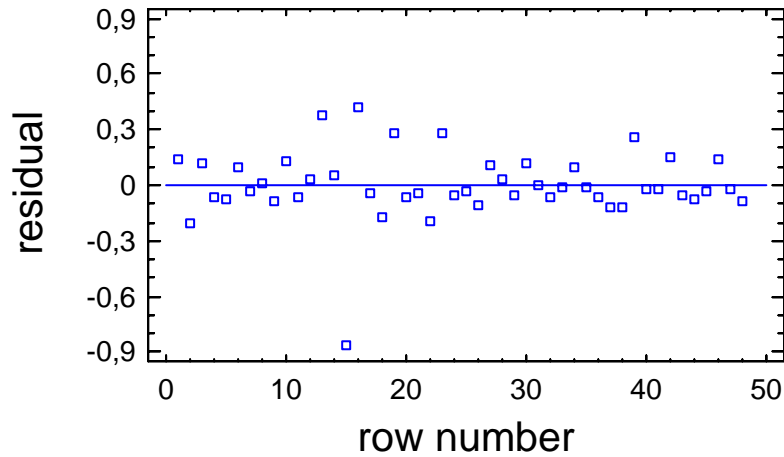


Figura IV-21. Residuos frente al orden de ejecución de los tratamientos del bloque Adelanto-Predictiva SMAGA-2-P13_50.

Una vez verificado el cumplimiento de las condiciones para el análisis ANOVA se obtiene y analiza la misma.

Analysis of Variance for LOG(IPSPVEPT+1) - Type III Sums of Squares

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
MAIN EFFECTS					
A:CEX	0,640294	3	0,213431	2,97	0,0529
B:K	16,1033	2	8,05166	112,10	0,0000
C:VA	0,026396	1	0,026396	0,37	0,5503
D:ALG	9,6187	1	9,6187	133,92	0,0000
INTERACTIONS					
AB	0,785419	6	0,130903	1,82	0,1389
AC	0,182843	3	0,0609477	0,85	0,4816
AD	1,04723	3	0,349077	4,86	0,0092
BC	0,652912	2	0,326456	4,55	0,0217
BD	0,744688	2	0,372344	5,18	0,0139
CD	0,127275	1	0,127275	1,77	0,1962
RESIDUAL	1,65199	23	0,0718258		
TOTAL (CORRECTED)	31,5811	47			

All F-ratios are based on the residual mean square error.

Tabla IV-10. ANOVA correspondiente IPSPVEPT del bloque Adelanto-Predictiva SMAGA-2-P13_50.

Como se puede ver en la Tabla IV-10 disponemos de 48 datos y tenemos 23 grados de libertad residuales. El contraste de hipótesis sobre la potencia (P-value) para un $\alpha=0,05$ implica que

son significativos los efectos principales de los factores B y D, y además lo son las interacciones AD, BC y BD.

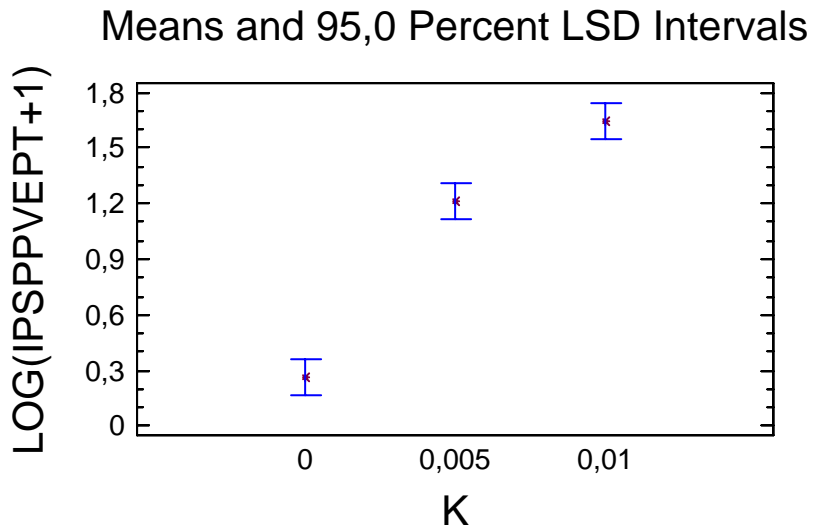


Figura IV-22. Gráfico del efecto principal del factor K del bloque Adelanto-Predictiva SMAGA-2-P13_50.

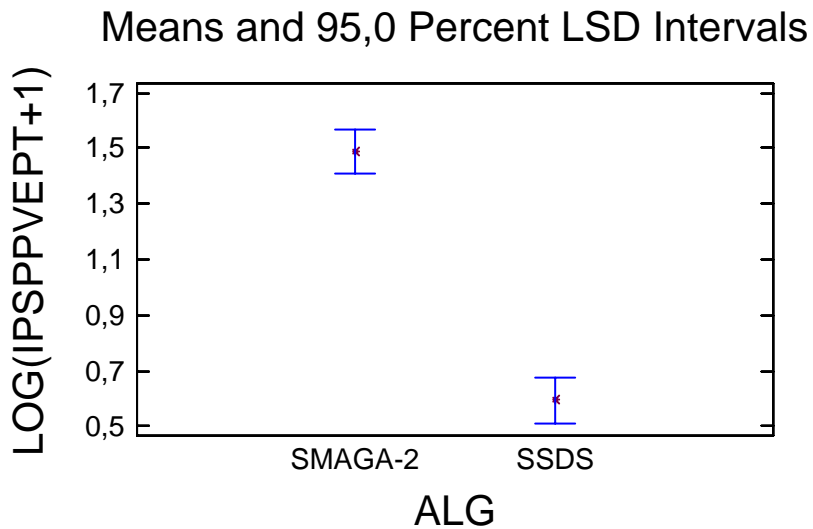


Figura IV-23. Gráfico del efecto principal del factor ALG del bloque Adelanto-Predictiva SMAGA-2-P13_50.

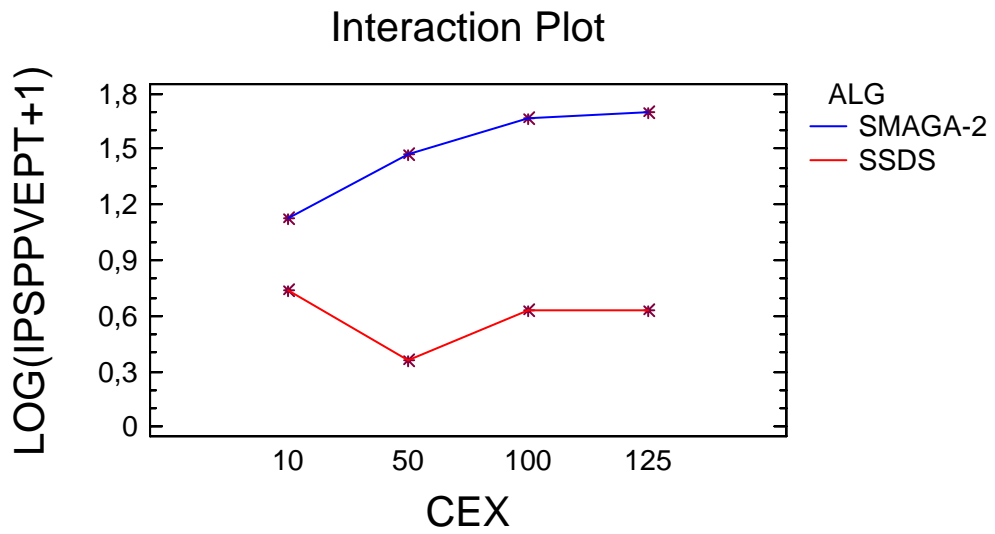


Figura IV-24. Gráfico del efecto de la interacción entre el factor CEX y el factor ALG del bloque Adelanto-Predictiva SMAGA-2-P13_50.

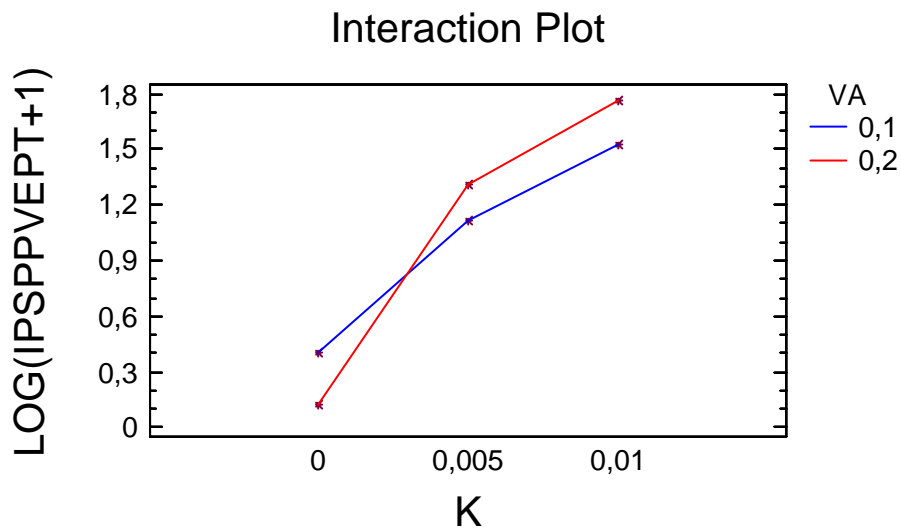


Figura IV-25. Gráfico del efecto de la interacción entre el factor K y el factor VA del bloque Adelanto-Predictiva SMAGA-2-P13_50.

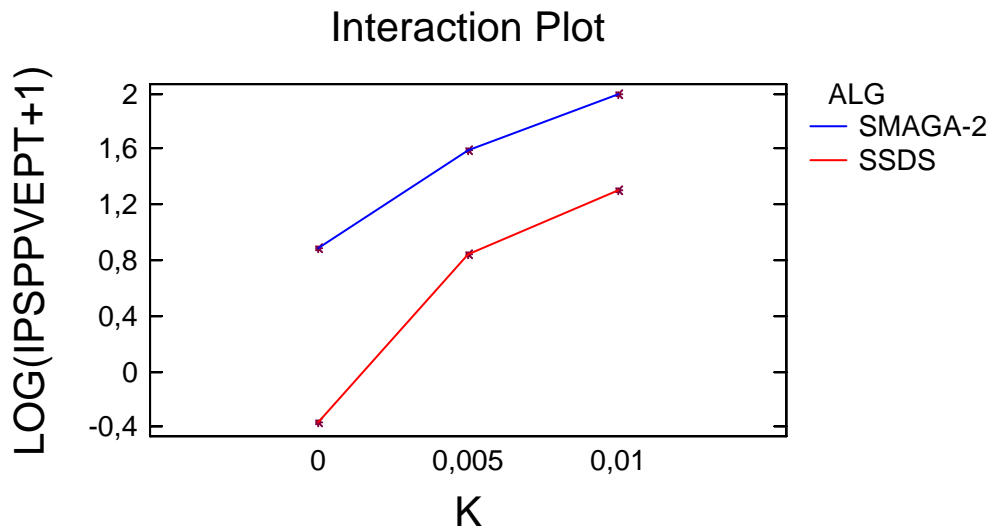


Figura IV-26. Gráfico del efecto de la interacción entre el factor K y el factor ALG del bloque Adelanto-Predictiva SMAGA-2-P13_50.

A-IV.1.4 Resultados para la Configuración P3 con Programación Predictiva basada en SMAGA-2

Conjunto Experimental	K=0 y VA=0,2		K=0,01 y VA=0,2		K=0,02 y VA=0,2	
	SSDS	SMAGA-2	SSDS	SMAGA-2	SSDS	SMAGA-2
SSD10_P3_20	0,291	0,725	1,868	1,679	3,344	2,766
SSD50_P3_20	0,654	2,285	2,027	3,663	3,642	4,957
SSD100_P3_20	0,336	3,676	1,410	4,960	2,560	6,931
SSD125_P3_20	0,643	3,790	1,662	4,904	2,844	6,549

Tabla IV-11. Valor de IPSPVEPT a partir de SMAGA-2 con adelanto para 20 trabajos, y un valor de VA de 0,2.

Conjunto Experimental	K=0 y VA=0,4		K=0,01 y VA=0,4		K=0,02 y VA=0,4	
	SSDS	SMAGA-2	SSDS	SMAGA-2	SSDS	SMAGA-2
SSD10_P3_20	0,304	0,739	1,948	1,956	3,470	3,324
SSD50_P3_20	0,612	2,139	2,263	3,922	3,745	5,328
SSD100_P3_20	0,286	3,448	1,699	5,494	2,894	7,385
SSD125_P3_20	0,611	3,803	2,027	5,337	3,153	7,454

Tabla IV-12. Valor de IPSPVEPT a partir de SMAGA-2 con adelanto para 20 trabajos, y un valor de VA de 0,4.

Conjunto Experimental	K=0 y VA=0,1		K=0,005 y VA=0,1		K=0,01 y VA=0,1	
	SSDS	SMAGA-2	SSDS	SMAGA-2	SSDS	SMAGA-2
SSD10_P3_50	0,966	2,049	1,687	3,847	2,669	5,702
SSD50_P3_50	0,919	3,814	1,613	5,944	2,428	7,236
SSD100_P3_50	1,217	4,791	1,895	6,771	2,395	8,468
SSD125_P3_50	1,309	6,069	1,956	7,743	2,596	9,754

Tabla IV-13. Valor de IPSPVEPT a partir de SMAGA-2 con adelanto para 50 trabajos, y un valor de VA de 0,1.

Conjunto Experimental	K=0 y VA=0,1		K=0,005 y VA=0,1		K=0,01 y VA=0,1	
	SSDS	SMAGA-2	SSDS	SMAGA-2	SSDS	SMAGA-2
SSD10_P3_50	0,828	2,207	1,989	4,673	3,060	7,099
SSD50_P3_50	1,011	3,685	1,914	6,438	2,720	9,151
SSD100_P3_50	1,183	5,317	2,054	7,644	2,783	10,631
SSD125_P3_50	1,411	5,998	2,395	8,523	2,912	10,770

Tabla IV-14. Valor de IPSPVEPT a partir de SMAGA-2 con adelanto para 50 trabajos, y un valor de VA de 0,2.

A-IV.1.5 Resultados para la Configuración P3 con Programación Predictiva basada en SMAGA-2 Modificado

Conjunto Experimental	K=0 y VA=0,2		K=0,01 y VA=0,2		K=0,02 y VA=0,2	
	SSDS	SMAGA-2	SSDS	SMAGA-2	SSDS	SMAGA-2
SSD10_P3_20	-0,141	-2,441	1,362	-1,230	2,936	0,251
SSD50_P3_20	-0,068	-2,343	1,259	-0,56	2,572	1,102
SSD100_P3_20	-0,135	-2,145	1,055	-0,205	2,139	1,655
SSD125_P3_20	-0,150	-3,038	0,904	-1,169	2,001	0,830

Tabla IV-15. Valor de IPSPVEPT a partir de SMAGA-2 Modificado con adelanto para 20 trabajos, y un valor de VA de 0,2.

Conjunto Experimental	K=0 y VA=0,4		K=0,01 y VA=0,4		K=0,02 y VA=0,4	
	SSDS	SMAGA-2	SSDS	SMAGA-2	SSDS	SMAGA-2
SSD10_P3_20	-0,181	-2,488	1,449	-1,107	3,045	0,385
SSD50_P3_20	-0,093	-2,329	1,449	-0,419	3,014	1,494
SSD100_P3_20	-0,103	-2,092	1,185	0,078	2,556	2,128
SSD125_P3_20	-0,124	-3,134	1,094	-0,870	2,273	1,361

Tabla IV-16. Valor de IPSPVEPT a partir de SMAGA-2 Modificado con adelanto para 20 trabajos, y un valor de VA de 0,4.

Conjunto Experimental	K=0 y VA=0,1		K=0,005 y VA=0,1		K=0,01 y VA=0,1	
	SSDS	SMAGA-2	SSDS	SMAGA-2	SSDS	SMAGA-2
SSD10_P3_50	-0,010	-3,262	0,996	-1,345	1,915	0,560
SSD50_P3_50	0,018	-3,062	0,740	-1,087	1,460	0,907
SSD100_P3_50	-0,023	-2,384	0,598	-0,420	1,233	1,542
SSD125_P3_50	0,130	-2,701	0,598	-0,731	1,236	1,303

Tabla IV-17. Valor de IPSPVEPT a partir de SMAGA-2 Modificado con adelanto para 50 trabajos, y un valor de VA de 0,1.

Conjunto Experimental	K=0 y VA=0,1		K=0,005 y VA=0,1		K=0,01 y VA=0,1	
	SSDS	SMAGA-2	SSDS	SMAGA-2	SSDS	SMAGA-2
SSD10_P3_50	-0,022	-3,337	1,254	-0,498	2,479	2,423
SSD50_P3_50	0,008	-3,043	0,940	-0,257	1,888	2,668
SSD100_P3_50	0,002	-2,412	0,778	0,591	1,527	3,589
SSD125_P3_50	0,013	-2,752	0,731	0,260	1,500	3,128

Tabla IV-18. Valor de IPSPVEPT a partir de SMAGA-2 Modificado con adelanto para 50 trabajos, y un valor de VA de 0,2.

A-IV.1.6 Análisis Estadístico para la Configuración P3 con Programación Predictiva basada en SMAGA-2

A-IV.1.6.1 Análisis del Bloque P3_20

En este bloque experimental se ha necesitado realizar una transformación de la variable dependiente debido a que se ha observado la existencia de alguna relación no lineal entre los factores. La transformación que se ha empleado ha sido el cálculo de logaritmo neperiano

(LOG en la aplicación Statgraphics) de la variable a la cual se le ha sumado dos unidades para evitar valores negativos que impidieran el cálculo del logaritmo. En todo caso, la significación de los factores y los test realizados son tan válidos como si se hubiesen realizado sin transformación.

En relación con la normalidad del experimento se ha obtenido un Chi-Square = 8,00013 con 4 d.f. y un P-Value = 0,0915733, lo que descarta la posibilidad de que la variable analizada no proceda de una distribución normal. Así mismo se puede observar en la Figura IV-27 como la variable se distribuye muy próxima a la línea continua diagonal, lo que implica su proximidad a la distribución Normal.

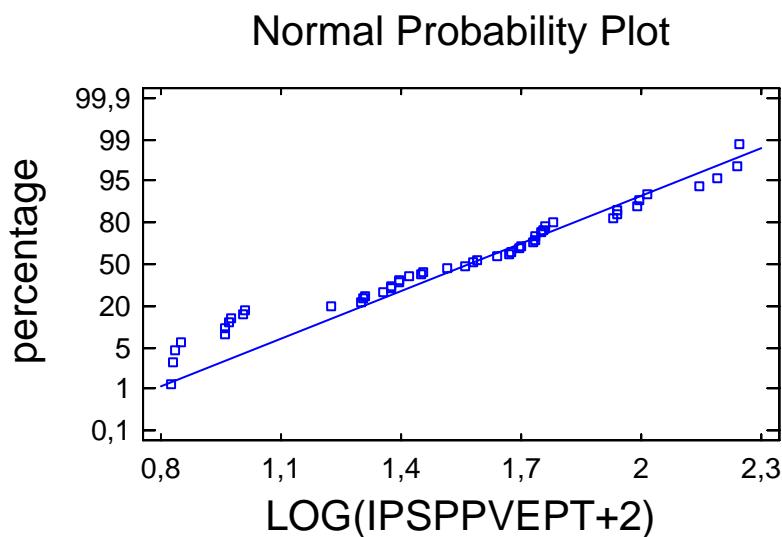


Figura IV-27. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPSPVEPT del bloque Adelanto-Predictiva SMAGA-2-P3_20 en relación a las frecuencias relativas acumuladas.

Por lo que respecta a la homogeneidad de las varianzas se ha realizado el test de Levene para la variable dependiente con respecto a las variables independientes obteniéndose los siguientes valores:

- IPSPVEPT respecto a CEX: test de Levene: 0,738387 P-Value = 0,534819
- IPSPVEPT respecto a K: test de Levene: 0,488775 P-Value = 0,616596
- IPSPVEPT respecto a VA: test de Levene: 0,0203471 P-Value = 0,887195
- IPSPVEPT respecto a ALG: test de Levene: 0,00165068 P-Value = 0,967768

Por lo que se descarta que las varianzas no sean homogéneas.

Por lo que se refiere a la linealidad se empleará el “gráfico de residuos (X_t, \bar{u}_t) y (\hat{Y}_t, \bar{u}_t)”, que permite disponer los residuos estudentizadosⁱ (\bar{u}_t), en el eje de ordenadas, frente a las variables exógenas (X_t), y endógena (\hat{Y}_t).

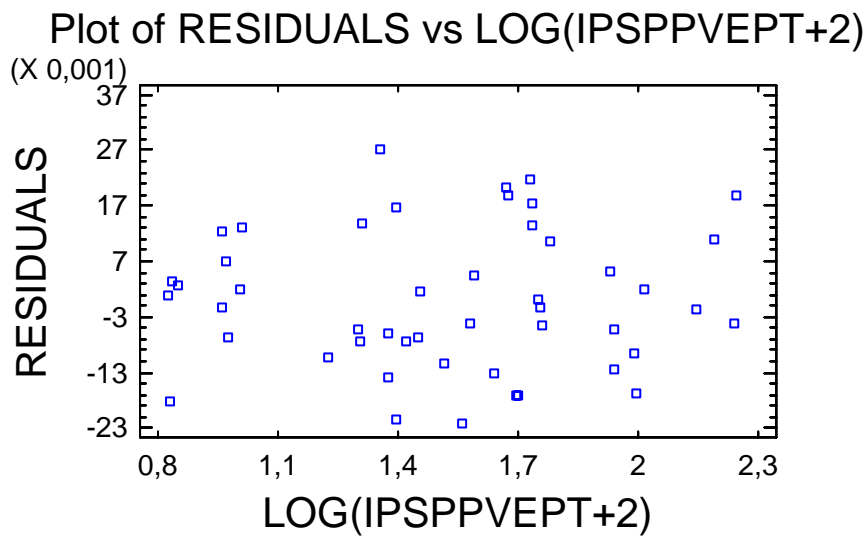


Figura IV-28. Gráfico de Residuos frente a los niveles de la Variable IPSPVEPT del bloque Adelanto-Predictiva SMAGA-2-P3_20.

ⁱ Consiste en el cálculo de la siguiente expresión para cada residuo: (residuo-media de los residuos)/desviación estándar de los residuos.

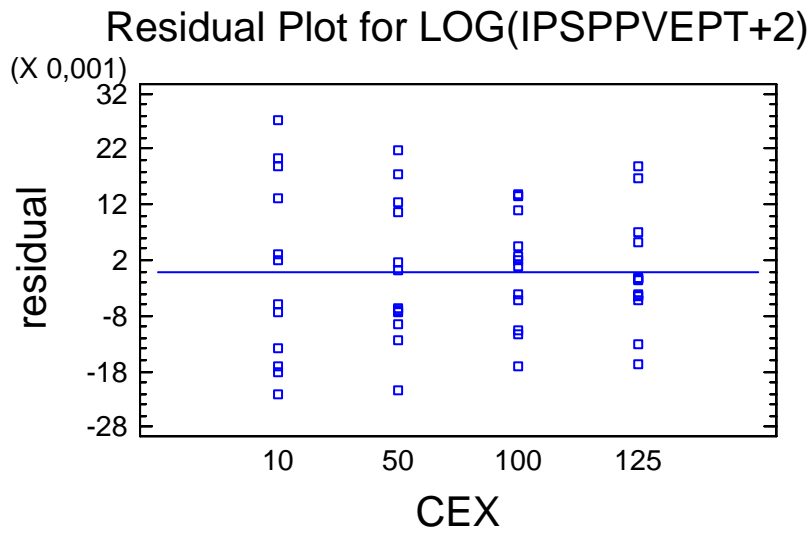


Figura IV-29. Gráfico de Residuos frente a los niveles del factor CEX del bloque Adelanto-Predictiva SMAGA-2-P3_20.

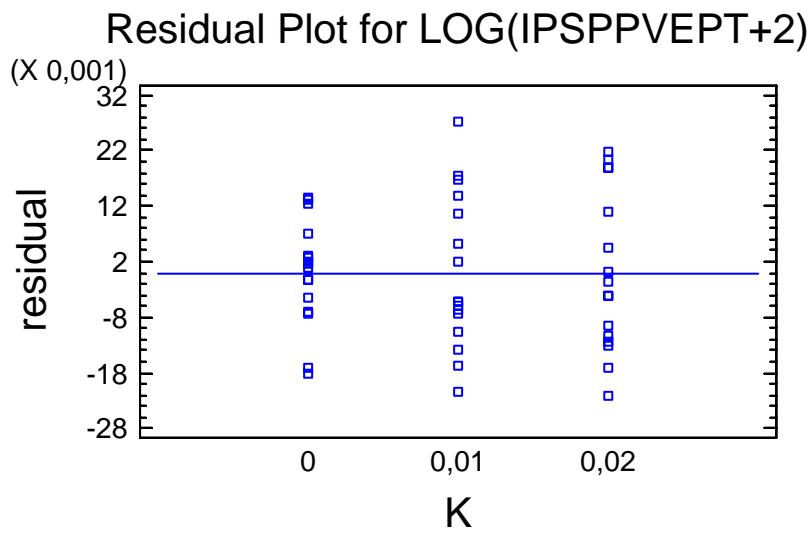


Figura IV-30. Gráfico de Residuos frente a los niveles del factor k del bloque Adelanto-Predictiva SMAGA-2-P3_20.

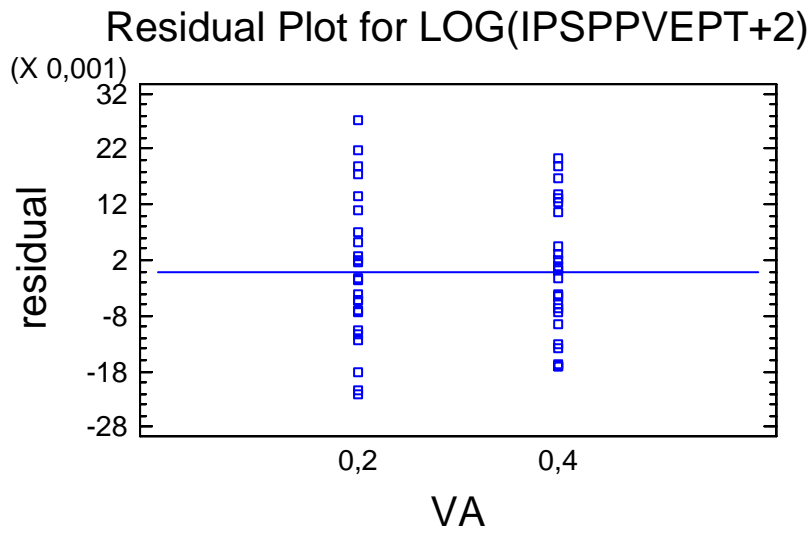


Figura IV-31. Gráfico de Residuos frente a los niveles del factor VA del bloque Adelanto-Predictiva SMAGA-2-P3_20.

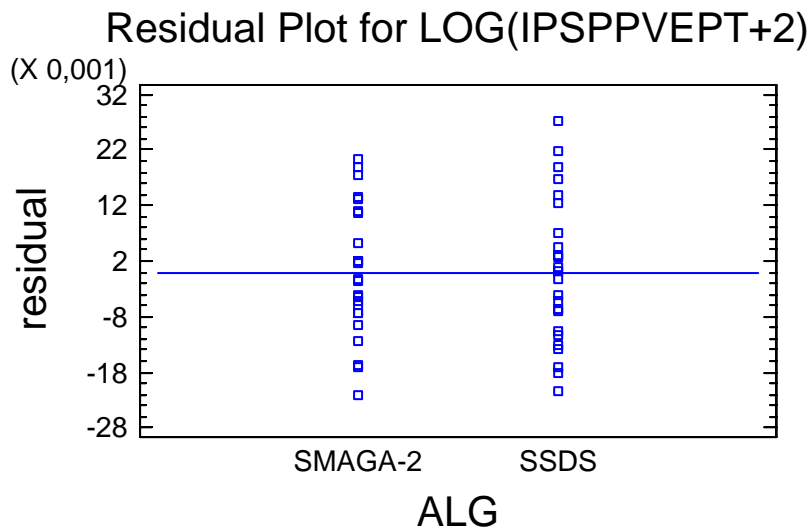


Figura IV-32. Gráfico de Residuos frente a los niveles del factor ALG del bloque Adelanto-Predictiva SMAGA-2-P3_20.

Como se observa en las figuras correspondientes a los residuos no se observa una falta de linealidad significativa en los mismos. Por lo que se descarta la ausencia de linealidad.

La correlación se detecta mediante el gráfico de residuos en relación al orden de ejecución de los tratamientos. Como se observa en la Figura IV-33 no se puede observar ninguna tendencia significativa en la distribución de los mismos por lo que se descarta la existencia de correlación.

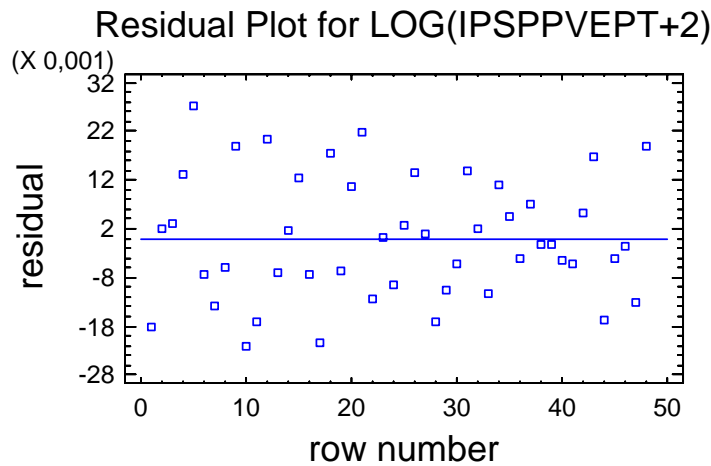


Figura IV-33. Residuos frente al orden de ejecución de los tratamientos del bloque Adelanto-Predictiva SMAGA-2-P3_20.

Una vez verificado el cumplimiento de las condiciones para el análisis ANOVA se obtiene y analiza la misma.

Analysis of Variance for LOG(IPSPVEPT+2) - Type III Sums of Squares

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
MAIN EFFECTS					
A:CEX	0,789322	3	0,263107	829,48	0,0000
B:K	3,227	2	1,6135	5086,79	0,0000
C:VA	0,0161097	1	0,0161097	50,79	0,0000
D:ALG	2,38907	1	2,38907	7531,87	0,0000
INTERACTIONS					
AB	0,0424005	6	0,00706674	22,28	0,0000
AC	0,00149223	3	0,000497412	1,57	0,2242
AD	0,981368	3	0,327123	1031,30	0,0000
BC	0,0154837	2	0,00774184	24,41	0,0000
BD	0,111071	2	0,0555353	175,08	0,0000
CD	0,000268663	1	0,000268663	0,85	0,3670
RESIDUAL	0,00729548	23	0,000317195		
TOTAL (CORRECTED)	7,58088	47			

All F-ratios are based on the residual mean square error.

Tabla IV-19. ANOVA correspondiente IPSPVEPT del bloque Adelanto-Predictiva SMAGA-2-P3_20.

Como se puede ver en la Tabla IV-19 disponemos de 48 datos y tenemos 23 grados de libertad residuales. El contraste de hipótesis sobre la potencia (P-value) para un $\alpha=0,05$ implica que son significativos todos efectos principales de todos los factores, y además lo son las interacciones AB, AD, BC y BD.

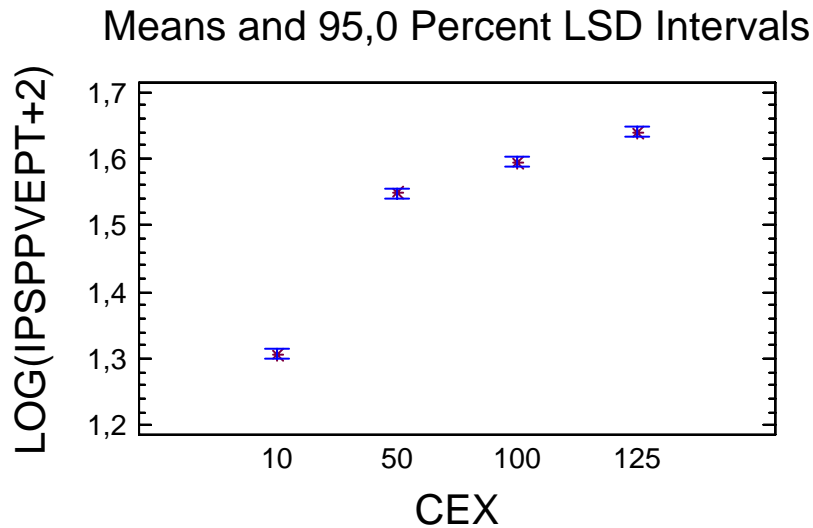


Figura IV-34. Gráfico del efecto principal del factor CEX del bloque Adelanto-Predictiva SMAGA-2-P3_20.

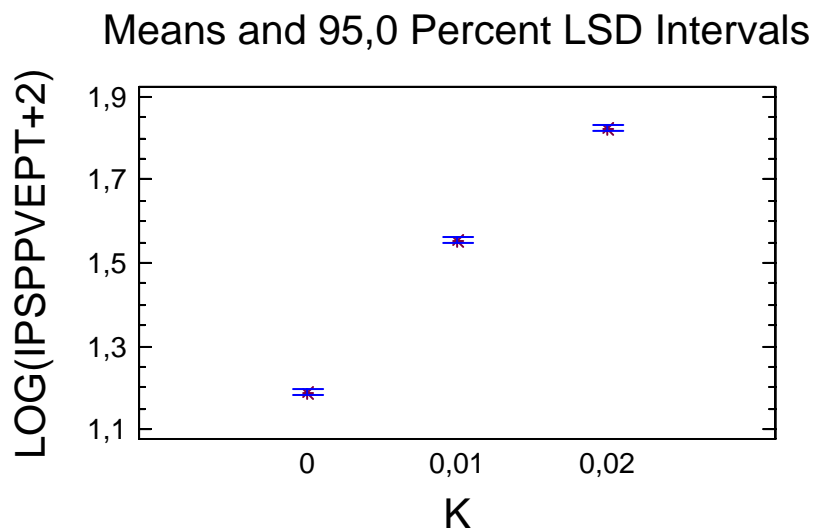


Figura IV-35. Gráfico del efecto principal del factor K del bloque Adelanto-Predictiva SMAGA-2-P3_20.

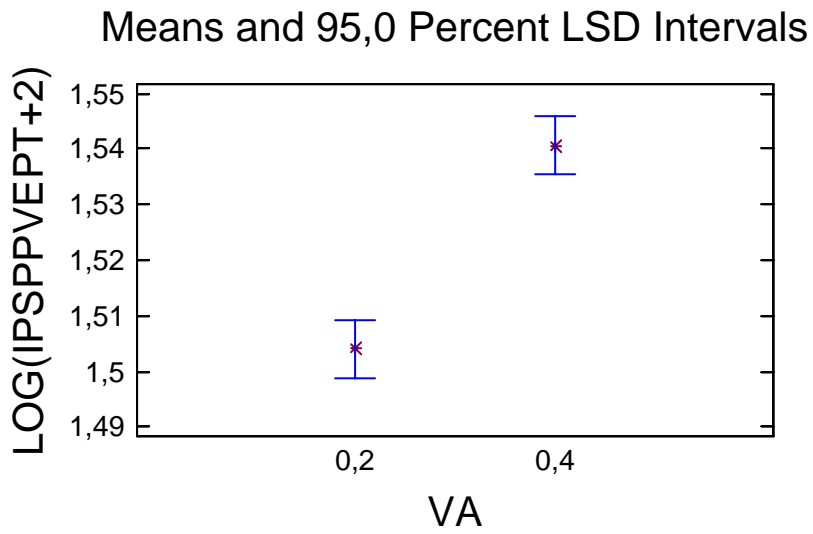


Figura IV-36. Gráfico del efecto principal del factor VA del bloque Adelanto-Predictiva SMAGA-2-P3_20.

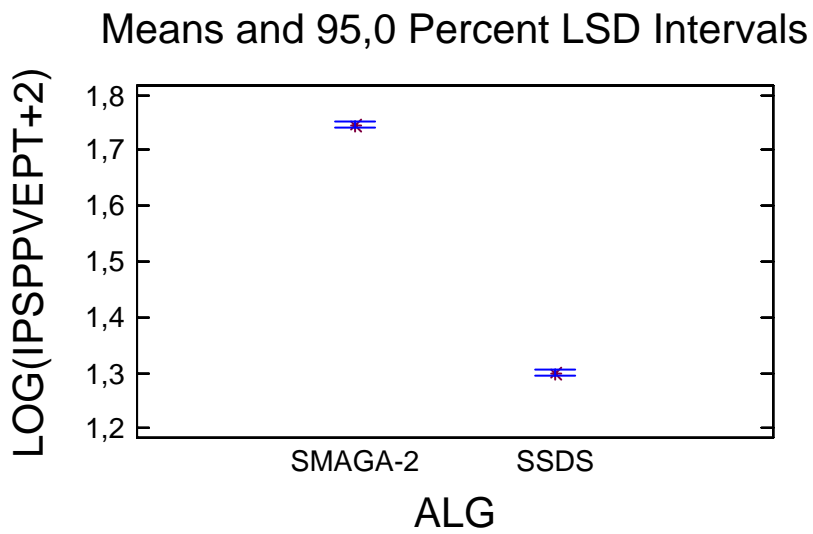


Figura IV-37. Gráfico del efecto principal del factor ALG del bloque Adelanto-Predictiva SMAGA-2-P3_20.

