

## Estudio del Comportamiento de los Sistemas de Fabricación Basados en Agentes: Comparativa entre diferentes arquitecturas.

José Alberto Araúzo Araúzo<sup>1</sup>, Juan José de Benito Martín<sup>1</sup>, Pedro Sanz Angulo<sup>1</sup>,  
Ricardo del Olmo Martínez<sup>2</sup>

1 ETSII. de la UVA, Paseo del Cauce s/n 47011 Valladolid, arauzo@eis.uva.es, debenito@eis.uva.es, psangulo@eis.uva.es.

2 EPS. de la UBU, Avenida Santander s/n 09006 Burgos, rdelolmo@ubu.es.

### Resumen

*Durante la última mitad del siglo XX la producción de bienes ha evolucionado desde sistemas de fabricación en masa hacia la fabricación personalizada. Tanto es así que en muchos sectores se tiende hacia la elaboración de lotes unitarios realizados a gusto del consumidor. Y todo ello, en una situación histórica donde la tecnología avanza tan rápidamente que la obsolescencia de productos y equipos es un hecho habitual. Un sistema productivo que pretenda ser competitivo en este entorno, debe poseer ciertas características que en otros momentos no eran tan relevantes. Aspectos como flexibilidad, adaptabilidad y rapidez de respuesta han pasado de ser deseables a convertirse en la clave del éxito de muchas empresas. Esta situación fomenta la aparición de nuevos paradigmas de gestión de la producción que evitan en la medida de lo posible los esquemas de control centralizados y jerárquicos, origen en muchas ocasiones de la falta de flexibilidad, adaptabilidad y robustez del sistema. Uno de estos nuevos paradigmas es la fabricación basada en agentes. En este artículo se presentan una serie de resultados obtenidos con una arquitectura de control de la fabricación basada en agentes que nos permitirán caracterizar el comportamiento de estos sistemas, en diferentes escenarios comparar y bajo diferentes modos de funcionamiento.*

**Palabras clave:** Sistemas de Fabricación, Sistemas Multiagente.

### 1. Sistemas de Fabricación Basados en Agentes

Los sistemas de fabricación basados en agentes pretenden explotar los conceptos, métodos y técnicas introducidas desde el paradigma informático de sistemas multiagente, con el fin de desarrollar sistemas de fabricación más competitivos en el entorno industrial actual.

Desde el punto de vista este paradigma, un agente es una entidad software autónoma y flexible, que es capaz de actuar en un entorno. Para ello debe poseer comportamientos pro-activos (actúa orientado a objetivos) y reactivos (reacciona ante estímulos percibidos del entorno) (Wooldridge 1995). Además, dado que en el entorno pueden existir más agentes, éstos deberán de ser capaces de interactuar entre sí (mediante comunicación), es decir, poseerán habilidades sociales. Estas características ofrecen a los desarrolladores de software la posibilidad de crear comunidades de agentes (sistema multiagente), que colaboran para conseguir un determinado objetivo (Sycara 1998). Debido a las características de los agentes, el paradigma multiagente proporciona de forma natural los métodos para diseñar e implementar complejos sistemas distribuidos.

Los sistemas de fabricación basados en agentes, pretenden integrar las actividades de programación, ejecución y control, propias del control de sistemas de fabricación, en un sistema software compuesto de varias entidades (agentes) autónomas, reactivas y pro-activas, que pueden interactuar haciendo uso de sus habilidades sociales. Aunque el sistema podría componerse de agentes dedicados a tareas concretas como pueden ser (programación, monitorización, control en tiempo real, etc.), esto no es lo usual en el SFC. La mayoría de los trabajos optan por una

descomposición del problema similar a la estructura física de la planta. Al ser la estructura del sistema software reflejo de la del entorno de fabricación, la adaptación del primero a modificaciones del segundo, será más sencilla (el sistema mejorará su flexibilidad, escalabilidad y reconfigurabilidad) (Giret 2005, Bongaerts 1998, Ryu 2003).

