

## UTILIZACIÓN DEL APRENDIZAJE INDUCTIVO EN LA TOMA DE DECISIONES. APLICACIÓN EN UN PROBLEMA DE SECUENCIACIÓN

Priore, P.  
De la Fuente, D.  
Pino, R.  
Puente, J.

Universidad de Oviedo

### RESUMEN:

El aprendizaje inductivo, que pertenece al campo de la inteligencia artificial, permite resolver un problema mediante el empleo de problemas resueltos en el pasado similares al planteado. En este artículo se describen las principales características del aprendizaje inductivo, así como las aplicaciones más comunes en las que se utiliza esta técnica. Asimismo, a modo de ejemplo, se presenta, de forma detallada, una aplicación en la que se emplea aprendizaje inductivo. En ésta, se resuelve un problema de secuenciación en un sistema de fabricación flexible (FMS, de Flexible Manufacturing System). En este tipo de sistemas, la forma habitual de secuenciar dinámicamente los trabajos es mediante el empleo de reglas de secuenciación. Sin embargo, el problema de este método es que el comportamiento de las reglas depende del estado del sistema de fabricación, y no existe una regla que supere a las demás en todos los posibles estados que puede presentar el mismo. Para resolver este inconveniente, se presenta una metodología que utiliza aprendizaje inductivo que permite usar, en cada momento, la regla de secuenciación más adecuada.

**PALABRAS CLAVE:** Secuenciación, Aprendizaje Inductivo, FMS, Simulación.

### ABSTRACT:

Inductive learning, which is part of the field of artificial intelligence, allows a problem to be solved by using the solutions to problems in the past that are similar to the problem at hand. This paper describes the hallmarks of inductive learning, and the most common applications when this technique is applied. Furthermore, and by way of example, an application involving inductive learning is presented in detail. In the application in question, a scheduling problem in a flexible manufacturing system (FMS) is solved. A common method of dynamically scheduling jobs in FMSs is to employ dispatching rules. However, the problem associated with this method is that the performance of the rules depends on the state of the system, but there is no rule that is superior to all the others for all the possible states the system might be in. To achieve this, an approach is presented that uses inductive learning that allows the best dispatching rule to be applied for each given moment in time.

**KEYWORDS:** Scheduling, Inductive Learning, FMS, Simulation.

## 1. INTRODUCCIÓN

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). Las principales técnicas dentro del aprendizaje automático son las redes neuronales, el aprendizaje inductivo y el razonamiento basado en casos (CBR, de Case-Based Reasoning). La diferencia fundamental entre estas técnicas 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.

Por otro lado, uno de los métodos más utilizados para resolver el problema de secuenciación en los FMSs es el empleo de reglas de secuenciación. Hasta la fecha, muchos investigadores (ver por ejemplo, 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 et al., 1993) han evaluado el comportamiento de las reglas de secuenciación en los FMSs, pudiéndose concluir a partir de estos estudios que el comportamiento de estas reglas depende del criterio de eficiencia elegido, de la configuración del sistema y de las condiciones de éste (carga global del sistema, equilibrio de las cargas en las distintas máquinas, la holgura de las fechas de entrega, etc.).

Por ello, sería interesante cambiar las reglas de secuenciación en el momento apropiado, para lo cual existen básicamente dos enfoques. En el primero, la regla de secuenciación se determina, para cada período de tiempo, simulando un conjunto de reglas de secuenciación predeterminadas y eligiendo la que presente mejor comportamiento (ver por ejemplo, Ishii y Talavage, 1991; Wu y Wysk, 1989). En el segundo enfoque, se emplean un conjunto de simulaciones previas del sistema de fabricación (ejemplos de entrenamiento o problemas resueltos en el pasado) para determinar, mediante un algoritmo de aprendizaje automático, cuál es la mejor de las reglas en cada posible estado del sistema (ver por ejemplo, Nakasuka y Yoshida, 1990; Shaw et al., 1992). Aytug et al. (1994) y Priore et al. (2001) muestran una revisión de trabajos en los cuales se aplica aprendizaje automático para resolver el problema de secuenciación. En este trabajo, se propone una nueva metodología de secuenciación para FMSs que utiliza aprendizaje inductivo.

