

Programación y Control de Sistemas de Fabricación Flexibles: un Enfoque Holónico

J.A. Araúzo^{a,*}, R. del-Olmo-Martínez^b, J.J. Laviós^b, J.J. de-Benito-Martín^a

^a *Departamento de Organización de Empresas y CIM, Universidad de Valladolid, Escuela de Ingenierías Industriales, Pº del Cauce, 59, E-47011, Valladolid.*

^b *Departamento de Ingeniería Civil, Universidad de Burgos, Escuela Politécnica Superior (Edificio La Milanera), C/ Villadiego s/n, E-09001, Burgos.*

Resumen

Los sistemas de fabricación holónicos constituyen un nuevo paradigma de producción. Incluidos dentro de las metodologías distribuidas, proponen nuevos esquemas de gestión de la fabricación que combinan auto-organización, jerarquías dinámicas y relaciones horizontales. Este enfoque es más cercano a la naturaleza de los sistemas de fabricación que las aproximaciones clásicas, centralizadas o estrictamente jerárquicas y permiten modelar los sistemas productivos de forma más realista, facilitando la implementación de sistemas más flexibles, escalables y robustos. En este artículo proponemos un sistema de programación y control para sistemas de fabricación flexibles desarrollado según el paradigma holónico. El sistema se basa en una nueva arquitectura que integra dos niveles de decisión: uno de programación basado en subastas, donde cada holón realiza su propio programa local; y otro de lanzamiento y control, a través del cual se coordinan las actividades de los holones. A diferencia de otros trabajos, donde los programas locales son integrados en un programa global que centraliza el sistema, en el sistema propuesto no ocurre así. Es el mecanismo de lanzamiento y control el que integra los programas locales. Además, el prototipo desarrollado puede funcionar con diferentes grados de descentralización, lo que permite valorar las ventajas que proporciona nuestra aproximación mediante simulación. Copyright © 2015 CEA. Publicado por Elsevier España, S.L. Todos los derechos reservados.

Palabras Clave:

Control de la producción, sistemas de producción, algoritmos de programación, sistemas de fabricación flexibles, sistemas de fabricación inteligentes.

1. Introducción

Durante la última mitad del siglo XX la producción de bienes ha evolucionado desde sistemas de fabricación en masa hacia la fabricación personalizada. 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, para ser competitivo en este entorno, debe poseer ciertas características que antes no eran tan relevantes. Flexibilidad, calidad, adaptabilidad y rapidez de respuesta se han convertido en la clave del éxito de muchas empresas (Nahmias 2013, Rosenzweig y Easton 2013).

Esta situación fomenta la aparición de nuevos paradigmas de gestión de la producción que evitan los esquemas centralizados y jerárquicos, origen, en muchas ocasiones, de la falta de flexibilidad, adaptabilidad y robustez (Leitao y Restivo 2000). La fabricación holónica es uno de esos nuevos conceptos. Proponen la distribución de la toma de decisión en unas entidades autónomas denominadas holones que poseen cierto grado de

independencia, pero que simultáneamente están sujetos al mando de múltiples autoridades. Con ello se pretende combinar la adaptabilidad, flexibilidad, robustez y rapidez de respuesta, típicas de los sistemas distribuidos no jerárquicos, y la eficiencia global asociada a las aproximaciones centralizadas y jerárquicas.

En este artículo se presenta el paradigma de fabricación holónica desde el punto de vista conceptual y se describirá un sistema software, que hemos desarrollado según esta aproximación, destinado a la programación y control de sistemas de fabricación flexibles. Este software, que aporta algunas novedades sobre las propuestas presentes en la literatura actual y que se detallarán más adelante, pretende mostrar el paso de los aspectos teóricos a la implementación. Además, permitirá obtener resultados sobre la idoneidad de esta aproximación en general, y sobre nuestra propuesta en particular.

El resto del artículo se organiza como sigue: en el segundo apartado se presenta el paradigma de fabricación holónica con sus postulados básicos. Al final de este apartado se concretará la aportación del presente trabajo. A continuación, en el apartado tres, se definirá formalmente el sistema de fabricación al que se pretende aplicar nuestra propuesta, que se describirá finalmente en el apartado cuatro. Se finalizará presentado unos resultados y unas conclusiones.

* Autor en correspondencia.

Correos electrónicos: arauzo@eii.uva.es (J.A. Araúzo),
rdeolmo@ubu.es (R. del Olmo-Martínez), jjlavios@ubu.es (J.J. Laviós), debenito@eii.uva.es (J.J. de Benito-Martín)

2. Sistemas de Fabricación Holónicos

La fabricación holónica surgió en el ámbito de un ambicioso programa de investigación: IMS (*Intelligent Manufacturing Systems* <http://www.ims.org/>). Lanzado por el Prof. Yoshikawa, de la Universidad de Tokio, se concibió como un programa de investigación básica apoyado por los gobiernos de los países industrializados, con la participación de sus industrias y centros académicos. En 1993-1994 se emprendió un estudio de viabilidad de este programa que comprendía seis proyectos o *test case*. El quinto, titulado *Holonic Manufacturing Systems* (HMS), trataba de trasladar los conceptos organizativos que subyacen tras las organizaciones espontáneas a los sistemas de fabricación (Höpf 1994). Con ello se pretendía superar las limitaciones de las arquitecturas jerárquicas y heterárquicas -distribuidas no jerárquicas- (Wyns 1999), intentando conservar por un lado la estabilidad que implica una jerarquía y proporcionado, además, la flexibilidad dinámica de las organizaciones heterárquicas (Bongaerst et al. 2000).

2.1. Marco Conceptual

El término holón lo introdujo el filósofo Koestler en 1967 (Koestler 1967), como combinación del término griego *holos* (entero), más el sufijo *on*, que indica partícula o parte. Koestler se basó en dos observaciones: (1) los sistemas complejos evolucionan a partir de sistemas simples mucho más rápidamente cuando hay formas intermedias estables, (2) las partes estables no están completamente definidas ni totalmente aisladas. Estas partes, identificables e inseparables, son a las que Koestler denominó holones (todo y parte).

Los holones son unidades independientes que no tienen la necesidad de una autoridad superior para poder realizar sus actividades, y que, simultáneamente, están sujetos a una o múltiples jerarquías que modulan su comportamiento (Chacón e Indriago 2010). La primera propiedad confiere a las organizaciones holónicas las características de las arquitecturas heterárquicas (flexibilidad, rapidez de respuesta, robustez y reconfigurabilidad). La segunda aporta estabilidad y orientación hacia objetivos globales, propio de las jerarquías (Botti y Giret 2008). Para trasladar estos conceptos holónicos al entorno fabril se propusieron una serie de definiciones (Höpf 1994):

- Holón: bloque autónomo y cooperativo constitutivo de un sistema de fabricación, destinado a la transformación, transporte, almacenamiento y/o validación de información.
- Autonomía: capacidad de una entidad para crear y controlar la ejecución de sus propios planes y/o estrategias.
- Cooperación: proceso mediante el cual varias entidades desarrollan y ejecutan planes conjuntamente.
- Holarquía: sistema de holones que pueden cooperar para lograr una meta. La holarquía define las reglas básicas para la cooperación y los límites de autonomía.
- Sistema de fabricación holónico: holarquía que integra todas las actividades del proceso de fabricación.

2.2. Aspectos de Implementación

Las definiciones del consorcio HMS son muy abstractas para guiar el desarrollo de sistemas de fabricación. Por ello es conveniente trasladar estos conceptos a requisitos prácticos (Bongaerst 1998). En los sistemas de fabricación holónicos se distinguen dos tipologías de holones: holones formados por una

unidad física de procesamiento (recurso del sistema productivo) y un software de control, y holones formados únicamente por una unidad de control. El desarrollo de un sistema holónico consiste básicamente en la identificación de los holones y en el diseño e implementación del software de cada holón y sus interacciones. Se distinguen, por lo tanto, dos niveles de diseño (Christensen 2003): el bajo nivel donde se especifica el funcionamiento interno de cada holón, y el alto nivel donde se concretan las relaciones entre holones.

En los holones (bajo nivel de diseño) se distinguen tres componentes (Figura 1): (1) una base de datos donde se registran los objetivos del holón, su estado y el conocimiento del medio; (2) una unidad de control responsable de la toma de decisiones; y (3) un módulo de coordinación mediante el cual el holón interactúa con otros holones.