En el caso del control de plantas de tipo *job shop*, muchas publicaciones proponen distribuir el sistema en dos clases básicas de agentes: agentes asociados a las órdenes de producción (agentes orden) y agentes asociados a los máquinas (agentes máquina). Los primeros, de los que existirá uno por cada orden, representarán a los artículos que se desean fabricar, poseerán información sobre los procesos de fabricación del artículo y sobre su estado de ejecución, y tendrán como misión la gestión de las actividades necesarias para fabricar el ítem asociado. Los agentes asociados a los recursos, de los que habrá tantos como recursos se consideren, serán sus controladores, poseerán, por lo tanto, información relativa a sus capacidades técnicas y sobre su estado, y gestionarán la cola de trabajos pendientes.

## 2. Problemática

La distribución del sistema de control según la estructura propuesta plantea el reto de la distribución de las funciones globales del sistema en estas entidades no funcionales; más concretamente, la distribución de la programación y el control *on-line* en agentes asociados a las órdenes (agentes orden) y a los recursos (de estos últimos sólo se considerarán las máquinas, ‘agentes máquina’).

Las propuestas más sencillas que incorporan esta arquitectura proponen que los acuerdos sean formalizados mediante un mecanismo de contrato en el que intervienen los agentes asociados a las órdenes y los asociados a los recursos máquinas. O bien los primeros solicitan a los segundos la ejecución de tareas, o los segundos solicitan trabajos (operaciones) a los primeros (Saad 1996). Cuando un agente debe optar entre varias propuestas utiliza criterios locales (longitud de las colas de espera en las máquinas, urgencia de los trabajos, etc.) para decidir el acuerdo que formaliza. Dentro de este tipo de propuestas destaca, por su implementación real, la que se puede encontrar en (Bussmann 2001), que utiliza este tipo de técnicas para la integración de una cadena de montaje de motores de *Daimler Chrysler*.

Aunque las nuevas arquitecturas del sistema de control que se proponen desde los nuevos paradigmas de producción (sobre todo desde la fabricación holónica y basada en agentes) dotan al sistema de flexibilidad, adaptabilidad, rapidez de respuesta y robustez, el alejamiento de los esquemas centralizados y jerárquicos, que estos nuevos esquemas de control proponen, plantea algunos problemas. El más importante estriba en la dificultad de conseguir un comportamiento globalmente eficiente.

En los sistemas centralizados y jerárquicos, al existir controladores que recopilan información global y regulan el comportamiento del resto de elementos, es posible elaborar programas de operaciones globalmente óptimos. Si una vez elaborado el programa éste se ejecuta tal y como ha sido establecido, el sistema se comportará de forma globalmente eficiente. En las arquitecturas distribuidas, que evitan la centralización para no introducir rigidez en el sistema, esta forma de actuar es imposible. No se pueden realizar programas globales por la sencilla razón de que ninguno de los agentes del sistema posee la información suficiente para realizarlo.

Para dirigir al sistema de fabricación hacia comportamientos globalmente óptimos, se han formulado varias alternativas. Algunos trabajos proponen la inclusión de agentes programadores que, a partir de información suministrada por el resto de agentes del sistema, realizan programas

globales que sirven de referencia para la toma de decisión de los agentes asociados a órdenes y recursos (Xue 2001, Bongaerst 1998). Otra posibilidad consiste en utilizar técnicas de programación distribuida (algunas de ellas basados en subastas) donde, en vez de elaborarse programas globales, se realizan programas locales por parte de los agentes que representan a las órdenes y a las máquinas (programas que sólo abarcan las operaciones asociadas a un agente) (Gou 1998, Davies 2005).

### **3. Distribución propuesta**

En esta tesis se han propuesto unos mecanismos de programación y control *on-line* distribuidos, para ser implementados en un sistema de control de una supuesta planta, con las siguientes características básicas: sólo se considera el recurso máquina, el tamaño de las colas en las máquinas es ilimitado, el flujo de materiales es de tipo *job shop*, existe la posibilidad de enrutamiento flexible y no se contemplan montajes ni particiones de piezas.