Este trabajo está organizado de la siguiente forma. En primer lugar, se describe el aprendizaje inductivo y una clasificación de las aplicaciones en las que se usa esta técnica. A continuación, se detalla el algoritmo de aprendizaje inductivo C4.5 (Quinlan, 1993) debido a que es el algoritmo de aprendizaje inductivo más utilizado y referenciado; asimismo, se enumeran las principales ventajas e inconvenientes del algoritmo C4.5. Posteriormente, se presenta una nueva metodología para secuenciar trabajos utilizando aprendizaje inductivo y se efectúa el estudio experimental. Finalmente, se describe un resumen de los resultados del trabajo.

## **2. APRENDIZAJE INDUCTIVO**

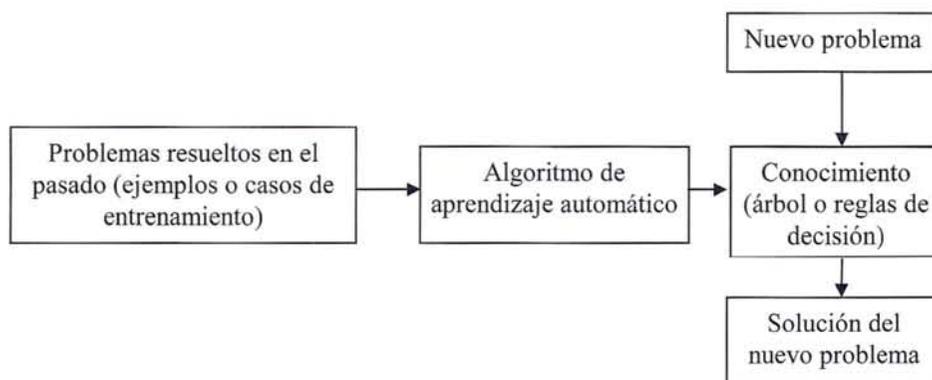
La mayoría de las aplicaciones prácticas de la inteligencia artificial están basadas en la construcción de un modelo del conocimiento utilizado por un experto humano. Este enfoque, que ha tenido un impacto importante a partir de los ochenta, se muestra en numerosos casos de estudio (Feigenbaum et al., 1988). En la mayor parte de éstos, la tarea que ejecuta el experto es de clasificación; por ejemplo, Feigenbaum et al. (1988) citan un sistema, desarrollado por American Express, para asistir a las personas que autorizan créditos.

Existen dos formas muy diferentes de construir los modelos del conocimiento; por una parte, estos se pueden obtener entrevistando al experto o expertos apropiados. La mayoría de los sistemas basados en el conocimiento han sido elaborados de esta forma, a pesar de las dificultades relacionadas con este enfoque (Michie, 1987, 1989). Alternativamente, se pueden examinar las clasificaciones realizadas en el pasado y construir un modelo mediante un algoritmo de aprendizaje automático que permita determinar la solución de un nuevo problema

a partir de los ejemplos o problemas resueltos en el pasado. Una representación esquemática de lo expuesto anteriormente se muestra en la figura 1. En este caso, el conocimiento se expresa en forma de un árbol o conjunto de reglas de decisión.

Los ejemplos o problemas resueltos en el pasado (también denominados ejemplos o casos de entrenamiento) se pueden representar mediante una tabla atributo-valor, como la que se muestra en la tabla 1, donde hay un atributo especial que se denomina la clase (la solución del ejemplo o problema resuelto). Los atributos representan las características del problema. El objetivo que tiene el aprendizaje inductivo (o cualquier otra técnica de aprendizaje automático) es tratar de aprender a clasificar nuevos problemas, similares a los de entrenamiento, de los que se conocen los valores de todos los atributos excepto la clase (la solución del problema). 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.

**Figura 1. Esquema general del problema de aprendizaje**



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

Ejemplos	Atributo 1	Atributo 2	.....	Atributo m	Clase
Ejemplo 1	$A_{11}$	$A_{12}$	.....	$A_{1m}$	$C_1$
Ejemplo 2	$A_{21}$	$A_{22}$	.....	$A_{2m}$	$C_2$
.....	.....	.....	.....	.....	.....
Ejemplo n	$A_{n1}$	$A_{n2}$	.....	$A_{nm}$	$C_n$

Sin embargo, es importante destacar que no todas las tareas de clasificación se prestan a este enfoque, y aún menos a los métodos particulares. Por ello, es necesario subrayar los principales requerimientos imprescindibles que deben cumplir las tareas de clasificación para poder aplicar este enfoque:

1. Descripción valor-atributo. Los datos que se han de analizar se deben expresar en términos de una colección fija de propiedades o atributos. Por lo tanto, no son adecuados dominios, en los cuales los datos tienen una estructura inherentemente variable.