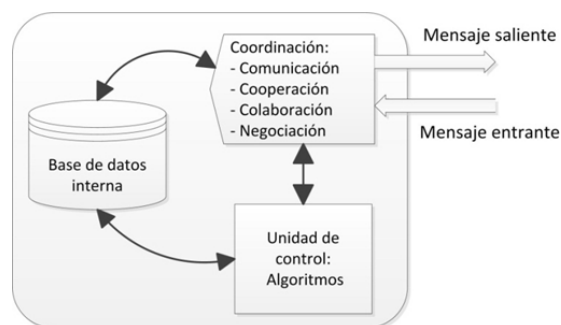


Figura 1: Arquitectura interna del holón (Babiceanu y Chen 2006)

En el diseño de alto nivel se definen los tipos de holones y sus interacciones. Estas últimas para conseguir coordinar las actividades de cada holón de forma que contribuyan al objetivo global del sistema. La coordinación abarca (Babiceanu y Chen 2006):

- Comunicación: intercambio de información entre los diferentes holones. Es necesario ya que la información está distribuida por el sistema.
- Cooperación: procedimiento, voluntario o no, mediante el que las acciones de cada holón se orientan hacia los objetivos globales.
- Colaboración: cooperación voluntaria.
- Negociación: mecanismo de resolución de los conflictos que aparecen cuando varios holones tienen intereses incompatibles.

2.3. Sistemas Holónicos para el Control de Plantas de Producción

Aunque la fabricación holónica es aplicable en muchos ámbitos de la producción (Borangiu et al. 2012), muchos de los trabajos actuales se centran en la programación y control de plantas de producción flexibles (Roulet-Dubonnet e Ystgaard 2011). En estos casos, la mayoría de trabajos adoptan una arquitectura órdenes-recursos (Figura 2). Ejemplos de estas arquitecturas son PROSA (Wyns 1999, Bongaerst 1998) y ADACOR (Leitão y Restivo 2006).

Las arquitecturas del tipo órdenes-recursos contienen fundamentalmente dos tipos de holones: holones orden y holones recurso. Los primeros representan los pedidos, contienen información sobre el producto a fabricar, sus operaciones, el

plazo de entrega, programa de producción previsto y estado de ejecución. Su objetivo es conseguir la fabricación del producto en el plazo especificado, y, para ello, contactan con recursos necesarios (máquinas en la Figura 2) formando holarquías. Los holones recurso cooperan con los holones órdenes y poseen información sobre el recurso físico que representan: capacidades (operaciones que pueden ejecutar, tiempo de ejecución, etc.), cola de trabajos, disponibilidad, etc.

La Figura 2 muestra un sistema de este tipo en el que se han emitido dos órdenes O1 y O2 -la primera se ha descompuesto en dos sub-órdenes O11 y O12-. Estas órdenes forman estructuras organizativas (holarquías) con las máquinas que necesitan para ejecutar sus operaciones. A diferencia de los sistemas jerárquicos, en los sistemas holónicos hay holones sujetos a múltiples autoridades, como por ejemplo, el holón M3 de la Figura 2 que pertenece a las holarquías promovidas por O12 y O2. Otra diferencia radica en que estas holarquías son dinámicas; cambian continuamente en función de los pedidos entrantes y de cambios en las características de los recursos.

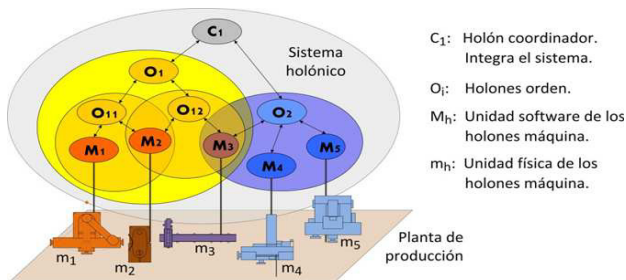


Figura 2: Representación de una arquitectura holónica órdenes-recurso.

Los mecanismos por los que se produce la cooperación en los sistemas holónicos de control de plantas de producción difieren en los diversos trabajos consultados, pero básicamente son combinaciones de dos procedimientos: (1) un holón de carácter centralizado elabora programas de producción eficientes que sirven como guía para la toma de decisiones de los holones orden y recurso, y (2) las órdenes acuerdan con las máquinas, de forma distribuida y asíncrona, la ejecución de las operaciones (colaboración). En este último caso, dado que puede haber órdenes que estén interesadas por la misma máquina en un mismo momento (conflicto), sería necesaria una negociación previa entre los holones orden implicados y el holón máquina.

El procedimiento más utilizado en la literatura para la asignación de las operaciones a los recursos, de forma totalmente distribuida y asíncrona, es el protocolo de contrato de red (Contract Net Protocol, CNP). Cuando una orden necesita algún recurso para ejecutar una operación, la orden solicita ofertas a los recursos, para, posteriormente, en función de las ofertas seleccionar el recurso más apropiado. Es un mecanismo muy reactivo por lo que es muy apropiado para sistemas muy dinámicos, pero a su vez puede resultar muy ineficiente. Probablemente la primera implementación de este mecanismo fue publicada por Parunak (1987). Implementaciones posteriores, con diferentes variantes, pueden encontrarse en Neligwa y Fletcher (2003), Babiceanu et al. (2004), y Hsieh (2008). Otra aproximación distribuida, basada en sistemas *pull*, puede verse en Ounnar y Pujo (2012). Otros trabajos que explotan esta idea pueden consultarse en Blanc et al. (2008) y en Roulet-Dubonnet e Ystgaard (2011).

Las arquitecturas más sofisticadas proponen holones de carácter centralizado que recopilan permanentemente información sobre el estado de los holones orden y recurso. Con esta información elaboran un programa global, que se modifica a medida que las condiciones de operación cambian, y que servirá de guía para la toma de decisiones entre los holones orden y máquina (Bongaerst 1998).

Dentro de las aproximaciones con holones centralizadores destacan las que utilizan mecanismos de asignación distribuidos basados en subastas (Gou et al. 1998, Wonga et al. 2006, Leitão y Restivo 2008). En éstas, un holón de carácter centralizador actúa como subastador, ofertando las unidades de tiempo de uso de los recursos a unos precios concretos. Basándose en estos precios, los holones orden calculan unos programas locales que incluyen únicamente las operaciones de la orden. Estos programas locales son integrados por el holón subastador, solucionando posibles conflictos, con el objetivo de obtener un programa global eficiente que sirva de guía para la toma de decisiones del resto de holones. Generalmente, después de una ronda de subasta, el subastador actualiza los precios y se vuelve a repetir el proceso de forma que los programas locales son cada vez más compatibles y el programa global resultante más eficiente.

2.4. Planteamiento del Trabajo

En este trabajo se describe un prototipo de sistema holónico para la programación y control de plantas flexibles, que aborda aspectos no presentes en trabajos anteriores. El sistema posee una nueva arquitectura que se diferencia de las ya documentadas en que cada holón posee dos niveles de decisión: uno de programación basado en subastas y otro de lanzamiento y control, implementado sobre una variante del CNP. Esto va a permitir que el sistema funcione en diferentes modos, con diversos grados de centralización y reactividad.

La aportación más novedosa se plasma en uno de los modos de funcionamiento. En él se realiza una programación distribuida basada en subastas que se adaptará dinámicamente a las condiciones de operación. A diferencia de las versiones ya publicadas, en este modo de funcionamiento no se integran los programas locales en un programa global central. Son los programas locales que realiza cada orden, los que sirven de guía a la toma de decisiones. Esto es posible gracias a la integración de los dos niveles de decisión propuestos, de forma que los posibles conflictos entre programas locales se resuelven en el nivel de lanzamiento y control. El procedimiento resultante es más distribuido que otras implementaciones basadas en subastas, con las ventajas que ello supone, pero a la vez es más eficiente que los sistemas más reactivos, los basados en CNP.

Esta arquitectura permitirá realizar estudios sobre uno de los aspectos centrales de los sistemas holónicos: cómo afecta el grado de centralización a la eficiencia del sistema y a su predictibilidad, así como al cumplimiento de los requisitos de reactividad, flexibilidad, adaptabilidad y robustez.