#### **3.1. Distribución del control on-line**

Para abordar el problema del control *on-line*, éste se ha subdividido en dos niveles: lanzamiento (formalización de acuerdos de ejecución de operaciones entre ‘agentes orden’ y ‘agentes máquina’) y ‘ejecución y control *on-line*’ (envío de instrucciones a máquinas físicas, y reajuste de los acuerdos formalizados cuando se produce alguna perturbación que impide ejecutarlos tal y como se han establecido). Respecto al lanzamiento distribuido, se han diseñado dos mecanismos alternativos: uno en el que los acuerdos se formalizan justo en el momento de comienzo de los trabajos (*LANZ-SR*), y el segundo donde los acuerdos se formalizan con cierta antelación a la ejecución (*LANZ-CR*). Mediante el mecanismo *LANZ-CR*, a diferencia de lo que sucede con el *LANZ-SR*, se producen reservas de máquinas para trabajos futuros, lo que en algunas ocasiones permite mejorar la eficiencia del sistema.

#### **3.2. Distribución de la programación**

En todos los procesos de lanzamiento, los ‘agentes orden’ y los ‘agentes máquina’ deben tomar una serie de decisiones relativas a la selección, de entre todos los posibles, del acuerdo que se pretende formalizar. Para llevarlas a cabo se han propuesto dos alternativas: toma de decisión a partir de la información local de los agentes implicados en los acuerdos (*TD-HEUR*), y toma de decisión a partir de programas, elaborados mediante un mecanismo de programación distribuida basado en subastas (*TD-RL*).

La arquitectura propuesta para el sistema de control permite el diseño de mecanismos de programación distribuida inspirados en los mercados competitivos. Basta con identificar a los ‘agentes orden’ como consumidores de recursos (periodos de tiempo de máquina), y los ‘agentes máquina’ como los vendedores (valorados a un precio determinado). Los primeros intentarán adquirir los recursos que maximizan su utilidad (fabricar los productos en el plazo impuesto, minimizando en la medida de lo posible el precio total de los recursos utilizados para ello); los segundos maximizar sus ingresos (ingresos totales derivados de la venta de los recursos).

A partir de la analogía anterior, se pueden idear mecanismos de subastas, que tras un proceso iterativo de solicitudes de recursos por parte de los ‘agentes orden’ (programas locales), y de asignación de precios a los recursos por parte de los ‘agentes máquina’, se llega a unos precios de equilibrio estables que contienen información global sobre la importancia que cada recurso tiene para el cumplimiento de los programas locales (uno por cada agente orden). Los programas locales, obtenidos teniendo en cuenta estos precios, serán bastante compatibles (no

existe asignación de una misma máquina en un mismo periodo de tiempo a dos o más programas locales diferentes) y globalmente eficientes.

El proceso de subasta anterior se puede diseñar de diferentes formas. En este trabajo se ha optado por utilizar los resultados derivados de la aplicación de la técnica de Relajación Lagrangiana al problema de programación *job shop* (Luh 1991, Gou 1998). Esta técnica permite descomponer el problema en varios subproblemas, más un problema dual que integra los resultados de los primeros. Los subproblemas por la información que manejan son fácilmente asignables a los 'agentes orden', y su resolución conduce a un programa local para la orden asociada. El problema dual permite obtener el precio de los recursos, y no es de fácil asignación a los agentes de la arquitectura propuesta. Aunque inicialmente cabe pensar en la distribución del procedimiento de resolución del problema dual, en partes que se puedan asociar a cada máquina (para que cada una de ellas pueda calcular sus precios), en esta primera aproximación no se ha considerado esa opción. Esta alternativa puede suponer algunos problemas, que de momento se prefiere dejar de lado, reservándola para los futuros desarrollos.

Se han establecido dos modalidades de la toma de decisión *TD-RL*, la centralizada (*TD-RL-C*) y la distribuida (*TD-RL-D*). Según el procedimiento *TD-RL-D*, las órdenes elaboran programas locales, a partir de los cuales seleccionan los parámetros que incluirán en las solicitudes de lanzamiento que se envían a las máquinas. En el caso de que el criterio de toma de decisión sea *TD-RL-C*, los parámetros incluidos en la solicitud de lanzamiento se seleccionarán en función de los consejos emitidos por un agente que los establece a partir de programas globales obtenidos de la unión de los programas locales realizados por las órdenes.

