

SECUENCIACIÓN DINÁMICA DE SISTEMAS DE FABRICACIÓN FLEXIBLE MEDIANTE APRENDIZAJE AUTOMÁTICO: ANÁLISIS DE LOS PRINCIPALES SISTEMAS DE SECUENCIACIÓN EXISTENTES

PAOLO PRIORE*

DAVID DE LA FUENTE*

JAVIER PUENTE*

ALBERTO GÓMEZ*

Una forma habitual de secuenciar de modo dinámico los trabajos en los sistemas de fabricación es mediante el empleo de reglas de secuenciación. Sin embargo, el problema que presenta este método es que el comportamiento del sistema de fabricación dependerá de su estado, y no existe una regla que supere a las demás en todos los posibles estados que puede presentar el sistema de fabricación. Por lo tanto, sería interesante usar en cada momento, la regla más adecuada. Para lograr este objetivo, se pueden utilizar sistemas de secuenciación que emplean aprendizaje automático que permiten, analizando el comportamiento previo del sistema de fabricación (ejemplos de entrenamiento), obtener el conocimiento necesario para determinar la regla de secuenciación más apropiada en cada instante. En el presente trabajo se realiza una revisión de los principales sistemas de secuenciación, existentes en la literatura, que utilizan aprendizaje automático para variar de forma dinámica la regla de secuenciación empleada en cada momento.

Dynamic scheduling of flexible manufacturing systems through machine learning: an analysis of the main scheduling systems

Palabras clave: Secuenciación dinámica, aprendizaje automático, sistemas de fabricación flexible, reglas de secuenciación, simulación

Clasificación AMS (MSC 2000): 68T20, 68M20

*Dpto. de Administración de Empresas y Contabilidad. Escuela Técnica Superior de Ingenieros Industriales e Informáticos de Gijón. Universidad de Oviedo.

–Recibido en diciembre de 1999.

–Aceptado en julio de 2001.

1. INTRODUCCIÓN

La secuenciación de trabajos, que forma parte del proceso de control en un sistema de fabricación, es necesaria cuando un conjunto común de recursos debe ser compartido, para fabricar una serie de productos durante el mismo período de tiempo. El objetivo de la secuenciación es la asignación eficiente de máquinas y otros recursos a los trabajos, o a las operaciones contenidas en éstos, y la determinación del momento en el que cada uno de estos trabajos debe procesarse (Shaw *et al.*, 1992).

Los tiempos de procesamiento en los sistemas de fabricación flexible (*FMSs*¹) son casi determinísticos puesto que las operaciones son controladas por ordenador y procesadas, en su mayor parte, por máquinas de control numérico; asimismo, las preparaciones entre operaciones consecutivas están automatizadas. Por lo tanto, si no hay perturbaciones en el *FMS*, se puede predecir el resultado y es suficiente un sistema de secuenciación fijo y «off-line».

Sin embargo, debido a la llegada de nuevos trabajos (en ocasiones urgentes), averías en las máquinas y otras perturbaciones, el estado de un *FMS* puede no ser predecible. Esta naturaleza incierta y dinámica sugiere que un sistema de secuenciación de tipo «off-line» no es el más adecuado. Asimismo, los *FMSs* son más sensibles a las perturbaciones que los sistemas de fabricación convencionales puesto que sus componentes están más sincronizados e integrados y son más interdependientes. Por lo tanto, se requiere una respuesta inmediata a los cambios en los estados del *FMS*, mediante la utilización de un sistema de secuenciación en tiempo real. Si los estados del *FMS* cambian de modo dinámico, la secuenciación de trabajos debe hacerse en función del estado actual de éste (Jeong y Kim, 1998).

El resto del artículo está organizado de la siguiente forma. En primer lugar, se define el aprendizaje automático y se describen los tipos principales de sistemas de secuenciación de *FMSs* existentes en la literatura. Posteriormente, para superar el problema que presenta las reglas de secuenciación cuando se emplean de forma estática, se definen dos métodos para modificar estas reglas de modo dinámico. Uno de ellos está basado en la utilización de un modelo de simulación, mientras que el otro emplea «conocimiento de secuenciación» del sistema de fabricación. A continuación, se realiza una revisión de los principales trabajos que utilizan este último método. Finalmente, se especifican una serie de carencias comunes de los sistemas de secuenciación basados en el conocimiento cuya resolución constituye una fuente de futuras líneas de investigación.

¹Del inglés, Flexible Manufacturing System.

2. APRENDIZAJE AUTOMÁTICO

El aprendizaje automático, que pertenece al campo de la inteligencia artificial, permite resolver problemas mediante el empleo del conocimiento obtenido de problemas resueltos en el pasado similares al actual (Michalski *et al.*, 1983). Una representación esquemática de lo expuesto anteriormente se muestra en la figura 1. Los tipos principales de algoritmos dentro del aprendizaje automático son las redes neuronales (Bishop, 1995), el aprendizaje inductivo (Quinlan, 1993) y el razonamiento basado en casos (CBR²; Watson, 1997). La diferencia fundamental entre estos tipos de algoritmos radica en la forma en que se almacena el conocimiento. Así, en las redes neuronales, el conocimiento se traduce en una serie de pesos y umbrales que poseen las neuronas. En cambio, en el aprendizaje inductivo, el conocimiento se transforma en un árbol de decisión o un conjunto de reglas. Por último, en el CBR, el conocimiento está formado por una base de casos compuesta por los problemas resueltos en el pasado.

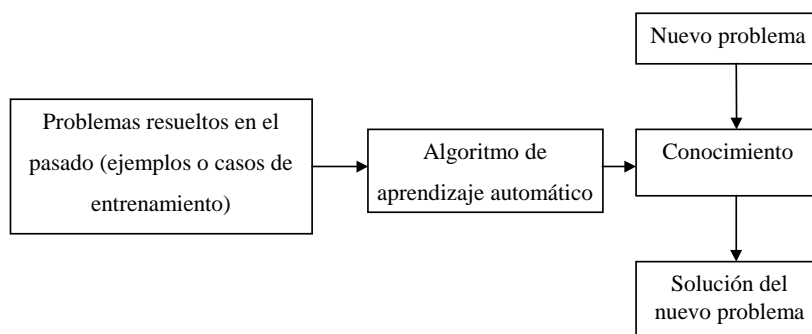


Figura 1. Esquema general del problema de aprendizaje.

Los casos o problemas resueltos en el pasado (también denominados ejemplos o casos de entrenamiento) se pueden representar por tablas atributo-valor, como la que se muestra en la tabla 1, donde hay un atributo especial que se denomina la clase (la solución del problema resuelto). Los atributos representan las características del problema. El objetivo que tiene un algoritmo de aprendizaje automático es tratar de aprender a clasificar nuevos casos, similares a los de entrenamiento, de los que se conocen los valores de todos los atributos excepto la clase. En esta situación, el término clasificar se utiliza en el sentido más literal; es decir, determinar cuál es la clase de un nuevo caso o ejemplo.

²Del inglés, Case-Based Reasoning.

Tabla 1. Tabla atributo-valor de los ejemplos de entrenamiento.

<i>Ejemplos</i>	<i>Atributo 1</i>	<i>Atributo 2</i>	<i>.....</i>	<i>Atributo m</i>	<i>Clase</i>
Ejemplo 1	A ₁₁	A ₁₂	A _{1m}	C ₁
Ejemplo 2	A ₂₁	A ₂₂	A _{2m}	C ₂
.....
Ejemplo n	A _{n1}	A _{n2}	A _{nm}	C _n

3. SISTEMAS DE SECUENCIACIÓN EN LOS FMSs

Los sistemas de secuenciación en los *FMSs* se pueden dividir en las siguientes categorías:

1. Sistemas basados en métodos analíticos.
2. Sistemas basados en métodos heurísticos.
3. Sistemas basados en simulación.
4. Sistemas basados en inteligencia artificial.

Los sistemas analíticos formulan el problema de secuenciación de un *FMS* como un modelo de optimización con restricciones, en términos de una función objetivo y restricciones explícitas. Posteriormente, se resuelve el modelo utilizando alguno de los algoritmos de resolución existentes (ver por ejemplo, Han *et al.*, 1989; Hutchison *et al.*, 1989; Kimemia y Gershwin, 1985; Lashkari *et al.*, 1987; Shanker y Rajamarthandan, 1989; Shanker y Tzen, 1985; Stecke, 1983; Wilson, 1989).

En general estos problemas son de tipo NP-completo (Garey y Johnson, 1979), por lo que normalmente se proponen algoritmos de tipo heurístico y «off-line» para su resolución (Chen y Yih, 1996; Cho y Wysk, 1993). Asimismo, estos modelos analíticos poseen simplificaciones que no siempre son válidas en la práctica (de hecho, Basnet y Mize (1994) afirman que algunos modelos son tan singulares que parece que los problemas son ideados para encajar en el modelo y no viceversa) y no son eficientes para problemas de tamaño razonable.