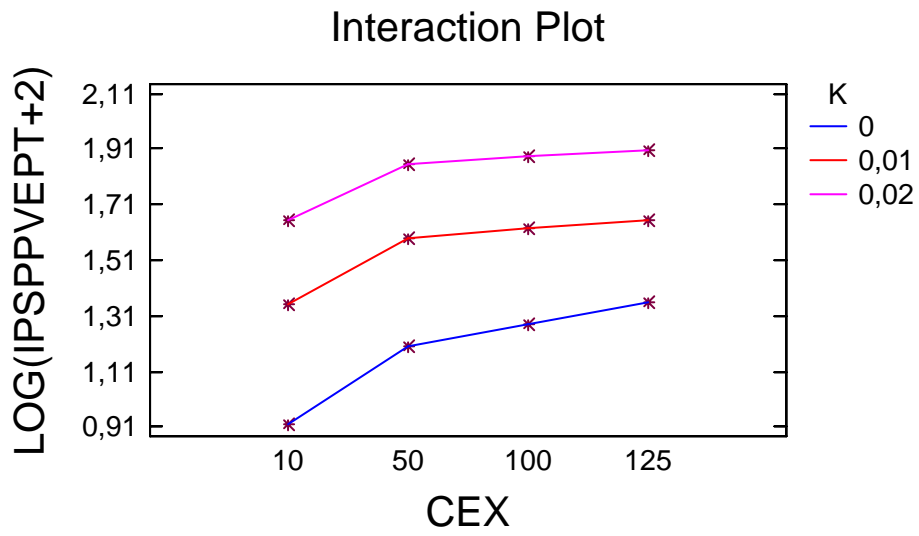


Figura IV-38. Gráfico del efecto de la interacción entre el factor CEX y el factor K del bloque Adelanto-Predictiva SMAGA-2-P3_20.

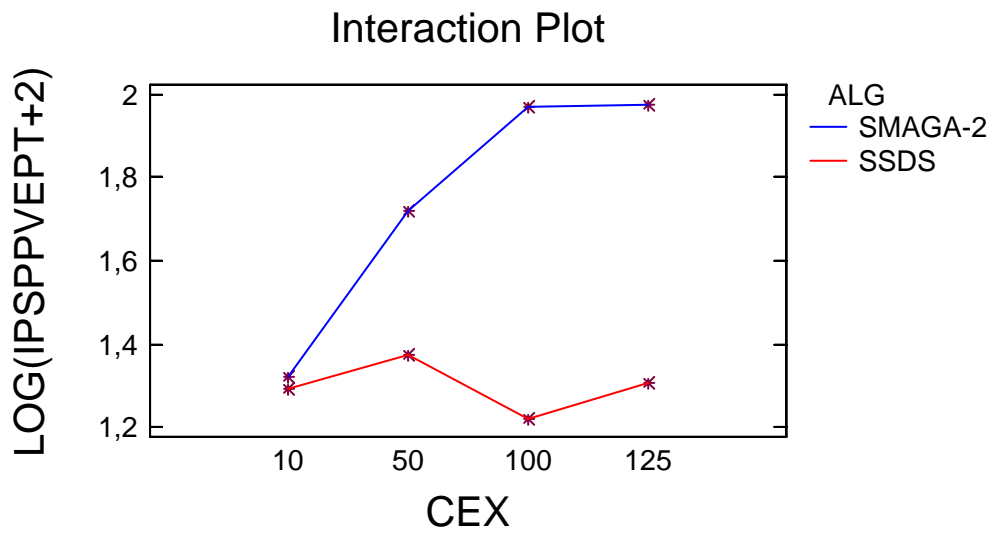


Figura IV-39. Gráfico del efecto de la interacción entre el factor CEX y el factor ALG del bloque Adelanto-Predictiva SMAGA-2-P3_20.

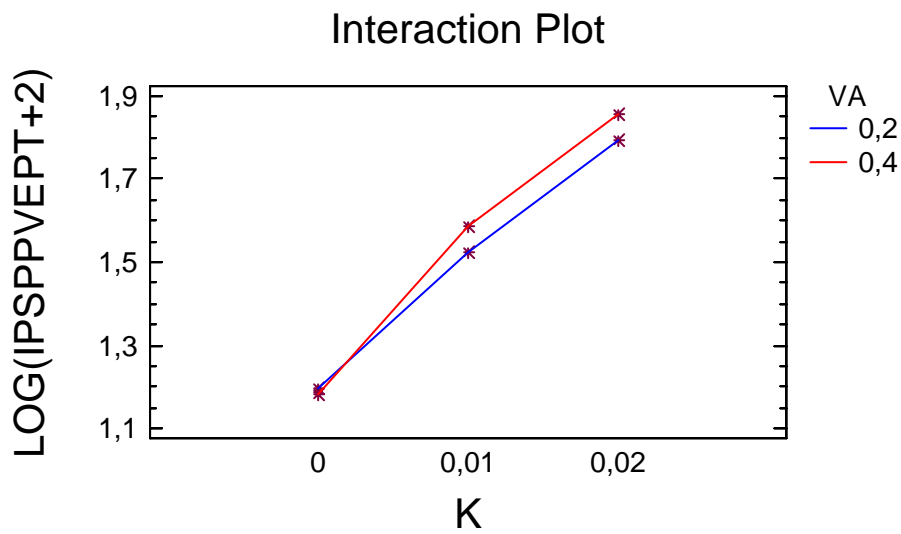


Figura IV-40. Gráfico del efecto de la interacción entre el factor K y el factor VA del bloque Adelanto-Predictiva SMAGA-2-P3_20.

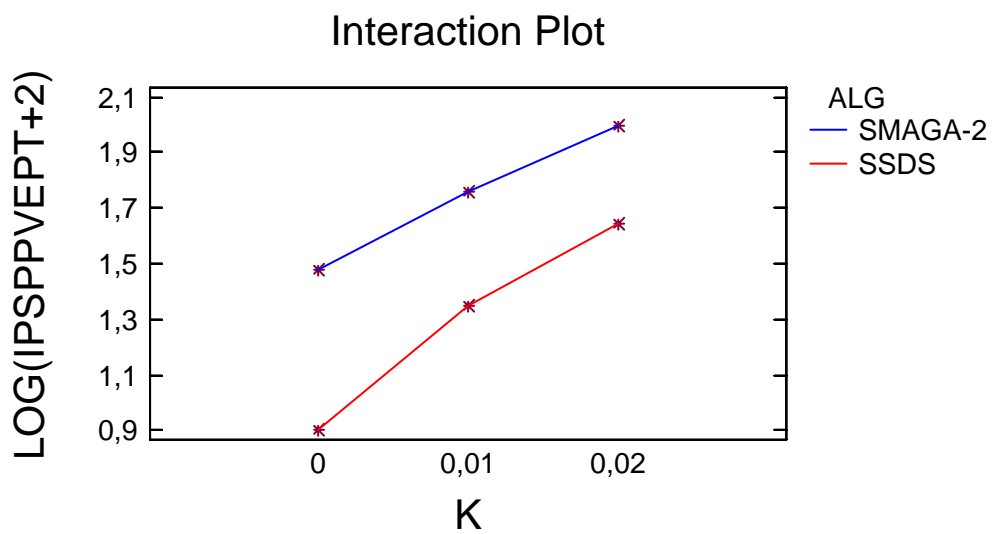


Figura IV-41. Gráfico del efecto de la interacción entre el factor K y el factor ALG del bloque Adelanto-Predictiva SMAGA-2-P3_20.

A-IV.1.6.2 Análisis del Bloque P3_50

En este bloque experimental se ha necesitado realizar una transformación de la variable dependiente debido a que se ha observado la existencia de alguna relación no lineal entre los factores. La transformación que se ha empleado ha sido el cálculo de logaritmo neperiano (LOG en la aplicación Statgraphics) de la variable a la cual se le ha sumado dos unidades para evitar valores negativos que impidieran el cálculo del logaritmo. En todo caso, la significación de los factores y los test realizados son tan válidos como si se hubiesen realizado sin transformación.

En relación con la normalidad del experimento se ha obtenido un Chi-Square = 8,58367 con 4 d.f., y un P-Value = 0,0723911, lo que descarta la posibilidad de que la variable analizada no proceda de una distribución normal. Así mismo se puede observa en la Figura IV-42 como la variable se distribuye muy próxima a la línea continua diagonal, lo que implica su proximidad a la distribución Normal.

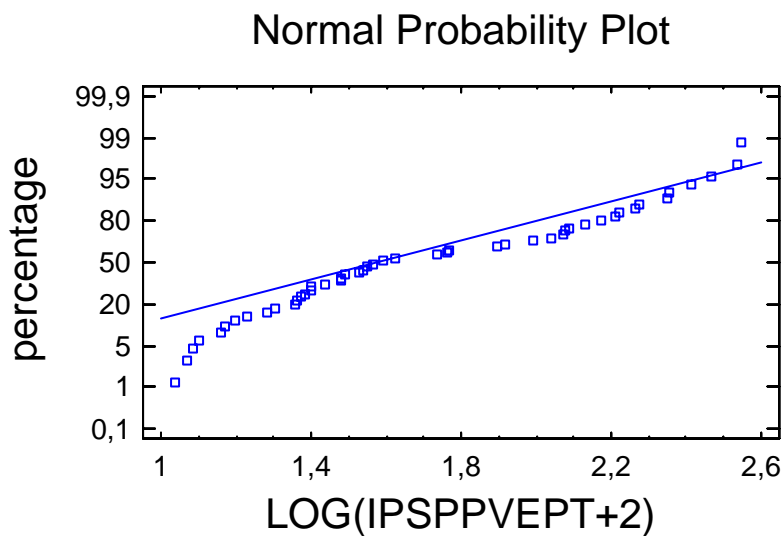


Figura IV-42. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPSPVEPT del bloque Adelanto-Predictiva SMAGA-2-P3_50 en relación a las frecuencias relativas acumuladas.

Por lo que respecta a la homogeneidad de las varianzas se ha realizado el test de Levene para la variable dependiente con respecto a las variables independientes obteniéndose los siguientes valores:

- IPSPVEPT respecto a CEX: test de Levene: 1,53992 P-Value = 0,21752

- IPSPVEPT respecto a K: test de Levene: 0,420167 P-Value = 0,659487
- IPSPVEPT respecto a VA: test de Levene: 0,0844773 P-Value = 0,772626
- IPSPVEPT respecto a ALG: test de Levene: 4,00912 P-Value = 0,0511707

Por lo que se descarta que las varianzas no sean homogéneas.

Por lo que se refiere a la linealidad se empleará el “gráfico de residuos (X_t, \bar{u}_t) y (\hat{Y}_t, \bar{u}_t)”, que permite disponer los residuos estudentizadosⁱⁱ (\bar{u}_t), en el eje de ordenadas, frente a las variables exógenas (X_t), y endógena (\hat{Y}_t).

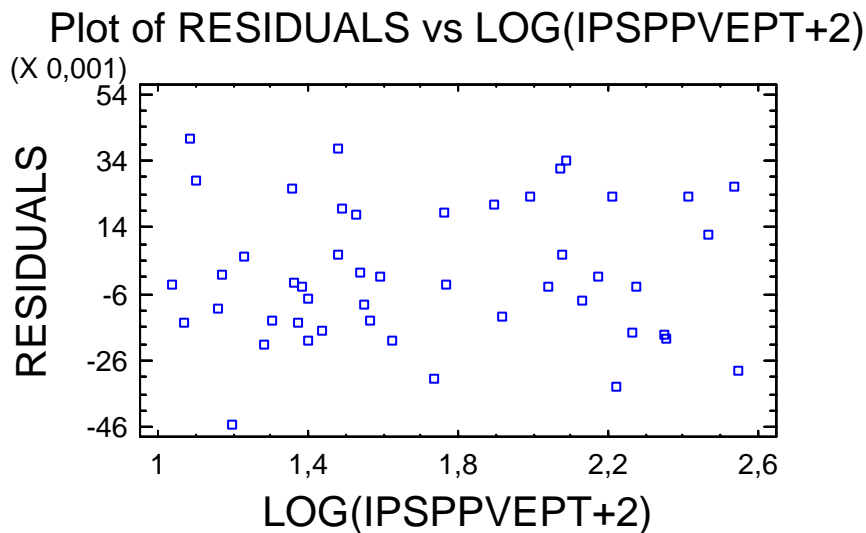


Figura IV-43. Gráfico de Residuos frente a los niveles de la Variable IPSPVEPT del bloque Adelanto-Predictiva SMAGA-2-P3_50.

ⁱⁱ Consiste en el cálculo de la siguiente expresión para cada residuo: (residuo-media de los residuos)/desviación estándar de los residuos.

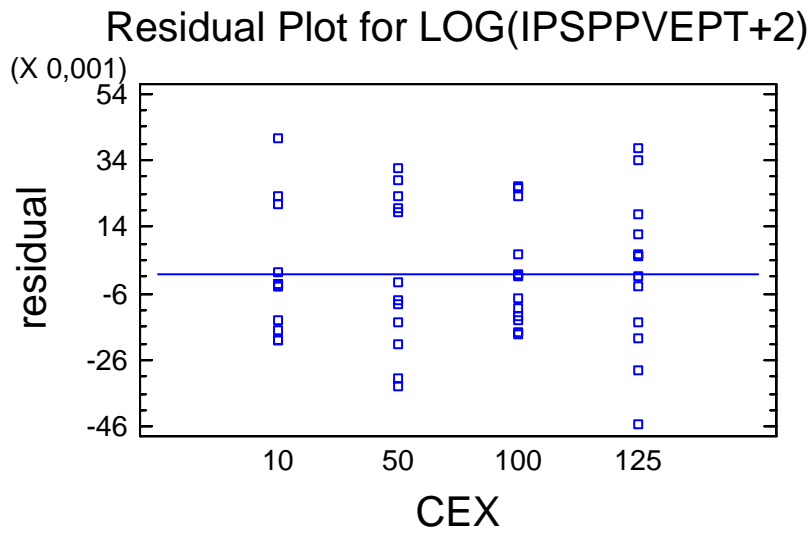


Figura IV-44. Gráfico de Residuos frente a los niveles del factor CEX del bloque Adelanto-Predictiva SMAGA-2-P3_50.

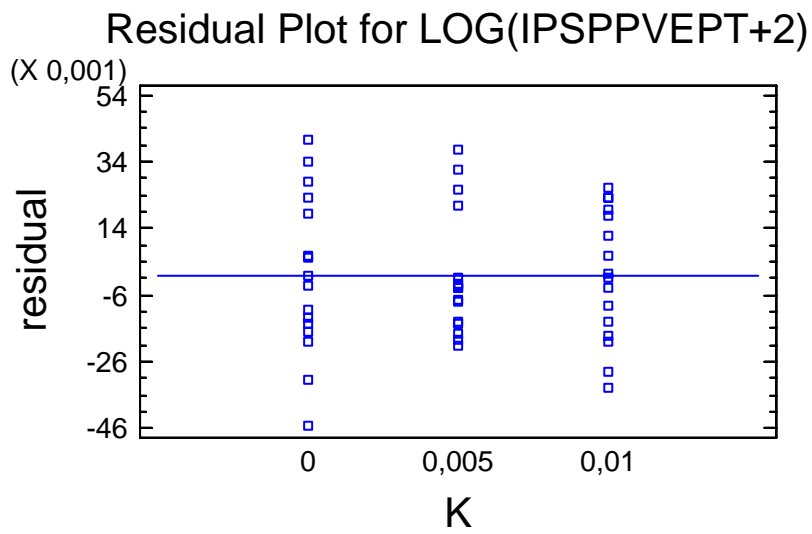


Figura IV-45. Gráfico de Residuos frente a los niveles del factor k del bloque Adelanto-Predictiva SMAGA-2-P3_50.

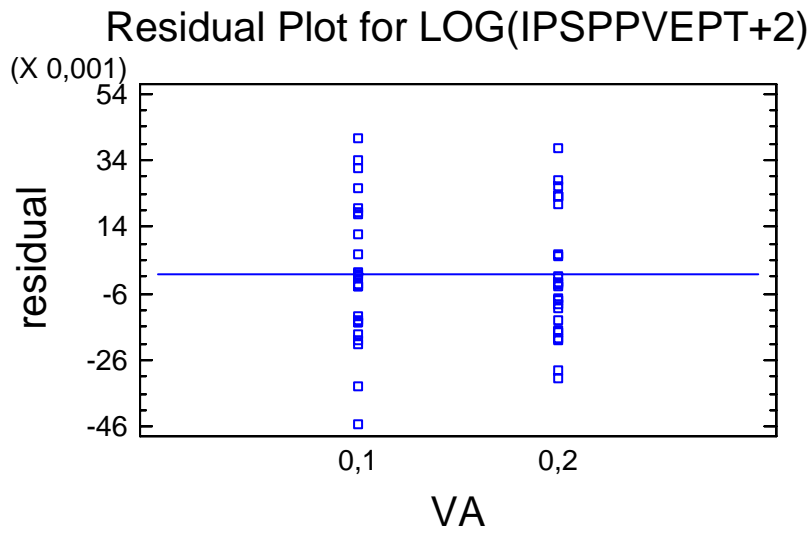


Figura IV-46. Gráfico de Residuos frente a los niveles del factor VA del bloque Adelanto-Predictiva SMAGA-2-P3_50.

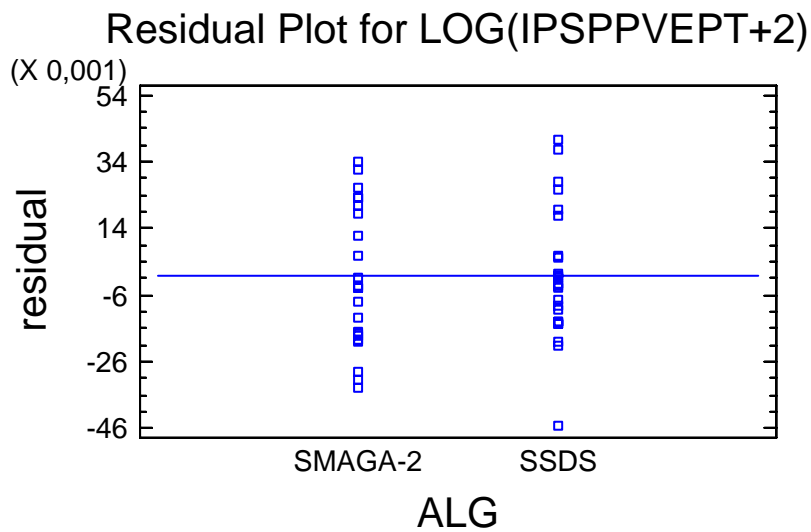


Figura IV-47. Gráfico de Residuos frente a los niveles del factor ALG del bloque Adelanto-Predictiva SMAGA-2-P3_50.

Como se observa en las figuras correspondientes a los residuos no se observa una falta de linealidad significativa en los mismos. Por lo que se descarta la ausencia de linealidad de la variable transformada.

La correlación se detecta mediante el gráfico de residuos en relación al orden de ejecución de los tratamientos. Como se observa en la Figura IV-48 no se puede observar ninguna tendencia significativa en la distribución de los mismos por lo que se descarta la existencia de correlación.

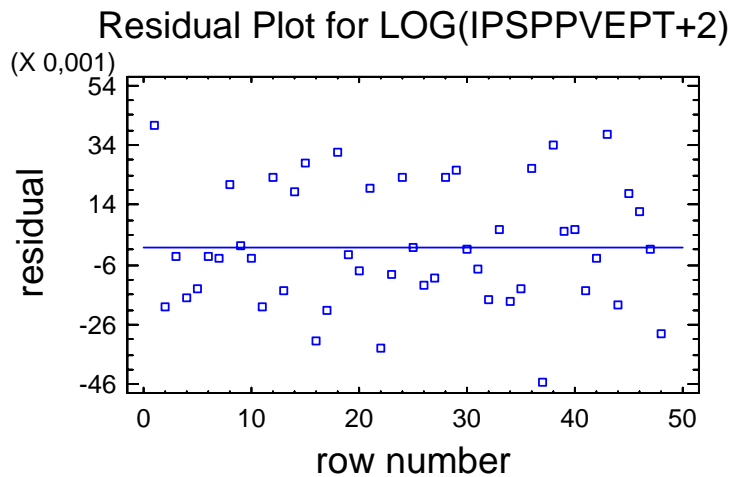


Figura IV-48. Residuos frente al orden de ejecución de los tratamientos del bloque Adelanto-Predictiva SMAGA-2-P3_50.

Una vez verificado el cumplimiento de las condiciones para el análisis ANOVA se obtiene y analiza la misma.

Analysis of Variance for LOG(IPSPVEPT+2) - Type III Sums of Squares

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
MAIN EFFECTS					
A:CEX	0,56819	3	0,189397	222,47	0,0000
B:K	1,86082	2	0,930409	1092,89	0,0000
C:VA	0,0583759	1	0,0583759	68,57	0,0000
D:ALG	6,56752	1	6,56752	7714,42	0,0000
INTERACTIONS					
AB	0,0608955	6	0,0101493	11,92	0,0000
AC	0,00069166	3	0,000230553	0,27	0,8458
AD	0,346795	3	0,115598	135,79	0,0000
BC	0,0230045	2	0,0115023	13,51	0,0001
BD	0,0354809	2	0,0177404	20,84	0,0000
CD	0,00459471	1	0,00459471	5,40	0,0294
RESIDUAL	0,0195806	23	0,00085133		
TOTAL (CORRECTED)	9,54595	47			

All F-ratios are based on the residual mean square error.

Tabla IV-20. ANOVA correspondiente IPSPVEPT del bloque Adelanto-Predictiva SMAGA-2-P3_50.

Como se puede ver en la Tabla IV-20 disponemos de 48 datos y tenemos 23 grados de libertad residuales. El contraste de hipótesis sobre la potencia (P-value) para un $\alpha=0,05$ implica que son significativos todos los efectos principales de los factores, y además lo son las interacciones AB, AD, BC, BD y CD.

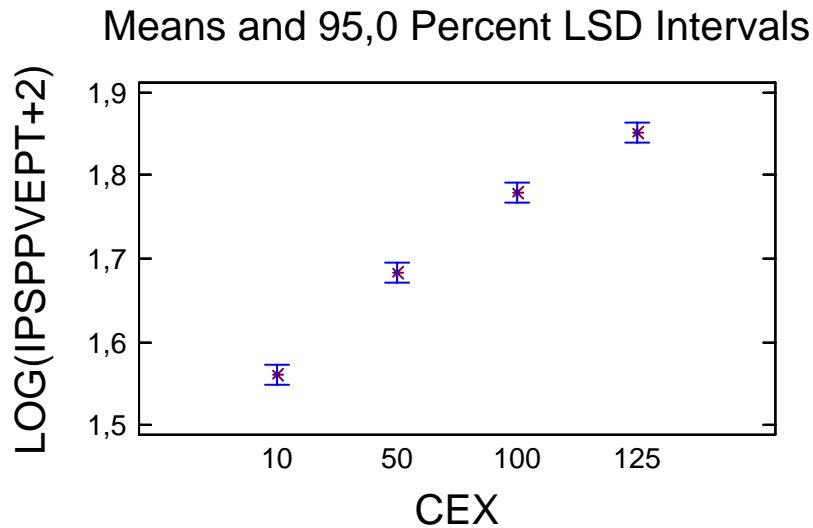


Figura IV-49. Gráfico del efecto principal del factor CEX del bloque Adelanto-Predictiva SMAGA-2-P3_50.

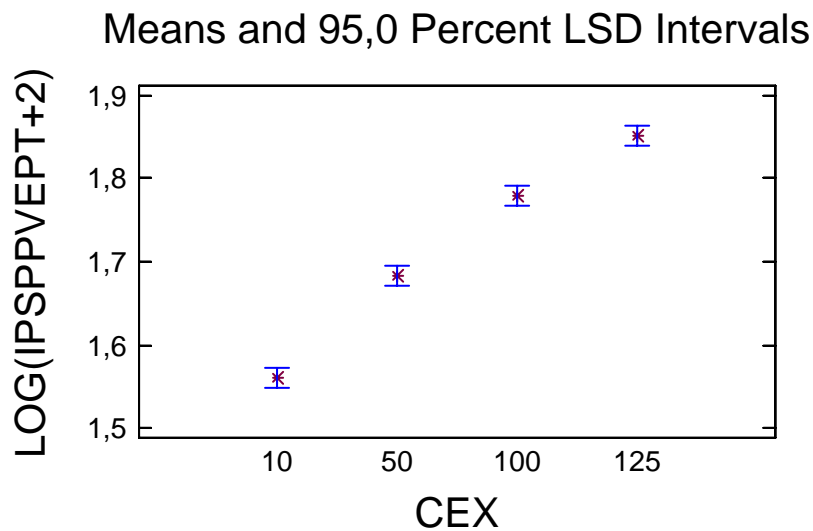


Figura IV-50. Gráfico del efecto principal del factor K del bloque Adelanto-Predictiva SMAGA-2-P3_50.

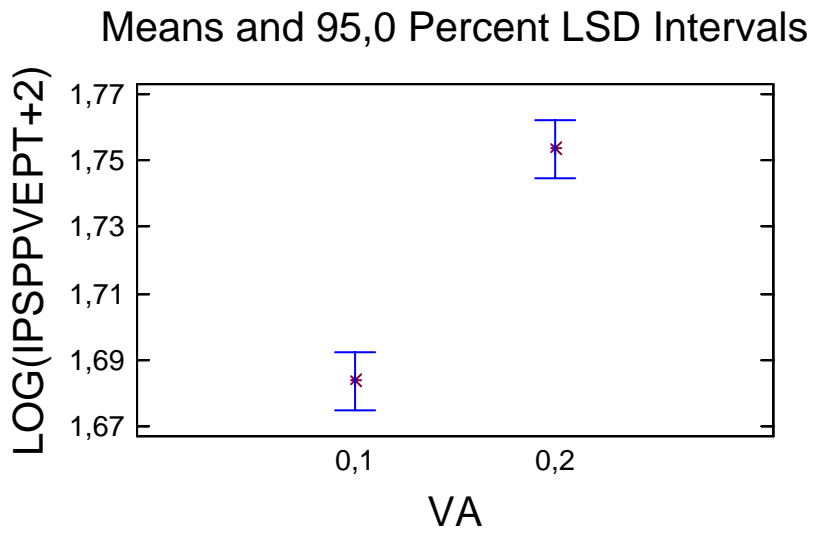


Figura IV-51. Gráfico del efecto principal del factor VA del bloque Adelanto-Predictiva SMAGA-2-P3_50.

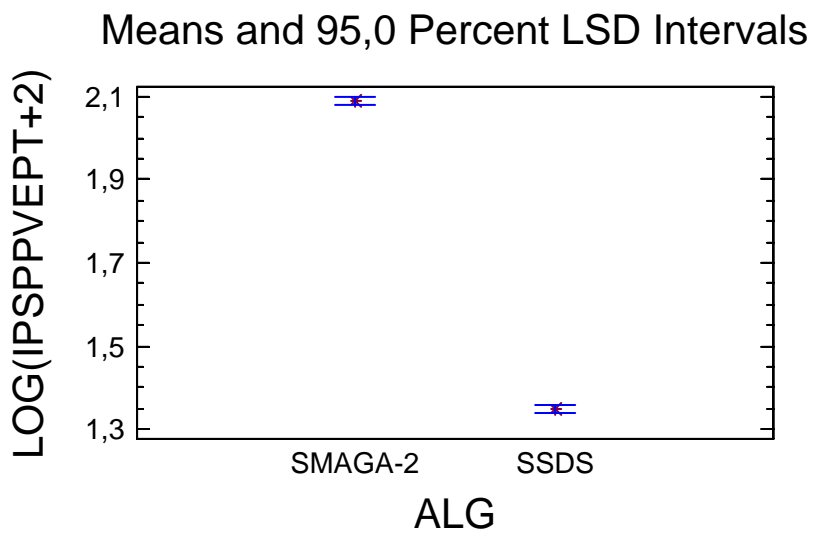


Figura IV-52. Gráfico del efecto principal del factor ALG del bloque Adelanto-Predictiva SMAGA-2-P3_50.

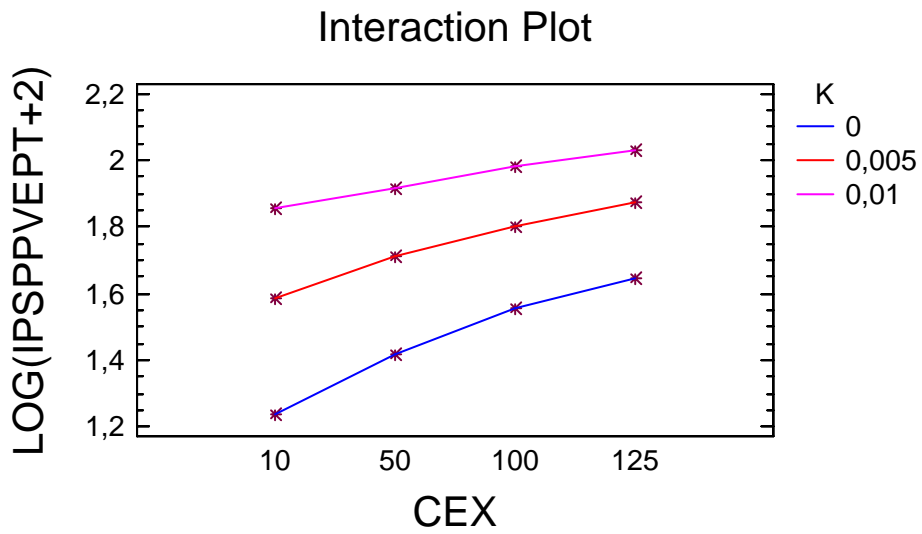


Figura IV-53. Gráfico del efecto de la interacción entre el factor CEX y el factor K del bloque Adelanto-Predictiva SMAGA-2-P3_50.

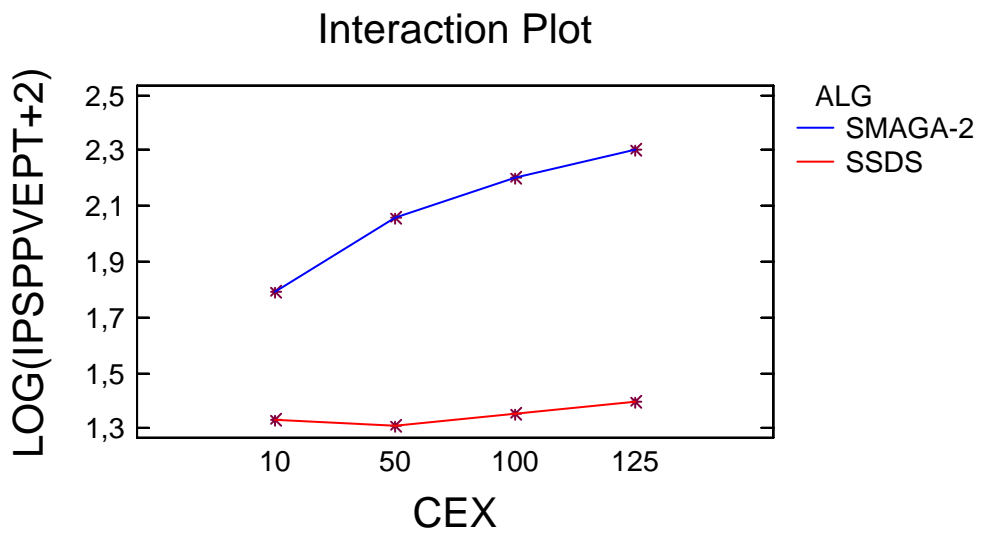


Figura IV-54. Gráfico del efecto de la interacción entre el factor CEX y el factor ALG del bloque Adelanto-Predictiva SMAGA-2-P3_50.

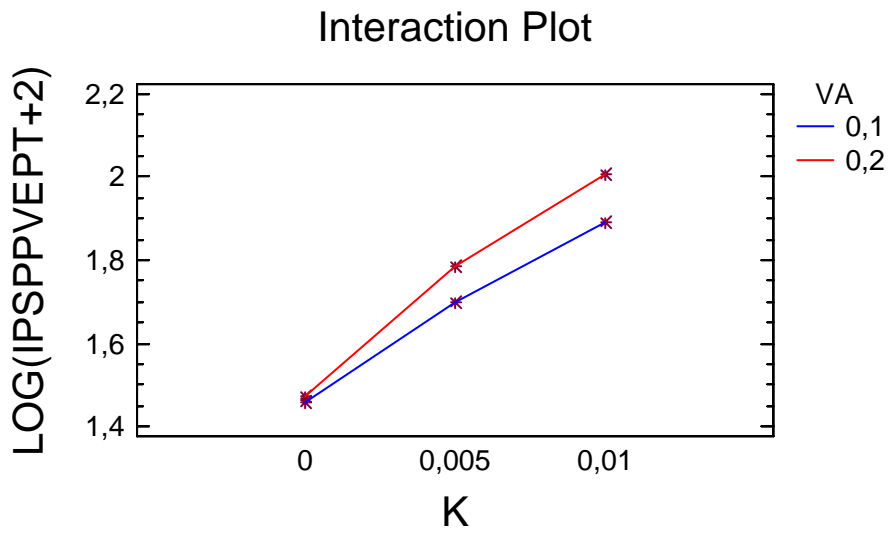


Figura IV-55. Gráfico del efecto de la interacción entre el factor K y el factor VA del bloque Adelanto-Predictiva SMAGA-2-P3_50.

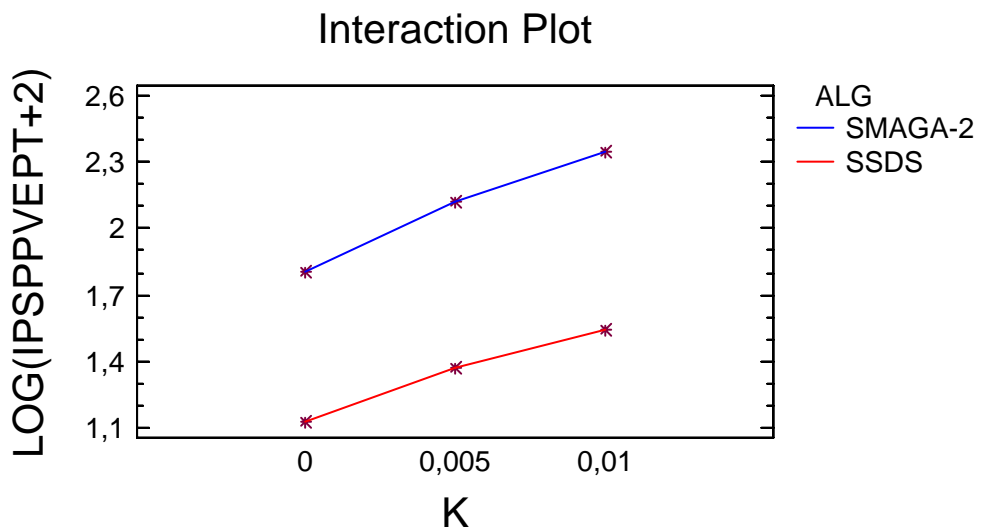


Figura IV-56. Gráfico del efecto de la interacción entre el factor K y el factor ALG del bloque Adelanto-Predictiva SMAGA-2-P3_50.

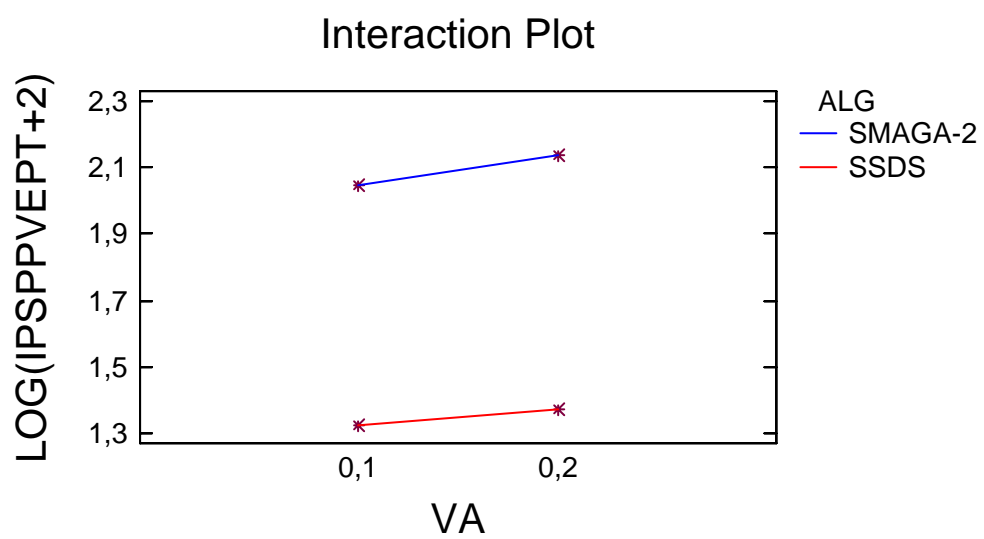


Figura IV-57. Gráfico del efecto de la interacción entre el factor VA y el factor ALG del bloque Adelanto-Predictiva SMAGA-2-P3_50.