### 3.3. Modos de funcionamiento

Cruzando los posibles mecanismos de lanzamiento con los posibles criterios de toma de decisión se llega a la tabla 1, que representa los posibles modos de funcionamiento del sistema que se describirán a continuación.

**Tabla 1.** Modos de funcionamiento del sistema de control de la fabricación.

		Lanzamiento sin retrasos	Lanzamiento con retrasos
Toma decisión heurística		HEUR-SR	HEUR-CR
Toma decisión RL	RL-D	RL-SR-D	RL-CR-D
	RL-C	RL-SR-C	RL-CR-C

## 4. Escenarios de experimentación

El sistema multiagente que se ha desarrollado integra programación, ejecución y control. Si para probar un sistema de programación, no es necesario disponer de la planta (es suficiente con haberla definido); para probar un sistema de ejecución y control (que debe necesariamente interactuar con el entorno en tiempo real) se necesitará que el sistema a controlar esté en funcionamiento (por lo menos de una forma virtual o simulada).

### 4.1. Planta de prueba

Con objeto de disponer de un sistema de control de planta, que cumpla con las hipótesis anteriormente indicadas, y que sirva de sistema piloto para testar nuestro sistema, se tienen

varias opciones: (1) construir o seleccionar un sistema físico con las características señaladas, (2) elaborar un sistema de fabricación virtual, separado del sistema SFC, que emule el comportamiento del sistema de fabricación objeto de control, o (3) emular, en el propio sistema SFC, la recepción de señales procedente desde la planta de producción (aunque ésta no esté implementada ni física ni virtualmente).

De las tres alternativas anteriores, nosotros hemos optado por la tercera. La primera parece bastante descabellada (alto coste, dificultad técnica, aparición de múltiples problemas, etc.). La segunda opción, aunque no resulta tan complicada, creemos que precisa de más trabajo que la tercera opción sin aportar ninguna ventaja adicional. Para llevar a cabo la segunda alternativa, o bien se implementa un sistema software que se comporte como el sistema definido anteriormente (lo cual implica un considerable esfuerzo), o bien se utiliza un simulador comercial (lo que exigiría, además del esfuerzo de modelado, el establecimiento de la comunicación entre el simulador y el sistema de control).

La planta de producción simulada poseerá una configuración *job shop* flexible, sin montajes ni particiones de piezas y con colas infinitas en las máquinas para impedir bloqueos por falta de espacio de almacenamiento. La flexibilidad se referirá a la posibilidad de enrutamiento flexible, que se podrá derivar, bien de la posibilidad de ejecutar una operación en varias máquinas (con diferente tiempo de procesamiento), o bien de la relajación de las restricciones de precedencia.

Aunque la planta es susceptible de funcionar de forma dinámica (las órdenes van entrando en tiempo de ejecución), y además se podrán considerar efectos estocásticos (duraciones aleatorias de las operaciones), los resultados obtenidos con estos casos no se expondrán en este artículo ya que preferimos ceñirnos a unos pocos escenarios.

#### 4.2. Problemas básicos

Para comprobar el funcionamiento del sistema de control diseñado sobre la planta propuesta anteriormente, se han generado aleatoriamente 12 problemas *job shop* estáticos y deterministas: cuatro de 5 órdenes por 5 máquinas (Prob5/1, Prob5/2, Prob5/3 y Prob5/4), cuatro de 7 órdenes por 5 máquinas (Prob7/1, Prob7/2, Prob7/3 y Prob7/4) y otros cuatro de 10 órdenes por 5 máquinas (Prob10/1, Prob10/2, Prob10/3 y Prob10/4). Estos problemas servirán para comparar los diferentes modos de funcionamiento del sistema en condiciones estáticas.

Para caracterizar el funcionamiento del sistema (con diferentes modos de funcionamiento), en condiciones dinámicas y estocásticas se modificarán los problemas básicos anteriores introduciéndolos diferentes grados de dinamismo y aleatoriedad, de forma que se pueda estudiar el efecto de estos factores en el desempeño del sistema.