Las dificultades de la aplicación de los sistemas analíticos en los problemas de secuenciación, han propiciado la investigación de múltiples heurísticas. Éstas, suelen tomar la forma de reglas de secuenciación y habitualmente se emplean para secuenciar los trabajos en los sistemas de fabricación de modo dinámico. Estas heurísticas ordenan los diversos trabajos que compiten por el uso de una máquina dada, mediante diferentes esquemas de prioridad; así, a cada trabajo se le asigna un índice de prioridad y aquel que posea el menor índice se selecciona en primer lugar.

Hasta la fecha, muchos investigadores (ver por ejemplo, Baker, 1984; Blackstone *et al.*, 1982; Kim, 1990; Panwalkar y Iskander, 1977; Ramasesh, 1990; Russel *et al.*, 1987; Vepsalainen y Morton, 1987) han evaluado, mediante simulación, el comportamiento de los sistemas de fabricación con diferentes reglas de secuenciación, concluyendo que dicho comportamiento depende de múltiples factores como el criterio de eficiencia, la configuración del sistema de fabricación, la carga de trabajo, etc. (Cho y Wysk, 1993). Con la llegada de los *FMSs*, surgen numerosos estudios que analizan el comportamiento de estos sistemas con las reglas de secuenciación (ver por ejemplo, Choi y Malstrom, 1988; Denzler y Boe, 1987; Egbelu y Tanchoco, 1984; Henneke y Choi, 1990; Montazeri y Van Wassenhove, 1990; Steckle y Solberg, 1981; Tang *et al.*, 1993).

Debido al comportamiento variable de los sistemas de fabricación, sería interesante modificar las reglas de secuenciación dinámicamente, en el momento apropiado, dependiendo de las condiciones del sistema de fabricación. A priori, se espera que este método sea superior a utilizar una regla de secuenciación de forma constante, por dos razones. En primer lugar, porque es capaz de identificar la mejor regla para un escenario de fabricación determinado. Así, debido a esta capacidad de selección, el sistema de fabricación debe comportarse al menos tan bien como con la mejor de las reglas de secuenciación candidatas consideradas. En segundo lugar, este método puede adaptar su selección de forma dinámica a los escenarios cambiantes. Esta adaptabilidad permite secuenciar los trabajos con una eficiencia incluso superior a la de la mejor regla de secuenciación (Shaw *et al.*, 1992).

Para modificar de forma dinámica las reglas existen, básicamente, dos tipos de sistemas de secuenciación en la literatura. En el primero, la regla se determina, en el momento apropiado, simulando un conjunto de reglas de secuenciación seleccionadas de antemano y eligiendo la mejor (ver por ejemplo, Ishii y Talavage, 1991; Jeong y Kim, 1998; Kim y Kim, 1994; Wu y Wysk, 1989). Los principales inconvenientes que presenta este sistema basado en simulación son los siguientes:

1. El tiempo necesario para realizar las simulaciones con el conjunto de reglas candidatas que puede dificultar la secuenciación en tiempo real.
2. Cambios en el sistema de fabricación muy frecuentes. Dado que la simulación con cada una de las reglas se hace hasta el final del período de simulación considerado, puede que no exista coincidencia entre la regla propuesta y la realmente necesaria ya que la regla elegida se utiliza durante un período de tiempo inferior al que se empleó durante la simulación.
3. No se dispone de mecanismos que eviten modificaciones innecesarias de las reglas de secuenciación ante cambios de tipo transitorio.
4. No se obtiene ningún tipo de conocimiento acerca del sistema de fabricación.

En el segundo tipo de sistema de secuenciación, perteneciente al campo de la inteligencia artificial, se emplea un conjunto de simulaciones previas del sistema de fabricación

(ejemplos de entrenamiento) para determinar cuál es la mejor de las reglas de secuenciación en cada posible estado del sistema de fabricación. Estos casos de entrenamiento se utilizan para entrenar un algoritmo de aprendizaje automático, con el objeto de obtener conocimiento acerca del sistema de fabricación. Finalmente, este conocimiento se utiliza para tomar decisiones inteligentes en tiempo real.

4. SISTEMAS DE SECUENCIACIÓN BASADOS EN EL CONOCIMIENTO

Para que un sistema de secuenciación en tiempo real, que modifique de forma dinámica las reglas de secuenciación, funcione adecuadamente debe cumplir dos características conflictivas:

1. La selección de reglas debe tener en cuenta una variedad de información, en tiempo real, acerca del sistema de fabricación.
2. La elección debe realizarse en un corto período de tiempo de modo que las operaciones no se retrasen.

Una forma de conseguir estas características es utilizar alguna clase de conocimiento acerca de las relaciones entre el estado del sistema de fabricación y la regla a emplear en ese momento. Por lo tanto, es útil emplear «conocimiento de secuenciación» del sistema de fabricación para ahorrar tiempo y alcanzar una respuesta rápida en un entorno que cambia dinámicamente (como es el caso de un *FMS*). Sin embargo, uno de los problemas más difíciles de resolver en un sistema de secuenciación basado en el conocimiento es la adquisición de éste. Para ello, se utilizan algoritmos de aprendizaje automático que reducen el esfuerzo en la determinación del conocimiento necesario para realizar las decisiones de secuenciación.

De todas formas, para que este conocimiento sea útil, es necesario que los ejemplos de entrenamiento y el propio algoritmo de aprendizaje sean los adecuados. Asimismo, para obtener los ejemplos de entrenamiento, son cruciales los atributos seleccionados (Chen y Yih, 1996). Sin embargo, existen, al menos, cuatro razones por las cuales un sistema de secuenciación basado en el conocimiento puede tener un comportamiento inferior a la mejor de las reglas utilizada de forma individual:

1. El conjunto de entrenamiento es un subconjunto del universo de todos los casos posibles. De todos modos, siempre se pueden observar los escenarios en los que el sistema de secuenciación no funciona de forma adecuada y añadirlos como ejemplos de entrenamiento.
2. El comportamiento del sistema de fabricación depende del número y rango de los atributos de control considerados para diseñar los ejemplos de entrenamiento.

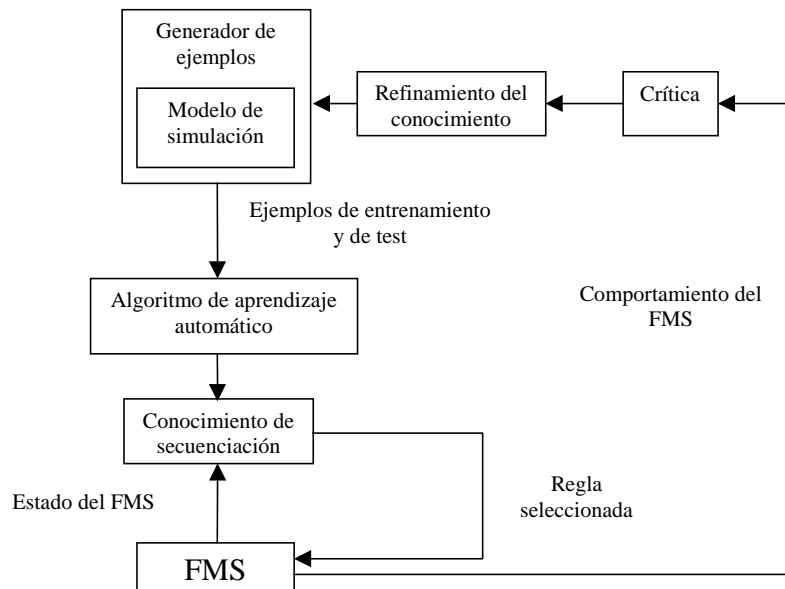


Figura 2. Esquema general de un sistema de secuenciación basado en conocimiento.

3. Una regla puede ser adecuada en una simulación durante un período de tiempo largo, para un conjunto dado de atributos, y no ser apropiada cuando se aplica de modo dinámico.
4. Existen escenarios, o estados del *FMS*, en los que el sistema de secuenciación no determina la regla adecuada que se debe utilizar.

El esquema general de un sistema de secuenciación basado en el conocimiento se muestra en la figura 2. Las etapas fundamentales de este sistema de secuenciación son las siguientes:

1. Creación de un conjunto de ejemplos de entrenamiento y de test mediante el generador de ejemplos. Para ello, es necesario definir los atributos adecuados que identifiquen el estado del sistema de fabricación. Obviamente, como no es posible tener en cuenta a todos ellos, se deben de elegir los más significativos. Los atributos seleccionados se denominan atributos de control siendo los valores utilizados de éstos, aquellos que se presentan con más frecuencia en el sistema de fabricación que se estudia. La clase o solución de cada ejemplo de entrenamiento o de test se obtiene a partir de la regla de secuenciación (o combinación de ellas, si existe más de un tipo de decisión que se debe tomar) que genere el mejor comportamiento en el sistema de fabricación. Para poder realizar lo anterior, se debe construir un modelo

de simulación del sistema de fabricación, y probar para cada conjunto de valores de los atributos de control (ejemplo de entrenamiento o de test) el comportamiento del sistema de fabricación con las diversas reglas de secuenciación que se pretenden utilizar.