2. Clases predefinidas. Las categorías que se asignan a los casos deben, en general, establecerse de antemano. En la terminología de aprendizaje automático, se habla de aprendizaje supervisado, en contraste con el aprendizaje no supervisado, en el que el agrupamiento de los casos se realiza por análisis (Fisher et al., 1991).
3. Clases discretas. Las clases deben estar perfectamente definidas; de esta forma, un caso pertenece, o no, a una clase particular, y debe haber bastantes más casos que clases. Sin embargo, algunas tareas no son de este tipo; por ejemplo, la predicción de valores continuos, tales como el precio del oro o la temperatura a la cual una aleación se funde. De todas formas, una clase continua se puede dividir en categorías vagas, tales como duro, bastante duro, flexible, bastante dúctil y dúctil.
4. Datos suficientes. La cantidad de datos requeridos depende de numerosos factores, como el número de propiedades y clases o la complejidad del modelo de clasificación, que al incrementarse hacen necesarios más datos para construir un sistema fiable. Un modelo simple se puede, a veces, identificar con pocos casos; sin embargo, un modelo de clasificación detallado requiere, normalmente, cientos o incluso miles de casos de entrenamiento.

El aprendizaje inductivo se ha utilizado en numerosas tareas de clasificación entre las que destacan (Langley y Simon, 1995):

1. Previsión. Por ejemplo, la predicción de un fallo en un equipo o de la calidad de un producto.
2. Valoración. Por ejemplo, la decisión de conceder o no un crédito a un cliente de una entidad financiera
3. Diagnóstico. Por ejemplo, la realización de diagnósticos médicos o de un equipo.
4. Control de procesos. Por ejemplo, la secuenciación de trabajos en un sistema de fabricación flexible.

### 3. EL ALGORITMO C4.5

Dentro de los algoritmos de aprendizaje inductivo, el más popular es el ID3 desarrollado por Quinlan (Quinlan, 1979, 1983, 1986). Además de ser un buen algoritmo, cuenta con el propio autor como gran difusor y con la transparencia científica que lo diferencia de otros algoritmos poco conocidos y herméticos como es el caso de los AQx (Michalski et al., 1983). Por lo que se refiere a C4.5, como señala Quinlan, es una evolución de ID3. Desde su aparición, C4.5 es el algoritmo de aprendizaje inductivo más utilizado y referenciado. El algoritmo más importante de C4.5 es el que genera un árbol de decisión inicial a partir de un conjunto de ejemplos de entrenamiento. La idea original procede de los trabajos de Hoveland y Hunt a finales de los cincuenta, que culminan en la siguiente década con los sistemas de aprendizaje de conceptos (Hunt et al., 1966). El esquema del método de Hunt para construir un árbol de decisión a partir de un conjunto  $T$  de ejemplos de entrenamiento es simple. Si el número de clases es  $k$ , existen tres posibilidades:

1. El conjunto  $T$  contiene uno o más ejemplos, todos pertenecientes a una única clase ( $C_j$ ). En este caso, el árbol de decisión para  $T$  es una hoja identificando la clase  $C_j$ .

2. El conjunto  $T$  no contiene ejemplos. En esta situación, el árbol de decisión es de nuevo una hoja; pero la clase de ésta, debe establecerse siguiendo un criterio determinado. Por ejemplo, la clase de la hoja se puede elegir en función del conocimiento que se posea del dominio. Así, C4.5 utiliza la clase más frecuente del padre de este nodo.
3. El conjunto  $T$  contiene ejemplos que pertenecen a distintas clases. En este caso, la idea es dividir  $T$  en subconjuntos, que sean o tiendan a grupos de ejemplos con una única clase. Para ello, se elige un test, basado en un único atributo, que tenga una, o más salidas mutuamente excluyentes  $\{O_1, O_2, \dots, O_n\}$ . Entonces,  $T$  se divide en  $n$  subconjuntos  $\{T_1, T_2, \dots, T_n\}$ , donde  $T_i$  contiene todos los casos de  $T$  que tienen como salida  $O_i$  en el test elegido. Así, el árbol de decisión para  $T$  consta de un nodo de decisión identificando el test y una rama para cada posible salida. Esta misma metodología se aplica de forma recursiva a cada subconjunto de ejemplos de entrenamiento. Una representación gráfica del método de Hunt se muestra en la figura 2.