3. Definición del Sistema de Fabricación

La configuración del sistema de fabricación es la conocida como *job shop* con enrutamiento flexible (Pinedo 2009): una serie de N órdenes, que denotaremos con el subíndice i , deben ser fabricadas dentro de un plazo (D_i , $i=1\dots N$). Cada orden i se compone de un conjunto de operaciones denotadas por j ($j=1\dots J_i$) cuya ejecución está limitada por restricciones de precedencia. Las

operaciones son ejecutadas por las máquinas del sistema que se denotarán con el subíndice h ($h=1\dots H$). Se supone que cada máquina sólo puede procesar una operación a la vez, y que una vez comenzada no puede interrumpirse (modo *non-preemptive*). Para simplificar el problema, supondremos que el transporte de material es instantáneo y los almacenes locales (colas o *buffers*) ilimitados. En futuros trabajos, estas simplificaciones se podrían eliminar fácilmente considerando los elementos de transporte y las colas como otros recursos más.

Dado que una de las ventajas de los sistemas holónicos es el aprovechamiento de la flexibilidad de la planta, se considerará enrutamiento flexible: cada operación se puede realizar en una o más máquinas con tiempo de ejecución diferente d_{ijh} (duración de la operación j de la orden i en la máquina h); además, las relaciones de precedencia no necesariamente conducen a una única ruta de fabricación.

Para medir la eficiencia del sistema se utiliza un criterio basado en los retrasos de las órdenes, entendiendo como retraso el tiempo que transcurre desde que se cumple el plazo hasta que se finaliza realmente la orden. Si la orden finaliza antes del plazo el retraso es cero. Dentro de estos criterios, que son ampliamente utilizados en este tipo de problemas, nosotros optamos por la suma de retrasos cuadráticos ponderados T^2 (1). El sistema deberá minimizar esta expresión. Se ha optado por este criterio porque penaliza más los grandes incumplimientos que los pequeños. De esta forma los retrasos tenderán a repartirse entre las órdenes.

$$T^2 = \sum_{i=1}^N w_i \cdot T_i^2 \tag{1}$$

con $T_i = \max(0, \max_j(c_{ij}) - D_i) \rightarrow$ retraso de la orden i
 $w_i \rightarrow$ Peso de la orden i
 $D_i \rightarrow$ Plazo de entrega de la orden i
 $c_{ij} \rightarrow$ Fecha de conclusión de la operación j de i

4. Sistema Holónico Propuesto

El sistema propuesto posee cuatro tipos de holones (Figura 3), dos de carácter centralizado (planificador y coordinador) y los dos típicos de la arquitectura (órdenes y máquinas):

- Holón planificador: es el encargado de crear los holones orden, previamente definidos por el usuario a través de una interfaz gráfica. Guarda información de las órdenes emitidas y cuándo finalizan, calcula los retrasos y con ellos la eficiencia del sistema según (1).
- Holón orden: gestiona la ejecución de cada pedido. Hay uno por cada orden activa. Contienen información sobre el pedido (operaciones, precedencias, peso y plazo), sobre los planes (programas de operaciones pendientes) y sobre el estado de ejecución. Su objetivo es fabricar el pedido, y para ello llega a acuerdos de ejecución de las operaciones con los holones máquina.
- Holón máquina: hay uno por cada máquina física. Posee una interfaz gráfica que facilita su configuración por el usuario (operaciones que puede ejecutar y tiempos de ejecución). Controla la máquina física, a la que encarga la ejecución de los acuerdos alcanzados con las órdenes.
- Holón coordinador: realiza las funciones de ayudar a la programación descentralizada, generar programas globales que sirven de consejo para la toma de decisión, y monitorizar el sistema.

Como se puede observar en la Figura 3, existen tres niveles de interacción entre holones orden y holones máquina: (1) formación de holarquías, a través de las que se forman asociaciones entre órdenes y máquinas; (2) negociación, mediante la que se realizan programas factibles y eficientes; y (3) colaboración, que permite la asignación final de operaciones a máquinas mediante acuerdos de ejecución. Estas interacciones son independientes y se realizan en paralelo.

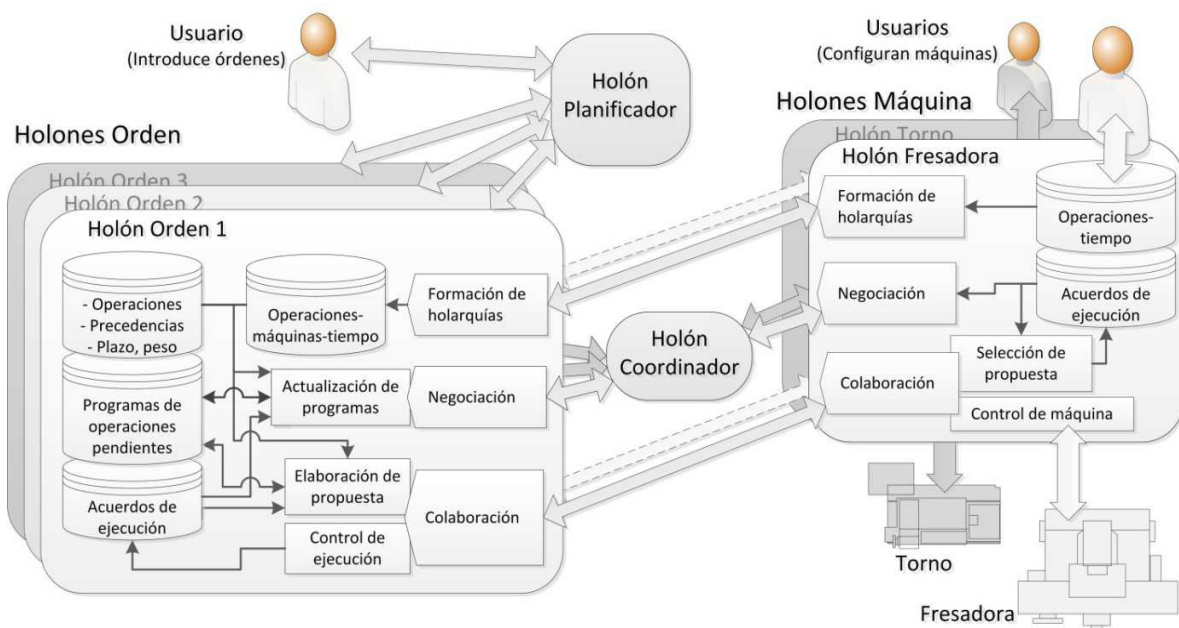


Figura 3: Representación esquemática del sistema propuesto para una planta con 2 máquinas (fresadora y torno) en una situación con tres órdenes activas.

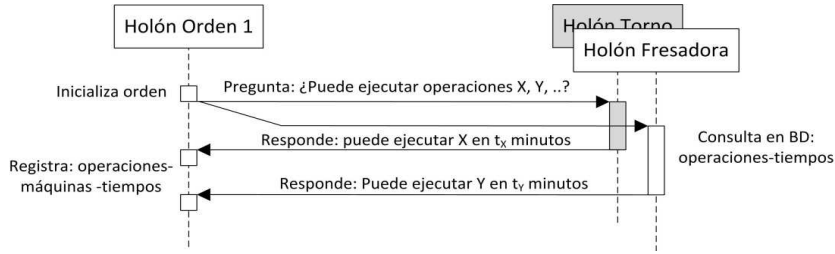


Figura 4: Procedimiento de formación de holarquías de un holón orden.

4.1. Formación de Holarquías

Con objeto de mejorar la flexibilidad, capacidad de reacción y extensibilidad del sistema, las operaciones de las órdenes no están asociadas a priori a las máquinas. Mediante el procedimiento de formación de holarquías es como se consigue esta asociación (Figura 4). El procedimiento lo inicia cada holón orden tras ser creado, preguntando a cada holón máquina si es capaz de fabricar sus operaciones. Tras consultar en sus bases de datos, los holones máquina responden indicando si pueden ejecutar la operación y, en su caso, el tiempo de ejecución. Finalmente, los holones orden lo registran estableciendo la holarquia (asociación de las órdenes con las máquinas que pueden ejecutar sus operaciones).

4.2. Colaboración: Formalización de Acuerdos de Ejecución

El mecanismo de colaboración (Figura 5) va a permitir la asignación de operaciones a las máquinas y su seguimiento, de forma dinámica, completamente distribuida (no se utiliza información global) y asíncrona (cada holón participa en el procedimiento cuando lo estima conveniente).