A-IV.2 Resultados y Análisis de los mismo en situaciones de Retrasos en las Operaciones

En este apartado se proporcionan los datos obtenidos durante la simulación de reprogramación de un programa predictivo sometido a eventos que suponen un retraso en la fecha de finalización de las operaciones. Los resultados que se muestran son el promedio de adelantos grandes, medianos y pequeños correspondientes a los tipos de eventos BAT_AS_10, BAT_AS_50 y BAT_AS_80. En concreto los datos se dividen en dos apartados, en el primero se muestran los datos obtenidos partiendo de un algoritmo predictivo obtenido con SMAGA-2, y en el siguiente apartado los obtenidos con un algoritmo SMAGA-2 cuya secuencia ha sido modificada intercambiando 2 elementos de la misma. En ambos casos se muestran los datos obtenidos para la combinación de los 4 factores considerados (CEX, K, VA y ALG). De igual manera, en ambos casos, los algoritmos considerados son 5, RS, SSDS, AOR, SMAGA-2s y SMAGA-2. Dado que de dichos algoritmos sólo SSDS y SMAGA-2 son sensibles al valor de K (empeorando cuanto más vale K), y que del os restantes sólo SMAGA-2s es sensible al valor de VA, se debe tener en cuenta que si un algoritmo no sensible al valor de K es el mejor cuando K vale 0, ni el algoritmo SSDS ni SMAGA-2 podrá superarlo cuando K tome valores superiores a 0. Por este motivo en algunos casos no se analizan, o se hace parcialmente, los resultados con valor de K superiores a 0.

También se realiza el análisis estadístico sólo de los datos obtenidos con el programa predictivo establecido con algoritmo SMAGA-2. En concreto el análisis se divide en 4 bloques, en función de la configuración del taller y número de trabajos: Bloque P13_20, P13_50, P3_20 y P3_50.

En primer lugar se proporcionan los datos de los bloques P13_20 y P13_50, y el análisis estadístico correspondiente, y posteriormente se muestran los datos de los bloques P3_20 y P3_50 y se realiza en estudio estadístico correspondiente.

A-IV.2.1 Resultados para la Configuración P13 con Programación Predictiva basada en SMAGA-2

Conjunto Experimental	K=0 y VA=0,2					K=0,01 y VA=0,2				
	RS	SSDS	AOR	SMAGA-2s	SMAGA-2	RS	SSDS	AOR	SMAGA-2s	SMAGA-2
SSD10_P13_20	1,301	1,155	1,305	1,146	1,201	1,301	1,345	1,305	1,146	
SSD50_P13_20	1,168	0,957	1,097	1,122	1,409	1,168	1,234	1,097	1,122	
SSD100_P13_20	0,940	0,850	0,863	1,054	1,622	0,940	0,951	0,863	1,054	
SSD125_P13_20	0,864	0,728	0,796	1,110	1,821	0,864	0,964	0,796	1,110	

Tabla IV-21. Valor de IPSPVEPT a partir de SMAGA-2 con retraso para 20 trabajos, y un valor de VA de 0,2.

Conjunto Experimental	K=0 y VA=0,4					K=0,01 y VA=0,4				
	RS	SSDS	AOR	SMAGA-2s	SMAGA-2	RS	SSDS	AOR	SMAGA-2s	SMAGA-2
SSD10_P13_20	1,301	1,155	1,305	1,064	1,201	1,301	1,389	1,305	1,064	
SSD50_P13_20	1,168	0,957	1,097	0,973	1,409	1,168	1,255	1,097	0,973	
SSD100_P13_20	0,940	0,850	0,863	0,901	1,622	0,940	0,987	0,863	0,901	
SSD125_P13_20	0,864	0,728	0,796	0,863	1,821	0,864	0,974	0,796	0,863	

Tabla IV-22. Valor de IPSPVEPT a partir de SMAGA-2 con retraso para 20 trabajos, y un valor de VA de 0,4.

Conjunto Experimental	K=0 y VA=0,1					K=0,005 y VA=0,1				
	RS	SSDS	AOR	SMAGA-2s	SMAGA-2	RS	SSDS	AOR	SMAGA-2s	SMAGA-2
SSD10_P13_50	0,656	0,627	0,680	0,687	0,795	0,656	0,704	0,680	0,687	
SSD50_P13_50	0,507	0,480	0,477	1,273	1,752	0,507	0,568	0,477	1,273	
SSD100_P13_50	0,392	0,359	0,394	2,003	2,727	0,392	0,423	0,394	2,003	
SSD125_P13_50	0,333	0,317	0,318	2,220	3,103	0,333	0,374	0,318	2,220	

Tabla IV-23. Valor de IPSPVEPT a partir de SMAGA-2 con retraso para 50 trabajos, y un valor de VA de 0,1.

Conjunto Experimental	K=0 y VA=0,2					K=0,005 y VA=0,2				
	RS	SSDS	AOR	SMAGA-2s	SMAGA-2	RS	SSDS	AOR	SMAGA-2s	SMAGA-2
SSD10_P13_50	0,656	0,627	0,680	0,606	0,795	0,656	0,747	0,680	0,606	
SSD50_P13_50	0,507	0,480	0,477	0,963	1,752	0,507	0,672	0,477	0,963	
SSD100_P13_50	0,392	0,359	0,394	1,384	2,727	0,392	0,538	0,394	1,384	
SSD125_P13_50	0,333	0,317	0,318	1,563	3,103	0,333	0,451	0,318	1,563	

Tabla IV-24. Valor de IPSPVEPT a partir de SMAGA-2 con retraso para 50 trabajos, y un valor de VA de 0,2.

A-IV.2.2 Resultados para la Configuración P13 con Programación Predictiva basada en SMAGA-2 Modificada

Conjunto Experimental	K=0 y VA=0,2					K=0,01 y VA=0,2				
	RS	SSDS	AOR	SMAGA-2s	SMAGA-2	RS	SSDS	AOR	SMAGA-2s	SMAGA-2
SSD10_P13_20	0,924	0,793	0,910	0,551	-0,465	0,924	1,035	0,910	0,551	0,473
SSD50_P13_20	0,633	0,525	0,644	0,084	-1,591	0,633	0,798	0,644	0,084	-0,523
SSD100_P13_20	0,445	0,422	0,455	-0,481	-2,909	0,445	0,644	0,455	-0,481	-1,670
SSD125_P13_20	0,374	0,323	0,353	-0,807	-3,325	0,374	0,521	0,353	-0,807	-2,191
	K=0,02 y VA=0,2									
SSD10_P13_20	0,924	1,285	0,910	0,551	1,512					
SSD50_P13_20	0,633	1,045	0,644	0,084	0,564					
SSD100_P13_20	0,445	0,916	0,455	-0,481	-0,426					
SSD125_P13_20	0,374	0,699	0,353	-0,807	-1,040					

Tabla IV-25. Valor de IPSPVEPT a partir de SMAGA-2 Modificado con retraso para 20 trabajos, y un valor de VA de 0,2.

Conjunto Experimental	K=0 y VA=0,4					K=0,01 y VA=0,4				
	RS	SSDS	AOR	SMAGA-2s	SMAGA-2	RS	SSDS	AOR	SMAGA-2s	SMAGA-2
SSD10_P13_20	0,950	0,765	0,893	0,695	-0,532	0,950	1,053	0,893	0,695	0,632
SSD50_P13_20	0,633	0,539	0,624	0,413	-1,540	0,633	0,783	0,624	0,413	-0,310
SSD100_P13_20	0,452	0,442	0,433	0,205	-2,906	0,452	0,678	0,433	0,205	-1,378
SSD125_P13_20	0,363	0,319	0,369	-0,042	-3,470	0,363	0,512	0,369	-0,042	-2,020
K=0,02 y VA=0,4										
SSD10_P13_20	0,950	1,280	0,893	0,695	1,911					
SSD50_P13_20	0,633	1,062	0,624	0,413	0,985					
SSD100_P13_20	0,452	0,962	0,433	0,205	0,165					
SSD125_P13_20	0,363	0,767	0,369	-0,042	-0,523					

Tabla IV-26. Valor de IPSPPVEPT a partir de SMAGA-2 Modificado con retraso para 20 trabajos, y un valor de VA de 0,4.

Conjunto Experimental	K=0 y VA=0,1					K=0,005 y VA=0,1				
	RS	SSDS	AOR	SMAGA-2s	SMAGA-2	RS	SSDS	AOR	SMAGA-2s	SMAGA-2
SSD10_P13_50	0,316	0,331	0,346	-1,522	-2,366	0,316	0,457	0,346	-1,522	-0,921
SSD50_P13_50	0,224	0,222	0,226	-3,108	-4,606	0,224	0,316	0,226	-3,108	-2,840
SSD100_P13_50	0,159	0,143	0,155	-4,543	-6,138	0,159	0,221	0,155	-4,543	-4,311
SSD125_P13_50	0,153	0,133	0,143	-4,304	-6,154	0,153	0,203	0,143	-4,304	-4,155
K=0,01 y VA=0,1										
SSD10_P13_50	0,316	0,542	0,346	-1,522	0,746					
SSD50_P13_50	0,224	0,404	0,226	-3,108	-1,171					
SSD100_P13_50	0,159	0,323	0,155	-4,543	-2,367					
SSD125_P13_50	0,153	0,254	0,143	-4,304	-2,201					

Tabla IV-27. Valor de IPSPPVEPT a partir de SMAGA-2 Modificado con retraso para 50 trabajos, y un valor de VA de 0,1.

Conjunto Experimental	K=0 y VA=0,2					K=0,005 y VA=0,2				
	RS	SSDS	AOR	SMAGA-2s	SMAGA-2	RS	SSDS	AOR	SMAGA-2s	SMAGA-2
SSD10_P13_50	0,361	0,321	0,349	-0,841	-2,424	0,361	0,567	0,349	-0,841	-0,068
SSD50_P13_50	0,237	0,221	0,243	-2,068	-4,534	0,237	0,394	0,243	-2,068	-1,860
SSD100_P13_50	0,159	0,149	0,166	-2,923	-6,162	0,159	0,302	0,166	-2,923	-3,127
SSD125_P13_50	0,144	0,135	0,147	-2,861	-6,028	0,144	0,248	0,147	-2,861	-2,962
	K=0,01 y VA=0,2									
SSD10_P13_50	0,361	0,780	0,349	-0,841	2,331					
SSD50_P13_50	0,237	0,601	0,243	-2,068	0,583					
SSD100_P13_50	0,159	0,376	0,166	-2,923	-0,288					
SSD125_P13_50	0,144	0,395	0,147	-2,861	-0,008					

Tabla IV-28. Valor de IPSPVEPT a partir de SMAGA-2 Modificado con retraso para 50 trabajos, y un valor de VA de 0,2.

A-IV.2.3 Análisis Estadístico para la Configuración P13 con Programación Predictiva basada en SMAGA-2

A-IV.2.3.1 Análisis del Bloque P13_20

En este bloque se ha tenido que dividir el análisis en dos partes. En una primera parte se ha estudiado que ocurre cuando el valor de K es 0, y en una segunda fase lo que ocurre cuando el valor de K superior a 0. Nuevamente el motivo de la división ha sido el de trabajar con datos con variable homogéneas. En el caso de K superior a 0 sólo se ha estudiado el valor 0,01 ya que se ha observado como el tipo de algoritmo que depende del valor de K se ve superado siempre por un algoritmo independiente de este valor cuando K es 0,01, y que por lo tanto responderá peor y será nuevamente superado por los segundas cuando K tomo el valor 0,02.

A-IV.2.3.1.1. Análisis del Bloque P13_20 (Valor de K=0)

En relación con la normalidad del experimento se ha obtenido un Chi-Square = 5,84977 con 4 d.f. y un P-Value = 0,210652, lo que descarta la posibilidad de que la variable analizada no proceda de una distribución normal. Así mismo se puede observa en la Figura IV-58 como la

variable se distribuye muy próxima a la línea continua diagonal, lo que implica su proximidad a la distribución Normal.

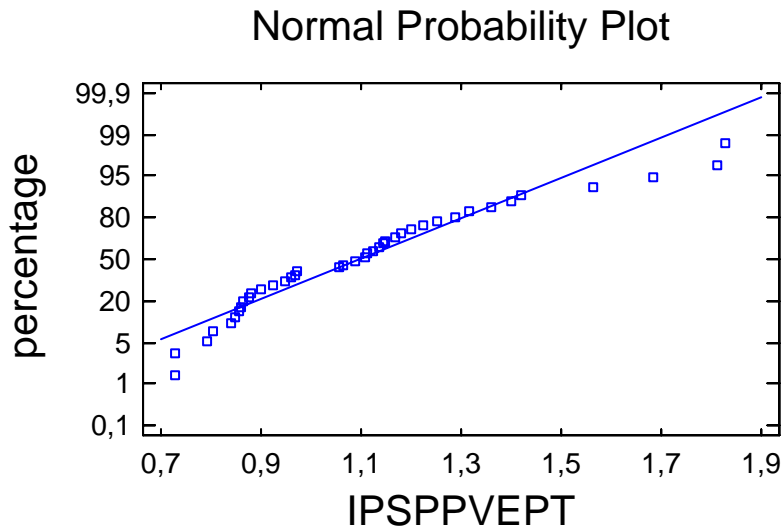


Figura IV-58. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPSPVEPT del bloque Retraso-Predictiva SMAGA-2-P13_20 ($K=0$) en relación a las frecuencias relativas acumuladas.

Por lo que respecta a la homogeneidad de las varianzas se ha realizado el test de Levene para la variable dependiente con respecto a las variables independientes obteniéndose los siguientes valores:

- IPSPVEPT respecto a CEX: test de Levene: 1,09575 P-Value = 0,363452
- IPSPVEPT respecto a VA: test de Levene: 0,117567 P-Value = 0,733578
- IPSPVEPT respecto a ALG: test de Levene: 2,45902 P-Value = 0,0634562

Por lo que se descarta que las varianzas no sean homogéneas.

Por lo que se refiere a la linealidad se empleará el “gráfico de residuos (X_t, \bar{u}_t) y (\hat{Y}_t, \bar{u}_t)”, que permite disponer los residuos, en el eje de ordenadas, frente a las variables exógenas (X_t), y endógena (\hat{Y}_t).

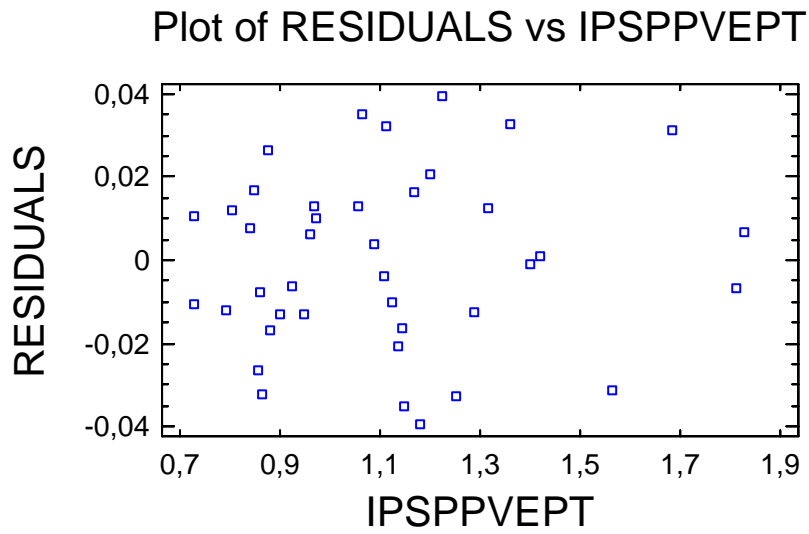


Figura IV-59. Gráfico de Residuos frente a los niveles de la Variable IPSPPVEPT del bloque Retraso-Predictiva SMAGA-2-P13_20 (K=0).

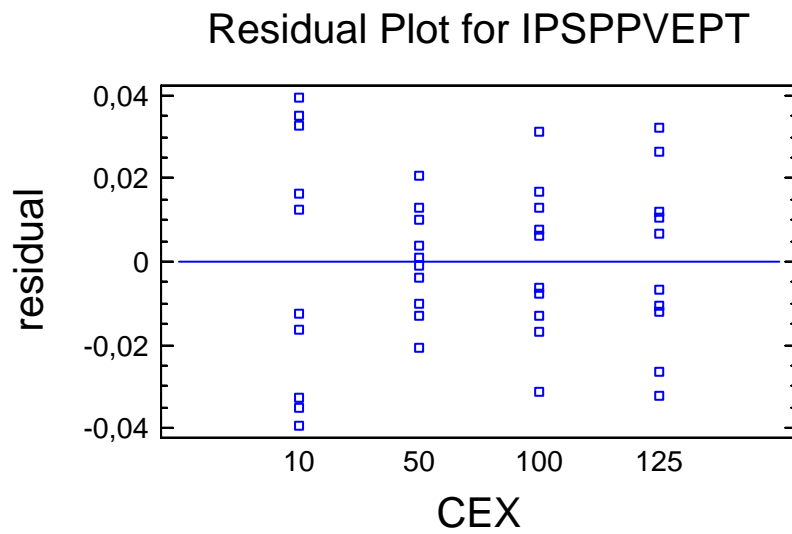


Figura IV-60. Gráfico de Residuos frente a los niveles del factor CEX del bloque Retraso-Predictiva SMAGA-2-P13_20 (K=0).

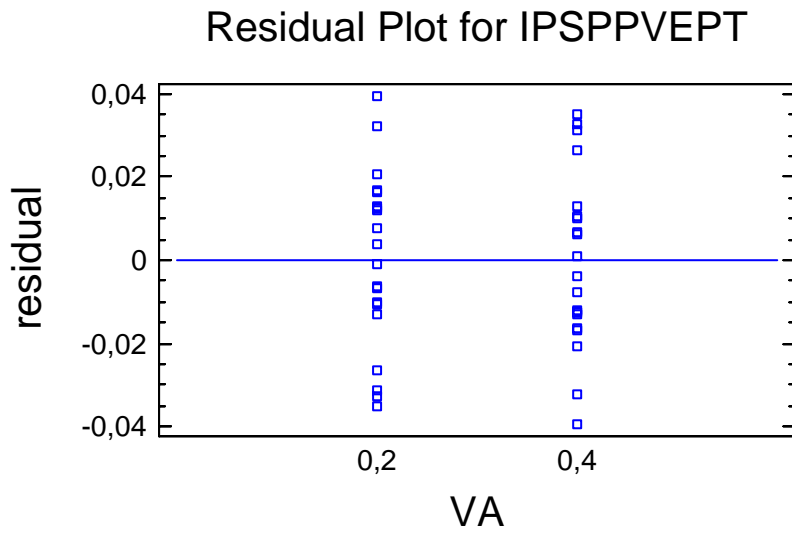


Figura IV-61. Gráfico de Residuos frente a los niveles del factor VA del bloque Retraso-Predictiva SMAGA-2-P13_20 (K=0).

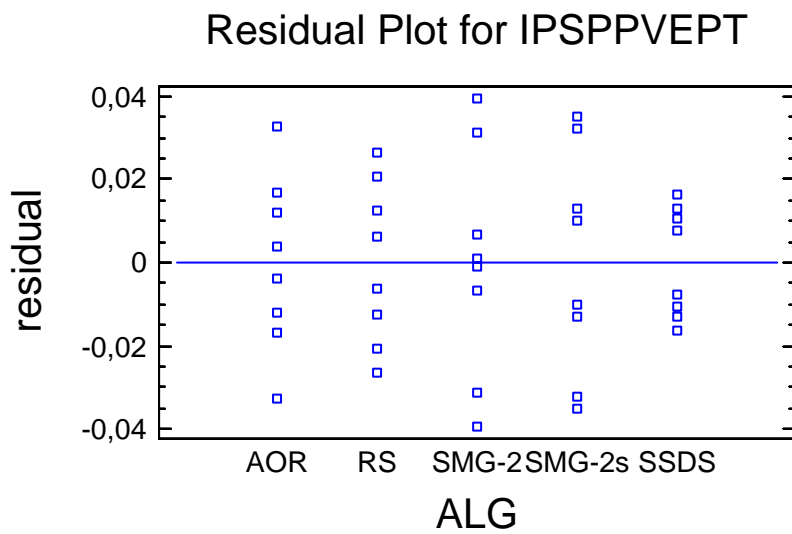


Figura IV-62. Gráfico de Residuos frente a los niveles del factor ALG del bloque Retraso-Predictiva SMAGA-2-P13_20 (K=0).

Como se observa en las figuras correspondientes a los residuos no se observa una falta de linealidad significativa en los mismos. Por lo que se descarta la ausencia de linealidad.

La correlación se detecta mediante el gráfico de residuos en relación al orden de ejecución de los tratamientos. Como se observa en la Figura IV-63 no se puede observar ninguna tendencia significativa en la distribución de los mismos por lo que se descarta la existencia de correlación.

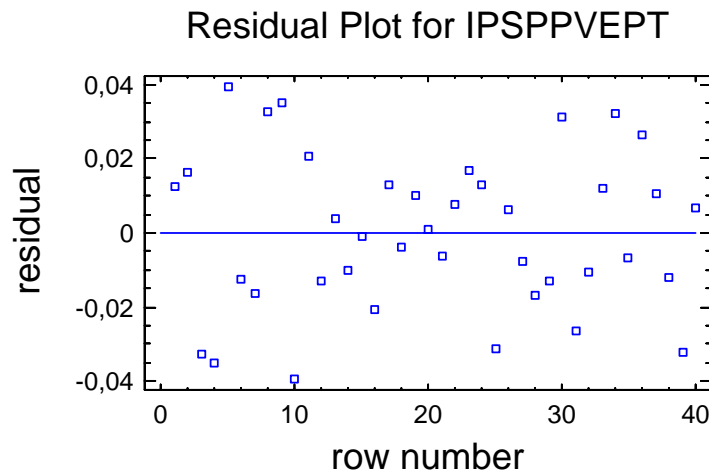


Figura IV-63. Residuos frente al orden de ejecución de los tratamientos del bloque Retraso-Predictiva SMAGA-2-P13_20 (K=0).

Una vez verificado el cumplimiento de las condiciones para el análisis ANOVA se obtiene y analiza la misma.

Analysis of Variance for IPSPVEPT - Type III Sums of Squares

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
MAIN EFFECTS					
A:CEX	0,199318	3	0,0664394	46,73	0,0000
B:VA	0,003816	1	0,003816	2,68	0,1273
C:ALG	1,72094	4	0,430236	302,58	0,0000
INTERACTIONS					
AB	0,00428665	3	0,00142888	1,00	0,4242
AC	1,0161	12	0,0846746	59,55	0,0000
BC	0,0506422	4	0,0126606	8,90	0,0014
RESIDUAL	0,0170626	12	0,00142188		
TOTAL (CORRECTED)	3,01217	39			

All F-ratios are based on the residual mean square error.

Tabla IV-29. ANOVA correspondiente IPSPVEPT del bloque Retraso-Predictiva SMAGA-2-P13_20 (K=0).

Como se puede ver en la Tabla IV-29 disponemos de 40 datos y tenemos 12 grados de libertad residuales. El contraste de hipótesis sobre la potencia (P-value) para un $\alpha=0,05$ implica que son significativos los efectos principales de los factores A y C, y además lo son las interacciones AC y BC.

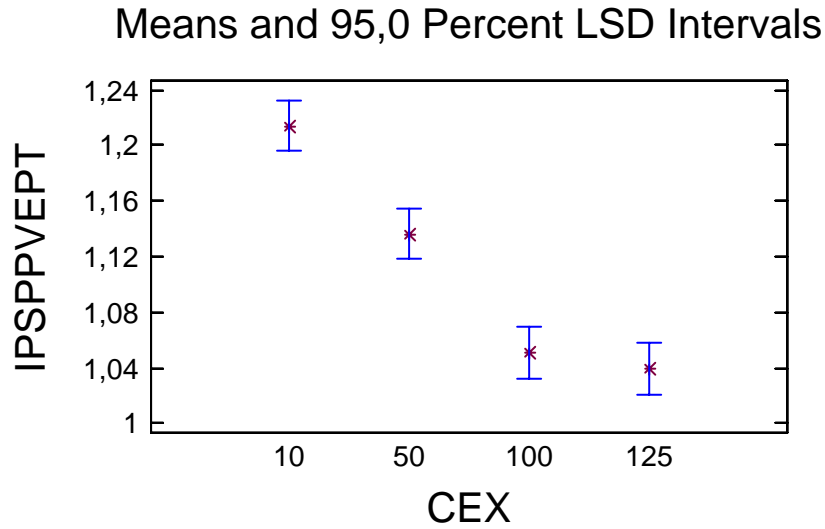


Figura IV-64. Gráfico del efecto principal del factor CEX del bloque Retraso-Predictiva SMAGA-2-P13_20 (K=0).

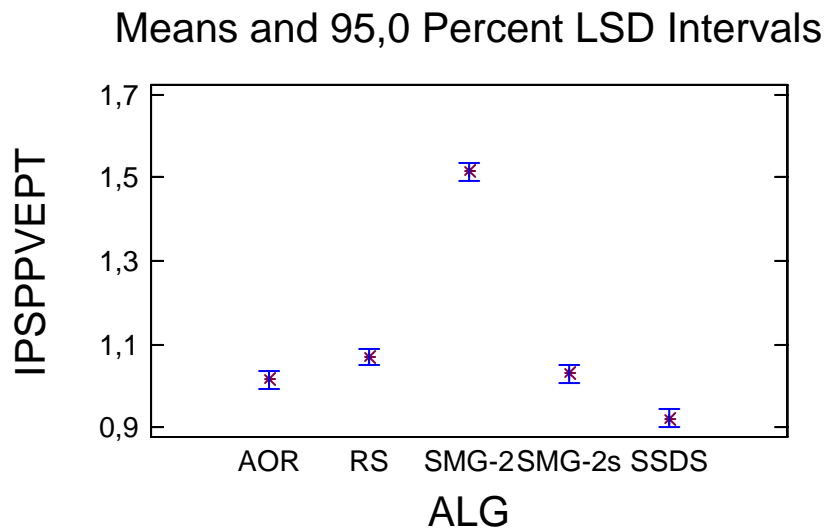


Figura IV-65. Gráfico del efecto principal del factor ALG del bloque Retraso-Predictiva SMAGA-2-P13_20 (K=0).

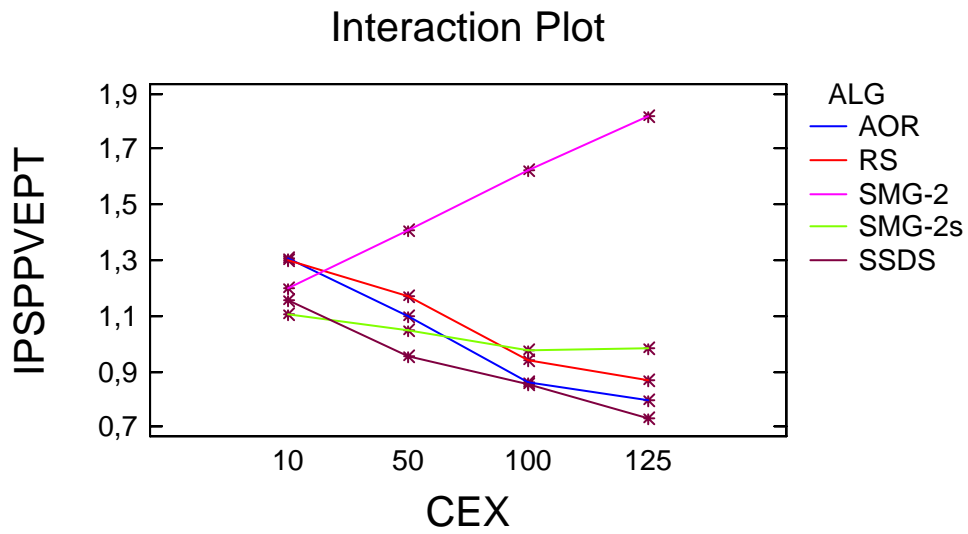


Figura IV-66. Gráfico del efecto de la interacción entre el factor CEX y el factor ALG del bloque Retraso-Predictiva SMAGA-2-P13_20 (K=0).

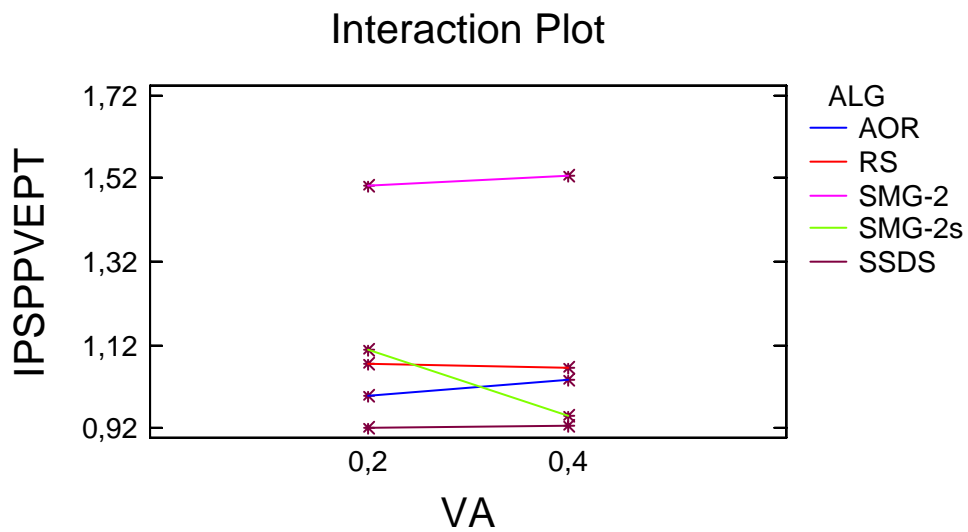


Figura IV-67. Gráfico del efecto de la interacción entre el factor VA y el factor ALG del bloque Retraso-Predictiva SMAGA-2-P13_20 (K=0).

A-IV.2.3.1.2. Análisis del Bloque P13_20 (Valor de K=0,01)

En este bloque experimental se ha necesitado realizar una transformación de la variable dependiente, y además en esta ocasión se no ha incluido en el análisis el algoritmo SMAGA-2. Ambas decisiones están orientadas a garantizar el cumplimiento de la condición de igualdad en la varianza. La transformación que se ha empleado ha sido el cálculo de la raíz cuadrada de la variable a la cual se le ha restado dos decimas. En todo caso, la significación de los factores y los test realizados son tan válidos como si se hubiesen realizado sin transformación. La no inclusión del algoritmo SMAGA-2 en el análisis es factible ya que ha demostrado no ser interesante para el caso de $K=0$, por lo que aún será menos interesante cuando el valor de K sea superior.

En relación con la normalidad del experimento se ha obtenido un Chi-Square = 4,75 con 4 d.f. y un P-Value = 0,313924, lo que descarta la posibilidad de que la variable analizada no proceda de una distribución normal. Así mismo se puede observa en la Figura IV-68 como la variable se distribuye muy próxima a la línea continua diagonal, lo que implica su proximidad a la distribución Normal.

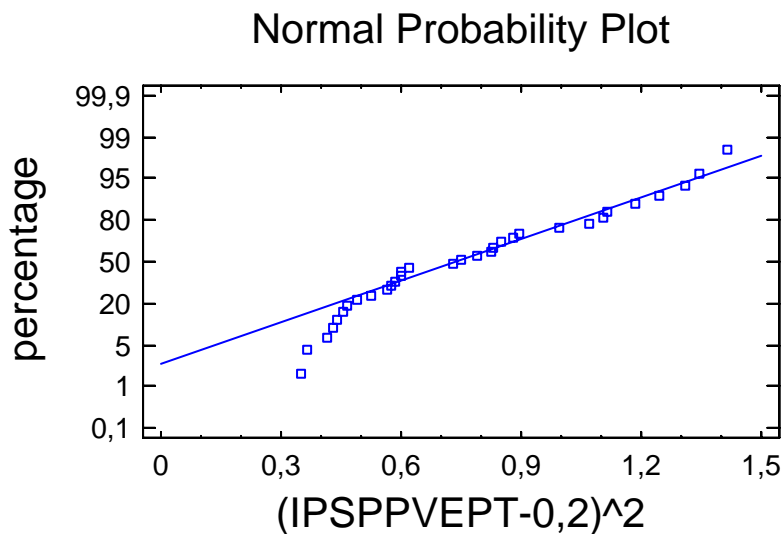


Figura IV-68. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPSPVEPT del bloque Retraso-Predictiva SMAGA-2-P13_20 ($K=0,01$) en relación a las frecuencias relativas acumuladas.

Por lo que respecta a la homogeneidad de las varianzas se ha realizado el test de Levene para la variable dependiente con respecto a las variables independientes obteniéndose los siguientes valores:

- IPSPVEPT respecto a CEX: test de Levene: 1,03092 P-Value = 0,393986
- IPSPVEPT respecto a VA: test de Levene: 0,0444746 P-Value = 0,834399
- IPSPVEPT respecto a ALG: test de Levene: 2,91589 P-Value = 0,0516274

Por lo que se descarta que las varianzas no sean homogéneas.

Por lo que se refiere a la linealidad se empleará el “gráfico de residuos (X_t, \bar{u}_t) y (\hat{Y}_t, \bar{u}_t) ”, que permite disponer los residuos, en el eje de ordenadas, frente a las variables exógenas (X_t) , y endógena (\hat{Y}_t) .

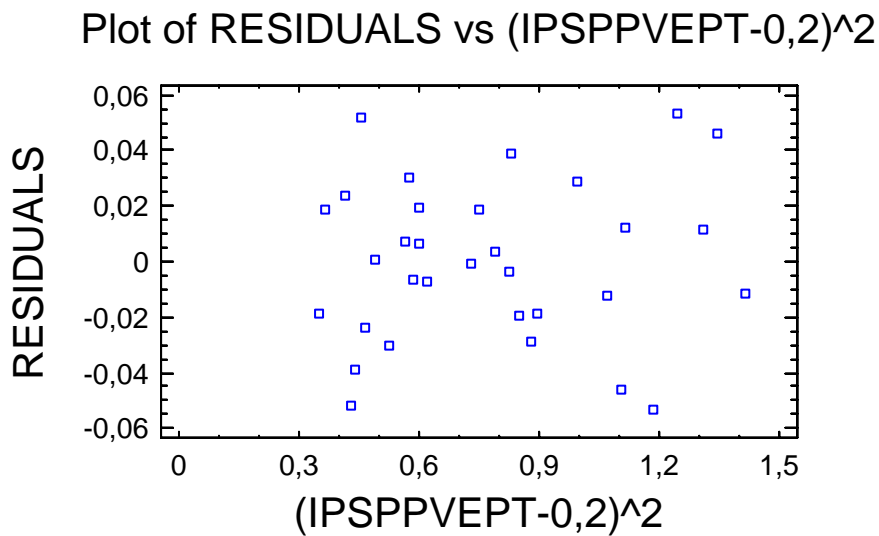


Figura IV-69. Gráfico de Residuos frente a los niveles de la Variable IPSPVEPT del bloque Retraso-Predictiva SMAGA-2-P13_20 ($K=0,01$).

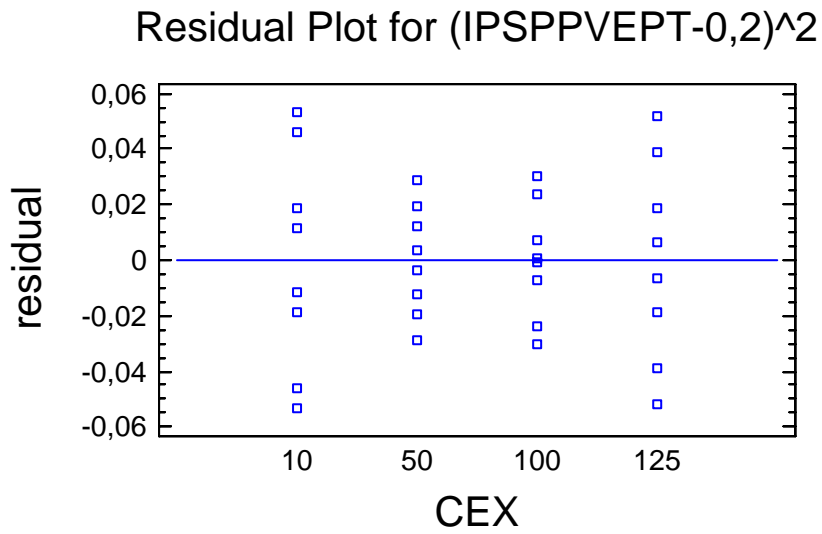


Figura IV-70. Gráfico de Residuos frente a los niveles del factor CEX del bloque Retraso-Predictiva SMAGA-2-P13_20 ($K=0,01$).