2. Determinación del «conocimiento de secuenciación» mediante un algoritmo de aprendizaje automático.
3. Determinación de la regla de secuenciación más adecuada (o combinación de ellas, si existe más de un tipo de decisión), mediante el «conocimiento de secuenciación», dependiendo de los valores que presenten los atributos (estado del sistema de fabricación) en cada momento.
4. Comparación del comportamiento del sistema de fabricación utilizando el «conocimiento de secuenciación» y la mejor regla de secuenciación, o combinación de ellas. Si este segundo método produce un comportamiento del sistema de fabricación superior, se debe regresar al primer paso para refinar el «conocimiento de secuenciación».

A continuación, se analizan diversos sistemas de secuenciación basados en el conocimiento, que modifican de forma dinámica la regla de secuenciación empleada en cada momento. Estos sistemas, según el tipo de algoritmo de aprendizaje automático utilizado, se pueden dividir en las siguientes categorías:

1. Sistemas que no utilizan algoritmos de adquisición de conocimiento.
2. Sistemas basados en aprendizaje inductivo.
3. Sistemas basados en redes neuronales.
4. Sistemas mixtos. En este caso se utiliza una combinación de distintos tipos de algoritmos de aprendizaje.

4.1. Sistemas de secuenciación que no utilizan algoritmos de adquisición de conocimiento

Thesen y Lei (1986) sugieren un sistema experto para secuenciar robots en un sistema de galvanizado flexible. Los autores realizan un conjunto de simulaciones previas utilizando diversas reglas de secuenciación para estudiar el comportamiento del sistema de fabricación en diversos escenarios, obteniendo, de este modo, 38 ejemplos de entrenamiento. Sin embargo, el conocimiento acerca del sistema de fabricación no se alcanza mediante ningún procedimiento de adquisición automática, sino que se efectúa de modo manual, inspeccionando directamente los resultados de las simulaciones. Los autores comprueban que el sistema de fabricación aumenta el número de piezas producidas en unos porcentajes que varían entre un 7% y un 30%.

Sarin y Salgame (1990) definen un sistema experto para realizar secuenciación de tipo dinámico. El sistema de secuenciación tiene, al comienzo de un período de tiempo determinado, una secuencia ordenada de los trabajos que se realizan durante ese período y actúa cuando se produce un cambio. Los cambios se clasifican en diferentes grupos: avería de una máquina, llegada de un trabajo urgente, llegada de un nuevo lote de trabajos, falta de material, absentismo laboral, finalización de un trabajo en una máquina y cambio de turno. El sistema de secuenciación propuesto consta de los siguientes elementos: un «conocimiento de secuenciación», una base de datos global, una interface con el usuario y un bloque de control. El conocimiento está dividido en varios grupos, cada uno de los cuales contiene reglas que se encargan de resolver distintos tipos de problemas en función del cambio que ocurra en el sistema de fabricación. Las reglas representan las heurísticas de un secuenciador de trabajos humano.

La base de datos global contiene información de los diferentes trabajos y turnos existentes en el momento actual. El grupo de reglas que necesite información, acude a la base de datos global. Por último, el bloque de control, en forma de árbol («meta-reglas» o «conocimiento acerca del conocimiento»), elige el grupo de reglas adecuado al nuevo problema que origina el cambio en el sistema de fabricación. Del mismo modo, los autores presentan un sistema de secuenciación integrado que consta de dos módulos. El primero, que se apoya en la programación matemática, determina una secuencia de trabajos de tipo predictivo como punto inicial de partida. El segundo, el sistema experto, ante cualquier cambio que se produzca, retoma el control para ejecutar una secuenciación dinámica o reactiva en función de la nueva situación. Finalmente, los autores señalan que este sistema de secuenciación integrado aún no ha sido implementado en un caso real.

Chandra y Talavage (1991) presentan un sistema de secuenciación denominado *EXPERT*, formado por un conjunto de reglas de decisión. La información utilizada en el proceso de decisión es el nivel de congestión del sistema de fabricación, la preferencia de una pieza por una máquina, la criticidad de la pieza (indica la capacidad de la pieza para cumplir sus fechas de entrega) y el objetivo actual del sistema de fabricación. Los autores afirman que, en principio, es interesante el objetivo de maximizar el ritmo de «progreso de trabajo», aunque exista el peligro de que algunos trabajos se retrasen (sobre todo si el sistema es muy utilizado y son muchos los trabajos críticos). La excesiva preocupación por los trabajos críticos, puede empeorar todo el sistema; por lo tanto, se elige el objetivo de maximizar el ritmo de «progreso de trabajo» como criterio primario.

Por otra parte, los trabajos se dividen en grupos (preferencia alta, mediana y baja) en lugar de clasificarlos de forma individual. El sistema de secuenciación propuesto elige el trabajo que se asigna a la máquina empezando por los de preferencia alta, persiguiendo el objetivo primario y buscando oportunidades para mejorar el secundario (minimizar el número de trabajos retrasados) al mismo tiempo. En determinados casos, se inspeccionan los trabajos disponibles en un futuro cercano. En caso de empate o si no se logra

una decisión clara se aplica la regla *SPT*³. En la etapa experimental se comprueba que el sistema *EXPERT* es superior a las reglas de secuenciación convencionales.

Sabuncuoglu y Hommertzhaim (1992) sugieren un algoritmo dinámico para secuenciar trabajos en máquinas y *AGVs*⁴. El algoritmo propuesto se basa en la idea de que un trabajo no debe ser asignado a una máquina si tiene que esperar por un *AGV* en la siguiente operación y viceversa. Para ello, se utilizan diversos esquemas de prioridad (o reglas) e información acerca del sistema de fabricación (niveles en las colas, número de piezas en el sistema, estado de las máquinas, etc.) y de los trabajos (tiempos de operación, número de operaciones, etc.). El algoritmo consta de dos partes fundamentales: un conjunto de procedimientos para secuenciar trabajos en las máquinas y otro para secuenciar los trabajos en los *AGVs*. Éstos últimos comprueban la existencia de estaciones bloqueadas o vacías de piezas en el «buffer» central.

Los autores comparan el algoritmo propuesto con las dos mejores combinaciones de reglas de secuenciación en máquinas y *AGVs*, utilizando como criterios de comportamiento el tiempo medio de una pieza en el sistema y el retraso medio. Para ello, proponen diversos escenarios variando los niveles de carga, la capacidad de las colas, la variable F ⁵ (Baker, 1984), el tipo de distribución de los tiempos de procesamiento y el criterio de comportamiento. Se comprueba que el algoritmo funciona mejor que las mejores combinaciones de reglas cuando la carga de las máquinas es elevada (si es baja, en las colas apenas hay piezas, con lo cual no influye la regla seleccionada) y el tamaño de las colas pequeño. En estas condiciones, la mejora obtenida es superior al 12%.

Pierreval y Mebarki (1997) presentan una metodología heurística, denominada *SFSR* (Shift From Standard Rules), para modificar de forma dinámica las reglas de secuenciación, teniendo en cuenta dos criterios de comportamiento (uno primario y otro secundario). La heurística *SFSR* chequea el estado del sistema de fabricación cuando se vuelve disponible un recurso o entra un nuevo trabajo. Mediante el empleo de unas reglas definidas de antemano que son función de unos parámetros que se deben optimizar, se puede detectar la presencia de determinadas anomalías en el sistema de fabricación (por ejemplo, la saturación de una máquina, un trabajo que permanece demasiado tiempo en el sistema, etc.). Los valores óptimos de los parámetros se calculan mediante el método de Hooke y Jeeves (1961). Si no existen anomalías en el sistema de fabricación se utilizan reglas de tipo estándar, tomadas de la literatura en función del criterio que se vaya a optimizar, que permiten determinar la regla de secuenciación que se empleará.

³Del inglés, Shortest Processing Time. Esta regla elige el trabajo que tiene el menor tiempo de procesamiento.

⁴Del inglés, Automatic Guided Vehicles.

⁵La variable F (flow allowance factor) determina la holgura de las fechas de entrega.

En caso contrario, se utilizan reglas definidas por los autores, que dependen de los criterios que se optimizan, de la anomalía detectada y del estado del sistema. En general, la metodología propuesta supera, en el criterio primario, a la alternativa de usar una regla de forma constante; y si no, lo compensa con el criterio secundario. Los porcentajes de mejora varían entre el 12.3% y el 33.8%. El principal inconveniente de la metodología propuesta es que las reglas de tipo estándar se definen de acuerdo con los resultados de investigaciones previas presentadas en la literatura. Un enfoque alternativo sería generarlas mediante aprendizaje inductivo, para tener en cuenta las particularidades del sistema de fabricación bajo estudio. También se podría utilizar aprendizaje inductivo para generar los otros tipos de reglas empleadas en *SFSR*.