Cualquier test que divida  $T$  de forma no trivial, de modo que al menos dos de los subconjuntos no estén vacíos, dará, al final, una partición en subconjuntos de clase única, aunque todos o casi todos contengan un único caso de entrenamiento. Sin embargo, el proceso de construcción del árbol no sólo tiene la intención de encontrar cualquier partición, sino construir un árbol que revele la estructura del dominio y, de este modo, tenga cierto poder de predicción. Para ello, se necesita un número significativo de casos en cada hoja o, dicho de otra manera, la partición debe producir un número mínimo de bloques. Por lo tanto, lo ideal sería elegir un test en cada etapa, de modo que el árbol final sea lo más pequeño posible.

En el caso de C4.5, la elección del atributo adecuado, para dividir el conjunto de ejemplos presentes en un nodo, se realiza siguiendo dos criterios denominados  $g_1$  y  $g_2$ , basados en la teoría de la información, de acuerdo con las siguientes expresiones (Quinlan, 1993; Whiston, 1994):

$$g_1 = - \sum_{j=1}^k \frac{\text{freq}(C_j, T)}{|T|} * \log_2 \left( \frac{\text{freq}(C_j, T)}{|T|} \right) + \sum_{i=1}^n \frac{|T_i|}{|T|} * \sum_{j=1}^k \frac{\text{freq}(C_j, T_i)}{|T_i|} * \log_2 \left( \frac{\text{freq}(C_j, T_i)}{|T_i|} \right)$$

$$g_2 = \frac{g_1}{- \sum_{i=1}^n \frac{|T_i|}{|T|} * \log_2 \left( \frac{|T_i|}{|T|} \right)}$$

donde:

$n$ : es el número de ramas formadas por el test

$k$ : es el número de clases existentes

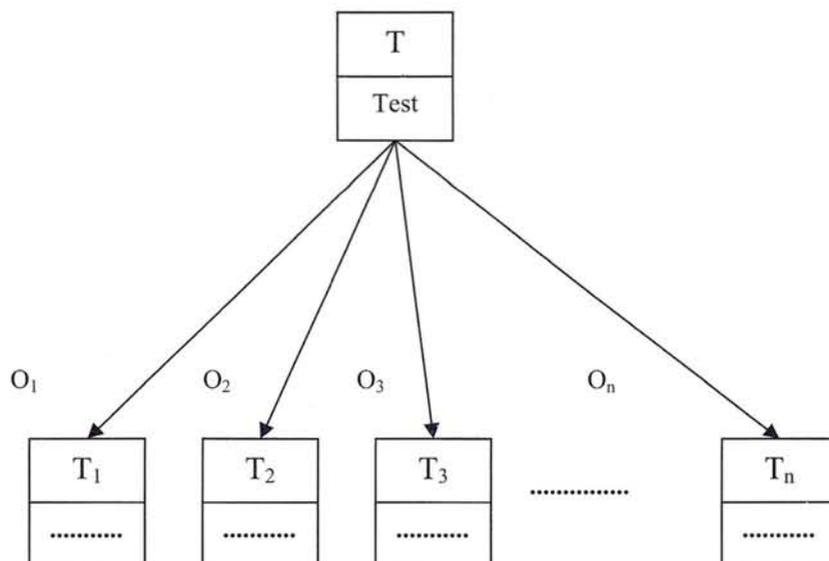
$|T|$ : es el número de casos en el conjunto  $T$

$\text{freq}(C_j, T)$ : es el número de casos de clase  $C_j$  en  $T$

$|T_i|$ : es el número de casos en el conjunto  $T_i$

$\text{freq}(C_j, T_i)$ : es el número de casos de clase  $C_j$  en  $T_i$

**Figura 2. Esquema general del método de Hunt**



La maximización de estos criterios favorece la elección de un atributo en el test, de forma que los conjuntos de ejemplos en cada una de sus ramas sean lo más homogéneos posibles. Un conjunto se dice que es homogéneo, cuando la mayor parte de sus casos pertenecen a una clase determinada. El segundo criterio ( $g_2$ ) es más robusto que el primero ( $g_1$ ) debido a que este último, favorece los tests con muchas ramas que, desde un punto de vista predictivo, son poco útiles (Quinlan, 1988). Este método de partición recursivo, para la construcción de árboles de decisión, continúa subdividiendo el conjunto de ejemplos de entrenamiento, hasta que cada uno de los subconjuntos contenga casos de una única clase, o no exista un test que ofrezca una mejora en los resultados.