El procedimiento lo inicia cualquier holón orden, con el objetivo de conseguir un acuerdo con un holón máquina para la ejecución de una operación. Para ello seleccionan una operación de entre aquellas que cumplan las dos condiciones siguientes: (1) no se han formalizado aún acuerdos de ejecución para ellas, y (2) sus precedentes tienen formalizados acuerdos de ejecución. Además, dado que es posible que la operación se pueda ejecutar en varias máquinas con diferente tiempo de procesamiento, se deberá elegir una de ellas (los criterios de selección se exponen en el apartado 4.4). Una vez seleccionada la operación y la máquina, el holón orden elabora una propuesta y se la envía al holón máquina, indicando la fecha más temprana de comienzo (aquella en la que todas sus precedentes inmediatas hayan finalizado), y una fecha deseada de finalización que dependerá de los criterios de decisión utilizados (apartado 4.4).

Cuando la máquina recibe una propuesta de ejecución, la registra como pendiente de respuesta y calcula una fecha de respuesta. Durante el intervalo de tiempo que transcurre entre la llegada de la propuesta y la fecha de respuesta, el holón máquina podrá recibir propuestas de otras órdenes. Éstas se tratarán de la misma forma, se registran como pendientes y se recalcula la fecha de respuesta.

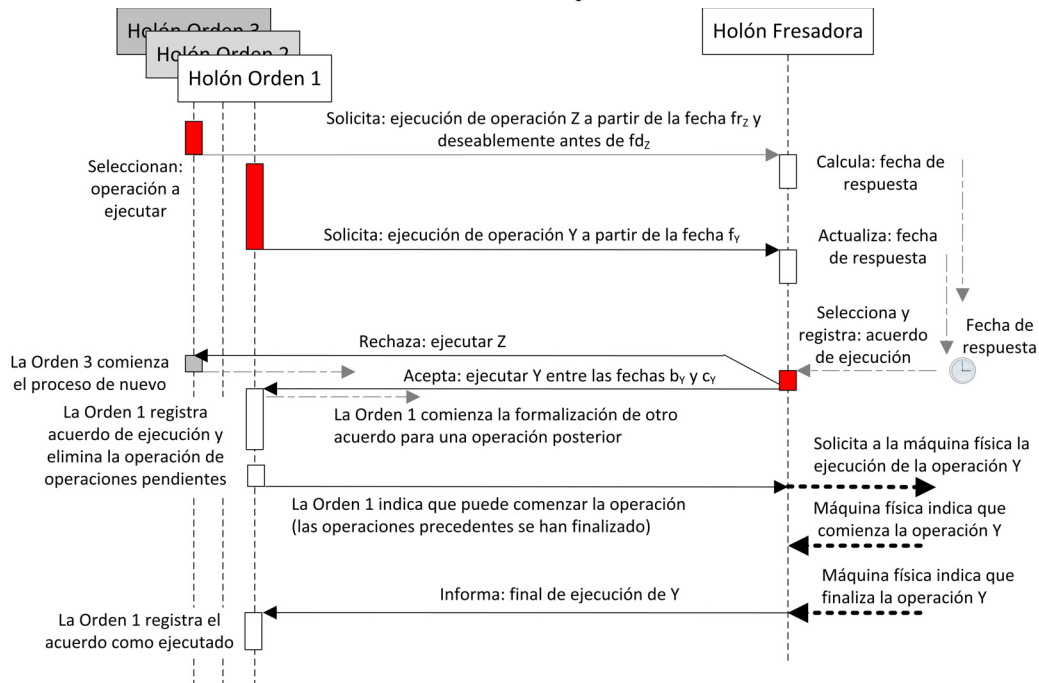


Figura 5: Procedimiento de asignación de operaciones a máquinas “acuerdos de ejecución”.

La fecha de respuesta se calcula como la fecha de finalización más temprana posible de las operaciones pendientes de respuesta, menos una constante. Esta constante debe ser igual para todas las máquinas y mayor que la duración de cualquier operación. Esto garantiza que la máquina espere lo suficiente para que se puedan acumular las solicitudes de varias órdenes, pudiendo así seleccionar entre más opciones. Cuando llega la fecha de respuesta, el holón máquina selecciona una propuesta (operación) de entre todas aquellas que están pendientes de respuesta. Se hace en dos fases: (1) se selecciona el conjunto de operaciones pendiente de respuesta que puedan comenzar antes de la fecha de finalización más temprana (la de la operación que antes pueda finalizar), y (2) de entre todas las propuestas que superen la primera selección se escoge aquella con plazo deseado de finalización más temprano. Este procedimiento es una versión propia, distribuida y en tiempo de ejecución del algoritmo de Giffler-Thompson de construcción de programas activos (Blazewicz et al. 2007).

Tras seleccionar la operación, el holón máquina programa su ejecución, considerando la fecha más temprana de comienzo indicada por la orden (fecha de disponibilidad de la operación), la fecha más temprana de disponibilidad de la máquina (según los acuerdos de ejecución que tenga pendientes) y la duración de la operación. Después se lo comunica a la orden correspondiente, mientras que indica al resto de órdenes el rechazo de su propuesta. Cuando las órdenes reciben la notificación, comienzan de nuevo el procedimiento, bien con otra operación, o bien con la misma en caso de haber sido rechazada su propuesta.

En el momento en que se pueda comenzar una operación acordada (porque han finalizado sus operaciones precedentes) la orden se lo indica al holón máquina, y este último se lo solicita a la máquina física, que comenzará la operación en cuanto pueda (cuando quede libre). Cuando posteriormente, la máquina física finalice la operación, se lo comunicará al holón máquina y este último al holón orden.

Este nivel de interacción es también el responsable del control on-line. Cuando en una máquina el proceso no va según lo previsto, el holón máquina actualiza las fechas de sus acuerdos de ejecución y se lo comunica a las órdenes implicadas. De esta forma, las perturbaciones se propagan por todo el sistema, adaptando cada holón las nuevas decisiones a la situación actual.

4.3. Negociación: Elaboración de Programas

El procedimiento de formalización de acuerdos permite la perfecta coordinación de actividades, pero no garantiza la eficiencia, que depende de la toma de decisiones de los holones orden. En esta toma de decisiones, que aparece en rojo en la Figura 5- se elegía la operación a lanzar, la máquina, y la fecha deseada de finalización. La selección de la operación y la máquina es importante ya que, por ejemplo, se puede descargar trabajo en cuellos de botella, aunque sea a costa de programar una operación en máquinas menos eficientes. El plazo deseado de finalización, que indica la orden en la propuesta de ejecución, tiene también gran influencia ya que, posteriormente, los holones máquina seleccionan las operaciones con plazos más ajustados.

Aunque las decisiones de los holones orden se podrían tomar con información local -por ejemplo: operación más corta, en máquina más eficiente y con plazo más reducido posible-, es muy difícil que esto conduzca a un comportamiento global eficiente. Por ello proponemos un mecanismo de negociación que permite la elaboración, de forma distribuida y en tiempo de ejecución, de programas eficientes que sirvan de guía para la toma de decisiones. Se elaborarán dos tipos de programas: (1) programas locales, elaborados por cada holón orden, que incluyen sólo sus operaciones pendientes (aquellas para las que aún no hay acuerdos de ejecución); y (2) programas globales, elaborados por el holón coordinador, que resultan de la unión de todos los programas locales.