4.2. Sistemas de secuenciación basados en aprendizaje inductivo

Pierreval y Ralambondrainy (1990) sugieren una técnica de aprendizaje inductivo, denominada *GENREG*, para obtener reglas heurísticas que permitan conocer el comportamiento del sistema de fabricación ante diferentes reglas de secuenciación y estados del sistema de fabricación. En la metodología propuesta se obtiene una regla con cada ejemplo de entrenamiento. Debido a que el número de reglas es muy grande, se utiliza posteriormente *GENREG* para generalizarlas y, de esta forma, reducir su número. La metodología se aplica en una configuración sencilla, de tipo «flow shop» con dos máquinas, utilizando 198 ejemplos de entrenamiento. Sin embargo, las reglas obtenidas mediante *GENREG* no se emplean de forma dinámica.

Shaw *et al.* (1992) presentan un sistema de secuenciación denominado *PDS* (Pattern-Directed Scheduling) para secuenciar trabajos en un *FMS* que utiliza aprendizaje inductivo. En este caso, el algoritmo de aprendizaje utilizado es *ID3* (Quinlan, 1986). En la etapa de adquisición de conocimiento se utilizan 130 ejemplos de entrenamiento, empleando como criterio de comportamiento el retraso medio de los trabajos. El sistema de secuenciación propuesto genera una reducción media en los retrasos de los trabajos del 11.5%.

Los autores observan que la máxima efectividad del sistema de secuenciación se obtiene cuando el número de cambios en los estados del sistema de fabricación (patterns) es de medio a razonablemente alto, y no existe un estado claramente dominante. Este rango también corresponde a valores de retraso que varían entre bajos y moderadamente altos. Asimismo, el número de máquinas alternativas para procesar una operación dada no debe de ser muy elevado. Los autores afirman que todas estas características se presentan en la mayor parte de los *FMSs* reales.

Nakasuka y Yoshida (1992) proponen un sistema de secuenciación denominado *LADS* (Learning-Aided Dynamic Scheduler) que incorpora un algoritmo de aprendizaje inductivo con dos características que lo diferencian de los algoritmos convencionales. En

primer lugar, un nuevo criterio para decidir como separar los grupos de datos, ya que debido al ruido que presentan éstos en los problemas de secuenciación, no interesa dividirlos de forma que pertenezcan a una sola «clase» pues el número de datos en cada grupo sería muy pequeño. La segunda característica del algoritmo propuesto, es la generación de combinaciones lineales de los atributos facilitados al sistema. El sistema de secuenciación propuesto se utiliza en un sistema de fabricación sencillo «flow shop» con 3 máquinas, con el objetivo de minimizar el «makespan» (tiempo total invertido para realizar todos los trabajos) y mantener el retraso por debajo de un valor determinado. Los autores observan que el sistema de secuenciación propuesto es superior (en los dos criterios) a utilizar una regla de forma constante.

Piramuthu *et al.* (1993) definen un sistema de secuenciación para secuenciar tareas de modo dinámico. El sistema propuesto, así como los ejemplos en los que lo aplica, es similar al presentado por los autores en otros trabajos (ver por ejemplo, Shaw *et al.*, 1992; Piramuthu *et al.*, 1994). Lo que aporta este artículo, con respecto a otros publicados por los mismos autores, es un marco teórico más elaborado de cada una de las partes que forman el sistema de secuenciación de trabajos.

Piramuthu *et al.* (1994) definen un sistema para secuenciar trabajos, mediante aprendizaje inductivo, en un sistema de fabricación flexible de tipo «flow-shop», utilizando como criterio de comportamiento el tiempo medio de una pieza en el sistema. Empleando C4.5 (Quinlan, 1993) como algoritmo de aprendizaje, se generan dos árboles de decisión, mediante 66 ejemplos de entrenamiento. El primer árbol se utiliza para secuenciar los trabajos en las propias máquinas, y el segundo, para decidir el orden de entrada de los trabajos en el sistema de fabricación. Del mismo modo, se presenta un procedimiento de refinamiento de los árboles de decisión, que consiste en incluir en el conjunto de entrenamiento, aquellos casos que el sistema clasifica mal.

Los autores observan que la incorporación de un árbol de decisión, para seleccionar la regla de secuenciación, no mejora los resultados de forma significativa con respecto a la alternativa de usar sólo el árbol de decisión para fijar el orden de entrada de las piezas, utilizando una regla de secuenciación de forma constante. Además, esta metodología es particularmente útil cuando el tamaño del «buffer» de entrada es limitado y pequeño, y existe una gran variación en los tiempos de procesamiento de las piezas en las máquinas cuello de botella.

4.3. Sistemas de secuenciación basados en redes neuronales

Chen y Yih (1996) definen una metodología para determinar los atributos más importantes como paso previo a la construcción de sistemas de secuenciación basados en el conocimiento. El mecanismo de identificación de atributos consta de tres pasos:

1. Recopilación de datos a través de la simulación del sistema de fabricación.

2. Construcción, mediante redes neuronales de tipo «backpropagation», de funciones de «mapeo atributo-comportamiento» del sistema de fabricación.
3. Selección de los atributos esenciales.

Para ello, se omite un atributo y se mide la diferencia entre la salida original y la obtenida con el atributo omitido. Los atributos que son importantes, tienen una diferencia mayor que aquellos con un nivel de significación más bajo. En el estudio experimental se usan 20 atributos candidatos (utilizando la variabilidad del atributo que se define como el cociente entre la varianza y la media de éste) tomados de la literatura (Nakasuka y Yoshida, 1992; Cho y Wysk, 1993; Chiu, 1994), 6 medidas de comportamiento y 10 reglas de secuenciación. Se generan 1300 ejemplos de entrenamiento para cada tipo de regla y la correspondiente red neuronal, seleccionando los 10 atributos más importantes.

Finalmente los autores comparan 3 redes neuronales con nodos de entrada formados por grupos alternativos de atributos (con los 10 atributos seleccionados, con los 20 atributos iniciales y con los 10 restantes) y nodos de salida correspondientes a las reglas que se eligen. Se verifica que la capacidad de generalización de la red obtenida con los atributos significativos es un 9% superior a la de la red formada por los 20 atributos iniciales. Los autores afirman que usar tantos atributos como sea posible, para construir una base de conocimiento, no mejora su capacidad de generalización o predicción. Cuantos más atributos se incluyan, el esfuerzo necesario para el desarrollo de la base de conocimiento aumenta, y su estructura se vuelve más compleja. Los mayores defectos de la metodología propuesta son que no identifica atributos importantes si éstos no están inicialmente considerados y que el proceso se debe repetir si cambian las medidas de comportamiento.

Sun y Yih (1996) emplean un sistema de secuenciación que utiliza una red neuronal de tipo «backpropagation» en cada una de las máquinas, para la elección de la regla de secuenciación más adecuada en un entorno multicriterio. En cada punto de decisión, cuando una máquina necesita seleccionar un nuevo trabajo, un módulo de ajuste determina, en función de los valores deseados y actuales, la importancia relativa de cada criterio de comportamiento. La red neuronal, tomando como entrada el valor proporcionado por el módulo de ajuste y el estado actual del sistema de fabricación, proporciona la regla de secuenciación más adecuada. Los autores utilizan alrededor de 1000 ejemplos de entrenamiento para cada red neuronal y comprueban que el sistema de secuenciación propuesto genera en el sistema de fabricación, como término medio, una mejora superior al 4.2% con respecto a la alternativa de utilizar la mejor de las reglas. Asimismo, el sistema de secuenciación presenta una gran adaptabilidad ante cambios en la elección de los criterios de comportamiento considerados como prioritarios. El mayor defecto de este estudio es que el sistema de fabricación considerado no tiene rutas flexibles y el número de piezas es limitado.

Min *et al.* (1998) proponen un sistema de secuenciación que utiliza redes neuronales «competitivas». En este caso, se emplean como atributos las diferencias entre los valores, en los diferentes intervalos de tiempo, de los criterios de comportamiento y de las variables de estado del sistema de fabricación. Simulando el sistema de fabricación durante un largo período de tiempo, los ejemplos de entrenamiento se obtienen con estas diferencias, modificando las reglas de forma aleatoria. En la etapa de entrenamiento se utilizan 3500 ejemplos y se definen 40 «clases» o nodos en la red. Posteriormente, se emplean las redes neuronales para obtener las «clases» a partir de los ejemplos de entrenamiento.