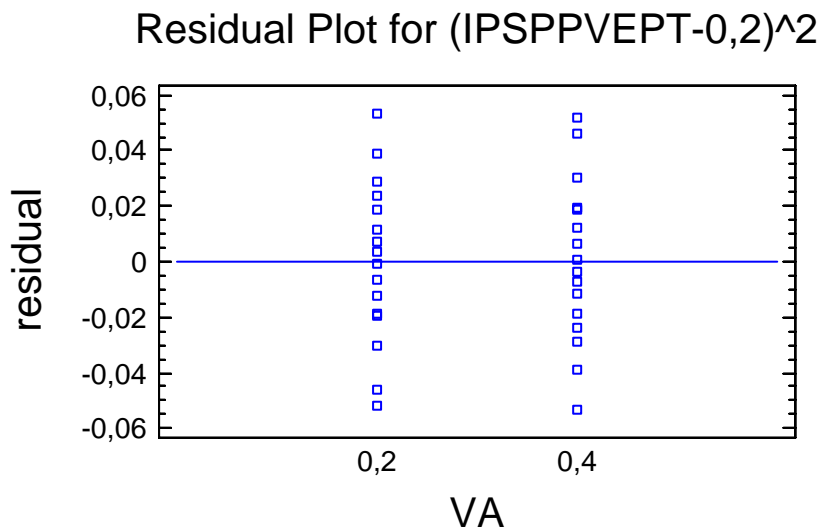


Figura IV-71. Gráfico de Residuos frente a los niveles del factor VA del bloque Retraso-Predictiva SMAGA-2-P13_20 ($K=0,01$).

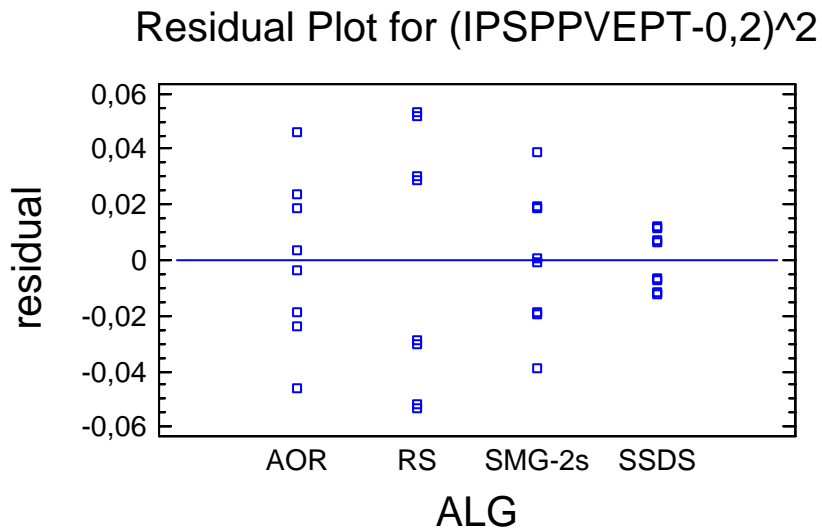


Figura IV-72. Gráfico de Residuos frente a los niveles del factor ALG del bloque Retraso-Predictiva SMAGA-2-P13_20 (K=0,01).

Como se observa en las figuras correspondientes a los residuos no se observa una falta de linealidad significativa en los mismos. Por lo que se descarta la ausencia de linealidad.

La correlación se detecta mediante el gráfico de residuos en relación al orden de ejecución de los tratamientos. Como se observa en la Figura IV-73 no se puede observar ninguna tendencia significativa en la distribución de los mismos por lo que se descarta la existencia de correlación.

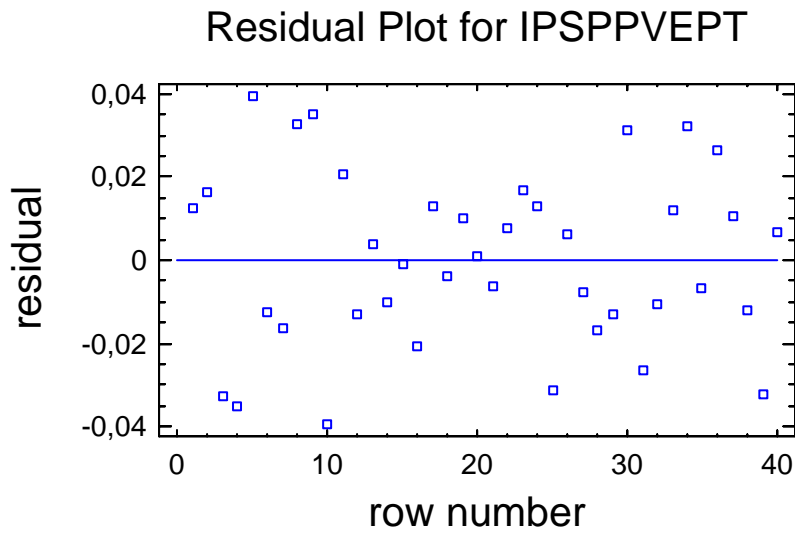


Figura IV-73. Residuos frente al orden de ejecución de los tratamientos del bloque Retraso-Predictiva SMAGA-2-P13_20 (K=0,01).

Una vez verificado el cumplimiento de las condiciones para el análisis ANOVA se obtiene y analiza la misma.

Analysis of Variance for (IPSPPEPT-0,2)^2 - Type III Sums of Squares

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
MAIN EFFECTS					
A:CEX	2,25782	3	0,752607	260,62	0,0000
B:VA	0,0114229	1	0,0114229	3,96	0,0779
C:ALG	0,23144	3	0,0771467	26,72	0,0001
INTERACTIONS					
AB	0,0188528	3	0,00628426	2,18	0,1606
AC	0,385864	9	0,0428737	14,85	0,0002
BC	0,14015	3	0,0467166	16,18	0,0006
RESIDUAL	0,0259896	9	0,00288773		
TOTAL (CORRECTED)	3,07154	31			

All F-ratios are based on the residual mean square error.

Tabla IV-30. ANOVA correspondiente IPSPPEPT del bloque Retraso-Predictiva SMAGA-2-P13_20 (K=0,01).

Como se puede ver en la Tabla IV-30 disponemos de 32 datos y tenemos 9 grados de libertad residuales. El contraste de hipótesis sobre la potencia (P-value) para un $\alpha=0,05$ implica que son significativos los efectos principales de los factores A y C, y además lo son las interacciones AC y BC.

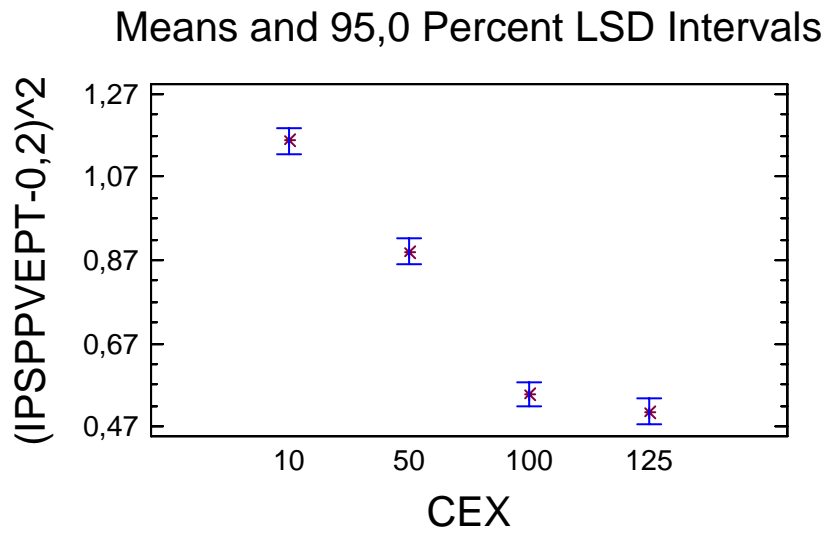


Figura IV-74. Gráfico del efecto principal del factor CEX del bloque Retraso-Predictiva SMAGA-2-P13_20 (K=0,01).

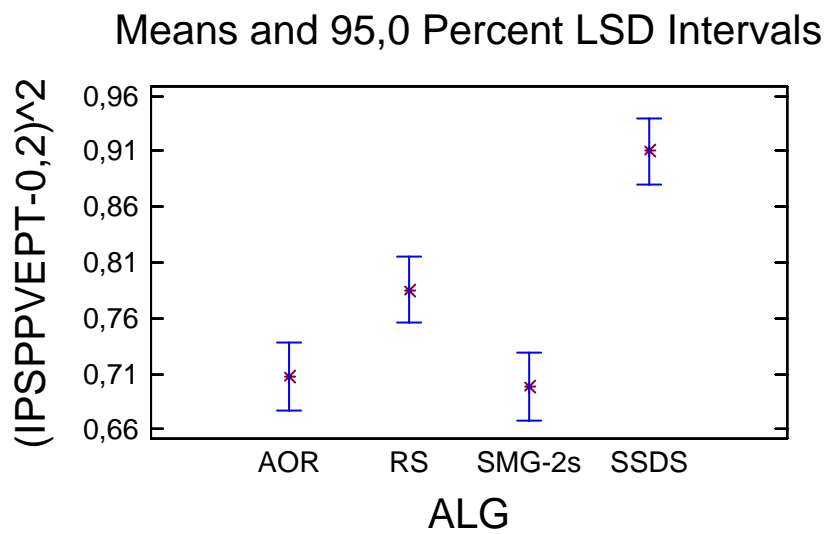


Figura IV-75. Gráfico del efecto principal del factor ALG del bloque Retraso-Predictiva SMAGA-2-P13_20 (K=0,01).

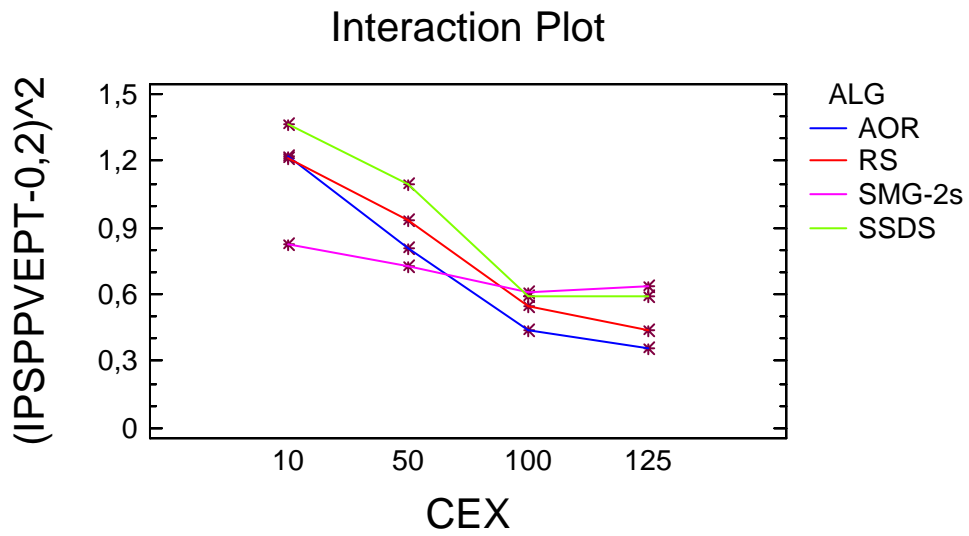


Figura IV-76. Gráfico del efecto de la interacción entre el factor CEX y el factor ALG del bloque Retraso-Predictiva SMAGA-2-P13_20 ($K=0,01$).

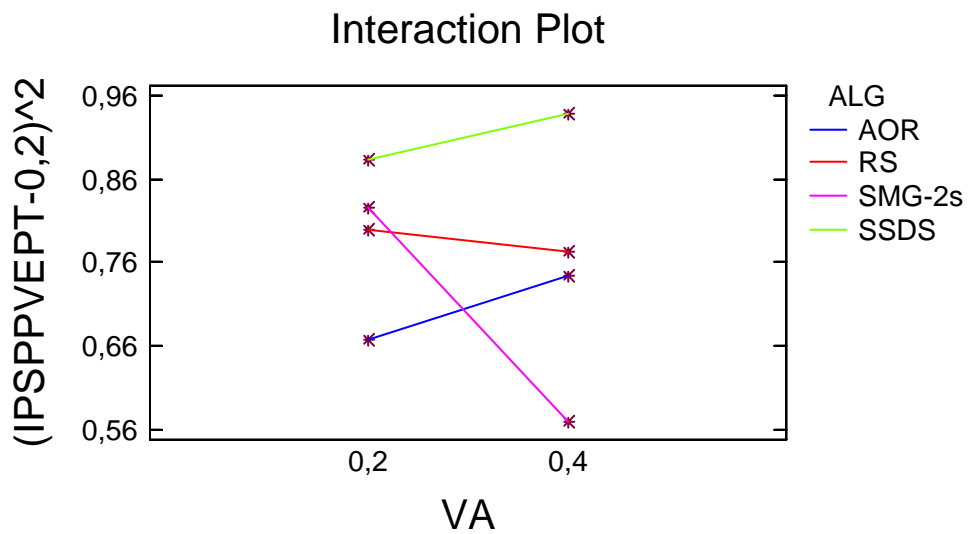


Figura IV-77. Gráfico del efecto de la interacción entre el factor VA y el factor ALG del bloque Retraso-Predictiva SMAGA-2-P13_20 ($K=0,01$).

A-IV.2.3.2 Análisis del Bloque P13_50

En este bloque se ha tenido que dividir el análisis en dos partes. En una primera parte se ha estudiado que ocurre cuando el valor de K es 0, y en una segunda fase lo que ocurre cuando el valor de K superior a 0. Nuevamente el motivo de la división ha sido el de trabajar con datos con variable homogéneas. En el caso de K superior a 0 sólo se ha estudiado el valor 0,01 ya que se ha observado como el tipo de algoritmo que depende del valor de K se ve superado siempre por un algoritmo independiente de este valor cuando K es 0,01, y que por lo tanto responderá peor y será nuevamente superado por los segundos cuando K tomo el valor 0,02.

A-IV.2.3.2.1. Análisis del Bloque P13_50 (Valor de K=0)

En este bloque experimental se ha necesitado realizar una transformación de la variable dependiente debido a que se ha observado la existencia de alguna relación no lineal entre los factores. La transformación que se ha empleado ha sido el cálculo de logaritmo neperiano (LOG en la aplicación Statgraphics) de la variable. En todo caso, la significación de los factores y los test realizados son tan válidos como si se hubiesen realizado sin transformación.

En relación con la normalidad del experimento se ha obtenido un Chi-Square = 8,64993 con 4 d.f. y un P-Value = 0,0704707, lo que descarta la posibilidad de que la variable analizada no proceda de una distribución normal. Así mismo se puede observa en la Figura IV-78 la variable se distribuye muy próxima a la línea continua diagonal, lo que implica su proximidad a la distribución Normal.

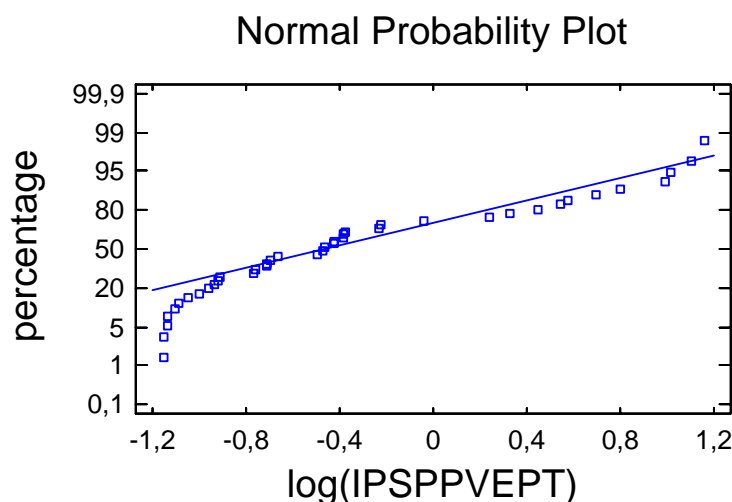


Figura IV-78. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPSPPEPT del bloque Retraso-Predictiva SMAGA-2-P13_50 ($K=0$) en relación a las frecuencias relativas acumuladas.

Por lo que respecta a la homogeneidad de las varianzas se ha realizado el test de Levene para la variable dependiente con respecto a las variables independientes obteniéndose los siguientes valores:

- IPSPVEPT respecto a CEX: test de Levene: 2,11697 P-Value = 0,115159
- IPSPVEPT respecto a VA: test de Levene: 0,127382 P-Value = 0,723136
- IPSPVEPT respecto a ALG: test de Levene: 1,5099 P-Value = 0,220525

Por lo que se descarta que las varianzas no sean homogéneas.

Por lo que se refiere a la linealidad se empleará el “gráfico de residuos (X_t, \bar{u}_t) y (\hat{Y}_t, \bar{u}_t)”, que permite disponer los residuos, en el eje de ordenadas, frente a las variables exógenas (X_t), y endógena (\hat{Y}_t).

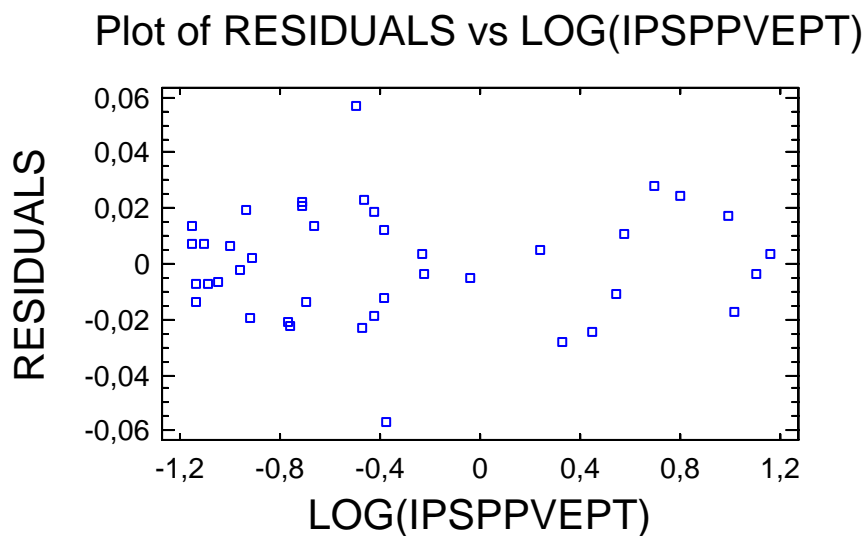


Figura IV-79. Gráfico de Residuos frente a los niveles de la Variable IPSPVEPT del bloque Retraso-Predictiva SMAGA-2-P13_50 (K=0).

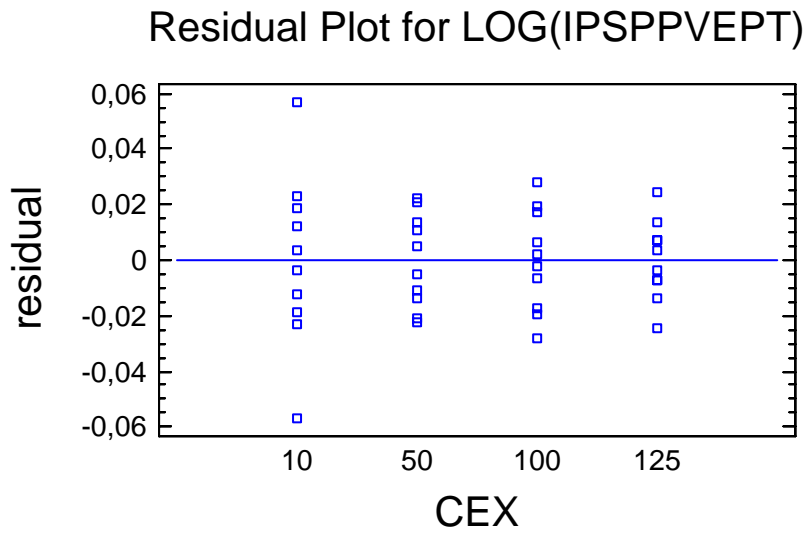


Figura IV-80. Gráfico de Residuos frente a los niveles del factor CEX del bloque Retraso-Predictiva SMAGA-2-P13_50 (K=0).

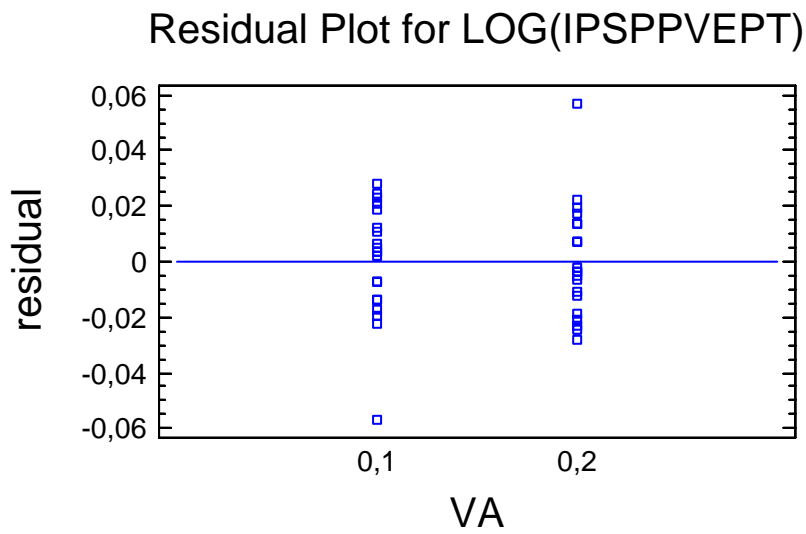


Figura IV-81. Gráfico de Residuos frente a los niveles del factor VA del bloque Retraso-Predictiva SMAGA-2-P13_50 (K=0).

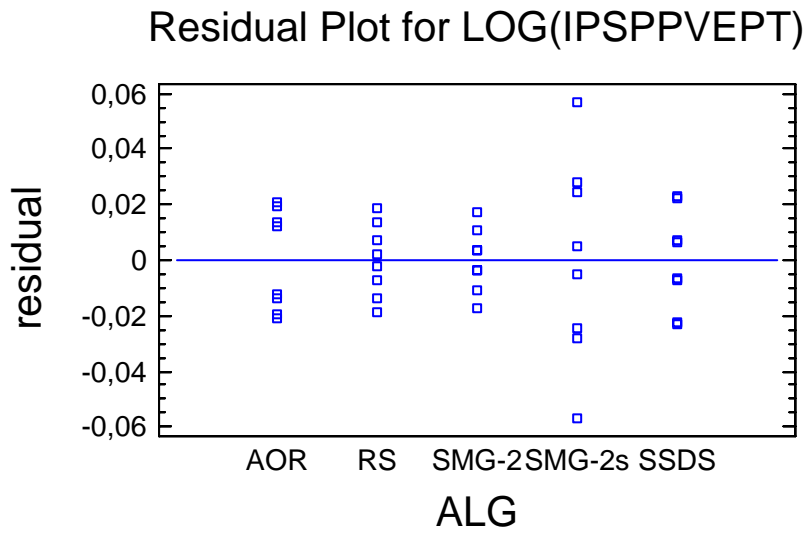


Figura IV-82. Gráfico de Residuos frente a los niveles del factor ALG del bloque Retraso-Predictiva SMAGA-2-P13_50 (K=0).

Como se observa en las figuras correspondientes a los residuos no se observa una falta de linealidad significativa en los mismos. Por lo que se descarta la ausencia de linealidad.

La correlación se detecta mediante el gráfico de residuos en relación al orden de ejecución de los tratamientos. Como se observa en la Figura IV-83 no se puede observar ninguna tendencia significativa en la distribución de los mismos por lo que se descarta la existencia de correlación.

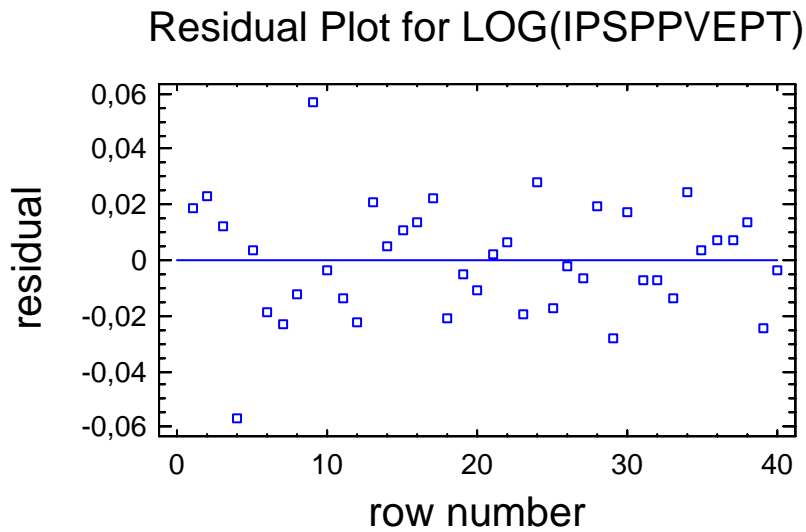


Figura IV-83. Residuos frente al orden de ejecución de los tratamientos del bloque Retraso-Predictiva SMAGA-2-P13_50 (K=0).

Una vez verificado el cumplimiento de las condiciones para el análisis ANOVA se obtiene y analiza la misma.

Analysis of Variance for LOG(IPSPVEPT) - Type III Sums of Squares

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
MAIN EFFECTS					
A:CEX	0,0710581	3	0,023686	17,92	0,0001
B:VA	0,0459647	1	0,0459647	34,78	0,0001
C:ALG	14,927	4	3,73174	2823,39	0,0000
INTERACTIONS					
AB	0,00883757	3	0,00294586	2,23	0,1374
AC	5,28708	12	0,44059	333,35	0,0000
BC	0,114099	4	0,0285247	21,58	0,0000
RESIDUAL	0,0158607	12	0,00132172		
TOTAL (CORRECTED)	20,4699	39			

All F-ratios are based on the residual mean square error.

Tabla IV-31. ANOVA correspondiente IPSPVEPT del bloque Retraso-Predictiva SMAGA-2-P13_50 (K=0).

Como se puede ver en la Tabla IV-31 disponemos de 40 datos y tenemos 12 grados de libertad residuales. El contraste de hipótesis sobre la potencia (P-value) para un $\alpha=0,05$ implica que son significativos los efectos principales de los factores A, B y C, y además lo son las interacciones AC y BC.

Means and 95,0 Percent LSD Intervals

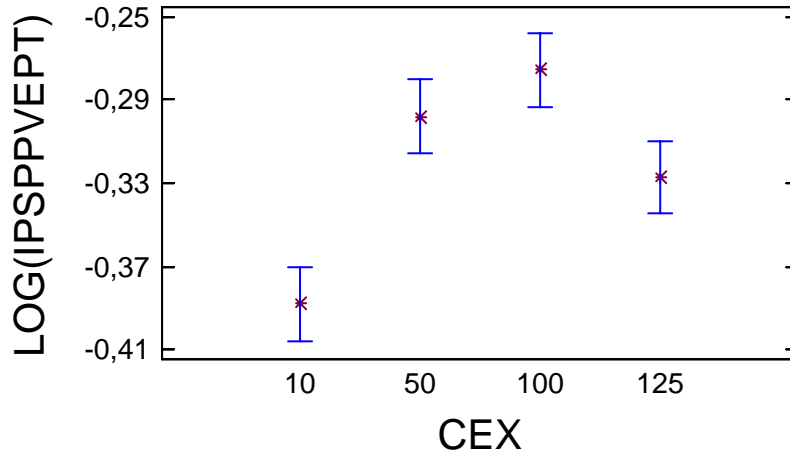


Figura IV-84. Gráfico del efecto principal del factor CEX del bloque Retraso-Predictiva SMAGA-2-P13_50 (K=0).

Means and 95,0 Percent LSD Intervals

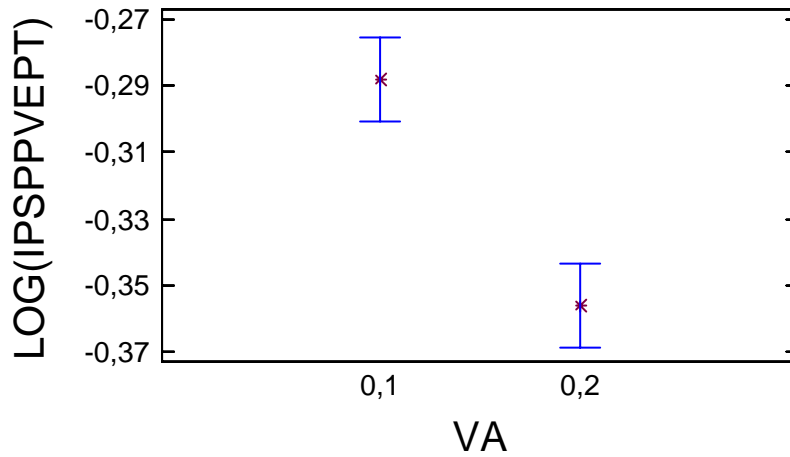


Figura IV-85. Gráfico del efecto principal del factor VA del bloque Retraso-Predictiva SMAGA-2-P13_50 (K=0).

Means and 95,0 Percent LSD Intervals

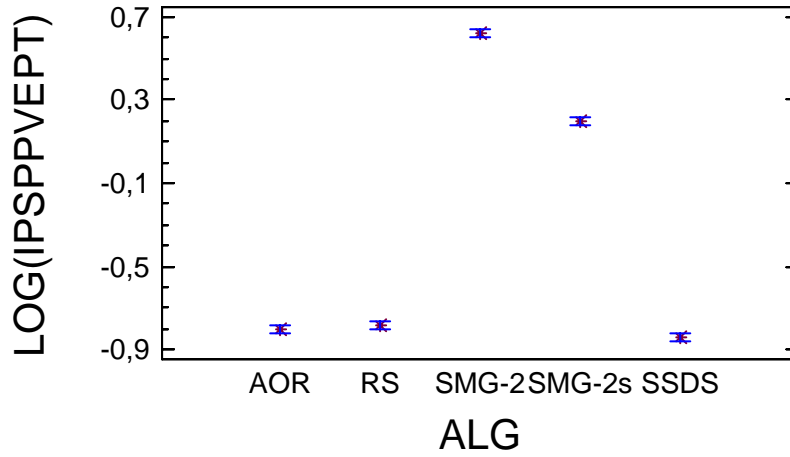


Figura IV-86. Gráfico del efecto principal del factor ALG del bloque Retraso-Predictiva SMAGA-2-P13_50 (K=0).

Interaction Plot

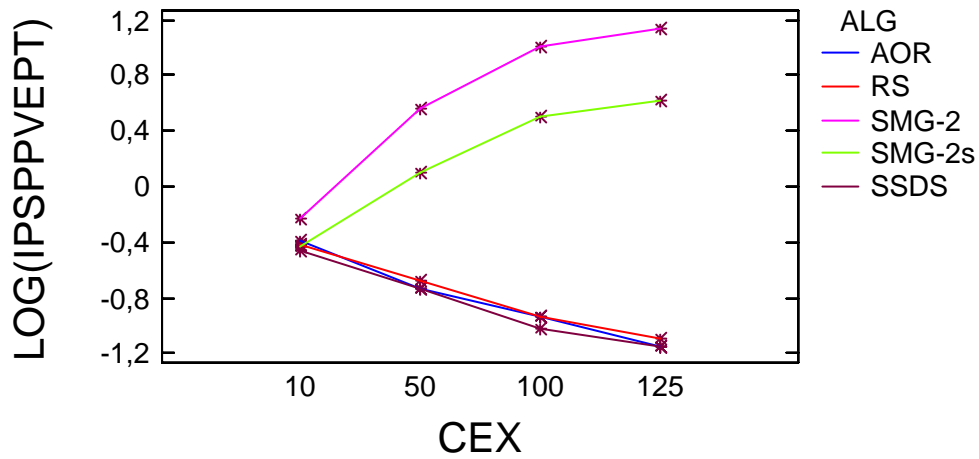


Figura IV-87. Gráfico del efecto de la interacción entre el factor CEX y el factor ALG del bloque Retraso-Predictiva SMAGA-2-P13_50 (K=0).

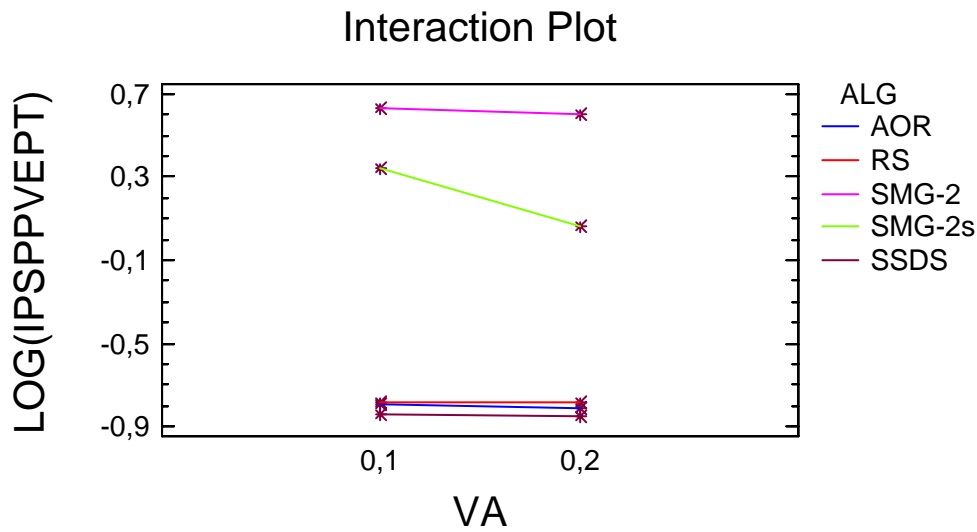


Figura IV-88. Gráfico del efecto de la interacción entre el factor VA y el factor ALG del bloque Retraso-Predictiva SMAGA-2-P13_50 (K=0).

A-IV.2.3.2.2. Análisis del Bloque P13_50 (Valor de K=0,005)

En este bloque experimental se ha necesitado realizar una transformación de la variable dependiente, y además en esta ocasión se no ha incluido en el análisis el algoritmo SMAGA-2. Ambas decisiones están orientadas a garantizar el cumplimiento de la condición de igualdad en la varianza. La transformación que se ha empleado ha sido el cálculo del logaritmo neperiano de la variable. En todo caso, la significación de los factores y los test realizados son tan válidos como si se hubiesen realizado sin transformación. La no inclusión del algoritmo SMAGA-2 en el análisis es factible ya que ha demostrado no ser interesante para el caso de K=0, por lo que aún será menos interesante cuando el valor de K sea superior.

En relación con la normalidad del experimento se ha obtenido un Chi-Square = 4,75008 con 4 d.f. y un P-Value = 0,313915, lo que descarta la posibilidad de que la variable analizada no proceda de una distribución normal. Así mismo se puede observa en la Figura IV-89 como la variable se distribuye muy próxima a la línea continua diagonal, lo que implica su proximidad a la distribución Normal.

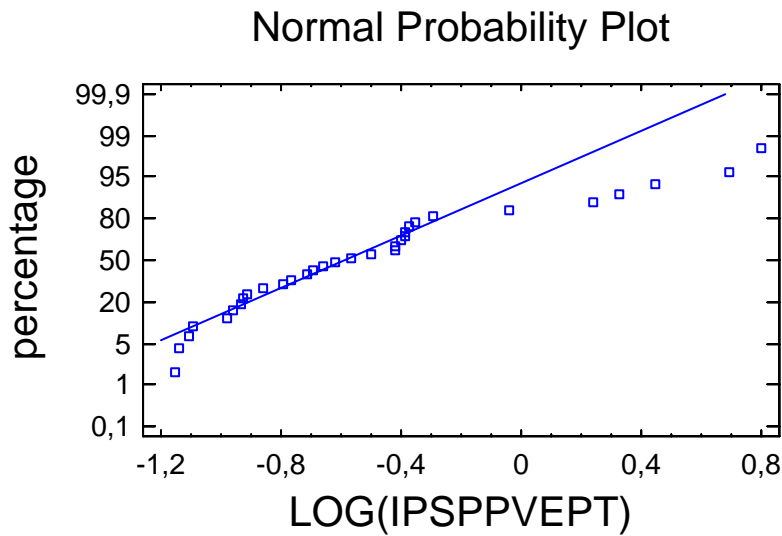


Figura IV-89. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPSPVEPT del bloque Retraso-Predictiva SMAGA-2-P13_50 ($K=0,005$) en relación a las frecuencias relativas acumuladas.

Por lo que respecta a la homogeneidad de las varianzas se ha realizado el test de Levene para la variable dependiente con respecto a las variables independientes obteniéndose los siguientes valores:

- IPSPVEPT respecto a CEX: test de Levene: 1,25666 P-Value = 0,308217
- IPSPVEPT respecto a VA: test de Levene: 0,366839 P-Value = 0,549287
- IPSPVEPT respecto a ALG: test de Levene: 1,23552 P-Value = 0,315414

Por lo que se descarta que las varianzas no sean homogéneas.

Por lo que se refiere a la linealidad se empleará el “gráfico de residuos (X_t, \bar{u}_t) y (\hat{Y}_t, \bar{u}_t)”, que permite disponer los residuos, en el eje de ordenadas, frente a las variables exógenas (X_t), y endógena (\hat{Y}_t).