#### 4.3. Medida del desempeño

Para medir el desempeño del sistema en cada escenario se utilizará la suma ponderada de los retrasos al cuadrado (expresión 1). Cuanto menor sea esta función mejor será el funcionamiento del sistema.

$$T^2 = \sum_{i=1}^I w_i \cdot T_i^2 \quad (1)$$

Con  $T_i$  retraso de finalización de la orden  $i$  respecto al plazo indicado.  
 $w_i$  peso asociado a la orden.

Se ha seleccionado esta medida por los siguientes motivos: (1) permite considerar la importancia de cada orden, (2) a diferencia de la suma de retrasos ponderados, la medida seleccionada (al incluir los retrasos al cuadrado) penaliza los retrasos extremos, y (3) estabiliza el método de programación distribuida que se utiliza para realizar la programación distribuida (también por incluir los retrasos al cuadrado).

Es de destacar que esta medida de eficiencia es aditiva respecto a las órdenes, es decir, la función objetivo es suma de las contribuciones individuales de cada orden. Esto la hace susceptible de ser utilizada en métodos de programación basados en subastas (o de descomposición que presentan una analogía con los de subasta), que requieren de este tipo de funciones objetivo. Todas estas características convierten a esta función en una medida muy adecuada para estimar la eficiencia del comportamiento del sistema de control que se pretende desarrollar.

## 5. Experimentación

La experimentación se ha realizado según la siguiente metodología: primero se comprobó el comportamiento del sistema bajo diferentes esquemas de control en escenarios estáticos, deterministas y sin presencia de flexibilidad; se continuó comprobando la capacidad de gestión de la flexibilidad que poseen los diferentes esquemas de control propuestos en condiciones estáticas y deterministas; y finalmente se estudió el comportamiento del sistema en escenarios dinámicos y estocásticos.

Los experimentos se realizaron en tiempo real, es decir, la ejecución de las actividades, aunque simulada, duró lo que duraría en un sistema real. Es necesario realizarlo de este modo, ya que parte de los experimentos tratarán de comprobar la capacidad que tiene el sistema de reajustarse a nuevas situaciones en tiempo de ejecución.

Aunque el sistema es susceptible de ejecutarse en una red de ordenadores, lo que podría suponer mejoras en el comportamiento global, en un principio los experimentos se han realizado en un único ordenador, buscando la diferencia de comportamiento entre los distintos modos de funcionamiento. En un futuro, si los resultados son prometedores, se podría estudiar la capacidad del sistema en situaciones límites (muchas operaciones y muchas máquinas), pudiéndose ejecutar los agentes en una o en varias máquinas.

### 5.1. Situación estática y determinista: procesos no flexibles

Interesa en esta sección comparar los diferentes esquemas de control propuestos, sobre todo los dos esquemas de control en los que se realiza programación distribuida (*TD-RL-C* y *TD-RL-C*), con aquellos en los que no (*TD-HEUR*). Con ellos se pretende mostrar como esta programación permite mejorar substancialmente los resultados obtenidos con los procedimientos que no usan esta técnica.

#### 5.1.1 Evaluación del comportamiento del sistema con *TD-RL-C*

En la Figura 1 se muestra los resultados obtenidos con la versión centralizada del criterio de toma de decisión *TD-RL*. Para cada problema se representa el resultado de dos experimentos, uno realizado con el mecanismo de lanzamiento *LANZ-CR* ('RL CR' en el gráfico) y otro con el *LANZ-CR* ('RL SR' en el gráfico). Además, con objeto de comparar el comportamiento de los dos criterios de toma de decisión propuestos, se ha representado junto a los dos resultados mencionados, el mejor de los obtenidos en sucesivos experimentos realizados con el criterio *TD-HEUR* ('HEUR' en el gráfico).