El sistema de secuenciación funciona en tiempo real de la siguiente forma. En primer lugar, el usuario fija unas diferencias como objetivo y se identifica la «clase» mediante la red neuronal. A continuación, dentro de todos los ejemplos de entrenamiento, se buscan aquellos que tengan la misma «clase» y las mismas variables de decisión (reglas de secuenciación) del intervalo anterior. Si no se encuentra este ejemplo, lo cual es probable, se elige para cada variable de decisión, la regla más utilizada dentro de la correspondiente «clase». Una característica interesante de este sistema de secuenciación es que se emplean las reglas de secuenciación anteriores para hallar las actuales. El sistema de secuenciación propuesto se compara con otro que elige las reglas de forma aleatoria (en realidad, se hacen 10 réplicas del sistema de secuenciación aleatorio y se elige la mejor de ellas), siendo superior el comportamiento del primero. Los principales defectos del sistema de secuenciación son la falta de un método que sistemáticamente busque un número óptimo de nodos de salida de la red neuronal y que se compare con un sistema de secuenciación aleatorio y no con la mejor combinación posible de las reglas propuestas.

4.4. Sistemas de secuenciación mixtos

Wu y Wysk (1988) proponen un sistema de control y secuenciación denominado *MPECS* (Multipass Expert Control System), que combina sistemas expertos, simulación y aprendizaje inductivo. El sistema de secuenciación propuesto consta de tres módulos: un módulo de secuenciación inteligente, un simulador del sistema de fabricación y un módulo de control de célula. El primer elemento, a su vez, está compuesto de una base conocimiento, un motor de inferencia y un módulo de aprendizaje. La base tiene conocimiento de tipo «declarativo» (información acerca del estado del sistema de fabricación, reglas y heurísticas de secuenciación) y de «procedimiento» (criterios de tipo general, en forma de reglas, para seleccionar las reglas de secuenciación). El motor de inferencia es un mecanismo de búsqueda para elegir las reglas adecuadas del conocimiento de «procedimiento». Por último, el módulo de aprendizaje genera un conjunto de reglas a través de ejemplos de entrenamiento que asocian reglas de secuenciación, medidas de comportamiento y características del sistema de fabricación. Las reglas creadas por el módulo de aprendizaje se envían al conocimiento de «procedimiento».

El módulo de secuenciación inteligente se activa cuando llega un nuevo trabajo, o si se produce una anomalía en el sistema de fabricación. El simulador tiene como misión examinar las reglas de secuenciación sugeridas por el módulo de secuenciación inteligente y elegir la mejor. Por último, el módulo de control posibilita la secuenciación en la célula física; además, la mayor parte de la información de ésta, que es esencial para el control de la célula, se obtiene y manipula a través de este módulo. Los autores señalan que el sistema de secuenciación propuesto produce una mejora en el comportamiento del sistema de fabricación que varía entre el 2.3% y el 29.3% cuando se compara con la alternativa de utilizar una regla de secuenciación de forma constante.

Rabelo y Alptekin (1989) definen un sistema de secuenciación, denominado *ISS/FMS* (Intelligent Scheduling System for *FMS*), para secuenciar trabajos en un *FMS* que consta de tres módulos básicos. El primero de ellos, es un sistema experto que basándose en determinada información (datos de los trabajos a realizar, restricciones impuestas por el taller, estado de la célula, etc.) decide la regla heurística que se debe utilizar. Para ello, tiene en cuenta los datos facilitados por una red neuronal y un módulo de análisis estadístico que estudian casos pasados obtenidos mediante un estudio de simulación. El segundo módulo ejecuta un procedimiento heurístico (Kiran y Alptekin, 1989) que depende de dos coeficientes determinados por una red neuronal, en función de las características del problema de secuenciación que se resuelva. El tercer componente elige la mejor de las soluciones determinadas por los dos anteriores módulos.

Cho y Wysk (1993) presentan un sistema de secuenciación denominado *IWC* (Intelligent Workstation Controller) que utiliza redes neuronales de tipo «backpropagation» y un simulador basado en el trabajo de Wu y Wysk (1989). La red neuronal tiene 7 nodos de entrada, correspondientes al estado del sistema de fabricación, y 9 nodos de salida, uno para cada regla de secuenciación considerada. La red se entrena, con 90 ejemplos tomados de la literatura, teniendo en cuenta diversas configuraciones de las capas ocultas y distintos ritmos de aprendizaje. El simulador, con las dos mejores reglas que proporciona la red, en función del estado del sistema de fabricación, selecciona la mejor de ellas. Asimismo, los autores calculan de forma experimental, mediante un conjunto de simulaciones, la ventana de simulación más adecuada para el criterio de comportamiento elegido. Se observa que *IWC* es superior al empleo de una regla de forma constante, aunque el porcentaje de mejora no supera nunca el 3%.

Li y She (1994) emplean un sistema de secuenciación que utiliza «análisis cluster» (Evert, 1980) y aprendizaje inductivo. Mediante 600 ejemplos de entrenamiento, que generan de forma uniforme del espectro de decisión, y «análisis cluster», se establecen 7 «clases» con valores de comportamiento similares. Posteriormente, con un algoritmo similar a *C4.5*, establecen dos conjuntos de reglas que determinan la «clase», en función de los atributos de decisión o de comportamiento. Una forma que proponen los autores de utilizar este «conocimiento de secuenciación» es fijar unas condiciones de comportamiento y determinar la «clase» a la cual corresponden. Con el otro conjunto

de reglas se determinan, una vez conocida la «clase», las variables de decisión que se van a tomar. Sin embargo, no se compara esta metodología con ninguna alternativa para comprobar su funcionamiento.

Chiu y Yih (1995) sugieren un sistema de secuenciación que utiliza aprendizaje inductivo y algoritmos genéticos. Éstos últimos se emplean para buscar un conjunto de ejemplos de entrenamiento que posea buena calidad. Para ello, en cada punto de decisión, se elige la mejor regla de secuenciación, y ésta, junto con el estado del sistema de fabricación, forma un caso de entrenamiento. Por otra parte, el algoritmo de aprendizaje puede modificar el árbol de decisión cuando se presentan nuevos ejemplos, sólo si el cambio es significativo. Los autores comprueban que el sistema de secuenciación propuesto es superior a utilizar una regla de secuenciación de forma constante. El mayor defecto del sistema de secuenciación presentado es la necesidad de cambiar el «conocimiento de secuenciación» inducido con pequeñas modificaciones en el sistema de fabricación.

Quiroga y Rabelo (1995) resuelven el problema de secuenciación de trabajos en una máquina, mediante aprendizaje inductivo (*ID3*), redes neuronales «backpropagation» y lógica «borrosa». Para ello, se utilizan 358 casos de entrenamiento y 198 de test, siendo el nivel de aciertos superior al 90% en las tres metodologías. El aprendizaje inductivo y la lógica «borrosa» presentan la ventaja de que generan reglas que son inteligibles para el ser humano, cosa que no ocurre con las redes neuronales. Sin embargo, éstas son las menos sensibles a ruidos o datos incompletos y presentan el mayor porcentaje de aciertos (98.8%).

Bowden y Bullington (1996) sugieren un sistema de secuenciación denominado *GARDS* (Genetic Algorithm Rule Discovery System), para determinar estrategias de control utilizando algoritmos genéticos. *GARDS* contiene tres bloques fundamentales:

1. Un modelo de simulación para analizar el comportamiento del sistema de fabricación con las distintas estrategias generadas.
2. Un algoritmo que determina la regla más adecuada, dentro de una estrategia o plan, para el estado actual del sistema de fabricación.
3. Un algoritmo genético que intenta, mediante los operadores tradicionales de cruce y mutación, mejorar los planes iniciales, eligiendo el mejor para el control del sistema de fabricación.

El sistema de secuenciación propuesto se prueba en dos configuraciones de distinta complejidad con el objetivo de minimizar el número de trabajos retrasados. Se observa que *GARDS* mejora el comportamiento del sistema de fabricación con respecto a diversos métodos clásicos heurísticos (por ejemplo, enviar un trabajo a la cola de la máquina con menor número de trabajos).

Lee *et al.* (1997) proponen un sistema de secuenciación que también emplea aprendizaje inductivo y algoritmos genéticos. La primera técnica se utiliza, generando un árbol de

Tabla 2. Clasificación de las referencias según la metodología empleada.