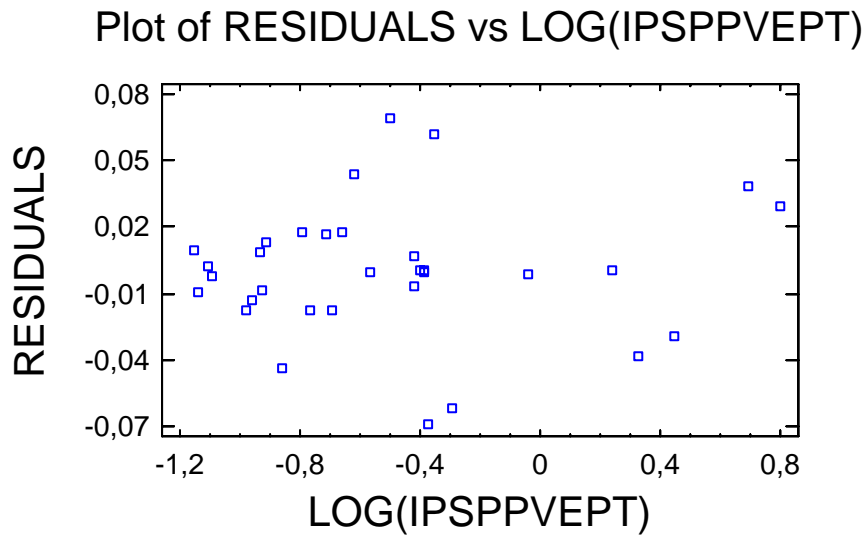


Figura IV-90. Gráfico de Residuos frente a los niveles de la Variable IPSPPEPT del bloque Retraso-Predictiva SMAGA-2-P13_50 ($K=0,005$).

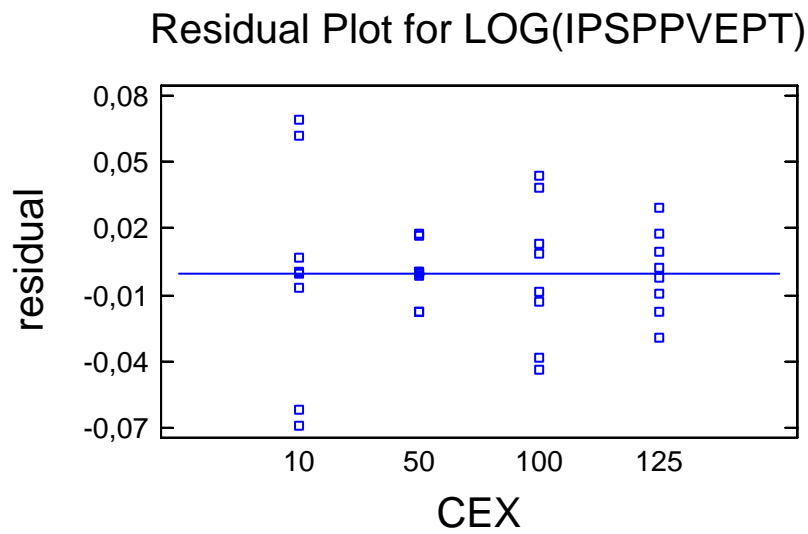


Figura IV-91. Gráfico de Residuos frente a los niveles del factor CEX del bloque Retraso-Predictiva SMAGA-2-P13_50 ($K=0,005$).

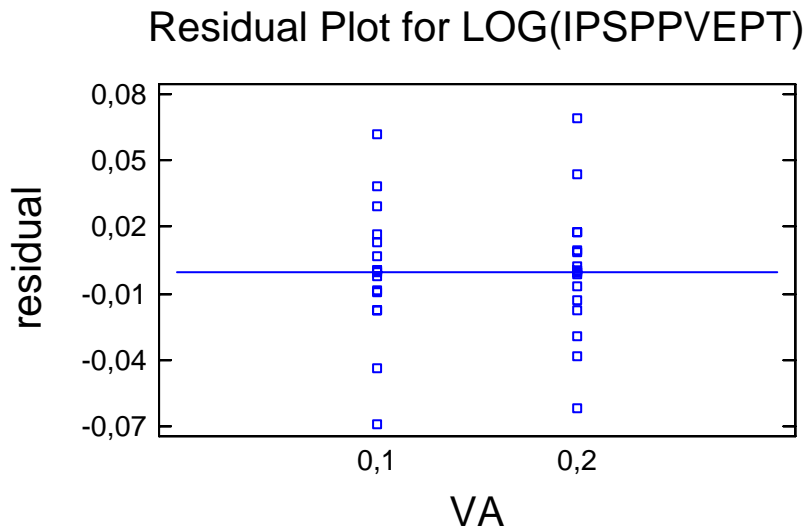


Figura IV-92. Gráfico de Residuos frente a los niveles del factor VA del bloque Retraso-Predictiva SMAGA-2-P13_50 ($K=0,005$).

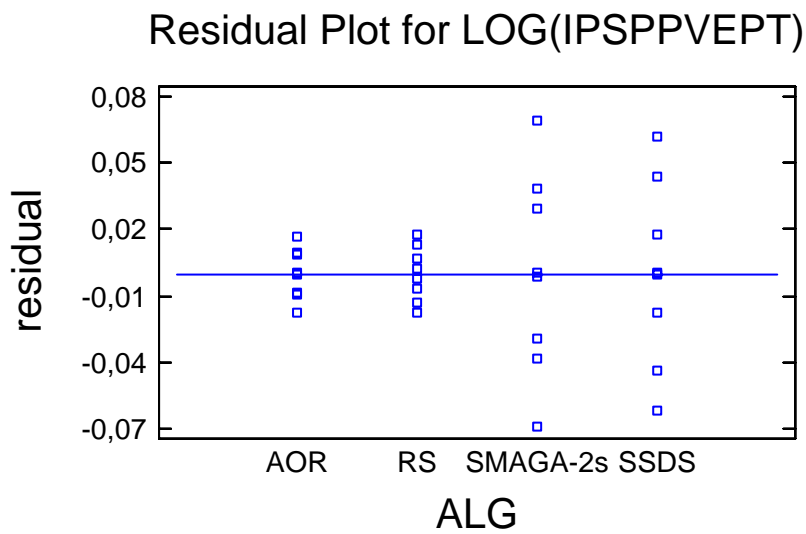


Figura IV-93. Gráfico de Residuos frente a los niveles del factor ALG del bloque Retraso-Predictiva SMAGA-2-P13_50 ($K=0,005$).

Como se observa en las figuras correspondientes a los residuos no se observa una falta de linealidad significativa en los mismos. Por lo que se descarta la ausencia de linealidad.

La correlación se detecta mediante el gráfico de residuos en relación al orden de ejecución de los tratamientos. Como se observa en la Figura IV-94 no se puede observar ninguna tendencia significativa en la distribución de los mismos por lo que se descarta la existencia de correlación.

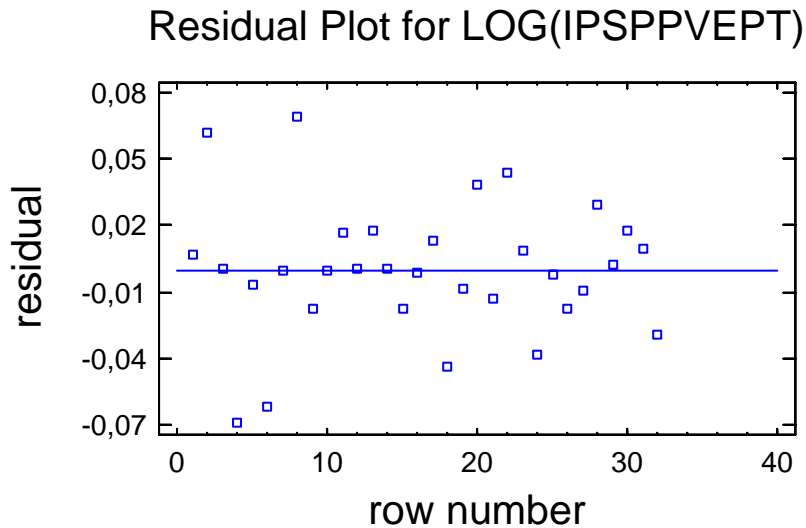


Figura IV-94. Residuos frente al orden de ejecución de los tratamientos del bloque Retraso-Predictiva SMAGA-2-P13_50 (K=0,005).

Una vez verificado el cumplimiento de las condiciones para el análisis ANOVA se obtiene y analiza la misma.

Analysis of Variance for LOG(IPSPVEPT) - Type III Sums of Squares

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
MAIN EFFECTS					
A:CEX	0,249696	3	0,0832321	26,79	0,0001
B:VA	0,00998923	1	0,00998923	3,21	0,1066
C:ALG	5,36834	3	1,78945	575,91	0,0000
INTERACTIONS					
AB	0,00134307	3	0,00044769	0,14	0,9309
AC	2,67369	9	0,297076	95,61	0,0000
BC	0,201665	3	0,0672216	21,63	0,0002
RESIDUAL	0,0279644	9	0,00310716		
TOTAL (CORRECTED)	8,53268	31			

All F-ratios are based on the residual mean square error.

Tabla IV-32. ANOVA correspondiente IPSPVEPT del bloque Retraso-Predictiva SMAGA-2-P13_50 (K=0,01).

Como se puede ver en la Tabla IV-32 disponemos de 32 datos y tenemos 9 grados de libertad residuales. El contraste de hipótesis sobre la potencia (P-value) para un $\alpha=0,05$ implica que son significativos los efectos principales de los factores A y C, y además lo son las interacciones AC y BC.

Means and 95,0 Percent LSD Intervals

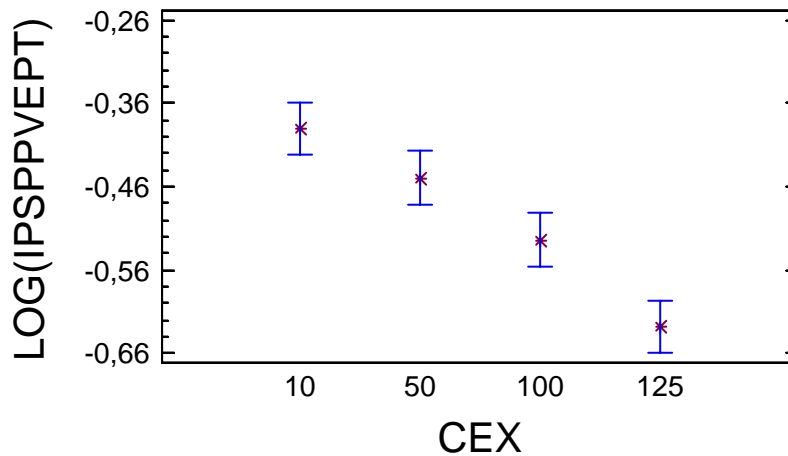


Figura IV-95. Gráfico del efecto principal del factor CEX del bloque Retraso-Predictiva SMAGA-2-P13_50 (K=0,005).

Means and 95,0 Percent LSD Intervals

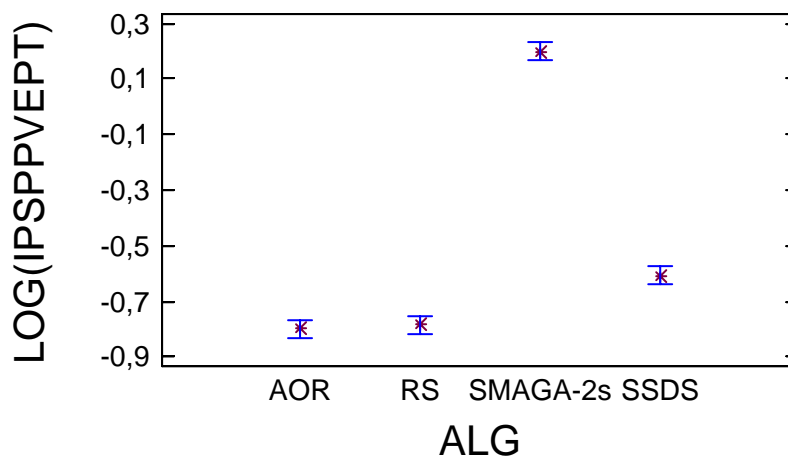


Figura IV-96. Gráfico del efecto principal del factor VA del bloque Retraso-Predictiva SMAGA-2-P13_50 (K=0,005).

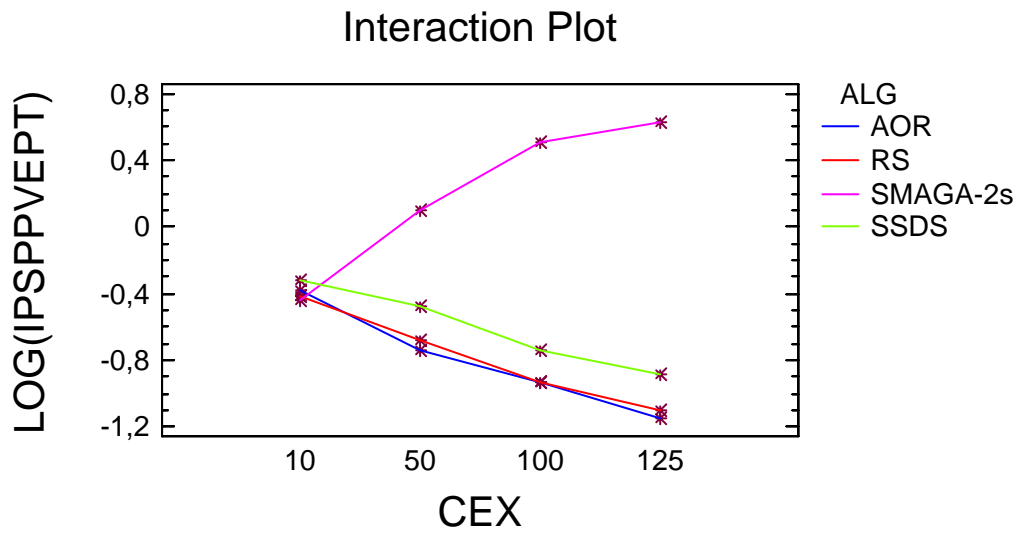


Figura IV-97. Gráfico del efecto de la interacción entre el factor CEX y el factor ALG del bloque Retraso-Predictiva SMAGA-2-P13_50 ($K=0,005$).

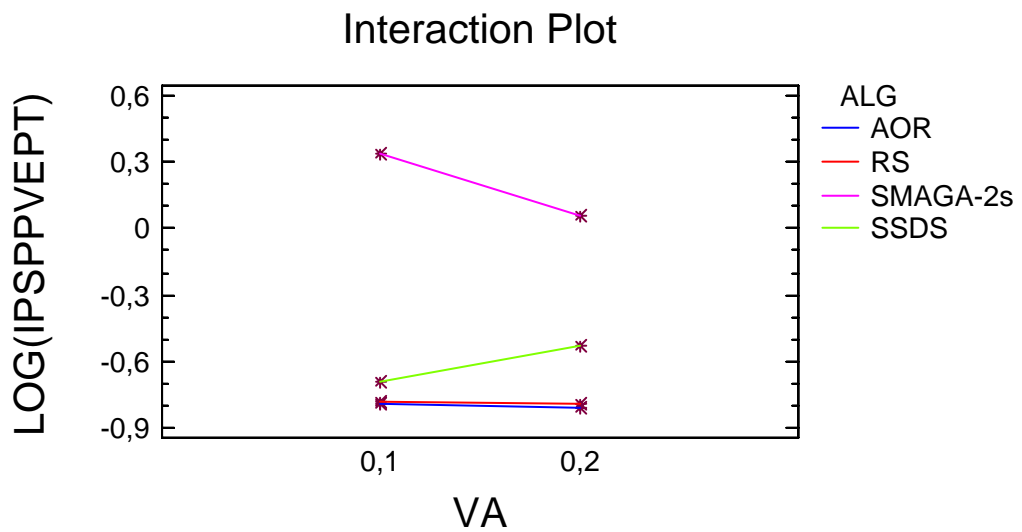


Figura IV-98. Gráfico del efecto de la interacción entre el factor VA y el factor ALG del bloque Retraso-Predictiva SMAGA-2-P13_50 ($K=0,005$).

A-IV.2.4 Análisis Estadístico para la Configuración P3 con Programación Predictiva basada en SMAGA-2

Conjunto Experimental	K=0 y VA=0,2				
	RS	SSDS	AOR	SMAGA-2s	SMAGA-2
SSD10_P3_20	4,328	2,914	3,144	2,627	2,644
SSD50_P3_20	4,087	2,858	2,254	2,834	3,756
SSD100_P3_20	4,724	3,847	1,791	3,673	5,470
SSD125_P3_20	4,492	3,994	1,573	4,250	5,865

Tabla IV-33. Valor de IPSPVEPT a partir de SMAGA-2 con retraso para 20 trabajos, y un valor de VA de 0,2.

Conjunto Experimental	K=0 y VA=0,4					K=0,01 y VA=0,4				
	RS	SSDS	AOR	SMAGA-2s	SMAGA-2	RS	SSDS	AOR	SMAGA-2s	SMAGA-2
SSD10_P3_20	4,340	2,868	3,145	2,880	2,644	4,340	3,360	3,145	2,880	3,618
SSD50_P3_20	3,937	2,704	2,160	2,851	3,695	3,937	3,273	2,160	2,851	4,907
SSD100_P3_20	4,855	3,949	1,915	3,740	5,187	4,855	4,214	1,915	3,740	6,191
SSD125_P3_20	4,248	3,997	1,593	3,999	5,997	4,248	4,233	1,593	3,999	7,187

Tabla IV-34. Valor de IPSPVEPT a partir de SMAGA-2 con retraso para 20 trabajos, y un valor de VA de 0,4.

Conjunto Experimental	K=0 y VA=0,1				
	RS	SSDS	AOR	SMAGA-2s	SMAGA-2
SSD10_P3_50	3,217	2,172	1,504	2,208	2,595
SSD50_P3_50	3,532	2,772	1,034	3,800	4,535
SSD100_P3_50	4,396	3,778	0,816	5,006	5,753
SSD125_P3_50	4,599	4,313	0,752	5,792	6,762

Tabla IV-35. Valor de IPSPVEPT a partir de SMAGA-2 con retraso para 50 trabajos, y un valor de VA de 0,1.

Conjunto Experimental	K=0 y VA=0,2				
	RS	SSDS	AOR	SMAGA-2s	SMAGA-2
SSD10_P3_50	3,230	2,17	1,496	1,978	2,611
SSD50_P3_50	3,414	2,816	1,073	3,208	4,386
SSD100_P3_50	4,202	3,752	0,812	4,057	5,596
SSD125_P3_50	4,682	4,283	0,726	4,966	6,773

Tabla IV-36. Valor de IPSPVEPT a partir de SMAGA-2 con retraso para 50 trabajos, y un valor de VA de 0,2.

A-IV.2.5 Resultados para la Configuración P3 con Programación Predictiva basada en SMAGA-2 Modificada

Conjunto Experimental	K=0 y VA=0,2					K=0,01 y VA=0,2				
	RS	SSDS	AOR	SMAGA-2s	SMAGA-2	RS	SSDS	AOR	SMAGA-2s	SMAGA-2
SSD10_P3_20	0,924	0,793	0,910	0,551	-0,465	0,924	1,035	0,910	0,551	0,473
SSD50_P3_20	0,633	0,525	0,644	0,084	-1,591	0,633	0,798	0,644	0,084	-0,523
SSD100_P3_20	0,445	0,422	0,455	-0,481	-2,909	0,445	0,644	0,455	-0,481	-1,670
SSD125_P3_20	0,374	0,323	0,353	-0,807	-3,325	0,374	0,521	0,353	-0,8077	-2,191
K=0,02 y VA=0,2										
SSD10_P3_20	0,924	1,285	0,910	0,551	1,512					
SSD50_P3_20	0,633	1,045	0,644	0,084	0,564					
SSD100_P3_20	0,445	0,916	0,455	-0,481	-0,426					
SSD125_P3_20	0,374	0,699	0,353	-0,807	-1,040					

Tabla IV-37. Valor de IPSPVEPT a partir de SMAGA-2 Modificado con retraso para 20 trabajos, y un valor de VA de 0,2.

Conjunto Experimental	K=0 y VA=0,4					K=0,01 y VA=0,4				
	RS	SSDS	AOR	SMAGA-2s	SMAGA-2	RS	SSDS	AOR	SMAGA-2s	SMAGA-2
SSD10_P3_20	0,950	0,765	0,893	0,695	-0,532	0,950	1,053	0,893	0,695	0,632
SSD50_P3_20	0,633	0,539	0,624	0,413	-1,540	0,633	0,783	0,624	0,413	-0,310
SSD100_P3_20	0,452	0,442	0,433	0,205	-2,906	0,452	0,678	0,433	0,205	-1,378
SSD125_P3_20	0,363	0,319	0,369	-0,042	-3,470	0,363	0,512	0,369	-0,042	-2,020
K=0,02 y VA=0,4										
SSD10_P3_20	0,950	1,280	0,893	0,695	1,911					
SSD50_P3_20	0,633	1,062	0,624	0,413	0,985					
SSD100_P3_20	0,452	0,962	0,433	0,205	0,165					
SSD125_P3_20	0,363	0,767	0,369	-0,042	-0,523					

Tabla IV-38. Valor de IPSPVEPT a partir de SMAGA-2 Modificado con retraso para 20 trabajos, y un valor de VA de 0,4.

Conjunto Experimental	K=0 y VA=0,1					K=0,005 y VA=0,1				
	RS	SSDS	AOR	SMAGA-2s	SMAGA-2	RS	SSDS	AOR	SMAGA-2s	SMAGA-2
SSD10_P3_50	0,990	0,220	0,897	-1,966	-2,817	0,990	0,756	0,897	-1,966	-1,241
SSD50_P3_50	0,723	0,114	0,628	-2,212	-3,003	0,723	0,536	0,628	-2,212	-1,161
SSD100_P3_50	0,515	0,113	0,416	-1,645	-2,039	0,515	0,357	0,416	-1,645	-0,359
SSD125_P3_50	0,585	0,363	0,374	-1,032	-1,270	0,585	0,566	0,374	-1,032	0,415
K=0,01 y VA=0,1										
SSD10_P3_50	0,990	1,166	0,897	-1,966	0,430					
SSD50_P3_50	0,723	0,984	0,628	-2,212	0,725					
SSD100_P3_50	0,515	0,730	0,416	-1,645	1,390					
SSD125_P3_50	0,585	0,861	0,374	-1,032	2,135					

Tabla IV-39. Valor de IPSPPVEPT a partir de SMAGA-2 Modificado con retraso para 50 trabajos, y un valor de VA de 0,1.

Conjunto Experimental	K=0 y VA=0,2					K=0,005 y VA=0,2				
	RS	SSDS	AOR	SMAGA-2s	SMAGA-2	RS	SSDS	AOR	SMAGA-2s	SMAGA-2
SSD10_P3_50	0,970	0,227	0,847	-1,228	-2,897	0,970	1,715	0,847	-1,228	1,990
SSD50_P3_50	0,685	0,093	0,604	-1,773	-2,900	0,685	1,340	0,604	-1,773	2,641
SSD100_P3_50	0,483	0,097	0,425	-1,271	-2,099	0,483	1,006	0,425	-1,271	3,237
SSD125_P3_50	0,509	0,348	0,374	-0,807	-1,370	0,509	1,134	0,374	-0,807	3,878
K=0,01 y VA=0,2										
SSD10_P3_50	0,970	0,940	0,847	-1,228	-0,464					
SSD50_P3_50	0,685	0,768	0,604	-1,773	-0,179					
SSD100_P3_50	0,483	0,549	0,425	-1,271	0,560					
SSD125_P3_50	0,509	0,748	0,374	-0,807	1,400					

Tabla IV-40. Valor de IPSPPVEPT a partir de SMAGA-2 Modificado con retraso para 50 trabajos, y un valor de VA de 0,2.

A-IV.2.5.1 Análisis del Bloque P3_20

En este bloque se ha tenido que dividir el análisis en dos partes. En una primera parte se ha estudiado que ocurre cuando el valor de K es 0, y en una segunda fase lo que ocurre cuando el valor de K superior a 0. Nuevamente el motivo de la división ha sido el de trabajar con datos con variable homogéneas. En el caso de K superior a 0 sólo se ha estudiado el valor 0,01 ya que

se ha observado como el tipo de algoritmo que depende del valor de K se ve superado siempre por un algoritmo independiente de este valor cuando K es 0,01, y que por lo tanto responderá peor y será nuevamente superado por los segundos cuando K tome el valor 0,02.

A-IV.2.5.1.1. Análisis del Bloque P3_20 (Valor de K=0)

En este bloque experimental se ha necesitado realizar una transformación de la variable dependiente, y además en esta ocasión se no ha incluido en el análisis el algoritmo RS. Ambas decisiones están orientadas a garantizar el cumplimiento de la condición de igualdad en la varianza. La transformación que se ha empleado ha sido el cálculo del logaritmo neperiano de la variable a la cual se le ha una unidad. En todo caso, la significación de los factores y los test realizados son tan válidos como si se hubiesen realizado sin transformación. La no inclusión del algoritmo RS en el análisis es factible ya que ha visto en los datos resultantes como nunca es un algoritmo interesante.

En relación con la normalidad del experimento se ha obtenido un Chi-Square = 3,43747 con 4 d.f. y un P-Value = 0,487449, lo que descarta la posibilidad de que la variable analizada no proceda de una distribución normal. Así mismo se puede observa en la Figura IV-99 como la variable se distribuye muy próxima a la línea continua diagonal, lo que implica su proximidad a la distribución Normal.

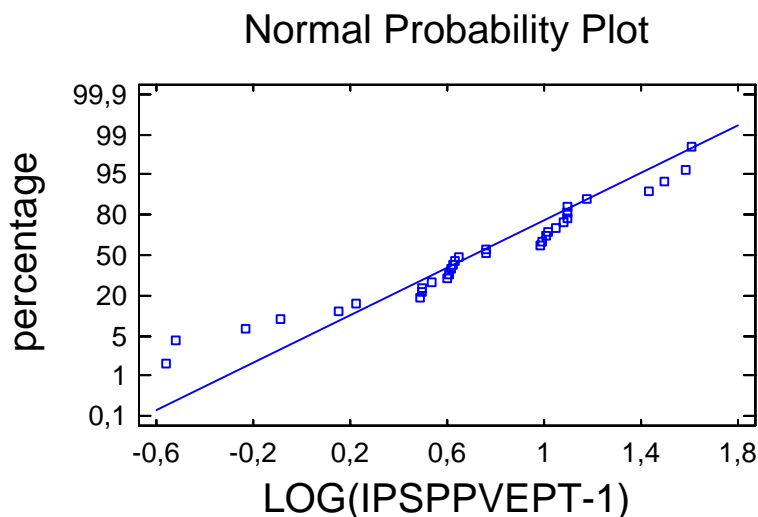


Figura IV-99. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPSPVEPT del bloque Retraso-Predictiva SMAGA-2-P3_20 ($K=0$) en relación a las frecuencias relativas acumuladas.

Por lo que respecta a la homogeneidad de las varianzas se ha realizado el test de Levene para la variable dependiente con respecto a las variables independientes obteniéndose los siguientes valores:

- IPSPVEPT respecto a CEX: test de Levene: 1,65432 P-Value = 0,199435
- IPSPVEPT respecto a VA: test de Levene: 0,0101382 P-Value = 0,920468
- IPSPVEPT respecto a ALG: test de Levene: 2,22833 P-Value = 0,106918

Por lo que se descarta que las varianzas no sean homogéneas.

Por lo que se refiere a la linealidad se empleará el “gráfico de residuos (X_t, \bar{u}_t) y (\hat{Y}_t, \bar{u}_t)”, que permite disponer los residuos, en el eje de ordenadas, frente a las variables exógenas (X_t), y endógena (\hat{Y}_t).

Plot of RESIDUALS vs LOG(IPSPVEPT-1)

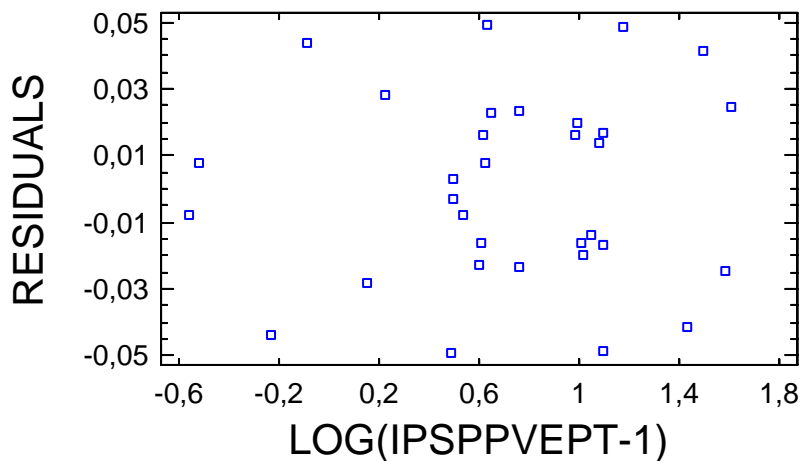


Figura IV-100. Gráfico de Residuos frente a los niveles de la Variable IPSPVEPT del bloque Retraso-Predictiva SMAGA-2-P3_20 (K=0).

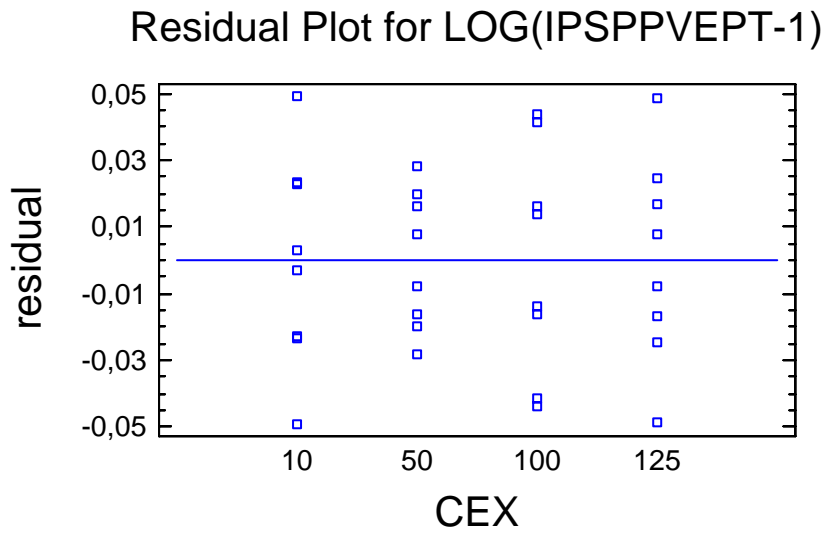


Figura IV-101. Gráfico de Residuos frente a los niveles del factor CEX del bloque Retraso-Predictiva SMAGA-2-P3_20 (K=0).

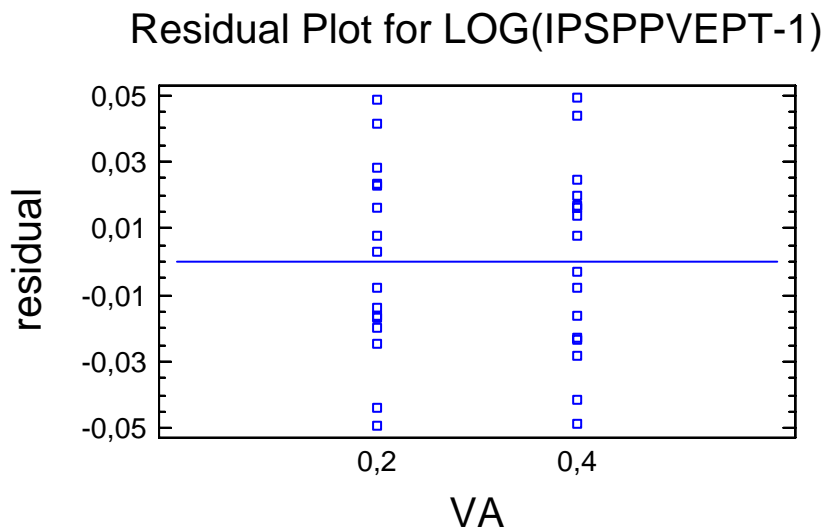


Figura IV-102. Gráfico de Residuos frente a los niveles del factor VA del bloque Retraso-Predictiva SMAGA-2-P3_20 (K=0).

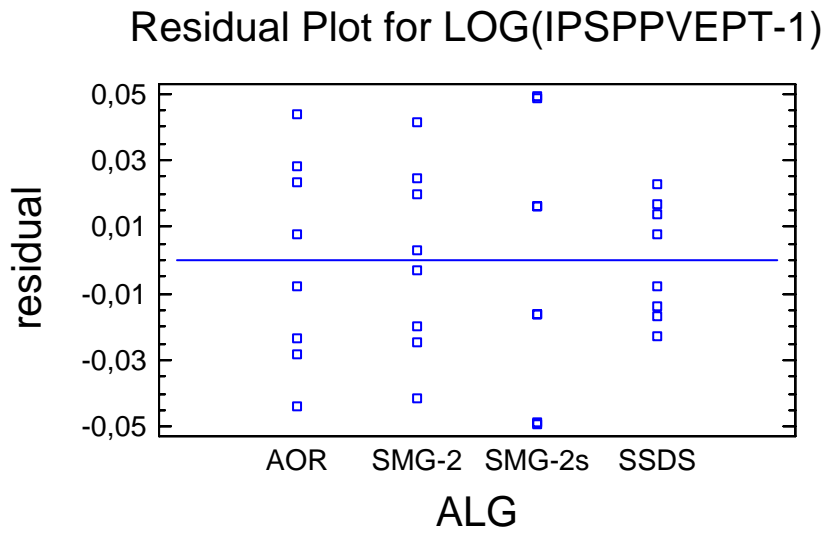


Figura IV-103. Gráfico de Residuos frente a los niveles del factor ALG del bloque Retraso-Predictiva SMAGA-2-P3_20 (K=0).

Como se observa en las figuras correspondientes a los residuos no se observa una falta de linealidad significativa en los mismos. Por lo que se descarta la ausencia de linealidad.

La correlación se detecta mediante el gráfico de residuos en relación al orden de ejecución de los tratamientos. Como se observa en la Figura IV-104 no se puede observar ninguna tendencia significativa en la distribución de los mismos por lo que se descarta la existencia de correlación.

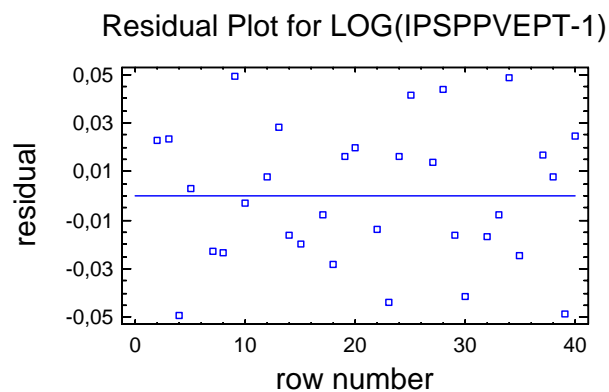


Figura IV-104. Residuos frente al orden de ejecución de los tratamientos del bloque Retraso-Predictiva SMAGA-2-P3_20 (K=0).

Una vez verificado el cumplimiento de las condiciones para el análisis ANOVA se obtiene y analiza la misma.

Analysis of Variance for LOG(IPSPPEPT-1) - Type III Sums of Squares

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
MAIN EFFECTS					
A:CEX	0,422442	3	0,140814	51,01	0,0000
B:VA	0,0000524739	1	0,0000524739	0,02	0,8934
C:ALG	5,07618	3	1,69206	612,95	0,0000
INTERACTIONS					
AB	0,00756839	3	0,0025228	0,91	0,4721
AC	3,85766	9	0,428628	155,27	0,0000
BC	0,00418833	3	0,00139611	0,51	0,6879
RESIDUAL	0,0248445	9	0,0027605		
TOTAL (CORRECTED)	9,39293	31			

All F-ratios are based on the residual mean square error.

Tabla IV-41. ANOVA correspondiente IPSPPEPT del bloque Retraso-Predictiva SMAGA-2-P3_20 (K=0).

Como se puede ver en la Tabla IV-41 disponemos de 40 datos y tenemos 12 grados de libertad residuales. El contraste de hipótesis sobre la potencia (P-value) para un $\alpha=0,05$ implica que son significativos los efectos principales de los factores A y C, y además lo es la interacción AC.

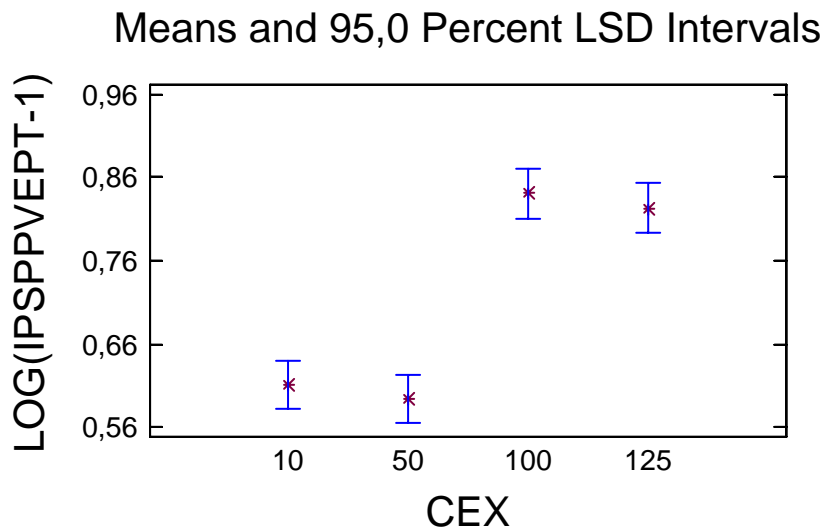


Figura IV-105. Gráfico del efecto principal del factor CEX del bloque Retraso-Predictiva SMAGA-2-P3_20 (K=0).