Posteriormente, si se desea, se puede transformar el árbol de decisión en un conjunto de reglas de decisión. Una vez obtenido el árbol o conjunto de reglas de decisión, se puede clasificar un nuevo caso, del que se conocen todos los atributos excepto la clase.

#### 4. VENTAJAS E INCONVENIENTES DEL ALGORITMO C4.5

Entre las numerosas ventajas que posee el algoritmo de aprendizaje inductivo C4.5, Quinlan (1993) destaca las siguientes:

1. El proceso de construcción de árboles es muy rápido y, por lo tanto, es capaz de procesar un elevado número de ejemplos de entrenamiento en muy poco tiempo.
2. No depende, a diferencia de otros enfoques estadísticos, de hipótesis sobre la distribución de los valores o la independencia de los atributos.
3. Numerosos autores (ver por ejemplo, Breiman et al., 1984; Buntine, 1991) han comparado clasificadores basados en árboles y reglas de decisión con otros enfoques que usan técnicas estadísticas y redes neuronales; de estos estudios, se

pueden extraer dos conclusiones generales. Por una parte, los métodos basados en árboles y redes neuronales tienden a ser más robustos que la mayor parte de las técnicas de tipo estadístico. Por otra parte, los clasificadores basados en árboles y redes, normalmente, tienen una exactitud similar. Sin embargo, las redes requieren mayor tiempo de cálculo y sus resultados son ininteligibles para el ser humano.

Por otra parte, desde un punto de vista geométrico, el árbol de decisión especifica cómo dividir el espacio de casos de entrenamiento en regiones asociadas a cada una de las clases. Debido al tipo de test utilizado en C4.5, las divisiones se realizan mediante hiperplanos que son ortogonales al eje del atributo empleado en el test y paralelos al resto de ejes. Por lo tanto, los árboles de decisión que produce C4.5 generan regiones en forma de hiperrectángulos. Debido a lo anterior, C4.5 no funciona adecuadamente en las siguientes situaciones (Quinlan, 1993):

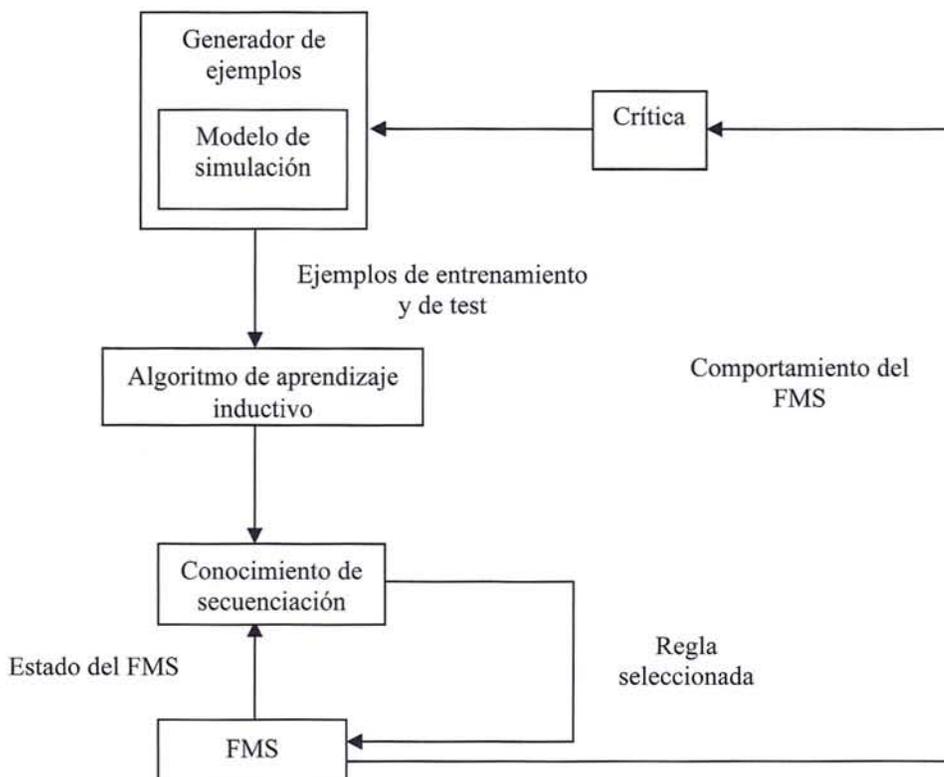
1. Regiones de clase no hiperrectangulares. En esta situación, el árbol de decisión intenta aproximar las regiones por hiperrectángulos. Cuando el número de casos de entrenamiento aumenta, la aproximación será más exacta, pero a cambio de un incremento substancial en el número de regiones y del propio árbol. La solución a este problema es buscar una combinación aritmética de los atributos e incluirla como un nuevo atributo.
2. Regiones de clase pobremente delineadas. Los árboles de decisión se construyen de forma iterativa, dividiendo las regiones existentes del espacio de casos de entrenamiento, con el propósito de generar regiones formadas por puntos de una única clase. Sin embargo, existen dos situaciones donde este objetivo no se consigue fácilmente: (1) cuando la densidad de puntos en algunas regiones es baja, de modo que hay una gran libertad en la posición de las superficies frontera; (2) la tarea de clasificación es esencialmente probabilística, lo que genera regiones objetivo con un número substancial de puntos que no pertenecen a la clase mayoritaria. En cualquiera de las dos situaciones, la naturaleza imprecisa de las regiones finales no facilita las divisiones iniciales del espacio de casos de entrenamiento.
3. Regiones de clase fragmentadas. En determinadas circunstancias, una descripción de clase que parece muy simple se traduce en un clasificador que requiere numerosas regiones. El número de casos de entrenamiento necesario para construir un clasificador, depende directamente del número de regiones precisas para especificarlo. Otra causa de regiones fragmentadas, no es el concepto en sí mismo, sino la presencia de atributos no relevantes. La solución a este problema, una vez construido el árbol con todos los atributos, es quitar aquellos atributos que no aparecen en el árbol o no están en los caminos de las hojas más pobladas. En ocasiones, la fragmentación del espacio se puede evitar introduciendo nuevos atributos, en vez de eliminarlos. La fragmentación, causada por atributos irrelevantes o la complejidad del concepto, es un obstáculo serio en la inducción de clasificadores eficientes y completos. Cualquier transformación del espacio, eliminando atributos o incluyendo combinaciones aritméticas o lógicas de ellos, que tenga el efecto de reducir el número de regiones necesarias para representar un clasificador, es beneficiosa. Si no se pueden encontrar dichas transformaciones, la única forma de superar la fragmentación es mediante la generación de más datos de entrenamiento.