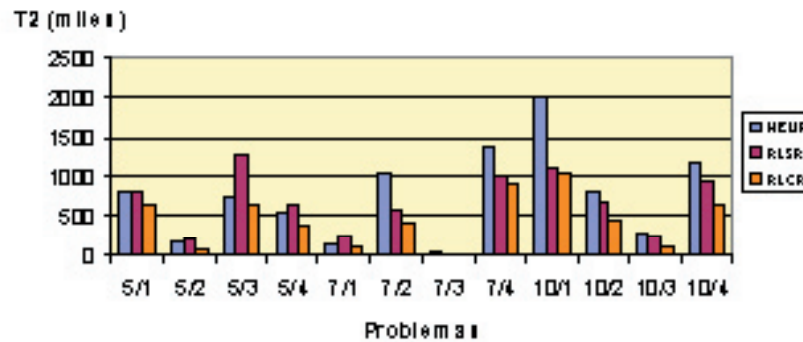


Figura 1. Comparación de TD-RL-C (LANZ-CR y LANZ-SR) frente al mejor obtenido con TD-HEUR.

En el gráfico anterior puede observarse cómo la incorporación de los precios (*TD-RL*) en la toma de decisiones, mejora significativamente la eficiencia del sistema. Además, en este caso, la posibilidad de lanzamiento con retrasos (*LANZ-CR*) supone mejoras sobre el modo *LANZ-SR*, en contra de lo que sucede con la toma de decisión *TD-HEUR*, que en muchas ocasiones es incapaz de aprovechar la posibilidad de retrasos.

Otro aspecto que se debe analizar en estos experimentos son los valores de los *gaps* de dualidad (parámetro derivado del mecanismo de programación distribuida que permite estimar la suboptimalidad de la solución). En la tabla 2 se muestra el valor del *gap* en diferentes en los experimentos realizados con el criterio de toma de decisión *TD-RL* para cada uno de los problemas. Cuanto más se acerque a cero más certeza se tendrá de que el comportamiento será cercano al óptimo, pero en caso contrario tampoco se puede asegurar que el comportamiento sea ineficiente. De hecho, en problemas como el 'Prob7/3', donde este *gap* es de 0.39, la mejora que se consigue al sustituir *TD-HEUR* por *TD-RL* es la mayor de las conseguidas en todos los experimentos. En los casos en los que *Gap* es elevado, una vez comenzada la fabricación éste suele disminuir con bastante celeridad. Además, aunque estos *gaps* sean altos, los precios se suelen estabilizar rápidamente en todos los experimentos, no apreciándose en ellos grandes oscilaciones en la evolución posterior.

Tabla 2. *Gaps* de dualidad en diferentes instantes.

	5/2	5/3	5/3	5/4	7/1	7/2	7/3	7/4	10/1	10/2	10/3	10/4
Gap	0.12	0.28	0.04	0.23	0.21	0.03	0.39	0.17	0.07	0.41	0.32	0.36

### 5.1.2 Evaluación del comportamiento del sistema con *TD-RL-D*

Para analizar el comportamiento del sistema con la versión distribuida de toma de decisión *TD-RL*, se han realizado cinco réplicas de cada experimento, seleccionando como mecanismo de lanzamiento el *LANZ-CR*. Los resultados obtenidos para cada problema en cada una de estas réplicas se muestran en la figura 2, junto a los resultados obtenidos con el *TD-RL-C* (RLC) y el mejor de los obtenidos con *TD-HEUR* (HEUR). Obsérvese que en la figura anterior los resultados obtenidos en réplicas del mismo experimento no son similares. La toma de decisión totalmente distribuida produce cierta inestabilidad en el sistema. Ese comportamiento era previsible, y por ello se decidió realizar varias réplicas.

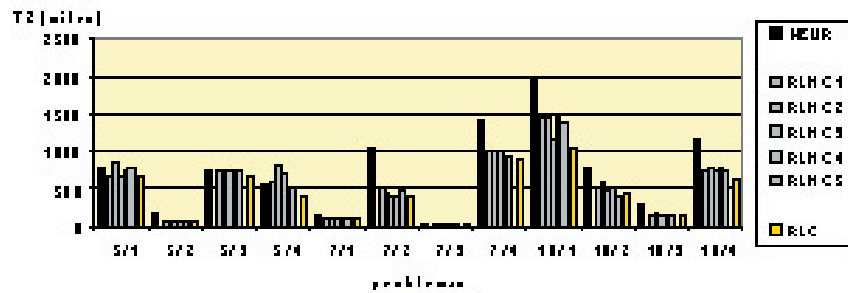


Figura 2. Comparación de TD-RL-C (LANZ-CR y LANZ-SR) frente al mejor obtenido con TD-HEUR.