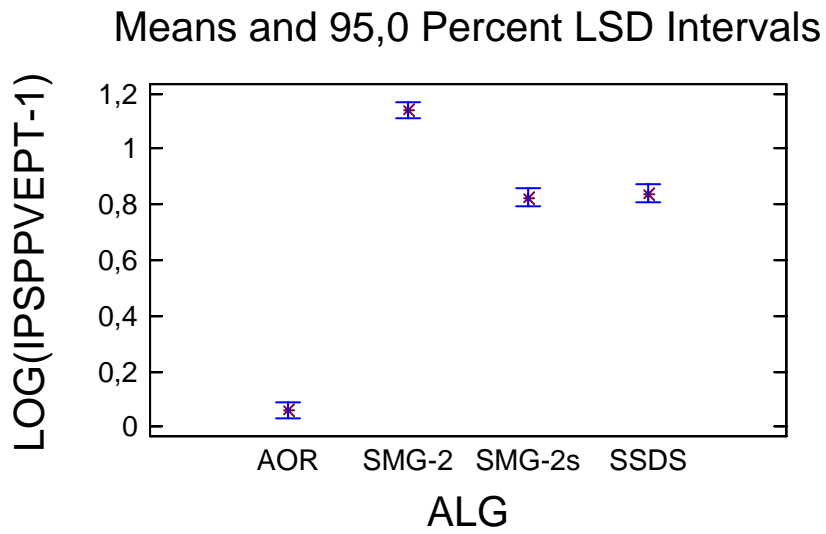


Figura IV-106. Gráfico del efecto principal del factor ALG del bloque Retraso-Predictiva SMAGA-2-P3_20 (K=0).

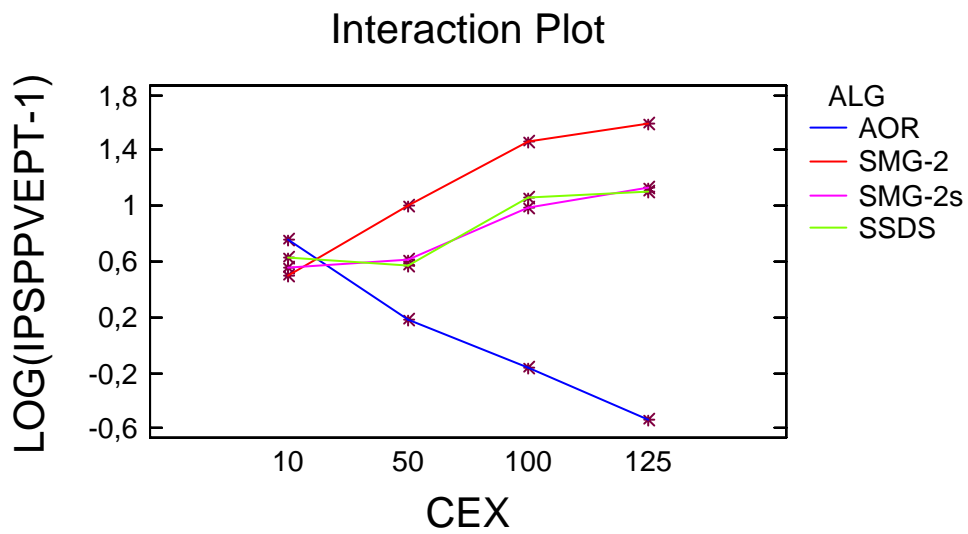


Figura IV-107. Gráfico del efecto de la interacción entre el factor CEX y el factor ALG del bloque Retraso-Predictiva SMAGA-2-P3_20 (K=0).

A-IV.2.5.1.2. Análisis del Bloque P3_20 (Valor de K=0,01)

En este bloque experimental se ha necesitado realizar una transformación de la variable dependiente, y además en esta ocasión se no ha incluido en el análisis el algoritmo RS y SSDS. Ambas decisiones están orientadas a garantizar el cumplimiento de la condición de igualdad en la varianza. La transformación que se ha empleado ha sido el cálculo del logaritmo neperiano de la variable a la cual se le ha restado una unidad. En todo caso, la significación de los factores y los test realizados son tan válidos como si se hubiesen realizado sin transformación. La no inclusión de los algoritmos RS y SSDS en el análisis es factible ya que el primero está dominado por otros para cualquier valor de K, y el segundo ha demostrado no ser interesante para el caso de K=0, por lo que aún será menos interesante cuando el valor de K sea superior.

En relación con la normalidad del experimento se ha obtenido un Chi-Square = 1,99991 con 4 d.f. y un P-Value = 0,572422, lo que descarta la posibilidad de que la variable analizada no proceda de una distribución normal. Así mismo se puede observa en la Figura IV-108 como la variable se distribuye muy próxima a la línea continua diagonal, lo que implica su proximidad a la distribución Normal.

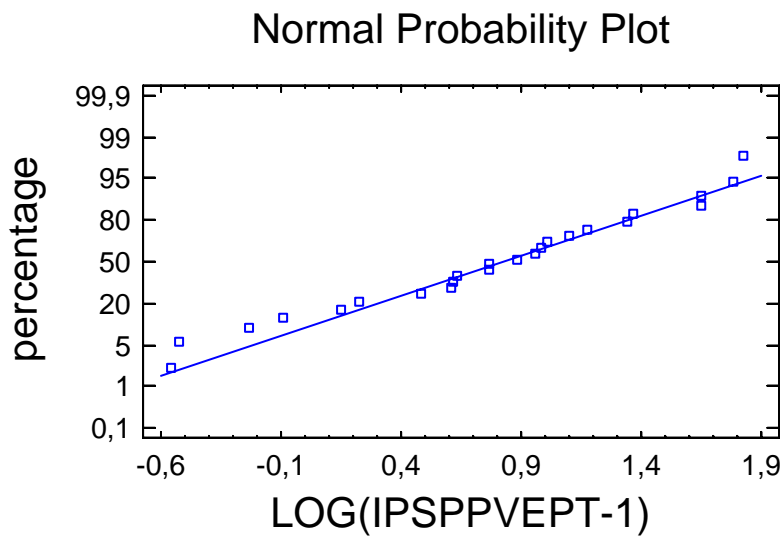


Figura IV-108. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPSPVEPT del bloque Retraso-Predictiva SMAGA-2-P3_20 (K=0,01) en relación a las frecuencias relativas acumuladas.

Por lo que respecta a la homogeneidad de las varianzas se ha realizado el test de Levene para la variable dependiente con respecto a las variables independientes obteniéndose los siguientes valores:

- IPSPVEPT respecto a CEX: test de Levene: 2,15758 P-Value = 0,124874
- IPSPVEPT respecto a VA: test de Levene: 0,00667089 P-Value = 0,935643
- IPSPVEPT respecto a ALG: test de Levene: 1,63396 P-Value = 0,219009

Por lo que se descarta que las varianzas no sean homogéneas.

Por lo que se refiere a la linealidad se empleará el “gráfico de residuos (X_t, \bar{u}_t) y (\hat{Y}_t, \bar{u}_t) ”, que permite disponer los residuos, en el eje de ordenadas, frente a las variables exógenas (X_t) , y endógena (\hat{Y}_t) .

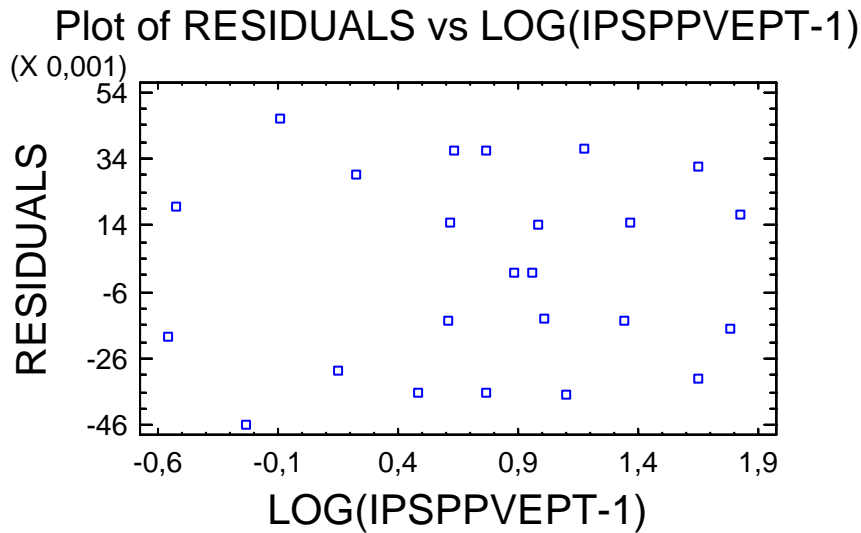


Figura IV-109. Gráfico de Residuos frente a los niveles de la Variable IPSPVEPT del bloque Retraso-Predictiva SMAGA-2-P3_20 (K=0,01).

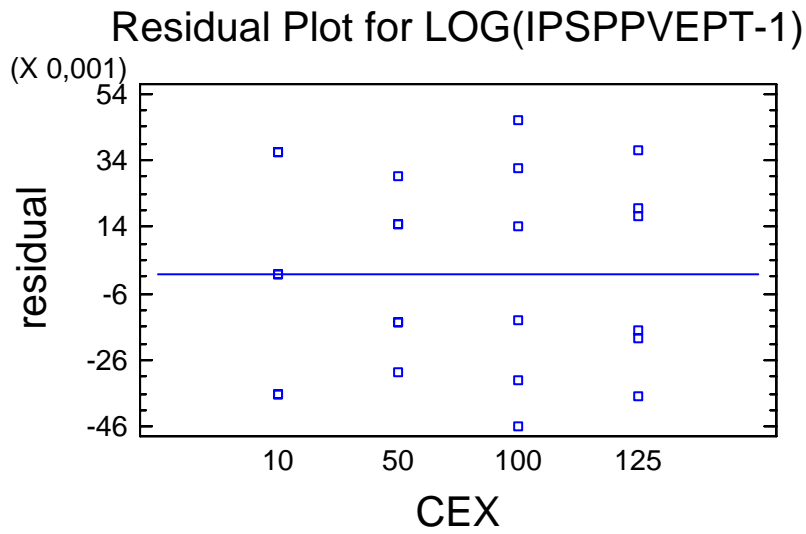


Figura IV-110. Gráfico de Residuos frente a los niveles del factor CEX del bloque Retraso-Predictiva SMAGA-2-P3_20 (K=0,01).

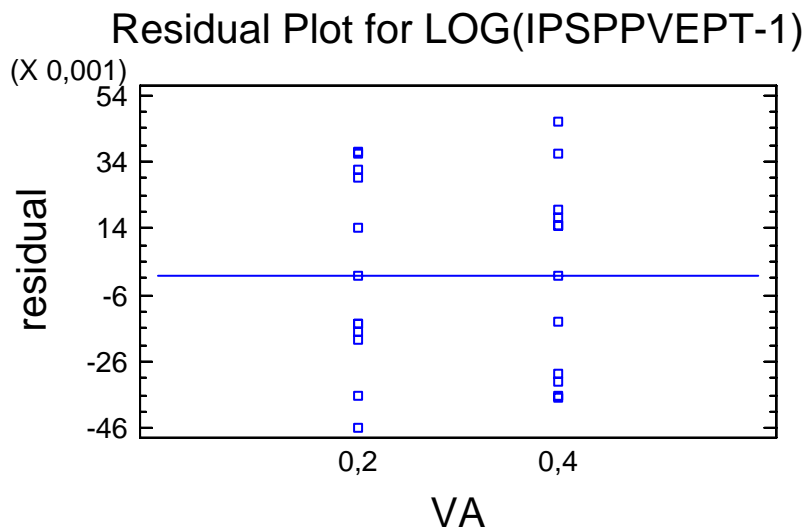


Figura IV-111. Gráfico de Residuos frente a los niveles del factor VA del bloque Retraso-Predictiva SMAGA-2-P3_20 (K=0,01).

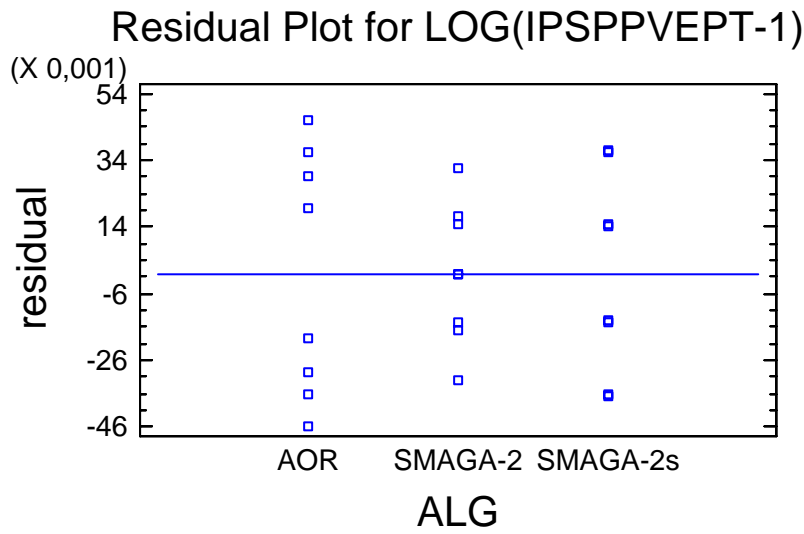


Figura IV-112. Gráfico de Residuos frente a los niveles del factor ALG del bloque Retraso-Predictiva SMAGA-2-P3_20 (K=0,01).

Como se observa en las figuras correspondientes a los residuos no se observa una falta de linealidad significativa en los mismos. Por lo que se descarta la ausencia de linealidad.

La correlación se detecta mediante el gráfico de residuos en relación al orden de ejecución de los tratamientos. Como se observa en la Figura IV-113 no se puede observar ninguna tendencia significativa en la distribución de los mismos por lo que se descarta la existencia de correlación.

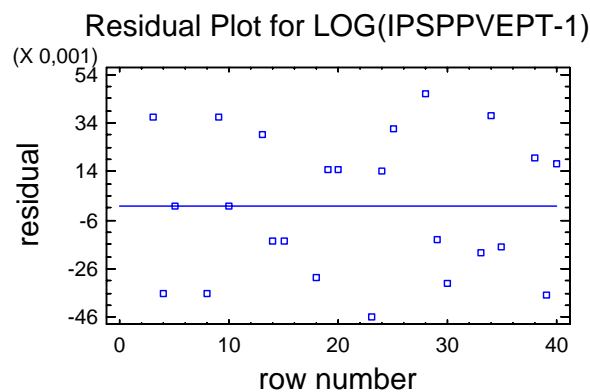


Figura IV-113. Residuos frente al orden de ejecución de los tratamientos del bloque Retraso-Predictiva SMAGA-2-P3_20 (K=0,01).

Una vez verificado el cumplimiento de las condiciones para el análisis ANOVA se obtiene y analiza la misma.

Analysis of Variance for LOG(IPSPPEPT-1) - Type III Sums of Squares

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
MAIN EFFECTS					
A:CEX	0,0446949	3	0,0148983	4,81	0,0489
B:VA	0,00493099	1	0,00493099	1,59	0,2540
C:ALG	7,52282	2	3,76141	1213,82	0,0000
INTERACTIONS					
AB	0,00902408	3	0,00300803	0,97	0,4658
AC	3,1835	6	0,530583	171,22	0,0000
BC	0,000146969	2	0,0000734847	0,02	0,9767
RESIDUAL	0,018593	6	0,00309883		
TOTAL (CORRECTED)	10,7837	23			

All F-ratios are based on the residual mean square error.

Tabla IV-42. ANOVA correspondiente IPSPPEPT del bloque Retraso-Predictiva SMAGA-2-P3_20 (K=0,01).

Como se puede ver en la Tabla IV-42 disponemos de 24 datos y tenemos 7 grados de libertad residuales. El contraste de hipótesis sobre la potencia (P-value) para un $\alpha=0,05$ implica que son significativos los efectos principales de los factores A y C, y además lo es la interacción AC.

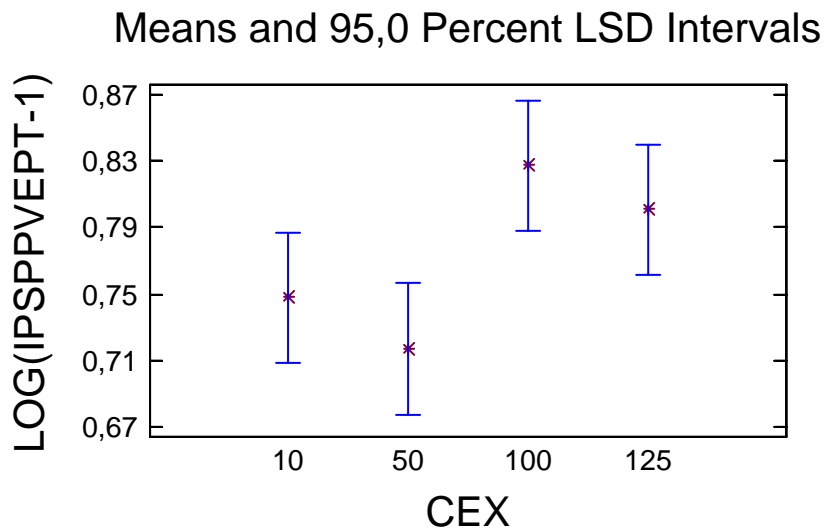


Figura IV-114. Gráfico del efecto principal del factor CEX del bloque Retraso-Predictiva SMAGA-2-P3_20 (K=0,01).

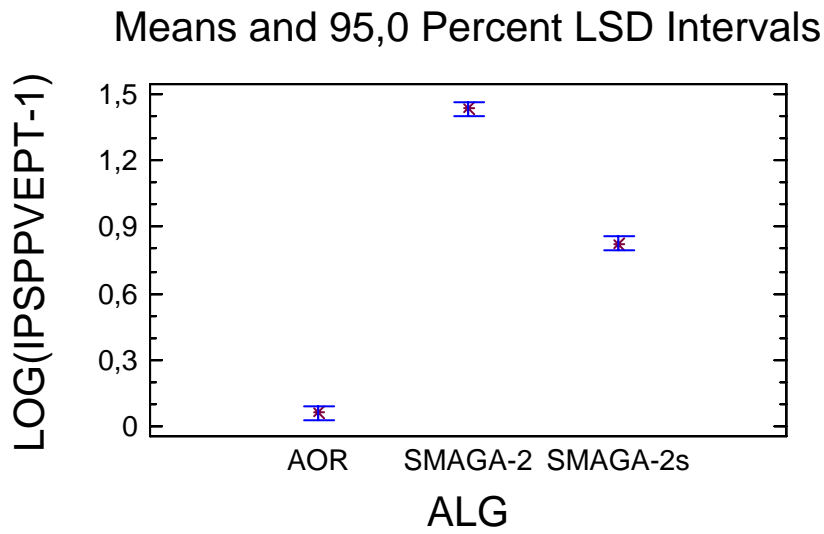


Figura IV-115. Gráfico del efecto principal del factor ALG del bloque Retraso-Predictiva SMAGA-2-P3_20 (K=0,01).

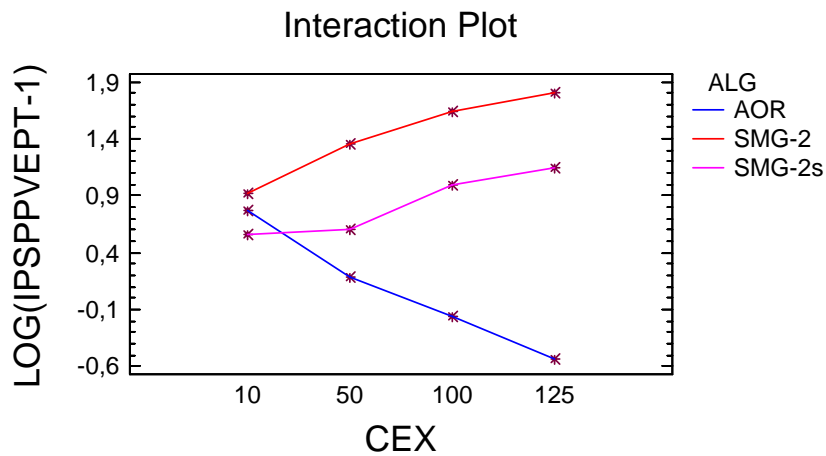


Figura IV-116. Gráfico del efecto de la interacción entre el factor CEX y el factor ALG del bloque Retraso-Predictiva SMAGA-2-P3_20 (K=0,01).

A-IV.2.5.2 Análisis del Bloque P3_50

En este bloque experimental se ha necesitado realizar una transformación de la variable dependiente debido a que se ha observado la existencia de alguna relación no lineal entre los factores. La transformación que se ha empleado ha sido el cálculo de logaritmo neperiano (LOG en la aplicación Statgraphics) de la variable. En todo caso, la significación de los factores y los test realizados son tan válidos como si se hubiesen realizado sin transformación.

Así mismo sólo se ha tenido en cuenta en el estudio de este bloque el caso en el que el factor K toma el valor 0. Esta decisión se basa en el hecho de que los algoritmos dependientes de dicho factor (SSDS y SMSGA-2) están dominados por el resto de algoritmos para el caso en el que toman los mejores valores, es decir con $K=0$, por lo que es obvio que para valores de K superiores aún obtendrán peores resultados en comparación con el resto de algoritmos.

En relación con la normalidad del experimento se ha obtenido un Chi-Square = 8,29992 con 4 d.f. y un P-Value = 0,0811894, lo que descarta la posibilidad de que la variable analizada no proceda de una distribución normal. Así mismo se puede observar en la Figura IV-117 como la variable se distribuye muy próxima a la línea continua diagonal, lo que implica su proximidad a la distribución Normal.

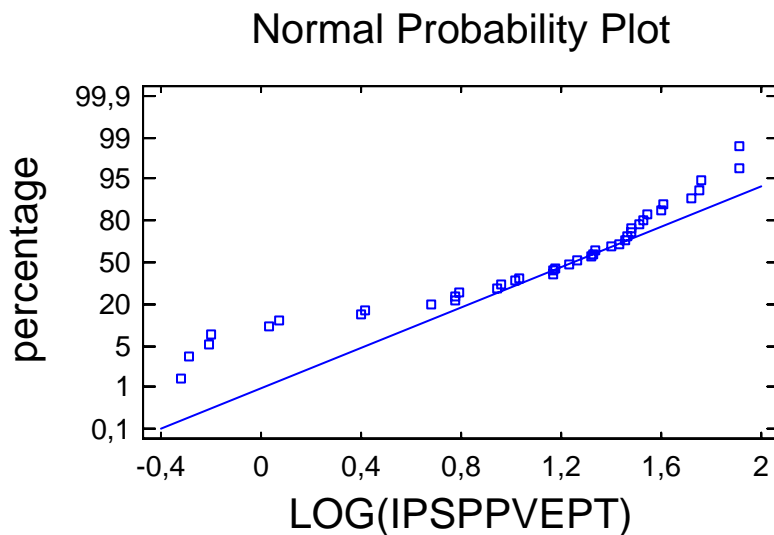


Figura IV-117. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPSPVEPT del bloque Retraso-Predictiva SMAGA-2-P3_50 ($K=0$) en relación a las frecuencias relativas acumuladas.

Por lo que respecta a la homogeneidad de las varianzas se ha realizado el test de Levene para la variable dependiente con respecto a las variables independientes obteniéndose los siguientes valores:

- IPSPVEPT respecto a CEX: test de Levene: 0,564817 P-Value = 0,641737
- IPSPVEPT respecto a VA: test de Levene: 0,0185165 P-Value = 0,89248
- IPSPVEPT respecto a ALG: test de Levene: 1,04674 P-Value = 0,397253

Por lo que se descarta que las varianzas no sean homogéneas.

Por lo que se refiere a la linealidad se empleará el “gráfico de residuos (X_t, \bar{u}_t) y (\hat{Y}_t, \bar{u}_t)”, que permite disponer los residuos, en el eje de ordenadas, frente a las variables exógenas (X_t), y endógena (\hat{Y}_t).

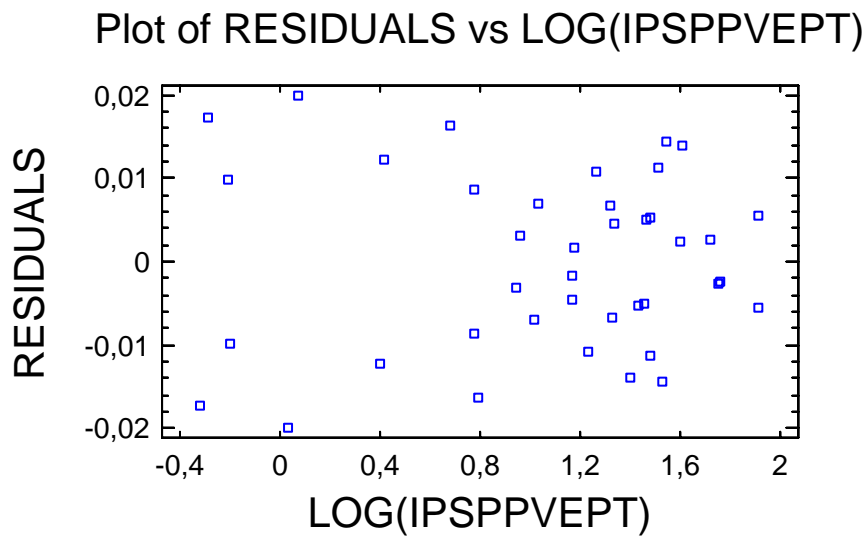


Figura IV-118. Gráfico de Residuos frente a los niveles de la Variable IPSPVEPT del bloque Retraso-Predictiva SMAGA-2-P3_50 ($K=0$).

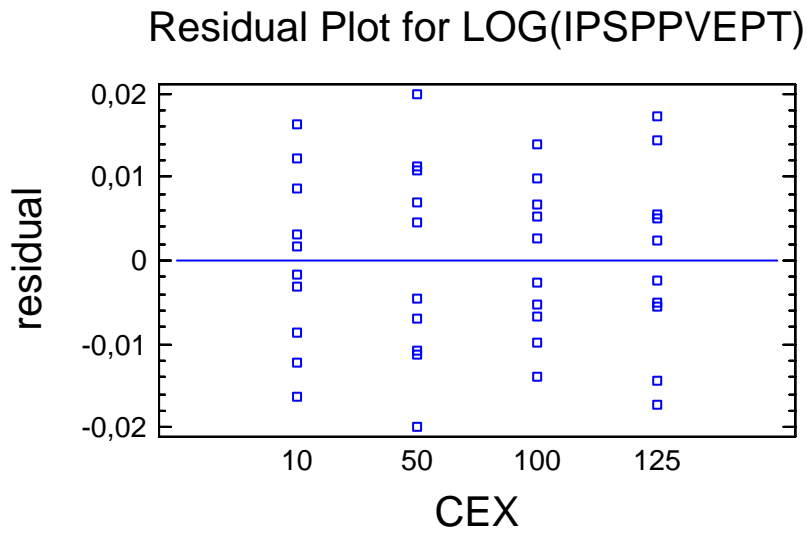


Figura IV-119. Gráfico de Residuos frente a los niveles del factor CEX del bloque Retraso-Predictiva SMAGA-2-P3_50 (K=0).

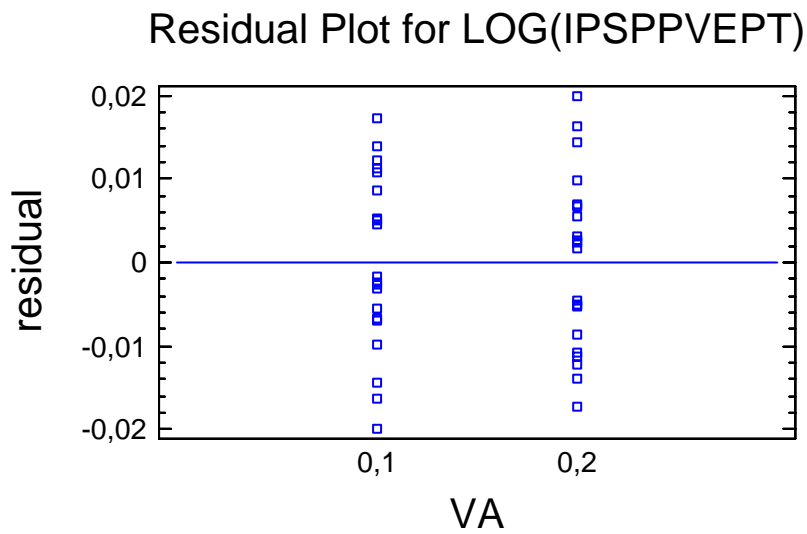


Figura IV-120. Gráfico de Residuos frente a los niveles del factor VA del bloque Retraso-Predictiva SMAGA-2-P3_50 (K=0).

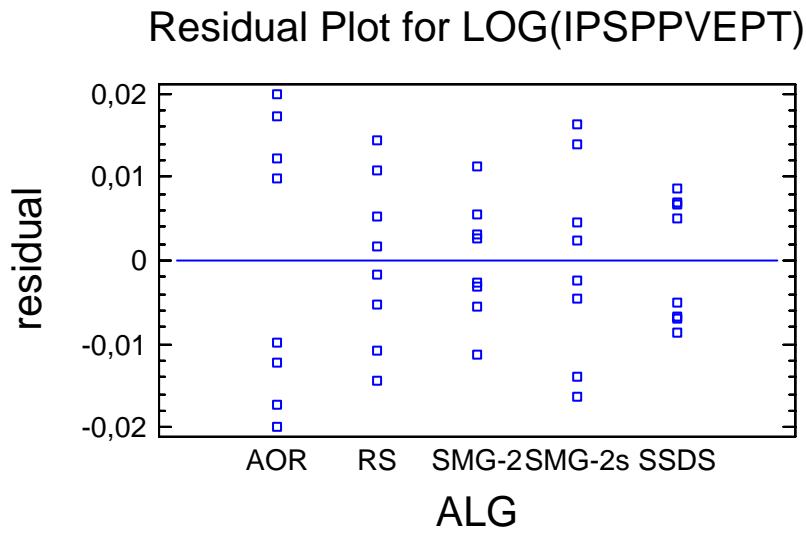


Figura IV-121. Gráfico de Residuos frente a los niveles del factor ALG del bloque Retraso-Predictiva SMAGA-2-P3_50 (K=0).

Como se observa en las figuras correspondientes a los residuos no se observa una falta de linealidad significativa en los mismos. Por lo que se descarta la ausencia de linealidad.

La correlación se detecta mediante el gráfico de residuos en relación al orden de ejecución de los tratamientos. Como se observa en la Figura IV-122 no se puede observar ninguna tendencia significativa en la distribución de los mismos por lo que se descarta la existencia de correlación.

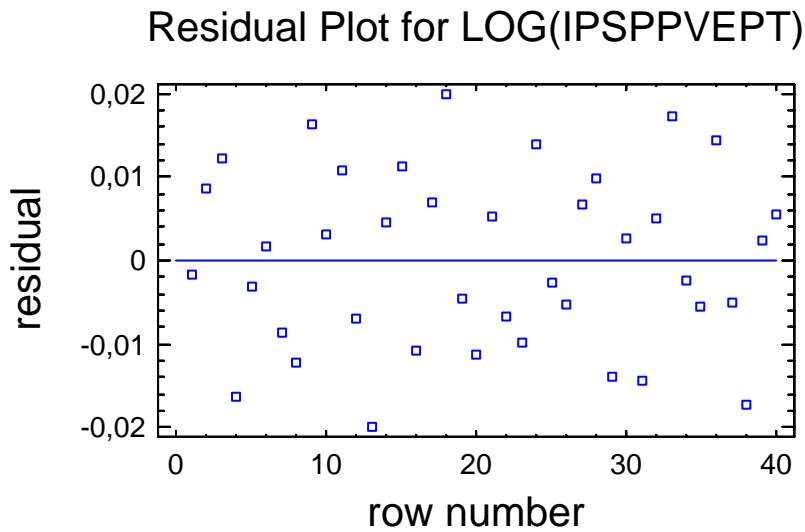


Figura IV-122. Residuos frente al orden de ejecución de los tratamientos del bloque Retraso-Predictiva SMAGA-2-P3_50 (K=0).

Una vez verificado el cumplimiento de las condiciones para el análisis ANOVA se obtiene y analiza la misma.

Analysis of Variance for LOG(IPSPVEPT) - Type III Sums of Squares

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
MAIN EFFECTS					
A:CEX	1,14668	3	0,382227	1067,15	0,0000
B:VA	0,0141687	1	0,0141687	39,56	0,0000
C:ALG	12,0892	4	3,02229	8438,02	0,0000
INTERACTIONS					
AB	0,00200554	3	0,000668512	1,87	0,1891
AC	2,25369	12	0,187808	524,35	0,0000
BC	0,0382065	4	0,00955163	26,67	0,0000
RESIDUAL	0,0042981	12	0,000358175		
TOTAL (CORRECTED)	15,5482	39			

All F-ratios are based on the residual mean square error.

Tabla IV-43. ANOVA correspondiente IPSPVEPT del bloque Retraso-Predictiva SMAGA-2-P3_50 (K=0).

Como se puede ver en la Tabla IV-43 disponemos de 40 datos y tenemos 12 grados de libertad residuales. El contraste de hipótesis sobre la potencia (P-value) para un $\alpha=0,05$ implica que son significativos los efectos principales de los factores A, B y C, y además lo son las interacciones AC y BC.

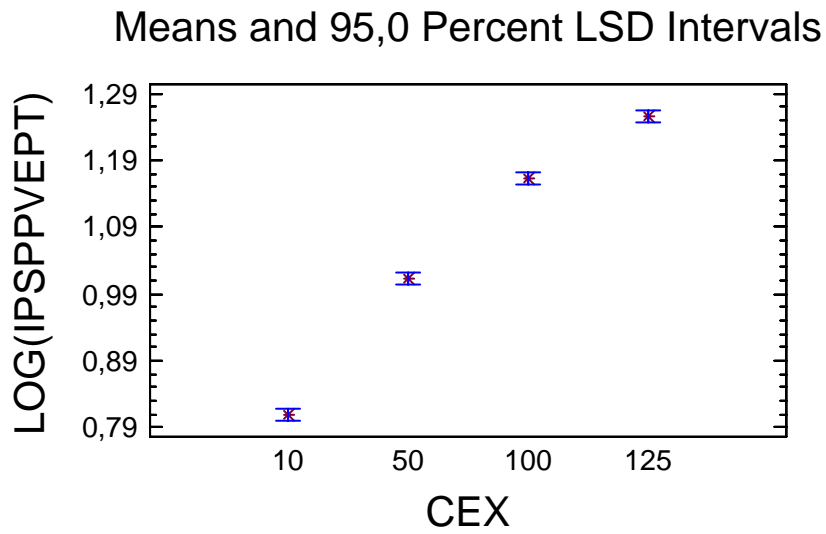


Figura IV-123. Gráfico del efecto principal del factor CEX del bloque Retraso-Predictiva SMAGA-2-P3_50 (K=0).

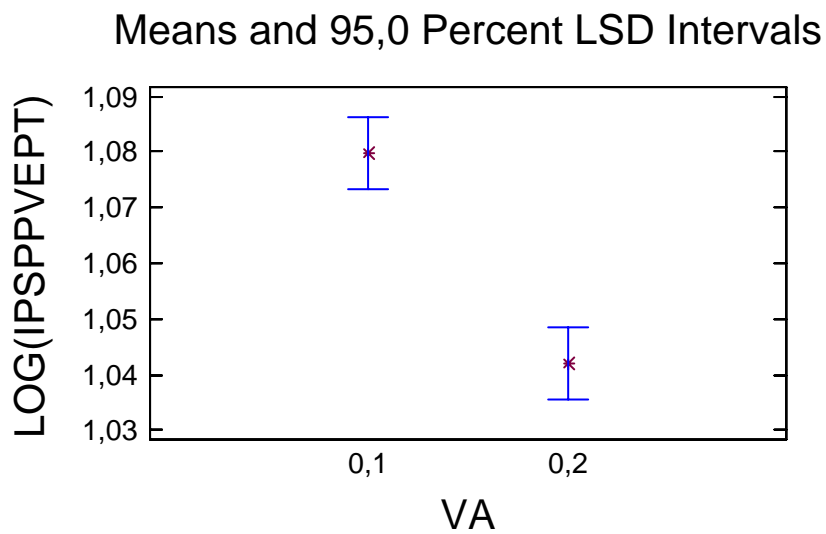


Figura IV-124. Gráfico del efecto principal del factor VA del bloque Retraso-Predictiva SMAGA-2-P3_50 (K=0).

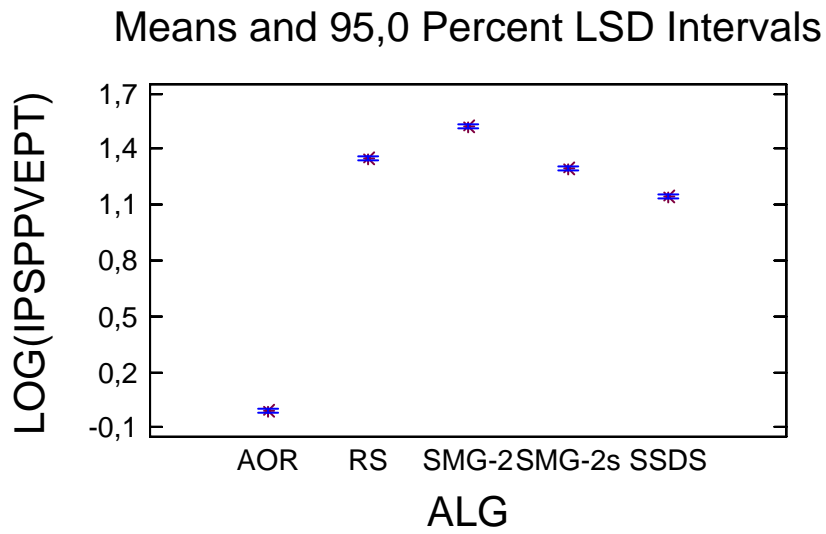


Figura IV-125. Gráfico del efecto principal del factor ALG del bloque Retraso-Predictiva SMAGA-2-P3_50 (K=0).