<i>Metodología</i>	<i>Referencias</i>
Sistemas basados en métodos analíticos	Han <i>et al.</i> (1989); Hutchison <i>et al.</i> (1989); Kimemia y Gershwin (1985); Lashkari <i>et al.</i> (1987); Shanker y Rajamarthandan (1989); Shanker y Tzen (1985); Stecke (1983); Wilson (1989).
Sistemas basados en métodos heurísticos	Choi y Malstrom (1988); Denzler y Boe (1987); Egbelu y Tanchoco (1984); Henneke y Choi (1990); Montazeri y Van Wassenhove (1990); Stecke y Solberg (1981); Tang <i>et al.</i> (1993)
Sistemas basados en simulación	Ishii y Talavage (1991); Jeong y Kim (1998); Kim y Kim (1994); Wu y Wysk (1989);
Sistemas basados en inteligencia artificial	Bowden y Bullington (1996); Chandra y Talavage (1991); Chen y Yih (1996); Chiu y Yih (1995); Cho y Wysk (1993); Kim <i>et al.</i> (1998); Lee <i>et al.</i> (1997); Li y She (1994); Min <i>et al.</i> (1998); Nakasuka y Yoshida (1992); Pierreval y Mebarki (1997); Pierreval y Ralambondrainy (1990); Piramuthu <i>et al.</i> (1993); Piramuthu <i>et al.</i> (1994); Quiroga y Rabelo (1995); Rabelo y Alptekin (1989); Sabuncuoglu y Hommertzhaim (1992); Sarin y Salgame (1990); Shaw <i>et al.</i> (1992); Sun y Yih (1996); Thesen y Lei (1986); Wu y Wysk (1988)

decisión mediante C4.5, para seleccionar la regla más adecuada que controle el flujo de entrada de trabajos en el sistema de fabricación. Por otra parte, los algoritmos genéticos se emplean para seleccionar las reglas de secuenciación más apropiadas en cada una de las máquinas del sistema de fabricación. Los autores verifican el sistema de secuenciación propuesto en dos sistemas de tipo «job shop» (uno de ellos con una máquina cuello de botella) utilizando el retraso medio como criterio de comportamiento, y comprueban que supera a la mejor combinación de reglas utilizadas de forma constante, en un porcentaje que varía entre el 20.34% y el 25.28%. Sin embargo, los tiempos requeridos (26 y 168 minutos para el primer y segundo caso, respectivamente) son bastante elevados para que el sistema de secuenciación funcione en tiempo real.

Kim *et al.* (1998) sugieren un sistema de secuenciación, ampliando un trabajo anterior (Min *et al.*, 1998), que emplea redes neuronales «competitivas» y aprendizaje inductivo. Se aplica esta última técnica, una vez obtenidas las «clases» mediante las redes neuronales, para expresar el conocimiento en forma de árbol y reglas de producción. Los autores utilizan 99.999 casos de entrenamiento y fijan una red con 100 grupos o «clases». El sistema de secuenciación funciona en tiempo real del mismo modo que el sistema presentado previamente por los mismos autores (Min *et al.*, 1998). La única diferencia es que la «clase» se identifica mediante las reglas de producción obtenidas del programa de aprendizaje inductivo C4.5. Los autores comparan este sistema de secuenciación con otro que solamente utiliza la red neuronal «competitiva», comprobando la

Tabla 3. Clasificación de las referencias según el algoritmo de aprendizaje automático empleado.

<i>Algoritmo de aprendizaje automático</i>	<i>Referencias</i>
No utiliza	Chandra y Talavage (1991); Pierreval y Mebarki (1997); Sabuncuoglu y Hommertzhaim (1992); Sarin y Salgame (1990); Thesen y Lei (1986)
Aprendizaje inductivo	Nakasuka y Yoshida (1992); Pierreval y Ralambondrainy (1990); Pirmuthu <i>et al.</i> (1993); Pirmuthu <i>et al.</i> (1994); Shaw <i>et al.</i> (1992)
Redes neuronales	Chen y Yih (1996); Min <i>et al.</i> (1998); Sun y Yih (1996)
Mixto	Bowden y Bullington (1996); Chiu y Yih (1995); Cho y Wysk (1993); Kim <i>et al.</i> (1998); Lee <i>et al.</i> (1997); Li y She (1994); Quiroga y Rabelo (1995); Rabelo y Alptekin (1989); Wu y Wysk (1988)

superioridad del primero debido al algoritmo de «podado» de los árboles de C4.5 que maneja de forma eficiente el ruido en los datos. Este sistema de secuenciación posee los mismos defectos que el sugerido en Min *et al.* (1998).

A modo de recapitulación, en la tabla 2 se presenta un resumen de las distintas aproximaciones existentes en la literatura, clasificadas según la metodología empleada. Por otro lado, en la tabla 3 se muestra una recopilación de los diferentes sistemas de secuenciación existentes que modifican de forma dinámica las reglas de secuenciación, clasificados según el tipo de algoritmo de aprendizaje automático utilizado.

5. LIMITACIONES DE LA LITERATURA EXISTENTE Y LÍNEAS DE INVESTIGACIÓN FUTURAS

A partir de los sistemas de secuenciación basados en el conocimiento presentados anteriormente, que utilizan algoritmos de aprendizaje automático, se detectan, en general, una serie de carencias o características deseables comunes. Éstas, pueden originar líneas de investigación en el campo de la secuenciación dinámica de sistemas de fabricación, mediante la modificación de la regla de secuenciación empleada. Estas carencias son las siguientes:

1. Comparación de diversos tipos de algoritmos de aprendizaje automático. En los sistemas de secuenciación presentados en la literatura se utiliza un algoritmo o, en determinados casos, una combinación de ellos. Sin embargo, no existe un estudio comparativo que determine cuál es el mejor de ellos. Por otra parte, debido a la gran disparidad de los *FMSs* utilizados en la literatura revisada, no es posible intuir cuál

de los algoritmos presentados es el más adecuado para la resolución de este tipo de problemas de secuenciación.

2. Utilización del *CBR* en los sistemas de secuenciación. Estos algoritmos poseen una gran eficacia de clasificación, a pesar de su sencillez. Sin embargo, ninguno de los sistemas de secuenciación presentados utiliza *CBR*; por lo tanto, sería interesante probar su idoneidad en los problemas de secuenciación.
3. Determinación del número de ejemplos de entrenamiento óptimo. En ninguno de los sistemas de secuenciación considerados se calcula el número de ejemplos necesario para entrenar el algoritmo de aprendizaje automático de forma óptima. Por otra parte, tampoco se especifica si los ejemplos de test son iguales, parecidos, o muy distintos de los de entrenamiento. Sin embargo, el error de clasificación del «conocimiento de secuenciación» y, por lo tanto, el comportamiento del sistema de fabricación, depende en gran medida del número de ejemplos de entrenamiento considerado. Por lo tanto, es necesario estudiar el error de clasificación en función del número de ejemplos considerado y elegir un tamaño adecuado del conjunto de entrenamiento.
4. Selección de un período de supervisión adecuado. En general, en la literatura existente, no se realiza estudio alguno para determinar el período de supervisión apropiado para cada criterio de comportamiento. Sin embargo, la frecuencia utilizada para chequear los atributos de control y decidir si se cambian, o no, las reglas de secuenciación, es un tema de vital importancia que determina el comportamiento del sistema de fabricación.
5. Determinación de un mecanismo o filtro que amortigüe los estados transitorios. En determinadas ocasiones, el sistema de fabricación alimentado con el «conocimiento de secuenciación» no se comporta tal como se esperaba, y es superado por la alternativa de utilizar la mejor combinación de reglas de secuenciación de forma constante. Este fenómeno se puede explicar por el hecho de que el sistema de secuenciación reacciona de forma precipitada ante cambios en los atributos de control que sólo son transitorios en el tiempo. Por ello, se propone utilizar filtros de tipo digital que permitan amortiguar estos escenarios transitorios en los atributos de control. En la mayor parte de los trabajos estudiados no se tiene en cuenta este mecanismo y, cuando se considera, no se analizan los diversos tipos de filtros digitales existentes y su relación con el período de supervisión.
6. Generación de nuevos atributos de control mediante un algoritmo que permita crear atributos que sean combinación de los iniciales. En algunos casos, para seleccionar las reglas de secuenciación más adecuadas, es preciso chequear relaciones del tipo: la utilización de la máquina 1 es menor que la de la máquina 2. Para poder lograr estas relaciones, sería necesario definir combinaciones aritméticas de los atributos básicos iniciales. Sin embargo, a menudo, tales combinaciones no son conocidas de antemano y sólo se encuentran, en sistemas de fabricación muy sencillos, después de examinar, en detalle, los resultados de simulación.

7. Incorporación de un simulador. El comportamiento del sistema de fabricación, podría mejorar si se incorporase un simulador que determine la mejor regla de entre las que el «conocimiento de secuenciación» considere más importantes. En ocasiones, el «conocimiento de secuenciación» determina que ante unos valores dados de los atributos de control existen dos, o más, reglas de secuenciación que podrían ser, en principio, adecuadas. En estos casos, en los cuales la decisión por parte del «conocimiento de secuenciación» no es clara, la incorporación del simulador sería bastante útil.
8. Refinamiento de la base de conocimiento. La base de conocimiento, una vez desarrollada, no es estática. Por lo tanto, sería interesante establecer un procedimiento que modifique el conocimiento automáticamente si se producen cambios importantes en el sistema de fabricación. La misión principal del módulo de refinamiento es descubrir deficiencias en la base de conocimiento y añadir casos de entrenamiento que cubran dichas deficiencias. Éstas, se pueden presentar en determinados rangos de valores de los atributos de control. Para solucionar este problema, se requiere «cubrir» estos rangos con nuevos casos de entrenamiento, de forma que el nuevo «conocimiento de secuenciación» obtenido, sea capaz de tratar estas situaciones.