## 5.2. Situación estática y determinista: procesos flexibles

Los resultados mostrados hasta el momento se han obtenido con experimentos realizados con problemas no flexibles (formados con productos que sólo se pueden fabricar mediante una ruta de fabricación). Aunque éstos muestran algunas características del comportamiento del sistema bajo diferentes configuraciones, no muestran la habilidad para la gestión de la flexibilidad. Con los resultados que se expondrán a continuación, se pretende ver las diferencias en el comportamiento ante problemas flexibles, en función del criterio de toma de decisión seleccionado. Para generar los experimentos se trabajará con los problemas planteados hasta el momento, pero con los procesos productivos ligeramente modificados para incluir varias rutas de fabricación.

Con objeto de caracterizar la flexibilidad de cada problema se ha definido un índice ( $If_p$ : índice de flexibilidad del problema  $p$ ) que se calcula como la suma de las flexibilidades de las órdenes del problema, dividido entre el número de órdenes del problema (expresión 2). La flexibilidad de cada orden será la flexibilidad del proceso del producto asociado a la orden ( $f_{ip}$ : flexibilidad de la orden  $i$  problema  $p$ ), entendiéndose ésta como el número de rutas alternativas de fabricación menos uno (expresión 3). Según esta definición, los procesos no flexibles poseerán flexibilidad cero, y los procesos flexibilizados según la técnica descrita (relajación de la restricción de precedencia entre dos operaciones) flexibilidad uno.

$$f_p = \frac{1}{I} \sum_{i=1}^I f_{ip} \quad I \rightarrow \text{número de órdenes} \quad (2)$$

$$f_{ip} = \text{número posibles rutas} - 1 \quad (3)$$

Para cada problema y grado de flexibilidad se han realizado dos experimentos uno con el criterio de toma de decisión  $TD-HEUR$  y otro con  $TD-RL-C$  (en ambos casos con el mecanismo de lanzamiento  $LANZ-CR$ ). Con los resultados obtenidos se ha realizado un gráfico por problema, donde se representa  $T^2$  frente a la flexibilidad, para ambos criterios de toma de decisión. En la figura se muestran los resultados para los problemas de 10.





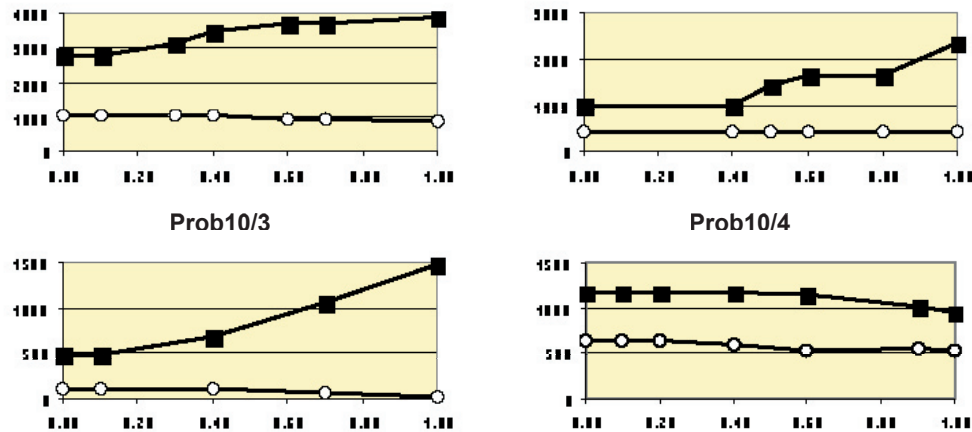


Figura 3.  $T^2$  frente a flexibilidad en para cada problema.