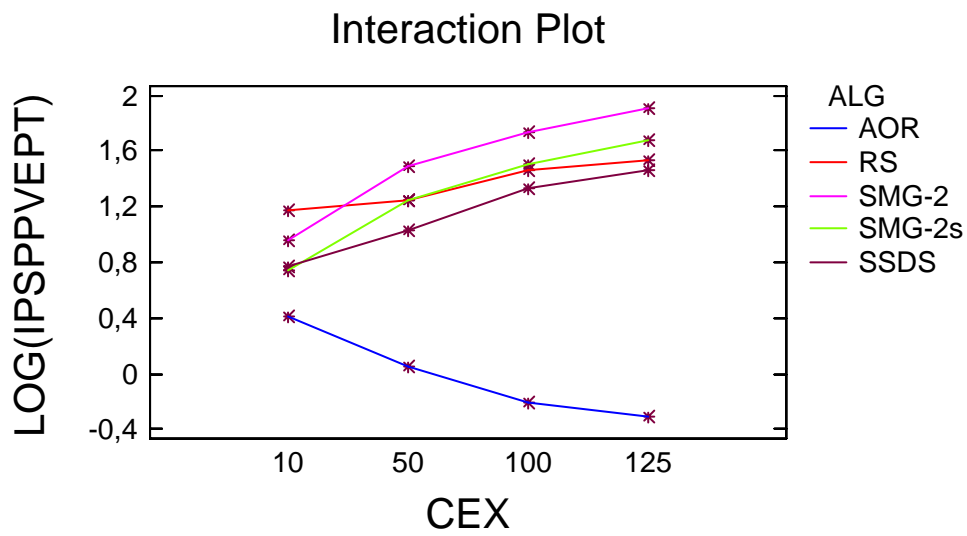


Figura IV-126. Gráfico del efecto de la interacción entre el factor CEX y el factor ALG del bloque Retraso-Predictiva SMAGA-2-P3_50 (K=0).

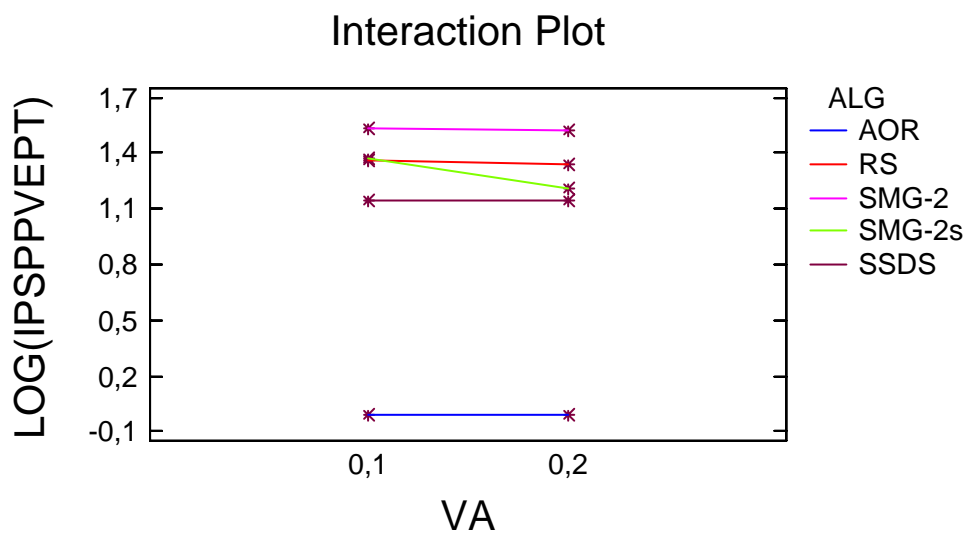


Figura IV-127. Gráfico del efecto de la interacción entre el factor VA y el factor ALG del bloque Retraso-Predictiva SMAGA-2-P3_50 ($K=0$).

**Productos Empleados para la
Simulación**

Anexo



A-IV.1 Productos

Durante la experimentación realizada sobre la plataforma IPSU-MAS se han empleado diversas combinaciones de producto, y en diversos tamaños de lote. En la Tabla IV-1 se muestran todos los productos, clasificados por formato. En el caso del formato 33x33 se han incluido dos familias de producto, una de tipo porcelánico, y otra de tipo gres. El conjunto de productos/referencias seleccionado, y mostrado en la tabla, es una selección representativa de los productos fabricados por la Empresa seleccionada para el taller analizado, a partir de los cuales se han podido generar diferentes conjuntos de trabajos de fabricación. Las órdenes de fabricación propuestas para cada periodo productivo, se ha elaborado intentando configurar una combinación de productos, atendiendo básicamente a su formato, y a su tamaño de lote. Para conseguir dicha mezcla se han empleado las frecuencias relativas a la referencia y al lote. La frecuencia relativa a la referencia establece la probabilidad de que una referencia de un determinado formato sea incluida en un programa productivo. Por su parte, la frecuencia relativa al tamaño de lote indica la probabilidad de que tiene un determinado valor, entre los tamaños normalmente manejados en dicha referencia, sea considerado el volumen a producir.

Formato	Nombre Referencia (frecuencia relativa)	Tamaño Lote Habitual (frecuencia relativa)
20x33	Europa (0.14)	7.000 m ² (0.5), 13.000 m ² (0.5)
20x33	Asia (0.59)	2.000 m ² (0.33), 3.000 m ² (0.33), 9.000 m ² (0.17), 10.000 m ² (0.17)
20x33	África (0.29)	3.000 m ² (0.25), 10.000 m ² (0.25), 13.000 m ² (0.25)
33x33	Blanco (0.5)	3.000 m ² (0.33), 4.000 m ² (0.5), 8.000 m ² (0.17)
33x33	Negro (0.5)	1.500 m ² (0.5), 4.000 m ² (0.5)
40x40	Yin (0.64)	3.000 m ² (0.11), 4.000 m ² (0.11), 5.000 m ² (0.11), 6.000 m ² (0.33), 8.000 m ² (0.17), 10.000 m ² (0.06), 12.000 m ² (0.06), 14.000 m ² (0.06)
40x40	Yan (0.36)	2.000 m ² (0.3), 3.000 m ² (0.2), 3.500 m ² (0.1), 4.000 m ² (0.2), 5.000 m ² (0.2)
25x40	Toledo (0.16)	3.000 m ² (0.18), 3.500 m ² (0.18), 4.000 m ² (0.09), 5.000 m ² (0.18), 7.000 m ² (0.27), 10.000 m ² (0.09)
25x40	Salamanca (0.03)	10.000 m ² (1)
25x40	Segovia (0.09)	4.000 m ² (0.83), 12.000 m ² (0.17)
25x40	Burgos (0.06)	3.000 m ² (0.33), 10.000 m ² (0.67)
25x40	Valladolid (0.09)	5.000 m ² (0.33), 10.000 m ² (0.33), 20.000 m ² (0.34)
25x40	Palencia (0.06)	4.000 m ² (0.5), 5.000 m ² (0.25), 8.000 m ² (0.25)

25x40	Albacete (0.48)	3.000 m ² (0.06), 4.000 m ² (0.11), 6.000 m ² (0.11), 8.000 m ² (0.11), 10.000 m ² (0.28), 12.000 m ² (0.17), 17.000 m ² (0.06), 18.000 m ² (0.11)
25x40	Soria (0.01)	5.000 (1)
25x40	Plasencia (0.03)	5.000 (0.5), 6.000 (0.5)
41x41	Ebro (0.2)	4.000 m ² (0.2), 5.000 m ² (0.2), 7.000 m ² (0.2), 10.000 m ² (0.2), 11.000 m ² (0.2)
41x41	Guadiana (0.65)	3.000 m ² (0.08), 5.000 m ² (0.08), 6.000 m ² (0.17), 8.000 m ² (0.08), 9.000 m ² (0.17), 10.000 m ² (0.08), 13.000 m ² (0.08), 15.000 m ² (0.08), 16.000 m ² (0.08), 18.000 m ² (0.1)
41x41	Tajo (0.07)	4.000 m ² (1)
41x41	Duero (0.08)	3.000 m ² (1)
33x33	Recaredo (0.05)	3.000 m ² (0.2), 4.000 m ² (0.8)
33x33	Liuva (0.05)	2.000 m ² (0.17), 3.000 m ² (0.33), 4.000 m ² (0.33), 5.000 m ² (0.17)
33x33	Witerico (0.08)	3.000 m ² (0.57), 4.000 m ² (0.14), 8.000 m ² (0.14), 9.000 m ² (0.14)
33x33	Gundemaro (0.04)	4.000 m ² (0.5), 11.000 m ² (0.5)
33x33	Sisebuto (0.03)	4.000 m ² (0.33), 5.000 m ² (0.67)
33x33	Suintila (0.07)	4.000 m ² (0.2), 5.000 m ² (0.4), 7.000 m ² (0.4)
33x33	Sisenando (0.07)	4.000 m ² (0.5), 5.000 m ² (0.5)
33x33	Chintila (0.04)	2.000 m ² (0.2), 3.000 m ² (0.4), 4.000 m ² (0.2), 5.000 m ² (0.2)
33x33	Tulga (0.05)	3.000 m ² (0.33), 4.000 m ² (0.67)
33x33	Chindasvinto (0.05)	7.000 m ² (0.67), 8.000 m ² (0.33)
33x33	Recesvinto (0.21)	4.000 m ² (0.17), 5.000 m ² (0.17), 6.000 m ² (0.25), 7.000 m ² (0.08), 8.000 m ² (0.17), 10.000 m ² (0.08), 12.000 m ² (0.08)
33x33	Wamba (0.04)	3.000 m ² (0.5), 5.000 m ² (0.25), 7.000 m ² (0.25)
33x33	Ervigio (0.14)	4.000 m ² (0.17), 5.000 m ² (0.17), 8.000 m ² (0.17), 10.000 m ² (0.17), 15.000 m ² (0.33)
33x33	Égica (0.06)	3.000 m ² (0.2), 4.000 m ² (0.2), 6.000 m ² (0.6)

Tabla IV-1. Productos empleados en la simulación clasificados por formato.

La forma de generar las órdenes de fabricación a partir de los datos de la tabla anterior se explica en el capítulo 9.

**Análisis Estadístico de los Resultados
Obtenidos en la Parametrización de
SMAGA-1 y SMAGA-2 para la
Plataforma IPSU-MAS**

Anexo

VI

A-IV.1 Análisis de los Resultados obtenidos en la Parametrización del Algoritmo SMAGA-1 para la plataforma IPSU-MAS.

En este apartado se analizarán los resultados obtenidos en la experimentación destinada a establecer los valores de los parámetros G_sM , FdM , y FmG del algoritmo SMAGA-2. Por lo tanto, el objetivo es detectar si dichos valores son significativos y seleccionar el valor más adecuado para cada uno de ellos.

Tanto en este caso, para el algoritmo SMAGA-1, como en el siguiente apartado, para el algoritmo SMAGA-2, se ha considerado el valor de los parámetros será el mismo para cualquier escenario con la misma configuración de Margen Operativo. Se considera por lo tanto que el valor que se seleccione debe ser adecuado en promedio para los 40 escenarios de cada configuración de Margen Operativo. Con ello se busca la máxima robustez del algoritmo.

A-IV.1.1.1 Análisis de los Conjuntos Experimentales con MO del 5%.

En relación con la normalidad del experimento se ha obtenido un Chi-Square = 2,00004 with 3 d.f. y un P-Value = 0,572394, lo que descarta la posibilidad de que la variable analizada no proceda de una distribución normal. Así mismo se puede observar en la Figura IV-1 como la variable se distribuye muy próxima a la línea continua diagonal, lo que implica su proximidad a la distribución Normal.

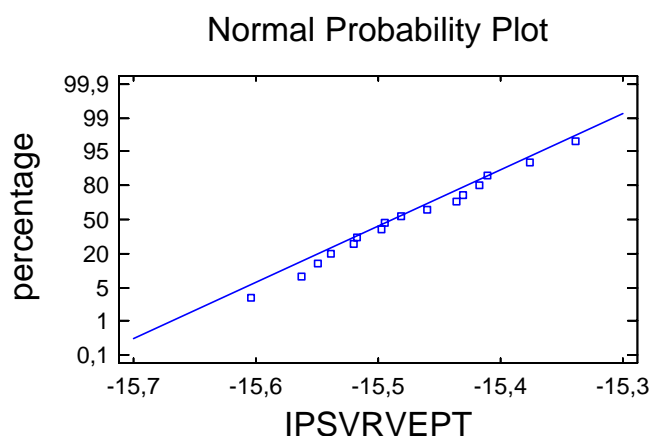


Figura IV-1. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPSPVRVEPT del promedio de Conjuntos Experimentales con Margen Operativo del 5% en relación a las frecuencias relativas acumuladas.

Por lo que respecta a la homogeneidad de las varianzas se ha realizado el test de Levene para la variable dependiente con respecto a las variables independientes obteniéndose los siguientes valores:

- IPSPVVEPT respecto a GsM: test de Levene: 0,208688 P-Value = 0,654805
- IPSPVVEPT respecto a FdM: test de Levene: 0,718971 P-Value = 0,410745
- IPSPVVEPT respecto a FmG: test de Levene: 0,293036 P-Value = 0,596791

Por lo que se descarta que las varianzas no sean homogéneas.

Por lo que se refiere a la linealidad se empleará el “gráfico de residuos (X_t, \bar{u}_t) y (\hat{Y}_t, \bar{u}_t)”, que permite disponer los residuos, en el eje de ordenadas, frente a las variables exógenas (X_t), y endógena (\hat{Y}_t).

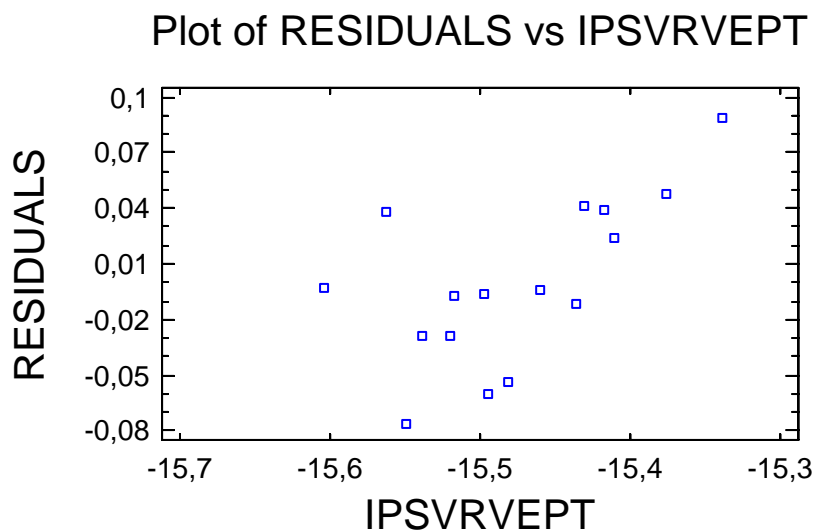


Figura IV-2. Gráfico de Residuos frente a los niveles de la Variable IPSVRVEPT del promedio de Conjuntos Experimentales con Margen Operativo del 5%.

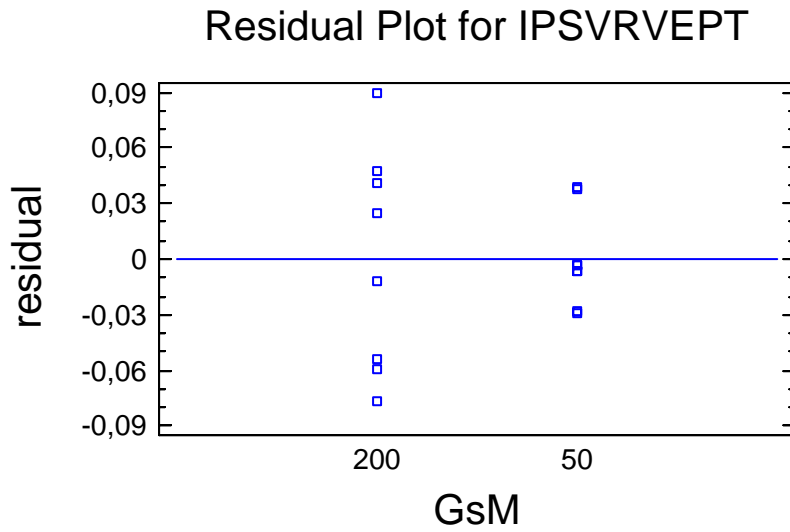


Figura IV-3. Gráfico de Residuos frente a los niveles del factor GsM del promedio de Conjuntos Experimentales con Margen Operativo del 5%.

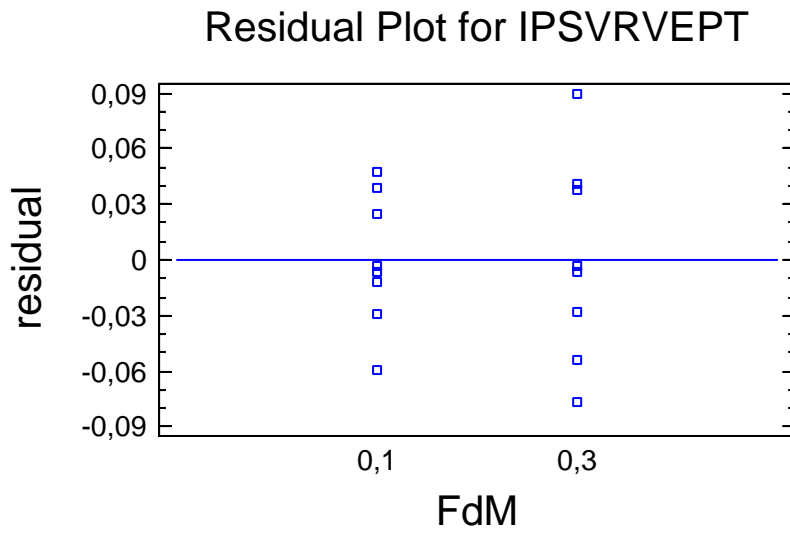


Figura IV-4. Gráfico de Residuos frente a los niveles del factor FdM del promedio de Conjuntos Experimentales con Margen Operativo del 5%.

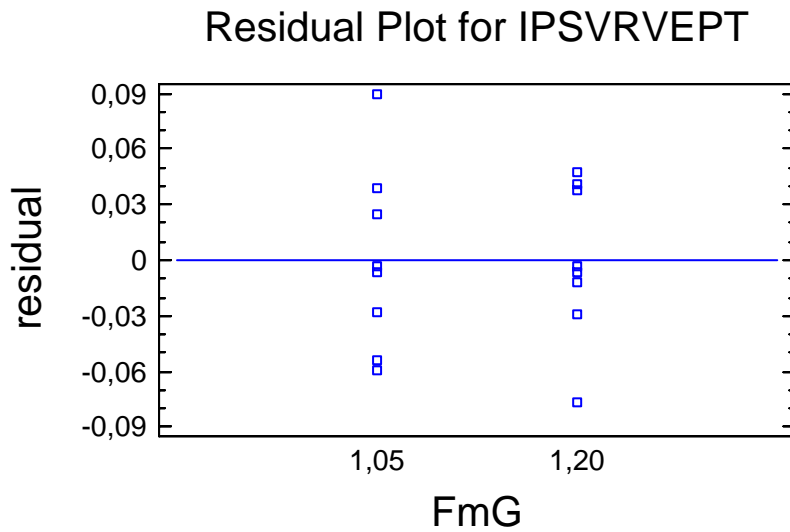


Figura IV-5. Gráfico de Residuos frente a los niveles del factor FmG del promedio de Conjuntos Experimentales con Margen Operativo del 5%.

En las figuras correspondientes a los residuos no se observa una falta de linealidad significativa en los mismos. Por lo que se descarta la ausencia de linealidad.

La correlación se detecta mediante el gráfico de residuos en relación al orden de ejecución de los tratamientos. Como se observa en la Figura IV-6 no se puede observar ninguna tendencia significativa en la distribución de los mismos por lo que se descarta la existencia de correlación.

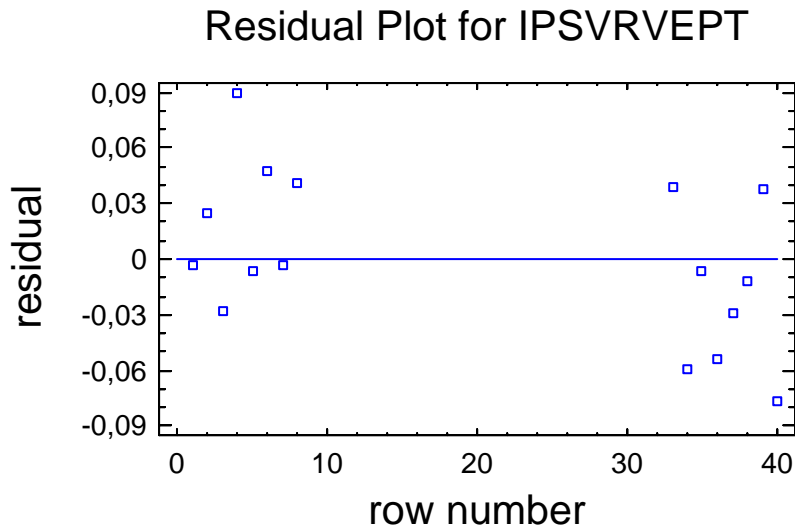


Figura IV-6. Residuos frente al orden de ejecución de los tratamientos de la variable IPSVRVEPT del promedio de Conjuntos Experimentales con Margen Operativo del 5%.

Una vez verificado el cumplimiento de las condiciones para el análisis ANOVA se obtiene y analiza la misma.

Analysis of Variance for IPSVRVEPT - Type III Sums of Squares

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
MAIN EFFECTS					
A:GsM	0,0227063	1	0,0227063	6,87	0,0277
B:FdM	0,0105465	1	0,0105465	3,19	0,1076
C:FmG	0,00630674	1	0,00630674	1,91	0,2004
INTERACTIONS					
AB	0,00384958	1	0,00384958	1,17	0,3085
AC	0,00204903	1	0,00204903	0,62	0,4512
BC	0,00314707	1	0,00314707	0,95	0,3546
RESIDUAL	0,0297353	9	0,00330392		
TOTAL (CORRECTED)	0,0783405	15			

All F-ratios are based on the residual mean square error.

Tabla IV-1. ANOVA correspondiente IPSVRVEPT del promedio de Conjuntos Experimentales con Margen Operativo del 5%.

Como se puede ver en la Tabla IV-1 disponemos de 16 datos y tenemos 9 grados de libertad residuales. El contraste de hipótesis sobre la potencia (P-value) para un $\alpha=0,05$ implica que sólo es significativo el efecto principales del factor GsM, mientras que ninguna de las interacciones entre factores es significativa.

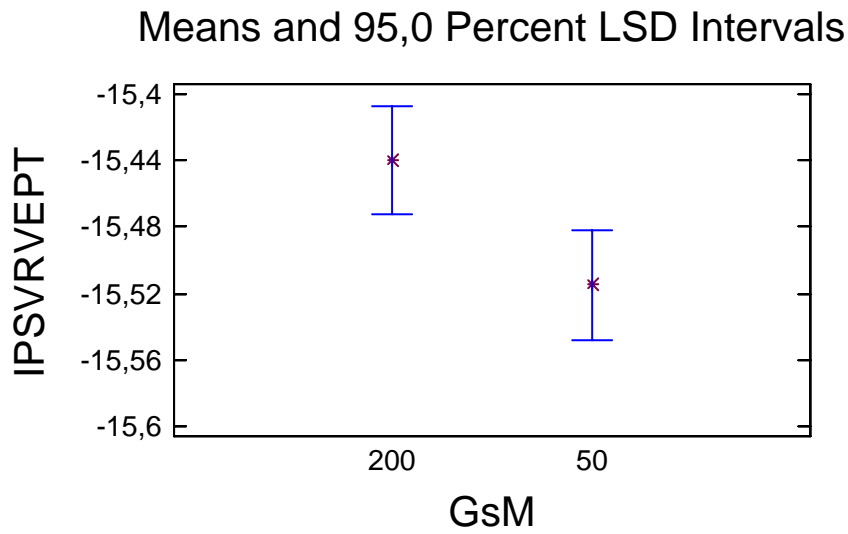


Figura IV-7. Gráfico del efecto principal del factor GsM del promedio de Conjuntos Experimentales con Margen Operativo del 5%.

A-IV.1.1.2 Análisis de los Conjunto Experimentales con MO del 10%.

En relación con la normalidad del experimento se ha obtenido un Chi-Square = 6,50007 with 3 d.f. y un P-Value = 0,0896578, lo que descarta la posibilidad de que la variable analizada no proceda de una distribución normal. Así mismo se puede observa en la Figura IV-8 como la variable se distribuye muy próxima a la línea continua diagonal, lo que implica su proximidad a la distribución Normal.

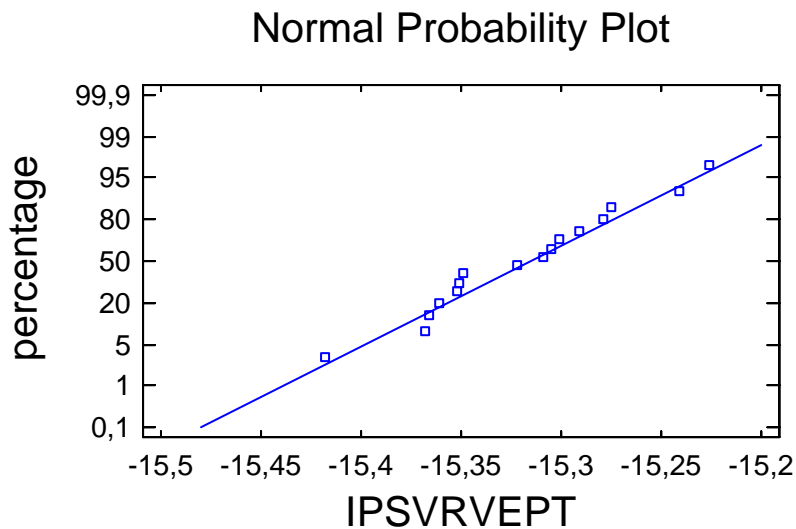


Figura IV-8. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPSPVRVEPT del promedio de Conjuntos Experimentales con Margen Operativo del 10% en relación a las frecuencias relativas acumuladas.

Por lo que respecta a la homogeneidad de las varianzas se ha realizado el test de Levene para la variable dependiente con respecto a las variables independientes obteniéndose los siguientes valores:

- IPSPVVEPT respecto a GsM: test de Levene: 0,14504 P-Value = 0,709038
- IPSPVVEPT respecto a FdM: test de Levene: 0,0916233 P-Value = 0,766574
- IPSPVVEPT respecto a FmG: test de Levene: 0,444372 P-Value = 0,515856

Por lo que se descarta que las varianzas no sean homogéneas.

Por lo que se refiere a la linealidad se empleará el “gráfico de residuos (X_t, \bar{u}_t) y (\hat{Y}_t, \bar{u}_t)”, que permite disponer los residuos, en el eje de ordenadas, frente a las variables exógenas (X_t), y endógena (\hat{Y}_t).

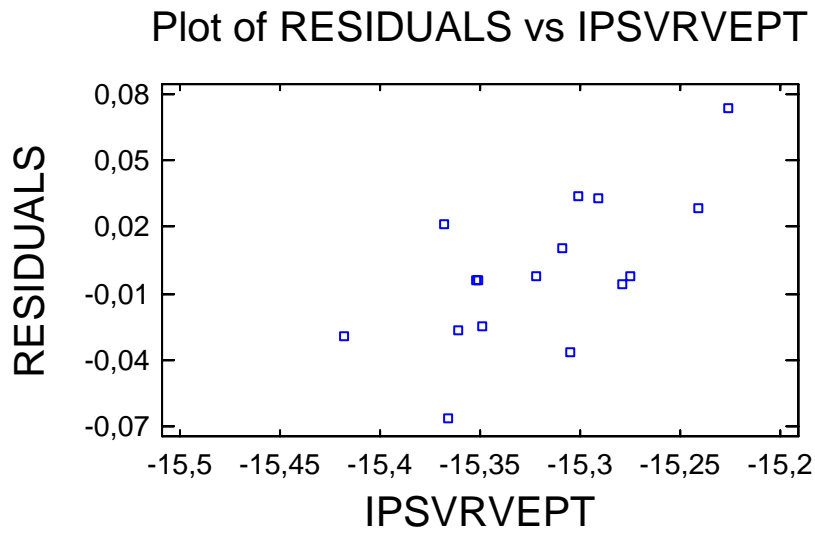


Figura IV-9. Gráfico de Residuos frente a los niveles de la Variable IPSVRVEPT del promedio de Conjuntos Experimentales con Margen Operativo del 10%.

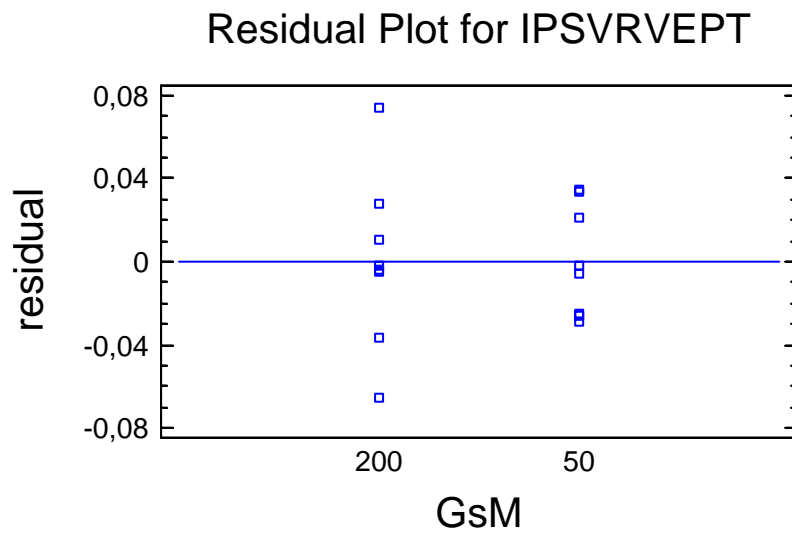


Figura IV-10. Gráfico de Residuos frente a los niveles del factor GsM del promedio de Conjuntos Experimentales con Margen Operativo del 10%.

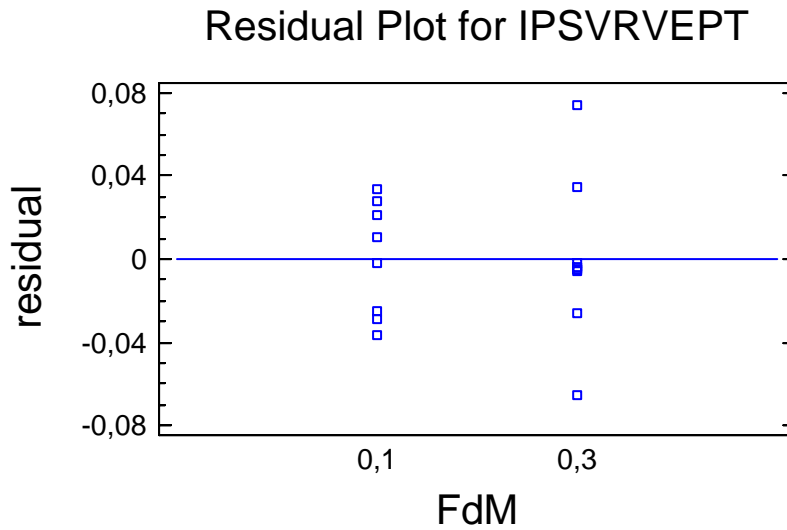


Figura IV-11. Gráfico de Residuos frente a los niveles del factor FdM del promedio de Conjuntos Experimentales con Margen Operativo del 10%.

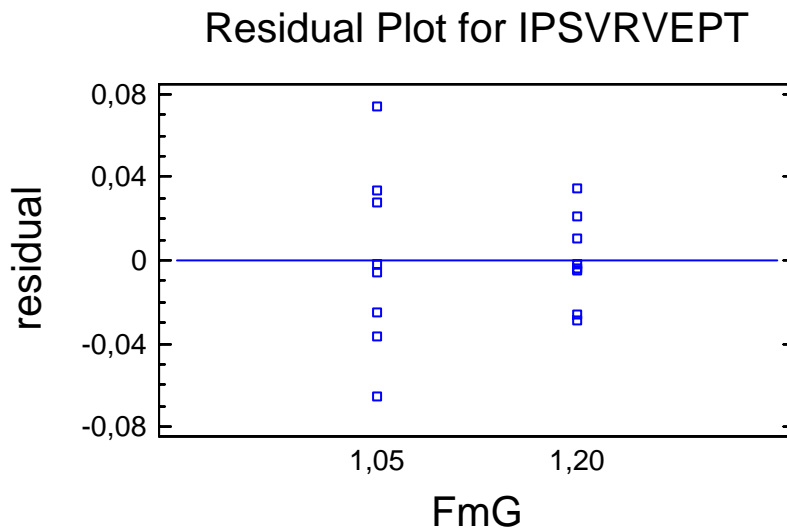


Figura IV-12. Gráfico de Residuos frente a los niveles del factor FmG del promedio de Conjuntos Experimentales con Margen Operativo del 10%.

En las figuras correspondientes a los residuos no se observa una falta de linealidad significativa en los mismos. Por lo que se descarta la ausencia de linealidad.

La correlación se detecta mediante el gráfico de residuos en relación al orden de ejecución de los tratamientos. Como se observa en la Figura IV-13 no se puede observar ninguna tendencia

significativa en la distribución de los mismos por lo que se descarta la existencia de correlación.

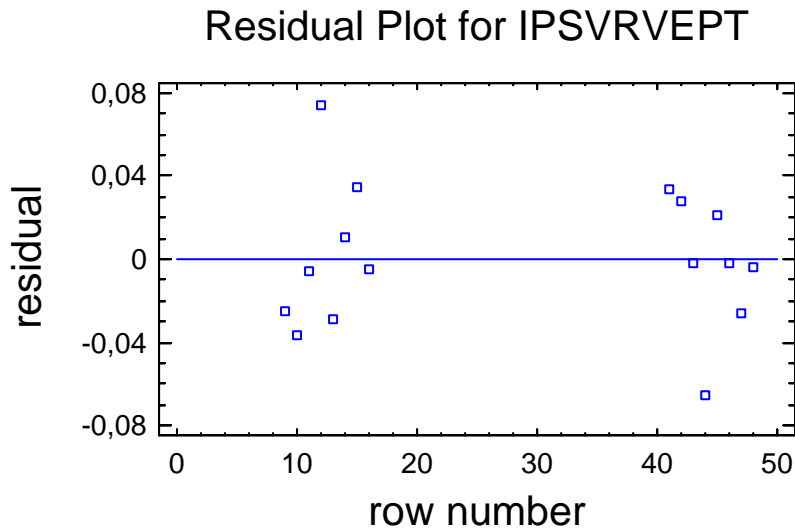


Figura IV-13. Residuos frente al orden de ejecución de los tratamientos de la variable IPSVRVEPT del promedio de Conjuntos Experimentales con Margen Operativo del 10%.

Una vez verificado el cumplimiento de las condiciones para el análisis ANOVA se obtiene y analiza la misma.

Analysis of Variance for IPSVRVEPT - Type III Sums of Squares

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
MAIN EFFECTS					
A:GsM	0,00186057	1	0,00186057	0,99	0,3452
B:FdM	0,000517591	1	0,000517591	0,28	0,6120
C:FmG	0,0127607	1	0,0127607	6,81	0,0283
INTERACTIONS					
AB	0,00676352	1	0,00676352	3,61	0,0900
AC	0,000212157	1	0,000212157	0,11	0,7443
BC	0,00000847902	1	0,00000847902	0,00	0,9479
RESIDUAL	0,0168731	9	0,00187479		
TOTAL (CORRECTED)	0,0389961	15			

All F-ratios are based on the residual mean square error.

Tabla IV-2. ANOVA correspondiente IPSVRVEPT del promedio de Conjuntos Experimentales con Margen Operativo del 10%.

Como se puede ver en la Tabla IV-2 disponemos de 16 datos y tenemos 9 grados de libertad residuales. El contraste de hipótesis sobre la potencia (P-value) para un $\alpha=0,05$ implica que sólo es significativo el efecto principales del factor FmG, mientras que ninguna de las interacciones entre factores es significativa.