6. CONCLUSIONES

En este artículo, se realiza una revisión de la literatura existente sobre secuenciación dinámica de *FMSs*, mediante aprendizaje automático, en la cual se modifica la regla de secuenciación empleada. Al efectuar esta revisión, se ha detectado una serie de carencias en los sistemas de secuenciación estudiados. En primer lugar, no se ha encontrado una comparación de los distintos algoritmos de aprendizaje automático existentes para determinar cuál es el mejor tipo de algoritmo para resolver esta clase de problemas de secuenciación. Asimismo, tampoco se ha utilizado el razonamiento basado en casos como algoritmo de aprendizaje automático, a pesar de su sencillez y gran eficacia de clasificación. Por otro lado, no se ha determinado el número óptimo de ejemplos de entrenamiento necesario para obtener un «conocimiento de secuenciación» con un error de clasificación pequeño.

Del mismo modo, en los sistemas de secuenciación estudiados, no se ha considerado, en general, la selección de un período de supervisión ni la utilización de filtros que amortigüen los estados transitorios del sistema de fabricación. De igual forma, tampoco se ha tenido en cuenta, en la mayor parte de los sistemas de secuenciación estudiados, la incorporación de un generador de nuevos atributos de control y de un simulador que apoye al «conocimiento de secuenciación» cuando éste no pueda determinar la regla de secuenciación que se debe utilizar. Asimismo, no se han encontrado, en general, módulos de refinamiento de la base de conocimiento que permitan modificar el

«conocimiento de secuenciación» si se producen cambios importantes en el sistema de fabricación. Por último, reseñar que sería interesante, como futuro trabajo, diseñar un sistema de secuenciación que incorpore las carencias anteriormente señaladas y medir el efecto de cada una de ellas en el comportamiento del sistema de fabricación.

7. REFERENCIAS

- Baker, K. R. (1984). «Sequencing rules and due-date assignments in a job shop». *Management Science*, 30, 9, 1093-1103.
- Basnet, C. Mize, J. H. (1994). «Scheduling and control of flexible manufacturing systems: a critical review». *International Journal Computer Integrated Manufacturing*, 7, 6, 340-355.
- Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford: Oxford University Press.
- Blackstone, J. H.; Phillips, D. T. & Hogg, G. L. (1982). «A state-of-the-art survey of dispatching rules for manufacturing job shop operations». *International Journal of Production Research*, 20, 1, 27-45.
- Bowden, R. & Bullington, S. F. (1996). «Development of manufacturing control strategies using unsupervised machine learning ». *IIE Transactions*, 28, 319-331.
- Chandra, J. & Talavage, J. (1991). «Intelligent dispatching for flexible manufacturing». *International Journal of Production Research*, 29, 11, 2259-2278.
- Chen, C. C. & Yih, Y. (1996). «Identifying attributes for knowledge-based development in dynamic scheduling environments». *International Journal of Production Research*, 34, 6, 1739-1755.
- Chiu, C. (1994). *A learning-based methodology for dynamic scheduling in distributed manufacturing systems*, PhD thesis. School of Industrial Engineering, Purdue University, West Lafayette, Indiana.
- Chiu, C. & Yih, Y. (1995). «A learning-based methodology for dynamic scheduling in distributed manufacturing systems». *International Journal of Production Research*, 33, 11, 3217-3232.
- Cho, H. & Wysk, R. A. (1993). «A robust adaptive scheduler for an intelligent workstation controller». *International Journal of Production Research*, 31, 4, 771-789.
- Choi, R. H. & Malstrom, E. M. (1988). «Evaluation of traditional work scheduling rules in a flexible manufacturing system with a physical simulator». *Journal of Manufacturing Systems*, 7, 1, 33-45.
- Denzler, D. R. & Boe, W. J. (1987). «Experimental investigation of flexible manufacturing system scheduling rules». *International Journal of Production Research*, 25, 7, 979-994.
- Egbelu, P. J. & Tanchoco, J. M. A. (1984). «Characterization of automated guided vehicle dispatching rules». *International Journal of Production Research*, 22, 3, 359-374.

- Evert, B. (1980). *Cluster Analysis*. New York: Heinemann.
- Garey, M. & Johnson, D. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: Freeman.
- Han, M.; Na, Y. K. & Hogg, G. L. (1989). «Real-time tool control and job dispatching in flexible manufacturing systems». *International Journal of Production Research*, 27, 1257-1267.
- Henneke, M. J. & Choi, R. H. (1990). «Evaluation of FMS parameters on overall system performance». *Computer Industrial Engineering*, 18, 1, 105-110.
- Hooke, R. & Jeeves, T. A. (1961). «Direct search solution of numerical and statistical problems». *Journal of the Association of Computer Machines*, 8, 212-229.
- Hutchison, J.; Leong, K.; Snyder, D. & Ward, F. (1989). «Scheduling for random job shop flexible manufacturing systems». *Proceedings of the Third ORSA/TIMS Conference on Flexible Manufacturing Systems*, 161-166.
- Ishii, N. & Talavage, J. (1991). «A transient-based real-time scheduling algorithm in FMS». *International Journal of Production Research*, 29, 12, 2501-2520.
- 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, 9, 2609-2626.
- Kim, Y.-D. (1990). «A comparison of dispatching rules for job shops with multiple identical jobs and alternative routings». *International Journal of Production Research*, 28, 5, 953-962.
- Kim, M. H. & Kim, Y.-D. (1994). «Simulation-based real-time scheduling in a flexible manufacturing system». *Journal of Manufacturing Systems*, 13, 2, 85-93.
- Kim, C.-O.; Min, H.-S. & Yih, Y. (1998). «Integration of inductive learning and neural networks for multi-objective FMS scheduling». *International Journal of Production Research*, 36, 9, 2497-2509.
- Kimemia, J. & Gershwin, S. B. (1985). «Flow optimization in flexible manufacturing systems». *International Journal of Production Research*, 23, 81-96.
- Kiran, A. & Alptekin, S. (1989). «A tardiness heuristic for scheduling flexible manufacturing systems». *15th Conference on Production Research and Technology: Advances in Manufacturing Systems Integration and Processes* (pp. 559-564), University of California at Berkeley, Berkeley, CA.
- Lashkari, R. S.; Dutta, S. P. & Padhye, A. M. (1987). «A new formulation of operation allocation problem in flexible manufacturing systems: mathematical modelling and computational experience». *International Journal of Production Research*, 25, 1267-1283.
- Lee, C.-Y.; Piramuthu, S. & Tsai, Y.-K. (1997). «Job shop scheduling with a genetic algorithm and machine learning». *International Journal of Production Research*, 35, 4, 1171-1191.
- Li, D.-C. & She, I.-S. (1994). «Using unsupervised learning technologies to induce scheduling knowledge for FMSs». *International Journal of Production Research*, 32, 9, 2187-2199.

- Michalski, R. S.; Carbonell, J. G. & Mitchell, T. M. (1983). *Machine Learning. An Artificial Intelligence Approach*. Palo Alto, CA: Tioga Press.
- Min, H.-S.; Yih, Y. & Kim, C.-O. (1998). «A competitive neural network approach to multi-objective FMS scheduling». *International Journal of Production Research*, 36, 7, 1749-1765.
- Montazeri, M. & Wassenhove, L. N. V. (1990). «Analysis of scheduling rules for an FMS». *International Journal of Production Research*, 28, 4, 785-802.
- Nakasuka, S. & Yoshida, T. (1992). «Dynamic scheduling system utilizing machine learning as a knowledge acquisition tool». *International Journal of Production Research*, 30, 2, 411-431.
- Panwalkar, S. S. & Iskander, W. (1977). «A survey of scheduling rules». *Operations Research*, 23, 5, 961-973.
- Pierreval, H. & Mebarki, N. (1997). «Dynamic selection of dispatching rules for manufacturing system scheduling». *International Journal of Production Research*, 35, 6, 1575-1591.
- Pierreval, H. & Ralambondrainy, H. (1990). «A simulation and learning technique for generating knowledge about manufacturing systems behaviour». *Journal of the Operational Research Society*, 41, 6, 461-474.
- Piramuthu, S.; Raman, N. & Shaw, M. J. (1994). «Learning-based scheduling in a flexible manufacturing flow line». *IEEE Transactions on Engineering Management*, 41, 2, 172-182.
- Piramuthu, S.; Raman, N.; Shaw, M. J. & Park, S. (1993). «Integration of simulation modeling and inductive learning in an adaptive decision support system». *Decision Support Systems*, 9, 127-142.
- Quinlan, J. R. (1986). «Induction of decision trees». *Machine Learning*, 1, 1, 81-106. Reprinted in J. W. Shavlik and T. G. Dietterich (eds.), *Readings in Machine Learning*. San Mateo, CA: Morgan Kaufmann, 1991. Reprinted in B. G. Buchanan and D. Wilkins (eds.), *Readings in Knowledge Acquisition and Learning*. San Mateo, CA: Morgan Kaufmann, 1992.
- Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann.
- Quiroga, L. A. & Rabelo, L. C. (1995). «Learning from examples: a review of machine learning, neural networks and fuzzy logic paradigms». *Computers & Industrial Engineering*, 29, 561-565.
- Rabelo, L. C. & Alptekin, S. (1989). «Integrating scheduling and control functions in computer integrated manufacturing using artificial intelligence». *Computers & Industrial Engineering*, 17, 101-106.
- Ramasesh, R. (1990). «Dynamic job shop scheduling: a survey of simulation studies». *OMEGA: The International Journal of Management Science*, 18, 1, 43-57.
- Russel, R. S.; Dar-El, E. M. & Taylor, B. W. (1987). «A comparative analysis of the COVERT job sequencing rule using various shop performance measures». *International Journal of Production Research*, 25, 10, 1523-1540.