El mecanismo de programación (Figura 6) se inspira en los mecanismos de asignación eficientes que subyacen en los

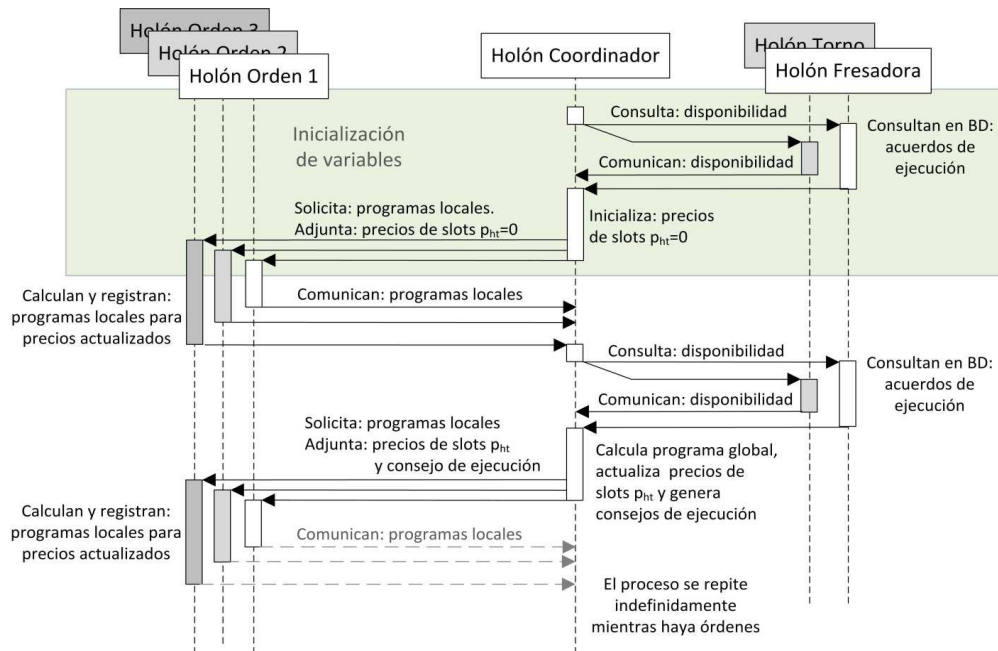


Figura 6: Procedimiento de programación distribuida.

mercados competitivos, en los que los precios son utilizados por los agentes económicos para tomar las decisiones óptimas. En nuestro caso, el bien a vender en el mercado es el tiempo de uso de las máquinas y los compradores de este bien son las órdenes que lo necesitan para ejecutar sus operaciones pendientes. Cada máquina podrá ofertar en el mercado un número determinado de unidades de tiempo, que denominamos *slots*, y que son resultado de dividir el horizonte de planificación en intervalos pequeños. La negociación se produce a través del holón coordinador, que actúa como mercado central donde los holones máquina ofertan sus *slots* disponibles. Nosotros hemos implementado una versión de este procedimiento derivado del método de Relajación Lagrangiana ya que tiene una formulación matemática rigurosa que permite controlar la convergencia y establecer la bondad de la solución obtenida (Liu et al. 2007).

El procedimiento (Figura 6) lo inicia el holón coordinador solicitando a las órdenes que propongan un paquete de *slots* donde asignar sus operaciones, para unos precios dados. Esto se concreta en lo que denominamos programa local: una asignación de máquinas, en determinados intervalos de tiempo, a las operaciones de una orden. Para seleccionar el paquete de *slots* adecuado, las órdenes utilizan un algoritmo enumerativo que evalúa el coste (2) de todos los posibles programas locales, eligiendo finalmente el más económico.

$$C_i(S_i) = \sum_{h \in S_i} p_{ht} + w_i \cdot T_i(S_i)^2 \quad (2)$$

con $S_i \rightarrow$ Paquete de *slots* selecciona dos por la orden i
 $C_i(S_i) \rightarrow$ Coste del paquete S_i
 $p_{ht} \rightarrow$ Precio del *slot* t de la máquina h
 $T_i(S_i) \rightarrow$ Retraso de la orden i suponiendo el programa derivado de S_i

En la Figura 7 puede verse un ejemplo de obtención del programa local óptimo de una orden para unos precios dados. La orden tiene dos operaciones Op1 y Op2 que se ejecutan respectivamente en las máquinas M1 y M2 con una duración de dos minutos. El periodo de planificación se ha dividido en 6 *slots* de 1 minuto cada uno. Para estos *slots* se suponen los siguientes precios: (2,2,1,1,0,0) en M1 y (2,2,2,2,0,0) en M2. Con estos precios el programa local óptimo de la orden es el mostrado en la figura. Si se adelantara la ejecución de las operaciones para que la orden terminara a tiempo se necesitaría utilizar *slots* con mayor precio, aumentando el coste total.

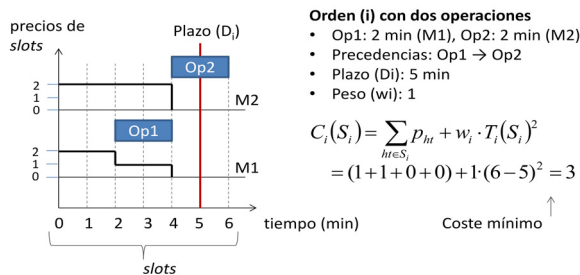


Figura 7: Ejemplo de obtención del programa local óptimo de una orden.

Cuando el holón coordinador recibe todas las asignaciones propuestas (programas locales), las integra obteniendo un programa global. Dado que los programas locales se realizan por separado por cada orden, es normal que se presenten conflictos

(asignaciones no factibles), es decir, que un *slot* de una máquina haya sido solicitado por varias órdenes. En ese caso, el coordinador dará prioridad, en la construcción del programa global, a la operación que antes finalice según el programa local.

Para evitar programas con conflictos en futuras rondas, el coordinador actualiza los precios de modo que la siguiente vez que las órdenes elaboren programas locales, eviten las asignaciones conflictivas. Para ello, aumenta el precio de los *slots* donde se han producido conflictos, haciendo que las órdenes poco importantes eviten esas asignaciones, y disminuye el precio de los *slots* que no han sido asignados (Lavios et al. 2010):

$$p_{ht} = \max(0, p_{ht}^* + \alpha \cdot (d_{ht} - 1)) \quad (3)$$

con $p_{ht}^* \rightarrow$ Precio previo de *slot* t de la máquina h
 $p_{ht} \rightarrow$ Precio actualizado de *slot* t de la máquina h
 $d_{ht} \rightarrow$ Demanda del *slot* t de la máquina h :
 número de órdenes que han programado el uso de ese *slot*
 $\alpha \rightarrow$ Paso : controla la convergencia

Una vez actualizados los precios, el holón coordinador solicita nuevos programas locales a cada orden, incluyendo en la solicitud los precios actualizados, y, además, un consejo de programación para la siguiente operación de la orden. Dicho consejo es obtenido del mejor programa global que previamente ha elaborado el holón coordinador, e incluye: operación, máquina y fechas de comienzo y finalización.

Cuando las órdenes reciben los nuevos precios, recalculan los programas locales y los envían al holón coordinador para que establezca un nuevo programa global y otros nuevos precios. Este proceso se repite continuamente y, con el tiempo, los precios convergen hacia una situación de equilibrio. Estos precios contienen información de la importancia que tiene los diferentes *slots* de las máquinas en la consecución de los objetivos globales. A medida que convergen, los programas locales elaborados por las órdenes serán generalmente más factibles, y los programas globales derivados, generalmente, más eficientes.

4.4. Modos de Funcionamiento

Según se ha explicado en el apartado 4.2, el comportamiento del sistema depende de la toma de decisiones que los holones orden realizan cuando solicitan la ejecución de una operación (zona roja en la Figura 5). La decisión consta de tres elementos: operación, máquina y fecha deseada de finalización. Nosotros hemos implementado tres criterios de toma de decisiones (HD, PD, PC) que supondrán diferentes grados de centralización.

Modo HD (Heurístico Distribuido). En este modo se suprime el mecanismo de programación distribuida y las decisiones se toman únicamente con información local. El holón orden selecciona la operación más corta, en la máquina más eficiente, y calcula la fecha deseada de finalización para esa operación (f_{di}) según el siguiente procedimiento:

- (1) Cálculo del hueco (s_i) o tiempo extra disponible para ejecutar las operaciones pendientes de la orden:

$$s_i = D_i - f_{ri} - (1 + \alpha_i \cdot w_i) \cdot DP_i \quad (4)$$

con $D_i \rightarrow$ Plazo de la orden i
 $f_{ri} \rightarrow$ Fecha de disponibilidad de la operación a programar
 $DP_i \rightarrow$ Duración pendiente de la orden i

α_i → Constante : disminuye la hogura de las órdenes con más peso
 w_i → Peso de la orden i

(2) Cálculo de la fecha deseada de finalización (fd_i):

$$fd_i = \begin{cases} D_i - (1 + \alpha_i \cdot w_i) \cdot DP_i + d_{ijh} \\ fr_i + d_{ijh} + s_i / Z_i \end{cases} \quad (5)$$

con d_{ijh} → Duración de la operación ij en la máquina h
 Z_i → N° de operaciones pendientes de la orden i

Esta forma de calcular el plazo de la operación, junto con el procedimiento de formalización de acuerdos del apartado 4.2 es equivalente al método constructivo de Giffier-Thomson, con una regla heurística basada en el huelgo (*minimum slack*), pero en este caso, de forma distribuida y en tiempo de ejecución. La corrección que se hace de la duración pendiente en función del peso ($1 + \alpha_i \cdot w_i$) tiene por objeto dar preferencia a las órdenes de mayor peso. Los valores del parámetro α_i se han estimado posteriormente mediante simulación, buscando obtener comportamientos eficientes.