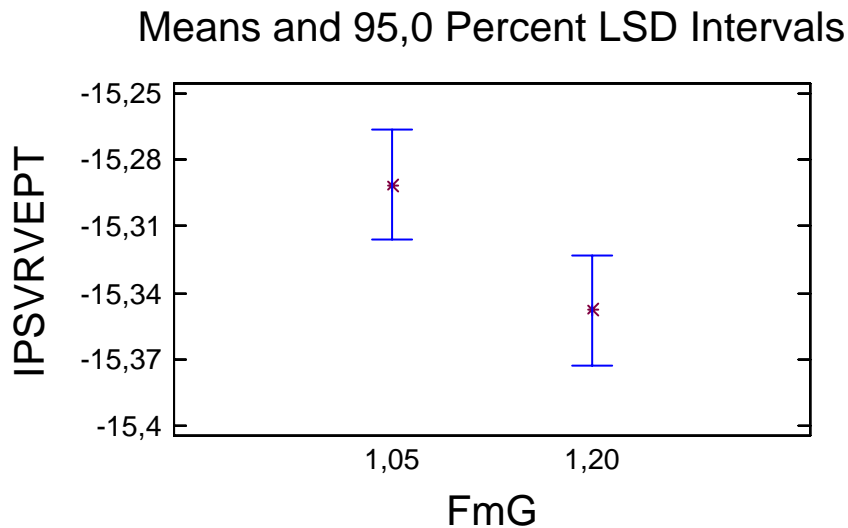


Figura IV-14. Gráfico del efecto principal del factor FmG del promedio de Conjuntos Experimentales con Margen Operativo del 10%.

A-IV.1.1.3 Análisis de los Conjunto Experimentales con MO del 15%.

En relación con la normalidad del experimento se ha obtenido un Chi-Square = 0,500053 with 3 d.f. y un P-Value = 0,918879, lo que descarta la posibilidad de que la variable analizada no proceda de una distribución normal. Así mismo se puede observa en la Figura IV-15 como la variable se distribuye muy próxima a la línea continua diagonal, lo que implica su proximidad a la distribución Normal.

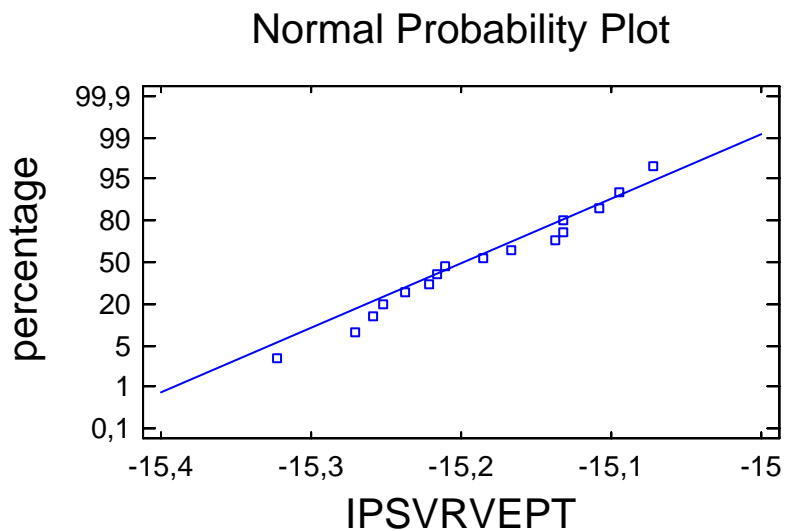


Figura IV-15. Gráfico de probabilidad normal (Y_i, F_i) para la variable $IPSPVREPT$ del promedio de Conjuntos Experimentales con Margen Operativo del 15% en relación a las frecuencias relativas acumuladas.

Por lo que respecta a la homogeneidad de las varianzas se ha realizado el test de Levene para la variable dependiente con respecto a las variables independientes obteniéndose los siguientes valores:

- $IPSPVREPT$ respecto a GsM: test de Levene: 0,176151 P-Value = 0,681072
- $IPSPVREPT$ respecto a FdM: test de Levene: 0,103585 P-Value = 0,752324
- $IPSPVREPT$ respecto a FmG: test de Levene: 0,103585 P-Value = 0,752324

Por lo que se descarta que las varianzas no sean homogéneas.

Por lo que se refiere a la linealidad se empleará el “gráfico de residuos (X_t, \bar{u}_t) y (\hat{Y}_t, \bar{u}_t) ”, que permite disponer los residuos, en el eje de ordenadas, frente a las variables exógenas (X_t) , y endógena (\hat{Y}_t) .

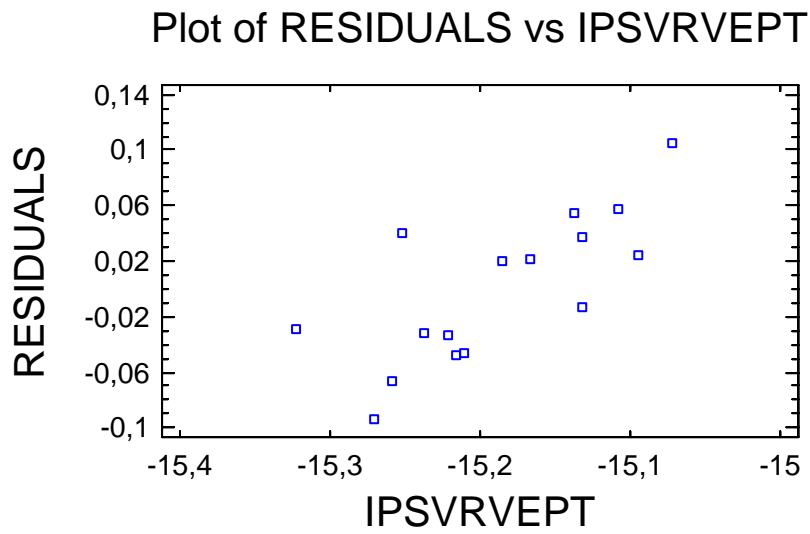


Figura IV-16. Gráfico de Residuos frente a los niveles de la Variable IPSVRVEPT del promedio de Conjuntos Experimentales con Margen Operativo del 15%.

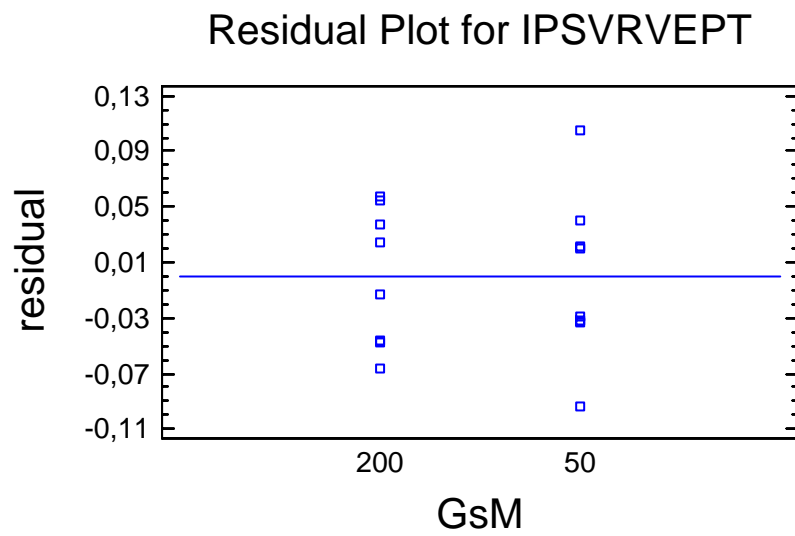


Figura IV-17. Gráfico de Residuos frente a los niveles del factor GsM del promedio de Conjuntos Experimentales con Margen Operativo del 15%.

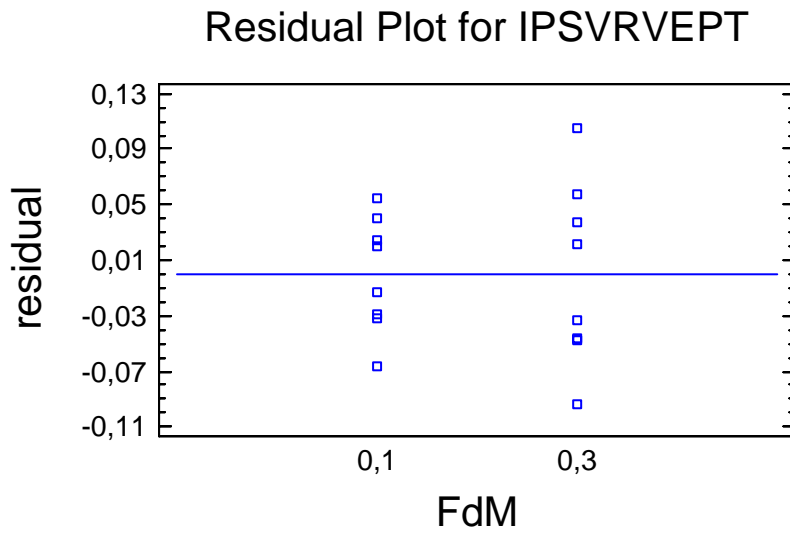


Figura IV-18. Gráfico de Residuos frente a los niveles del factor FdM del promedio de Conjuntos Experimentales con Margen Operativo del 15%.

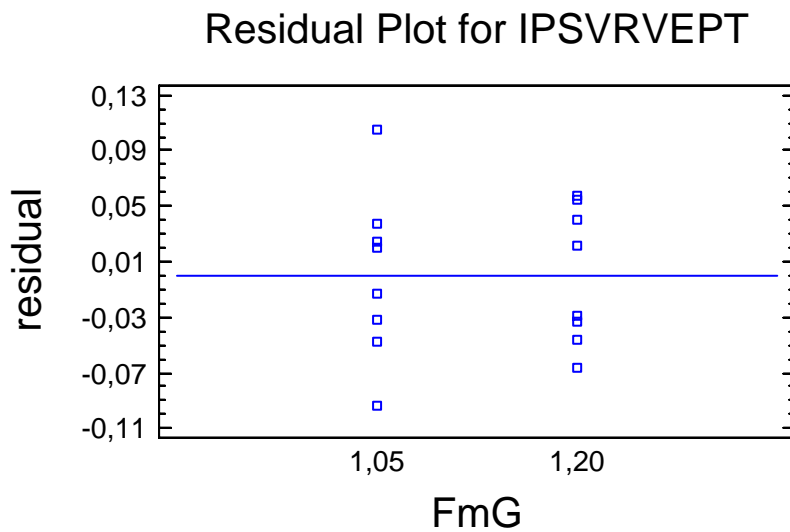


Figura IV-19. Gráfico de Residuos frente a los niveles del factor FmG del promedio de Conjuntos Experimentales con Margen Operativo del 15%.

En las figuras correspondientes a los residuos no se observa una falta de linealidad significativa en los mismos. Por lo que se descarta la ausencia de linealidad.

La correlación se detecta mediante el gráfico de residuos en relación al orden de ejecución de los tratamientos. Como se observa en la Figura IV-20 no se puede observar ninguna tendencia significativa en la distribución de los mismos por lo que se descarta la existencia de correlación.

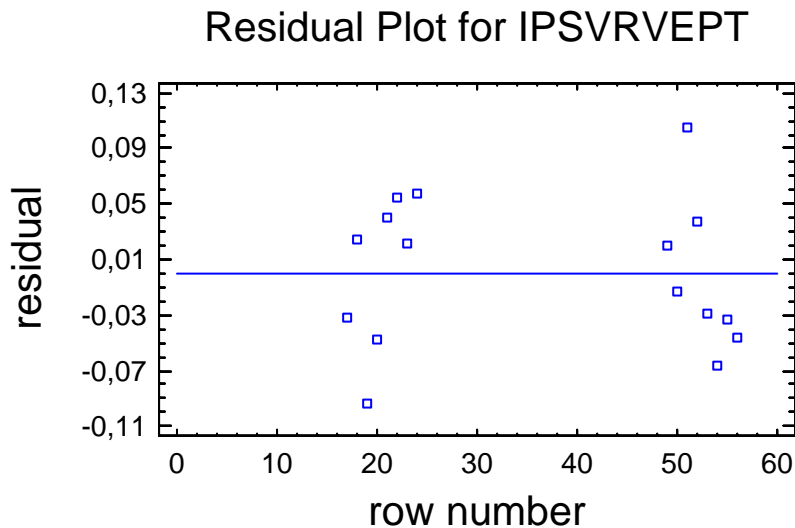


Figura IV-20. Residuos frente al orden de ejecución de los tratamientos de la variable IPSVRVEPT del promedio de Conjuntos Experimentales con Margen Operativo del 15%.

Una vez verificado el cumplimiento de las condiciones para el análisis ANOVA se obtiene y analiza la misma.

Analysis of Variance for IPSVRVEPT - Type III Sums of Squares

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
MAIN EFFECTS					
A:GsM	0,0120501	1	0,0120501	2,56	0,1441
B:FdM	0,00308407	1	0,00308407	0,66	0,4392
C:FmG	0,00697319	1	0,00697319	1,48	0,2545
INTERACTIONS					
AB	0,00601177	1	0,00601177	1,28	0,2877
AC	0,000204115	1	0,000204115	0,04	0,8397
BC	0,00583209	1	0,00583209	1,24	0,2945
RESIDUAL	0,0423698	9	0,00470776		
TOTAL (CORRECTED)	0,0765252	15			

All F-ratios are based on the residual mean square error.

Tabla IV-3. ANOVA correspondiente IPSVRVEPT del promedio de Conjuntos Experimentales con Margen Operativo del 15%.

Como se puede ver en la Tabla IV-3 disponemos de 16 datos y tenemos 9 grados de libertad residuales. El contraste de hipótesis sobre la potencia (P-value) para un $\alpha=0,05$ implica que no es significativo ningún efecto principal así como tampoco lo es ninguna de las interacciones entre factores.

A-IV.1.1.4 Análisis de los Conjunto Experimentales con MO del 20%.

En relación con la normalidad del experimento se ha obtenido un Chi-Square = 2,00004 with 3 d.f. y un P-Value = 0,572394 lo que descarta la posibilidad de que la variable analizada no proceda de una distribución normal. Así mismo se puede observar en la Figura IV-21 como la variable se distribuye muy próxima a la línea continua diagonal, lo que implica su proximidad a la distribución Normal.

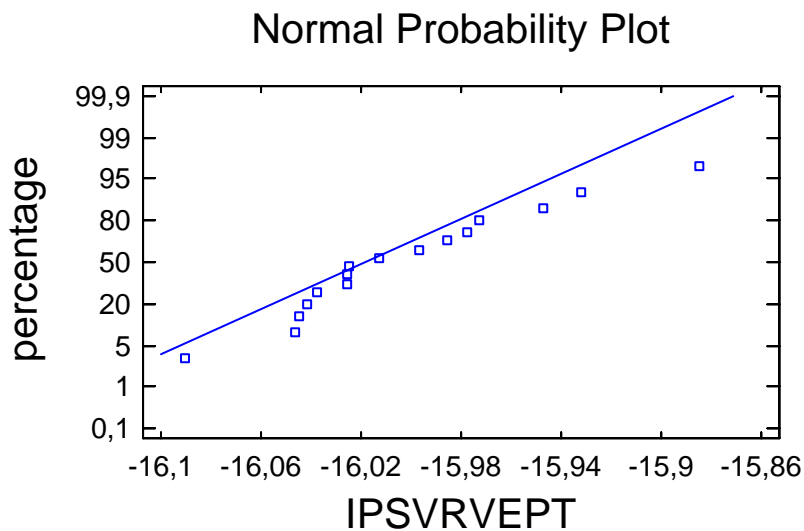


Figura IV-21. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPSPVREPT del promedio de Conjuntos Experimentales con Margen Operativo del 20% en relación a las frecuencias relativas acumuladas.

Por lo que respecta a la homogeneidad de las varianzas se ha realizado el test de Levene para la variable dependiente con respecto a las variables independientes obteniéndose los siguientes valores:

- IPSPVREPT respecto a GsM: test de Levene: 0,146057 P-Value = 0,708071
- IPSPVREPT respecto a FdM: test de Levene: 0,121214 P-Value = 0,732902

- IPSPVVEPT respecto a FmG: test de Levene: 2,53908 P-Value = 0,13338

Por lo que se descarta que las varianzas no sean homogéneas.

Por lo que se refiere a la linealidad se empleará el “gráfico de residuos (X_t, \bar{u}_t) y (\hat{Y}_t, \bar{u}_t)”, que permite disponer los residuos, en el eje de ordenadas, frente a las variables exógenas (X_t), y endógena (\hat{Y}_t).

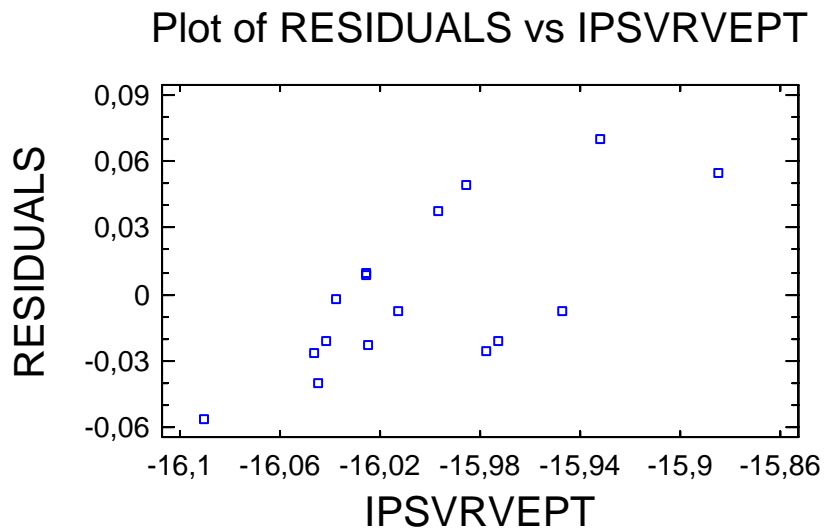


Figura IV-22. Gráfico de Residuos frente a los niveles de la Variable IPSPVVEPT del promedio de Conjuntos Experimentales con Margen Operativo del 20%.

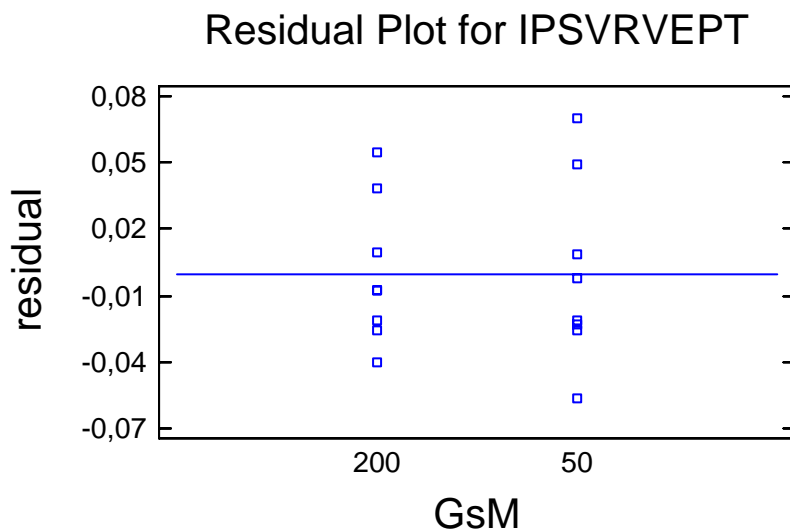


Figura IV-23. Gráfico de Residuos frente a los niveles del factor GsM del promedio de Conjuntos Experimentales con Margen Operativo del 20%.

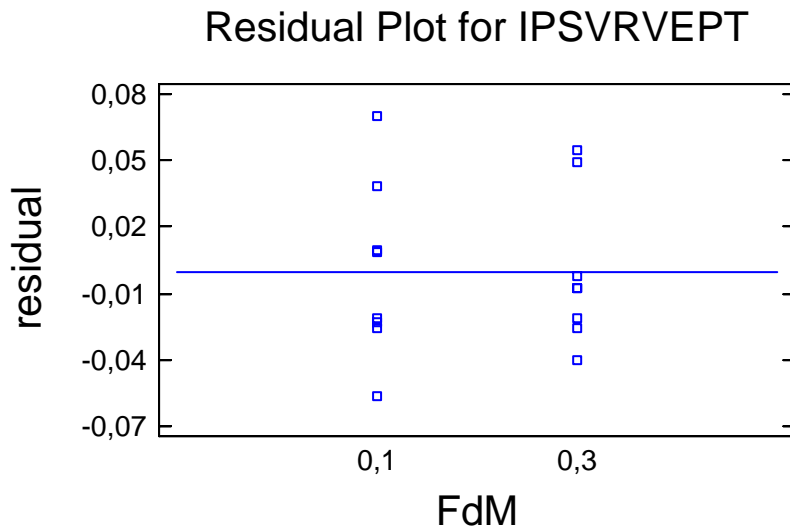


Figura IV-24. Gráfico de Residuos frente a los niveles del factor FdM del promedio de Conjuntos Experimentales con Margen Operativo del 20%.

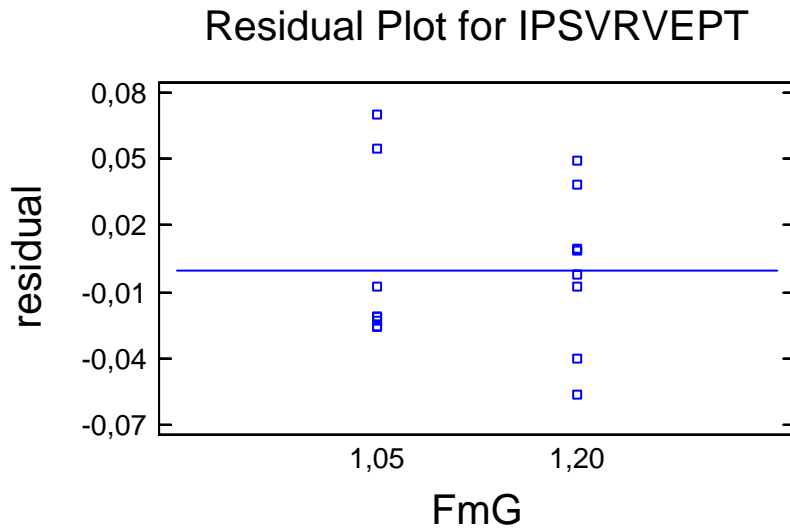


Figura IV-25. Gráfico de Residuos frente a los niveles del factor FmG del promedio de Conjuntos Experimentales con Margen Operativo del 20%.

En las figuras correspondientes a los residuos no se observa una falta de linealidad significativa en los mismos. Por lo que se descarta la ausencia de linealidad.

La correlación se detecta mediante el gráfico de residuos en relación al orden de ejecución de los tratamientos. Como se observa en la Figura IV-26 no se puede observar ninguna tendencia significativa en la distribución de los mismos por lo que se descarta la existencia de correlación.

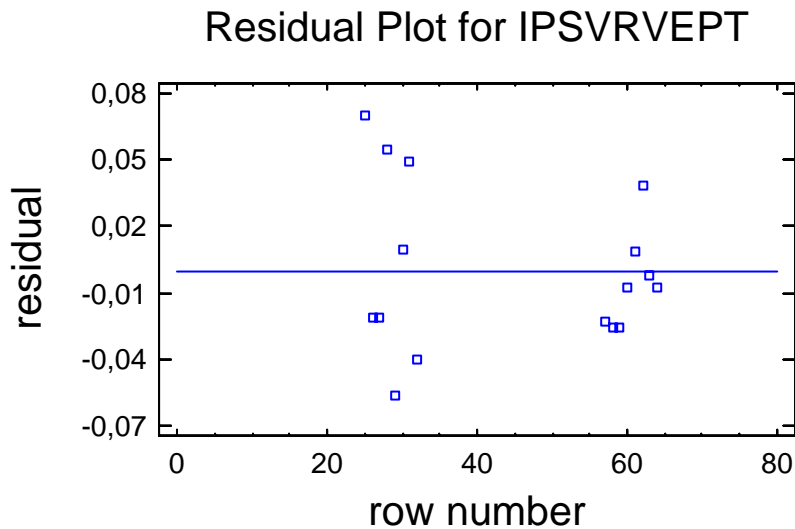


Figura IV-26. Residuos frente al orden de ejecución de los tratamientos de la variable IPSVRVEPT del promedio de Conjuntos Experimentales con Margen Operativo del 20%.

Una vez verificado el cumplimiento de las condiciones para el análisis ANOVA se obtiene y analiza la misma.

Analysis of Variance for IPSVRVEPT - Type III Sums of Squares

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
MAIN EFFECTS					
A:GsM	0,00643425	1	0,00643425	2,97	0,1187
B:FdM	0,000129932	1	0,000129932	0,06	0,8119
C:FmG	0,00966264	1	0,00966264	4,46	0,0638
INTERACTIONS					
AB	0,000914382	1	0,000914382	0,42	0,5319
AC	0,0025529	1	0,0025529	1,18	0,3057
BC	0,00031644	1	0,00031644	0,15	0,7110
RESIDUAL	0,0194771	9	0,00216412		
TOTAL (CORRECTED)	0,0394876	15			

All F-ratios are based on the residual mean square error.

Tabla IV-4. ANOVA correspondiente IPSPVRVEPT del promedio de Conjuntos Experimentales con Margen Operativo del 20%.

Como se puede ver en la Tabla IV-4 disponemos de 16 datos y tenemos 9 grados de libertad residuales. El contraste de hipótesis sobre la potencia (P-value) para un $\alpha=0,05$ implica que no es significativo ningún efecto principal así como tampoco lo es ninguna de las interacciones entre factores.

A-IV.2 Análisis de los Resultados obtenidos en la Parametrización del Algoritmo SMAGA-2 para la plataforma IPSU-MAS.

En este apartado se evalúa si el factor RM es significativo, y en caso de serlo cual o cuales son los valores más adecuados.

En relación con la normalidad del experimento se ha obtenido un Chi-Square = 4,25001 with 3 d.f. y un P-Value = 0,235699 lo que descarta la posibilidad de que la variable analizada no proceda de una distribución normal. Así mismo se puede observa en la Figura IV-27 como la variable se distribuye muy próxima a la línea continua diagonal, lo que implica su proximidad a la distribución Normal.

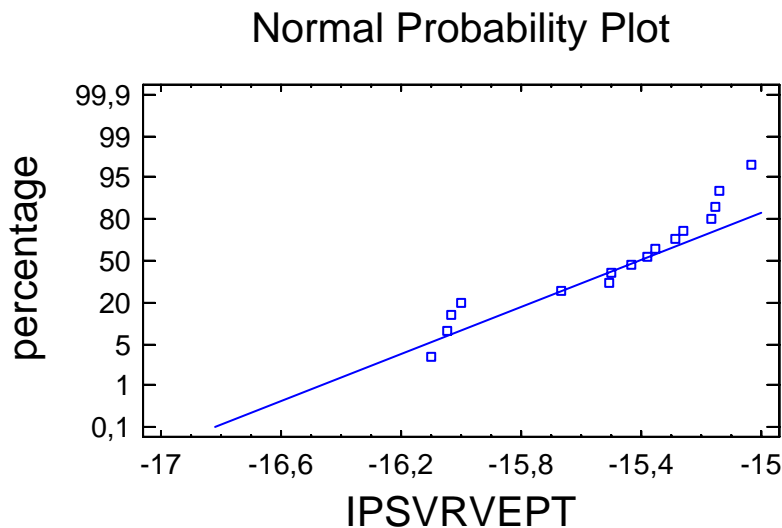


Figura IV-27. Gráfico de probabilidad normal (Y_i, F_i) para la variable IPSPVRVEPT del promedio de Conjuntos Experimentales con Margen Operativo del 5%, 10%, 15% y 20% en relación a las frecuencias relativas acumuladas.

Por lo que respecta a la homogeneidad de las varianzas se ha realizado el test de Levene para la variable dependiente con respecto a las variables independientes obteniéndose los siguientes valores:

- IPSPVVEPT respecto a CEX: test de Levene: 0,29686 P-Value = 0,827002
- IPSPVVEPT respecto a RM: test de Levene: 0,0025981 P-Value = 0,960068

Por lo que se descarta que las varianzas no sean homogéneas.

Por lo que se refiere a la linealidad se empleará el “gráfico de residuos (X_t, \bar{u}_t) y (\hat{Y}_t, \bar{u}_t)”, que permite disponer los residuos, en el eje de ordenadas, frente a las variables exógenas (X_t), y endógena (\hat{Y}_t).

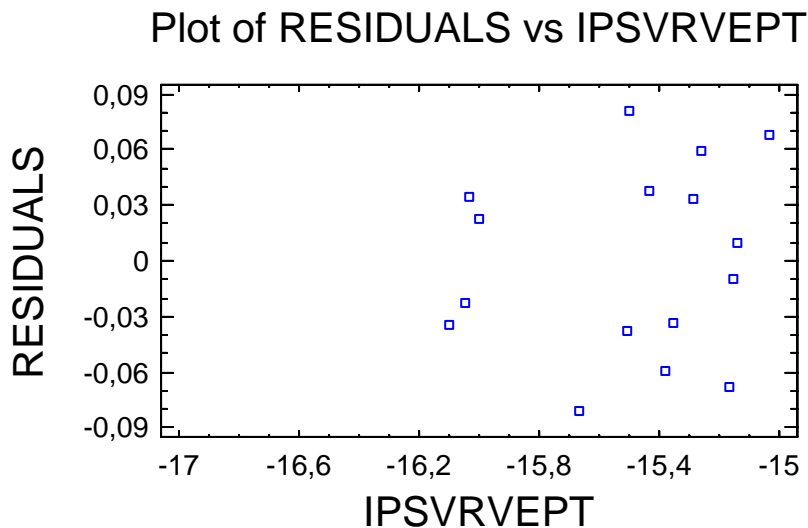


Figura IV-28. Gráfico de Residuos frente a los niveles de la Variable IPSPVVEPT del promedio de Conjuntos Experimentales con Margen Operativo del 5%, 10%, 15% y 20%.

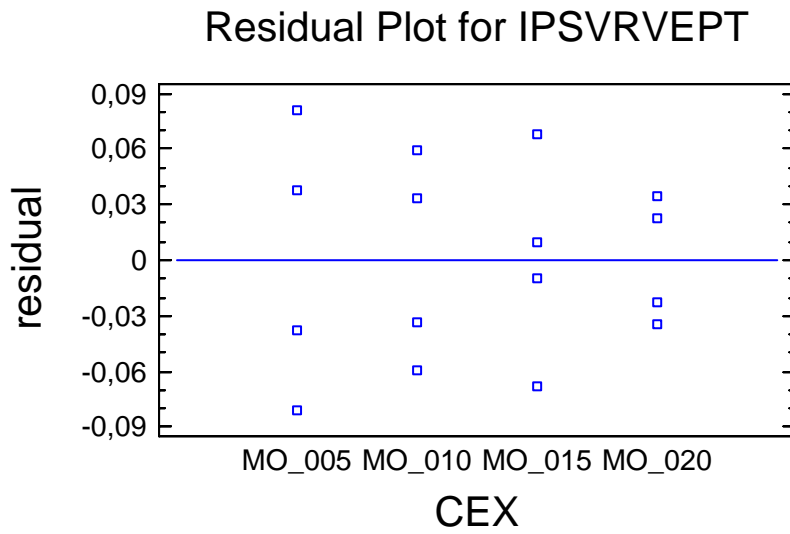


Figura IV-29. Gráfico de Residuos frente a los niveles del factor CEX del promedio de Conjuntos Experimentales con Margen Operativo del 5%, 10%, 15% y 20%.

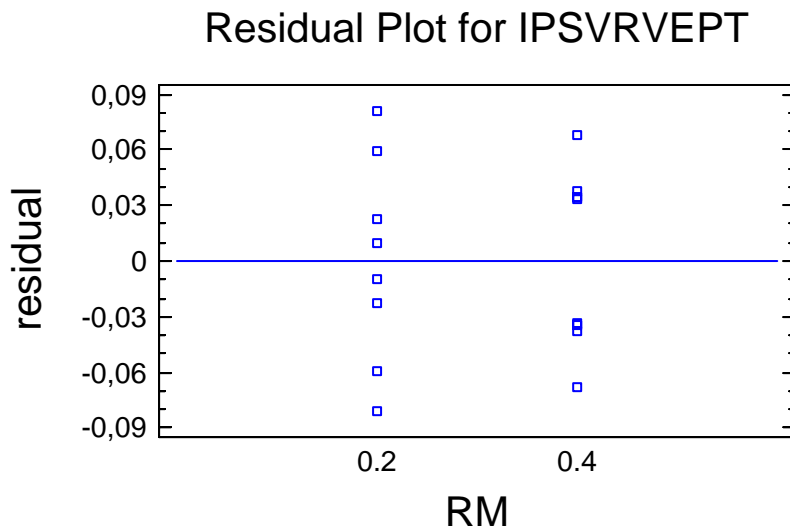


Figura IV-30. Gráfico de Residuos frente a los niveles del factor RM del promedio de Conjuntos Experimentales con Margen Operativo del 5%, 10%, 15% y 20%.

En las figuras correspondientes a los residuos no se observa una falta de linealidad significativa en los mismos. Por lo que se descarta la ausencia de linealidad.

La correlación se detecta mediante el gráfico de residuos en relación al orden de ejecución de los tratamientos. Como se observa en la Figura IV-31 no se puede observar ninguna tendencia significativa en la distribución de los mismos por lo que se descarta la existencia de correlación.

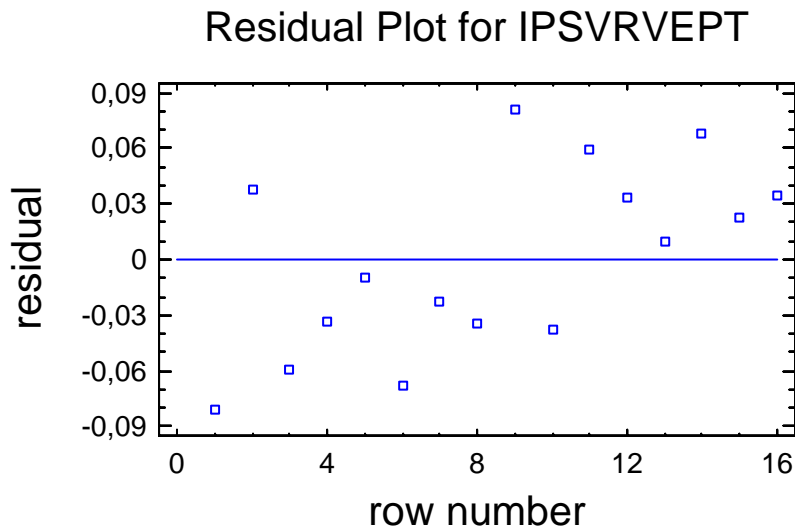


Figura IV-31. Residuos frente al orden de ejecución de los tratamientos de la variable IPSVRVEPT del promedio de Conjuntos Experimentales con Margen Operativo del del 5%, 10%, 15% y 20%.

Una vez verificado el cumplimiento de las condiciones para el análisis ANOVA se obtiene y analiza la misma.

Analysis of Variance for IPSVRVEPT - Type III Sums of Squares

Source	Sum of Squares	Df	Mean Square	F-Ratio	P-Value
MAIN EFFECTS					
A:CEX	1,88218	3	0,627392	131,16	0,0000
B:RM	0,00301133	1	0,00301133	0,63	0,4504
INTERACTIONS					
AB	0,0142419	3	0,0047473	0,99	0,4441
RESIDUAL	0,0382662	8	0,00478327		
TOTAL (CORRECTED)	1,9377	15			

All F-ratios are based on the residual mean square error.

Tabla IV-5. ANOVA correspondiente IPSVRVEPT del promedio de Conjuntos Experimentales con Margen Operativo del 5%, 10%, 15% y 20%.

Como se puede ver en la Tabla IV-5 disponemos de 16 datos y tenemos 8 grados de libertad residuales. El contraste de hipótesis sobre la potencia (P-value) para un $\alpha=0,05$ implica que sólo es significativo el efecto principales del factor CEX, y no es significativa la interacción entre factores. Por lo tanto, se puede concluir el análisis diciendo que RM no es un parámetro significativo.

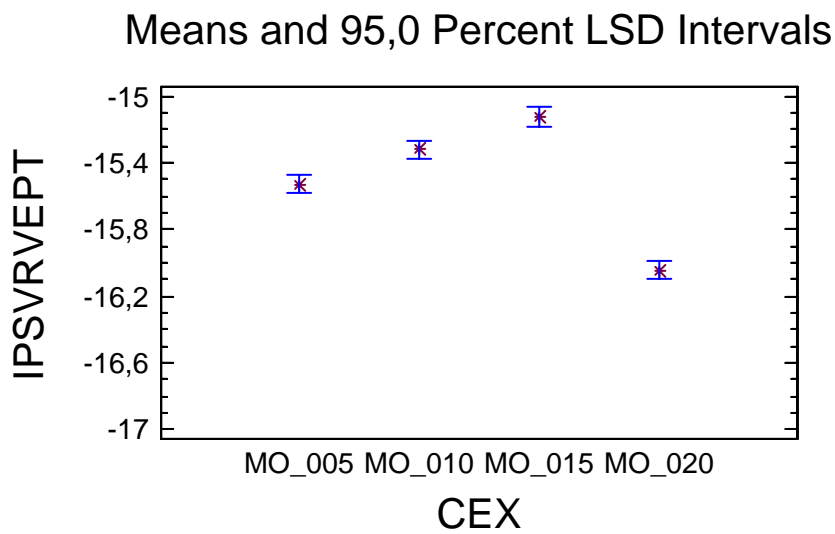


Figura IV-32. Gráfico del efecto principal del factor CEX del promedio de Conjuntos Experimentales con Margen Operativo del 5%, 10%, 15% y 20%.