- Sabuncuoglu, I. & Hommertzheim, D. L. (1992). «Dynamic dispatching algorithm for scheduling machines and automated guided vehicles in a flexible manufacturing system». *International Journal of Production Research*, 30, 5, 1059-1079.
- Sarin, S. C. & Salgame, R. R. (1990). «Development of a knowledge-based system for dynamic scheduling». *International Journal of Production Research*, 28, 8, 1499-1512.
- Shanker, K. & Rajamarthandan, S. (1989). «Loading problem in FMS: part movement minimization». *Proceedings of the Third ORSA/TIMS Conference on Flexible Manufacturing Systems*, 99-104.
- Shanker, K. & Tzen, Y. J. (1985). «A loading and dispatching problem in a random flexible manufacturing system». *International Journal of Production Research*, 23, 579-595.
- Shaw, M. J.; Park, S. & Raman, N. (1992). «Intelligent scheduling with machine learning capabilities: the induction of scheduling knowledge». *IIE Transactions*, 24, 2, 156-168.
- Stecke, K. E. (1983). «Formulation and solution of nonlinear integer production planning problems for flexible manufacturing systems». *Management Science*, 29, 3, March, 273-288.
- Stecke, K. E. & Solberg, J. (1981). «Loading and control policies for a flexible manufacturing system». *International Journal of Production Research*, 19, 5, 481-490.
- Sun, Y.-L. & Yih, Y. (1996). «An intelligent controller for manufacturing cells». *International Journal of Production Research*, 34, 8, 2353-2373.
- Tang, L.-L.; Yih, Y. & Liu, C.-Y. (1993). «A study on decision rules of a scheduling model in an FMS». *Computer in Industry*, 22, 1-13.
- Thesen, A. & Lei, L. (1986). «An expert system for scheduling robots in a flexible electroplating system with dynamically changing workloads». *Proceedings of the Second ORSA/TIMS Conference on Flexible Manufacturing Systems*, (pp. 555-566). Amsterdam: Elsevier Science Publishers.
- Vepsalainen, A. P. J. & Morton, T. E. (1987). «Priority rules for job shops with weighted tardiness costs». *Management Science*, 33, 8, 1035-1047.
- Watson, I. (1997). *Applying Case-Based Reasoning: Techniques for Enterprise Systems*. San Francisco, CA: Morgan Kaufmann Publishers.
- Wilson, J. M. (1989). «An alternative formulation of the operation-allocation problem in flexible manufacturing systems». *International Journal of Production Research*, 27, 1405-1412.
- Wu, S.-Y. D. & Wysk, R. A. (1988). «Multi-pass expert control system-a control/scheduling structure for flexible manufacturing cells». *Journal of Manufacturing Systems*, 7, 2, 107-120.
- Wu, S.-Y. D. & Wysk, R. A. (1989). «An application of discrete-event simulation to on-line control and scheduling in flexible manufacturing». *International Journal of Production Research*, 27, 9, 1603-1623.

ENGLISH SUMMARY

DYNAMIC SCHEDULING OF FLEXIBLE MANUFACTURING SYSTEMS THROUGH MACHINE LEARNING: AN ANALYSIS OF THE MAIN SCHEDULING SYSTEMS

PAOLO PRIORE*
DAVID DE LA FUENTE*
JAVIER PUENTE*
ALBERTO GÓMEZ*

A common way of dynamically scheduling jobs in a flexible manufacturing system (FMS) is by means of dispatching rules. The drawback of this method is that the performance of the manufacturing system depends on the state the manufacturing system is in at each moment, and no one rule exists that overrules the rest in all the possible states that the manufacturing system may be in. It would therefore be interesting to use the most appropriate dispatching rule at each moment. To achieve this goal, a scheduling system which uses machine learning can be used. By means of this technique, and by analysing the previous performance of the manufacturing system (training examples), knowledge is generated that can be used to decide which is the most appropriate dispatching rule at each moment in time. This paper provides a review of the main scheduling systems that use machine learning to vary the dispatching rule dynamically that have been described in the literature.

Keywords: Dynamic Scheduling, Machine Learning, Flexible Manufacturing Systems, Dispatching Rules, Simulation

AMS Classification (MSC 2000):

*Dpto. de Administración de Empresas y Contabilidad. Escuela Técnica Superior de Ingenieros Industriales e Informáticos de Gijón. Universidad de Oviedo.

–Received December 1999.

–Accepted July 2001.

A common way of dynamically scheduling jobs in a flexible manufacturing system (FMS) is by means of dispatching rules. The problem of this method is that the performance of these rules depends on the state the system is in at each moment, and no one rule exists that overrules the rest in all the possible states that the system may be in. It would therefore be interesting to use the most appropriate dispatching rule at each moment. To achieve this goal, a scheduling approach which uses machine learning can be used. By means of this technique, analysing the previous performance of the system (training examples), a knowledge is generated that can be used to decide which is the most appropriate dispatching rule at each moment in time. In this paper, a review of the main machine learning-based scheduling approaches described in the literature is presented.

Basically, two approaches can be found in the literature to modify dispatching rules dynamically. Firstly, the rule is selected at the appropriate moment by simulating a set of pre-established dispatching rules and selecting the one that provides the best performance. In the second approach, belonging to the field of artificial intelligence, a set of earlier system simulations (training examples) is used to determine which is the best rule for each possible system state. These training cases are used to train a machine learning module to acquire knowledge about the manufacturing system. Such knowledge is then used to make intelligent decisions in real time. These systems are normally said to be knowledge-based.

Several knowledge-based approaches that dynamically modify the dispatching rule being used at a specific instance are reviewed next. Depending on the type of machine learning algorithm used, these approaches can be divided into the following categories:

1. Approaches that do not use knowledge acquisition algorithms.
2. Inductive learning-based approaches.
3. Neural network-based approaches.
4. Mixed approaches. Here a combination of different types of learning algorithm is applied.

Inductive learning algorithms or neural networks are used in many of the above approaches to acquire knowledge. The advantage of neural networks over inductive learning is that the class need not be predefined (when the number of decision variables is high, the number of simulations to be done to determine the class is great). However, the knowledge of the network is not intelligible for the user and it is difficult for him or her to modify the knowledge. It is similarly easy to identify which attributes really affect performance of the dispatching rules considered by using an inductive learning algorithm (Shaw *et al.*, 1992).

Moreover, a series of common defects are generally noticeable in the above-mentioned knowledge-based approaches that use machine learning algorithms. These defects might

suggest lines of research in the field of dynamic scheduling of manufacturing systems, based on modifying the dispatching rule used. These research lines are the following:

1. A comparison of the different machine learning methodologies. One methodology, or in some cases a combination of them, is used in the approaches presented in the literature. However, there is no comparative study to determine which is the best of them. Furthermore, it would be interesting to use case-based reasoning (CBR) as a machine learning methodology in scheduling systems.
2. Selection of an adequate period of monitoring. In the literature available no study is generally made to calculate the monitoring period for each performance criterion.
3. The determination of a mechanism that mitigates transitory states. In most works that were studied this mechanism is not taken account of, and when it is, variable time is not used.
4. Determination of the optimum number of training examples. In most approaches that were considered the number of examples required to train the machine learning algorithm optimally is not calculated. On the other hand, it's not specified if test examples are similar or very different to the training examples.
5. Generation of new attributes. Development of an algorithm that serves to create new attributes that are linear combinations of the initial ones.
6. Incorporation of a simulator. The scheduling system's performance could be improved if a simulator were incorporated to determine the best rule from amongst the machine learning system considers most important.
7. Refinement of the knowledge base. It would be interesting to establish a procedure to automatically modify knowledge when important changes in the manufacturing system occur.