Este modo de funcionamiento es una versión distribuida de los procedimientos utilizados actualmente para la programación de operaciones en sistemas productivos muy dinámicos y estocásticos, por lo que se podrá utilizar como referencia para evaluar el funcionamiento de los mecanismos propuestos en este trabajo y que se describen a continuación.

Modo PD (Programación Distribuida). Los holones orden toman las decisiones según el programa local realizado mediante el procedimiento de programación distribuida del apartado 4.3. De él obtienen la operación a ejecutar, la máquina, y la fecha de finalización deseada para esa operación.

Modo PC (Programación Centralizada). El procedimiento es similar al anterior, exceptuando que los holones orden toman la decisión en función del consejo emitido por el holón coordinador, que lo obtiene del mejor programa global elaborado hasta el momento.

5. Implementación y Resultados Experimentales

El sistema descrito se ha implementado utilizando la tecnología software de sistemas multiagente, ampliamente aceptada por la comunidad de sistema holónicos (Leitão y Vrba 2011). Más concretamente, se ha recurrido a JADE (Bellifemine et al. 2007), una plataforma Java orientada a dicha tecnología. JADE proporciona un entorno de ejecución para agentes software, una arquitectura acorde con las especificaciones FIPA, y una serie de librerías Java que facilitan implementar los principales componentes de los agentes: los comportamientos y las interacciones. Nosotros hemos implementado cada tipo de holón como agentes JADE, usando para ello la clase *Agent* proporcionada para tal efecto. Esta clase, que tiene ya implementadas las funcionalidades básicas de los agentes (búsqueda de otros agentes, comunicación, gestión de comportamientos, gestión de la cola de menajes, etc.), se ha extendido para crear las clases que representan cada tipo holón de nuestro sistema (planificador, coordinador, orden y máquina), añadiendo en cada caso los comportamientos e interacciones descritos en el apartado anterior. JADE también proporciona clases que soportan, mediante extensión, la implementación de

comportamientos e interacciones, y todo ello, según los estándares FIPA.

5.1. Planteamiento de la Experimentación

Como primer paso para probar el sistema holónico propuesto se ha optado por simular su interacción con el sistema físico (máquinas) en vez de conectarlo a sistema físico real. Esta interacción, que se produce a través de los holones máquina, está incluida en las interacciones de colaboración (formalización de acuerdos) representadas en la Figura 5, y consisten en lo siguiente. Cuando un holón máquina ha acordado la ejecución de una operación con un holón orden, el primero encarga a la máquina física su ejecución. Posteriormente la máquina física comunicará a su holón, tanto el momento de comienzo como el momento de finalización de la operación. Dado que no se dispone de máquinas físicas reales, este intercambio de mensajes se ha simulado: cuando un holón máquina acuerda la ejecución de una operación con un holón orden y llega el momento en que puede comenzar la operación, el holón máquina programa un disparador de eventos para que una vez transcurrido el tiempo que dura la operación se dispare una señal que es interpretada por el holón máquina como “operación finalizada”.

Los objetivos de flexibilidad, adaptabilidad, rapidez de respuesta y robustez, están asegurados por la propia arquitectura distribuida del sistema y las herramientas utilizadas para su implementación (Leitão y Restivo 2008, Blanc et al. 2008). Pero dado que al distribuir el sistema, es posible que se pierda eficiencia y repetitividad, las pruebas realizadas están destinadas a evaluar estos últimos aspectos respecto a la función objetivo seleccionada. Y ello, comparando los resultados obtenidos con los modos propuestos por nosotros, el PC y el PD, con los obtenidos con el modo HD, similar este último a los métodos basados en reglas de prioridad utilizados en la actualidad en los sistemas dinámicos y estocásticos.

Para probar el sistema se han definido 12 problemas tipo *job shop* con cinco máquinas. Para ello se han establecido 26 procesos de fabricación, cada uno de ellos destinado a fabricar un tipo de producto. Cada proceso tiene de 3 a 5 operaciones, que se ejecutan en una máquina seleccionada al azar y con unas duraciones generadas aleatoriamente según una distribución uniforme entre 20 y 140 segundos. De los 12 problemas, 4 de ellos tienen 5 órdenes, otros 4 tienen 7 órdenes, y los 4 últimos 10 órdenes. Cada orden consiste en la fabricación de un producto (seleccionado al azar entre los 26 generados), con un plazo de finalización (D_i) y un peso (w_i). Los pesos de las órdenes se han generado también de forma aleatoria entre 1 y 5. Lo mismo se ha hecho con los plazos de entrega, que están entre 5 y 7 minutos en los problemas pequeños -los de 5 órdenes-, entre 5 y 10 en los medianos y entre 5 y 15 en los grandes.

Se han realizado tres grupos de experimentos. Un primer grupo en el que utilizando los problemas estáticos, se evalúa la eficiencia y la predictibilidad de los diferentes modos de funcionamiento del sistema. Un segundo grupo en el que se evalúa el funcionamiento del sistema en entornos dinámicos, donde en cada problema aparecen nuevas órdenes en tiempo de ejecución. Y finalmente, un tercer grupo con problemas estocásticos, aquellos en los que las duraciones de las operaciones pueden desviarse de lo previsto.

Los experimentos se han realizado en tiempo real, es decir, la ejecución de las actividades, aunque simulada, durará lo que duraría en un sistema real. Se ha hecho así para comprobar el

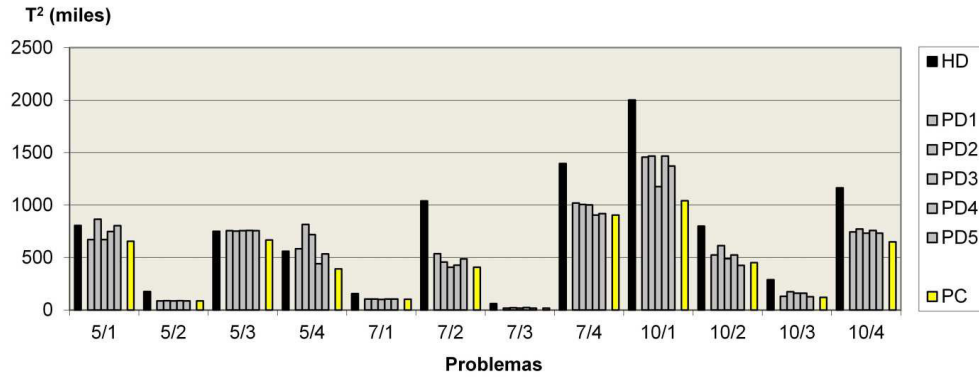


Figura 8: Valor de T^2 obtenido en cada problema estático en diferentes modos de funcionamiento

funcionamiento del sistema en circunstancias lo más parecidas posibles a la realidad, sobre todo en los experimentos dinámicos y estocásticos que tratan de comprobar la capacidad que tiene el sistema de reajustarse a nuevas situaciones en tiempo de ejecución.

5.2. Caso Estático

Simulando la ejecución de cada problema estático, con los diferentes modos de funcionamiento propuestos, se han obtenido los resultados que se muestran en la Figura 8. En ella se representa el valor de la función de evaluación T^2 (suma ponderada de retrasos al cuadrado) para cada problema en los distintos modos de funcionamiento. El primer dígito de la notación de cada problema indica el número de órdenes en ese problema, el segundo dígito es un índice correlativo.

La barra negra del gráfico representa el mejor de los resultados obtenidos con el modo HD, con el que se han realizado diferentes simulaciones con diversos valores de α_1 (0.1, 0.2, 0.5 y 0.8) y se ha seleccionado la mejor de ellas. Este valor constituye una cota superior para comprobar hasta qué punto los métodos de programación propuestos mejoran los resultados.