Como puede apreciarse en las gráficas anteriores, el comportamiento del sistema mientras que con el procedimiento de toma de decisión *TM-RL* el sistema es capaz de aprovechar la flexibilidad para mejorar la eficiencia, con el *TM-HEUR* esto no siempre ocurre.

### 5.3. Conclusiones

El paradigma multiagente ha aportado una nueva forma de abordar el problema de control de planta. Pero la novedad fundamental no se encuentra en las herramientas, ni en las especificaciones, sino en la forma de pensar y de analizar el problema. La idea de distribuir el sistema en entidades autónomas, en estrecha relación con la estructura de la planta, modifica radicalmente el desarrollo posterior del sistema.

Mientras que en los sistemas de control tradicionales existen elementos centralizadores que poseen un modelo global de sistema, que es utilizado para la toma de decisión y la integración del sistema, en la solución multiagente no tiene por qué ocurrir: (1) los agentes poseen un modelo local (cada orden de su proceso de producción, y cada máquina de sus capacidades), (2) realizan sus propios planes (programas locales), teniendo en cuenta para ello los precios de los recursos que han resultado de un proceso de negociación (mecanismo de programación distribuida), (3) los planes se intentan llevar a cabo formalizando acuerdos con otros agentes (procesos de lanzamiento), y (4) cuando algún suceso hace imposible que se cumplan los acuerdos y los planes, los agentes los reajustan comunicando sus decisiones al resto de agentes involucrados.

Como resultado de esta forma de pensar, el sistema se integra verticalmente en cada uno de los agentes (coherencia entre programación, lanzamiento, ejecución y control *on-line*), y horizontalmente a través de las interacciones entre agentes (compatibilidad de los diferentes programas locales, acuerdos y acciones). Los resultados obtenidos con el prototipo desarrollado son esperanzadores, ya que con la configuración más completa (incluyendo programación distribuida), el funcionamiento del sistema ha sido el adecuado en todos los escenarios propuestos.

### Referencias

Bongaerst, L. (1998) Integration Of Scheduling And Control In Holonic Manufacturing Systems. Ph.D. Dissertation, K.U.Leuven, PMA Division.

Bussmann, S.; Sieverding, J. (2001). Holonic Control of an Engine Assembly Plant – An Industrial Evaluation, Proceedings of IEEE Conf. on SMC. Tucson.

Davies, R.W. (2005). Distributed Generalized Vickrey Auctions Based on the Dantzig-Wolfe and Benders Decomposition Methods for Linear Programs, Thesis, Bachelor of Arts, Harvard College, Cambridge, Massachusetts.

Giret, A. (2005) ANEMONA: Una metodología Multiagente para Sistemas Holónicos de Fabricación. Tesis Doctoral, Universidad Politécnica de Valencia, Valencia, España.

Gou, L.; Luh, P.B.; Kyoya, Y. (1998). Holonic Manufacturing Scheduling: Architecture, Cooperation Mechanism and Implementation. Computers In industry, Vol. 37, pp. 213-231.

Luh, P.B.; Hoiomt, D.J. (1991). Scheduling of Manufacturing Systems Using the Lagrangian Relaxation Technique. IFAC Work Shop on Discrete Event System Theory and Applications in Manufacturing and Social Phenomena, Shenyang, China.

Ryu, K.; Son, Y.; Jung, M. (2003). Modeling and Specification of Dynamic Agents in Fractal Manufacturing Systems. Computers in Industry, Vol 52, pp 161-182.

Saad, A.; Kawamura, K.; Biswas, G. (1996). Performance Evaluation of Contract Net-Based Heterarchical Scheduling for Flexible Manufacturing Systems. Intelligent Autonomous and Soft Computing, Vol. 3, N° 3, pp 229-248.

Sycara, K.P. (1998). Multiagent Systems AI magazine Vol. 19, No.2, pp 79-92.

Wooldridge, M.; Jennings, N.R. (1995). Intelligent Agents: Theory and Pactice. The Knowledge Engineering Review, Vol 10, N° 2, pp. 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, Vol. 46, pp 209-231.