Se observan diferencias de comportamiento entre el modo de programación centralizada PC y el distribuido PD. Mientras que con el primero los resultados sobre el mismo problema son repetibles, con el segundo, el sistema muestra un comportamiento variable en la mayoría de los problemas. En ese caso, nunca se mejoran los resultados conseguidos con la versión centralizada del mecanismo de toma de decisión, dándose incluso algunas situaciones -pocas- donde el comportamiento global es peor que el mejor de los exhibidos con el modo HD. Ese comportamiento era previsible y por ello se decidió mostrar los resultados de varias réplicas del mismo experimento. Esto se debe a la naturaleza del método de programación distribuida, donde es prácticamente imposible que en dos ejecuciones del mismo experimento los precios sean exactamente iguales cuando las órdenes toman la decisión. Todo esto se traduce en cierta aleatoriedad.

5.3. Caso Dinámico

Después de las pruebas estáticas, se han realizado pruebas dinámicas sobre cada problema, añadiendo órdenes en tiempo de ejecución que se suman a las ya lanzadas en el modo estático.

Tradicionalmente, para programar las operaciones en entornos dinámicos se ha utilizado métodos de lanzamiento parecidos a los

procedimientos que hay detrás del modo HD de nuestro sistema. Nos interesa comprobar hasta qué punto los modos con programación pueden mejorar los resultados de estos métodos en entornos dinámicos. Para ello vamos a comparar las soluciones obtenidas con el modo HD y el modo PC.

La Figura 9 muestra los resultados obtenidos para 11 problemas derivados del caso 7/1, cada uno de ellos con un grado de dinamismo diferente: en el problema 1 se emite una orden en tiempo de ejecución, en el problema 2 se emiten dos órdenes, y así sucesivamente hasta el problema 11. Los órdenes se lanzan cada 60 segundos. En la Figura 9 se muestran los resultados, como porcentaje de la solución obtenida con el modo PC frente a la obtenida con el modo HD. Se puede observar cómo la eficiencia del procedimiento PC sigue siendo significativamente mejor frente al método de HD, incluso en condiciones de alto dinamismo. Este resultado es similar si se coge otro problema de partida en vez del 7/1. También se han realizado simulaciones con el modo PD obteniéndose el mismo comportamiento que en el caso estático: los resultados se acercan en calidad a los obtenidos con el modo PC, pero al igual que en los problemas estáticos, no son repetibles.

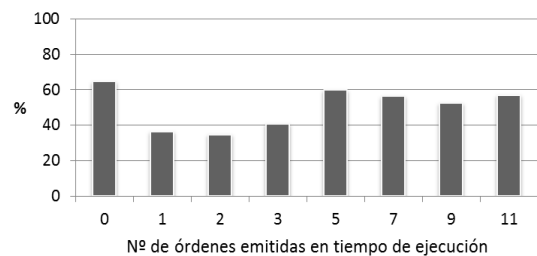


Figura 9: Comparación de resultados obtenidos en el modo PC frente a los obtenidos en el modo HD para 11 variaciones del problema 7/1. Las variaciones se derivan de introducir diferentes cantidades de órdenes en tiempo de ejecución.

5.4. Caso Estocástico

Para evaluar el comportamiento en entornos estocásticos se introduce un factor de aleatoriedad en tiempo de proceso de las operaciones. Cada operación puede sufrir un alargamiento inesperado con una probabilidad que denotamos por Pa . Una vez que se ha producido el alargamiento, éste será de un 30% de la duración prevista. Con esto se pretende estudiar el

comportamiento del sistema en escenarios con diferentes probabilidades de perturbación.

La experimentación se ha realizado del siguiente modo: (1) para el problema 10/1 se han generado 15 escenarios de perturbación, cinco con probabilidad de alargamiento $P_a=0.1$, otros cinco con $P_a=0.3$, y otros cinco con $P_a=0.5$, y (2) para cada uno de esos escenarios se ha simulado el comportamiento del sistema en los modos HD y PC. De esta forma, las perturbaciones en cada uno de los escenarios son iguales para las simulaciones realizadas con HD y PC, y los resultados son comparables.

La Figura 10 muestra los resultados de estos experimentos. Cada punto representa la solución obtenida con el modo PC en porcentaje sobre a la obtenida con el modo HD para un escenario de perturbaciones determinado. La línea central de la figura muestra la media para los escenarios generados con la misma probabilidad de alargamiento. Los resultados revelan que las ventajas del comportamiento PC se van perdiendo a medida que la probabilidad de perturbación aumenta.

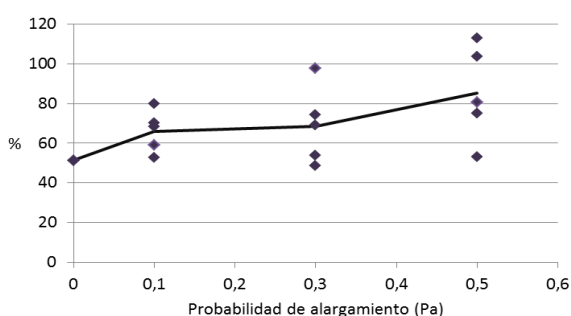


Figura 10: Comparación de resultados obtenidos en el modo PC frente a los obtenidos en el modo HD para 15 escenarios estocásticos derivados del problema 10/1.

6. Conclusiones

El paradigma de fabricación holónica aporta una nueva forma de abordar la programación y control de sistemas de fabricación flexibles. La idea de distribuir el sistema en entidades autónomas, en estrecha relación con la estructura de la planta y los productos que allí se fabrican, modifica radicalmente el desarrollo posterior del sistema. Mientras que en las aproximaciones tradicionales existen elementos centralizadores que poseen un modelo global de la planta, que la integra y la conduce a funcionamientos eficientes, en la aproximación holónica no es así. La toma de decisiones es distribuida, aunque pudiendo estar, en algunos casos, dirigida por consejos emitidos por holones de carácter centralizado.

La arquitectura propuesta, distribuida según órdenes y máquinas, así como la tecnología utilizada para implementar el sistema, lo dotan de las características deseadas de flexibilidad, reconfigurabilidad, adaptabilidad, rapidez de respuesta, robustez y mantenibilidad:

- **Flexibilidad:** la asignaciones de trabajos a máquinas se realizan en tiempo de ejecución pudiéndose cambiar la ruta de fabricación cuando sea conveniente, lo que mejora la flexibilidad del proceso. El sistema es capaz de gestionar eficientemente el enrutamiento flexible.
- **Reconfigurabilidad, adaptabilidad:** en el sistema se pueden incluir nuevos equipos y nuevas órdenes en tiempo de

ejecución. El sistema tiene la capacidad para establecer de forma automática las relaciones que implican la nueva situación.

- **Rapidez de respuesta:** si falla una máquina, el propio sistema puede sustituirla rápidamente en tiempo de ejecución aprovechando la capacidad de enrutamiento flexible.
- **Robustez:** el fallo de un holón orden o máquina no supone el fallo global del sistema, solo de aquellos procesos en los que interviene esa máquina o esa orden.
- **Mantenibilidad del software, adaptabilidad:** al haber una correspondencia entre la estructura del sistema de fabricación y la estructura del sistema software se facilita la evolución del software y mejora su reusabilidad. Así, si es necesario introducir órdenes o máquinas con nuevas características, se podrían programar solamente esos nuevos holones de forma que interactúen con los ya existentes según los protocolos establecidos.

Pero al distribuir la toma de decisión en varias entidades, queda por comprobar si el sistema es predecible y globalmente eficiente. Los resultados obtenidos con los problemas estáticos muestran claramente que el modo distribuido PD, aunque con un comportamiento aceptable, es más inestable e ineficiente que el centralizado PC, el cual, por su filosofía, sacrifica algunas de las ventajas propias de los sistemas holónicos indicadas en el párrafo anterior. De media, la mejora obtenida con el método PC respecto a la solución obtenida con el método HD es del 42%, mientras que la mejora obtenida con el método PD es del 32%. Aun así, los resultados obtenidos con el modo de programación distribuida (PD) mejoran, en muchos casos, a los obtenidos con el modo HD. De hecho, la mejora se produce en 51 de las 66 simulaciones realizadas con el método PD respecto a los resultados obtenidos con el método HD. Esta mejora es más significativa, un 37% de media en el valor de T^2 , en los problemas de más tamaño (10 órdenes), más parecidos a los casos reales. Estas diferencias son estadísticamente significativas con un nivel de confianza del 95% según el test U de Mann-Whitney.

Los resultados aportados muestran el buen comportamiento del sistema en casos dinámicos y estocásticos, donde los modos PD y PC, superan visiblemente a las técnicas de lanzamiento, representadas por el modo HD, y típicamente usadas en este tipo de entornos. En este caso, los resultado obtenidos con el modo PD se acercan en calidad a los obtenidos con el modo PC, pero al igual que en los problemas estáticos, estos últimos no son repetibles. Las mejoras obtenidas con respecto al modo HD llegan a ser del 60% en algún caso. Cabe destacar que en los problemas aleatorios, las mejoras obtenidas con nuestras propuestas de programación, se hacen más irrelevantes cuando la aleatoriedad es muy elevada, pero en promedio los resultados no son peores que los obtenidos con el modo HD.

Como investigación futura sería interesante trabajar sobre el modo PD diseñando nuevos mecanismos distribuidos de programación y criterios de toma de decisión más predecibles y eficientes. Esto permitiría disponer de un procedimiento con características holónicas muy marcadas, con las ventajas que ello supone, y a la vez muy eficiente y predecible.

Otra posible línea de investigación sería introducir más complejidad en el sistema, considerando nuevos aspectos como transporte de materiales, tiempos de preparación dependientes de la secuencia o colas de máquina finitas. También cabe la posibilidad de probar el software en una planta real.

English Summary

Scheduling and Control of Flexible Manufacturing Systems: a Holonic Approach

Abstract

Holonic Manufacturing Systems are a new manufacturing paradigm. They are included into distributed methodologies and propose new manufacturing management schemes that combine self-organization, dynamic hierarchies and horizontal relationships. This approach is closer to manufacturing system nature than centralized or strictly traditional hierarchical approaches. It allows modeling manufacturing systems in a more realistic way. So, it make easier to implement more flexible, scalable, responsiveness, reconfigurable, and robust production management systems. In this paper we propose a holonic system for scheduling and control of flexible manufacturing systems. The system is based on a new architecture that integrates two decision levels: a level of auction-based scheduling, where each holon performs its own local schedule, and another level of dispatching and control that coordinate holons activities. Unlike other studies in which local schedules are integrated into a global schedule that centralizes the system, the proposed system is not so. The dispatching and control mechanism is responsible to integrate the local programs. In addition, the developed prototype can operate with different degrees of decentralization, which allows evaluating the advantages that our approach provides by means of simulation.

Keywords:

Production control, production systems, scheduling algorithms, flexible manufacturing systems, intelligent manufacturing systems.

Agradecimientos

Los autores agradecen el apoyo de la Junta de Castilla y León, Proyecto VA056A12-2.

Referencias

- Babiceanu RF, Chen FF (2006). Development and applications of holonic manufacturing systems: A survey. *Journal of Intelligent Manufacturing*, 17(1):111–131.
- Babiceanu RF, Chen FF, Sturges RH (2004). Framework for the control of automated material handling systems using the holonic manufacturing approach. *International Journal of Production Research*, 42(17):3551–3564.
- Bellifemine FL, Caire G, Greenwood D (2007). *Developing multi-agent systems with JADE*. Chichester: Wiley.
- Blanc P, Demongodin I, Castagna P (2008). A holonic approach for manufacturing execution system design: An industrial application. *Engineering Applications of Artificial Intelligence*, 21(3):315–330.
- Blazewicz J, Ecker K, Pesch E, Schmidt G, Weglarz J (2007). *Handbook on scheduling: Models and methods for advanced planning*. New York: Springer.
- Bongaerst L (1998). *Integration of scheduling and control in holonic manufacturing systems*. Ph.D. Dissertation. Leuven: PMA Division of Leuven K.U.
- Bongaerst L, Monostori L, McFarlane M, Kádár B (2000). Hierarchy in distributed shop floor control. *Computers in Industry*, 43(2):123–137.
- Borangiu T, Thomas A, Trentesaux D (2012). *Service orientation in holonic and multi-agent manufacturing control*. London: Springer.
- Botti V, Giret A (2008). *Anemona: A multi-agent methodology for holonic manufacturing systems*. London: Springer.
- Chacón E, Indriago M (2010). Implementación de Supervisores en una Arquitectura de Referencia Basadas en Sistemas de Manufactura Holónicos. *Revista Iberoamericana de Automática e Informática Industrial*, 7(3):72–82.
- Christensen JH (2003). HMS/FB architecture and its implementation. In SM Deen (ed.), *Agent-Based Manufacturing: Advances in the Holonic Approach*, 53–88. Berlin: Springer.
- Fisher, M.L., (2004). The Lagrangian Relaxation Method for Solving Integer Programming Problems. *Management Science*, 50(12), pp.1861–1871.
- Gou L, Luh PB, Kyoya Y (1998). Holonic manufacturing scheduling: architecture, cooperation mechanism, and implementation. *Computers in Industry*, 37(3):213–231.
- Höpf M (1994). Holonic manufacturing systems (HMS). The basic concept and a report of IMS test Case 5. In JK Knudsen, PA MacConaill, J Bastos (eds.), *Advances in Design and Manufacturing 5*, 84–93. Amsterdam: IOS Press.
- Hsieh FS (2008). Holarchy formation and optimization in holonic manufacturing systems with contract net. *Automatica*, 44(4):959–970.
- Koestler A (1967). *The ghost in the machine*. London: Hutchinson.
- Lavios JJ, del Olmo R, Araújo JA, Galán JM (2010). Price updating in combinatorial auctions for coordination of manufacturing multiagent systems. In Y Demazeau et al. (eds.), *Trends in Practical Applications of Agents and Multiagent Systems*, 201–207. Berlin: Springer.
- Leitão P., Restivo F. (2006). ADACOR: A holonic architecture for agile and adaptive manufacturing control. *Computers in Industry* 57(2):121–130.
- Leitão P., Restivo F. (2008). A holonic approach to dynamic manufacturing scheduling. *Robotics and Computer-Integrated Manufacturing*, 24(5):625–634.
- Leitão P., Restivo, F. (2000). A framework for distributed manufacturing applications. *Proceeding of 2000 Advanced Summer Institute – Life Cycle Approaches to Production Systems: Management, Control and Supervision*, 75–80.
- Leitão P., Vrba P. (2011). Recent developments and future trends of industrial Agents. *Lecture Notes in Computer Science*, 6867:15–28.
- Liu N, Abdelrahman MA, Ramaswamy S (2007). A complete multiagent framework for robust and adaptable dynamic job shop scheduling. *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, 37(5):904–916.
- Luh, P.B., Hoitomt, D.J. (1993). Scheduling of manufacturing systems using the Lagrangian relaxation technique. *IEEE Transactions on Automatic Control*, 38(7), pp. 1066–1079.
- Nahmias S (2013). *Production and Operations Analysis*, 6th Revised edition. London: McGraw Hill Higher Education.
- Neligwa T, Fletcher M (2003). An HMS operational model. In SM Deen (ed.), *Agent-Based Manufacturing: Advances in the Holonic Approach*, 163–191. Berlin: Springer.
- Ounnar F, Pujo P (2012). Pull control for job shop: holonic manufacturing system approach using multicriteria decision-making. *Journal of Intelligent Manufacturing*, 23(1):141–153.
- Parunak VD (1987). Manufacturing experience with the contract net. In MN Huhns (ed.), *Distributed Artificial Intelligence*, 285–310. London: Pitman.
- Pinedo ML (2009). *Planning and scheduling in manufacturing and services*. New York: Springer.
- Rosenzweig ED, Easton GS (2013). Tradeoffs in Manufacturing? A Meta-Analysis and Critique of the Literature. *Production and Operations Management*, 19(2):127–141
- Roulet-Dubonnet O, Ystgaard P (2011). An application of the holonic manufacturing system to a flexible assembly cell. *Lecture Notes in Computer Science*, 6867:29–38.
- Wonga TN, Leunga CW, Maka KL, Fungb RYK (2006). Dynamic shopfloor scheduling in multi-agent manufacturing systems. *Expert Systems with Applications*, 31(3):486–494.
- Wyns J (1999). *Architecture for holonic manufacturing systems - The key to support evolution and reconfiguration*. Ph.D. Dissertation. Leuven: PMA Division of Leuven K.U.