## 5. SECUENCIACIÓN DINÁMICA MEDIANTE APRENDIZAJE INDUCTIVO

La metodología que se propone en este trabajo, para secuenciar tareas utilizando aprendizaje inductivo, se puede dividir en cuatro etapas, tal como se muestra en la figura 3:

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 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 frente a las diversas reglas de secuenciación que se pretenden utilizar.

**Figura 3. Secuenciación dinámica mediante aprendizaje inductivo**



2. Determinación del “conocimiento de secuenciación” (en forma de árbol o reglas de decisión) mediante un algoritmo de aprendizaje inductivo.
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 aprendizaje inductivo y la mejor regla de secuenciación, o combinación de ellas. Si este segundo método produce un comportamiento del sistema superior, se debe regresar al primer paso.

## 6. ESTUDIO EXPERIMENTAL

El FMS considerado es similar al propuesto por Min et al. (1998) y Kim et al. (1998), y está formado por cuatro centros de mecanizado, un sistema de manipulación de materiales y un almacén para piezas semiacabadas con una capacidad máxima de 32 piezas. Cada centro de mecanizado tiene su propio “buffer” de entrada y de salida. Asimismo, debido a la flexibilidad del sistema, existe la posibilidad de realizar cualquier operación en dos máquinas alternativas. En la figura 4, se muestra un esquema del FMS propuesto. En este sistema se consideran dos tipos de decisiones; en primer lugar, la selección de las piezas asignadas a un centro de mecanizado por parte de éste. Para ello, las reglas de secuenciación utilizadas en esta configuración son: SPT (Shortest Processing Time); EDD (Earliest Due Date); MDD (Modified Job Due Date) y SRPT (Shortest Remaining Processing Time). Se han seleccionado estas reglas debido a que presentan un buen comportamiento en diversos estudios realizados previamente (Kim et al., 1998; Min et al., 1998; Shaw et al., 1992). Estas reglas ordenan los diversos trabajos, que compiten por el uso de una máquina dada, mediante diferentes esquemas de prioridad. A cada trabajo se le asigna un índice de prioridad y aquel que posea el menor índice será seleccionado en primer lugar. El cálculo del índice de prioridad para cada regla se define del siguiente modo:

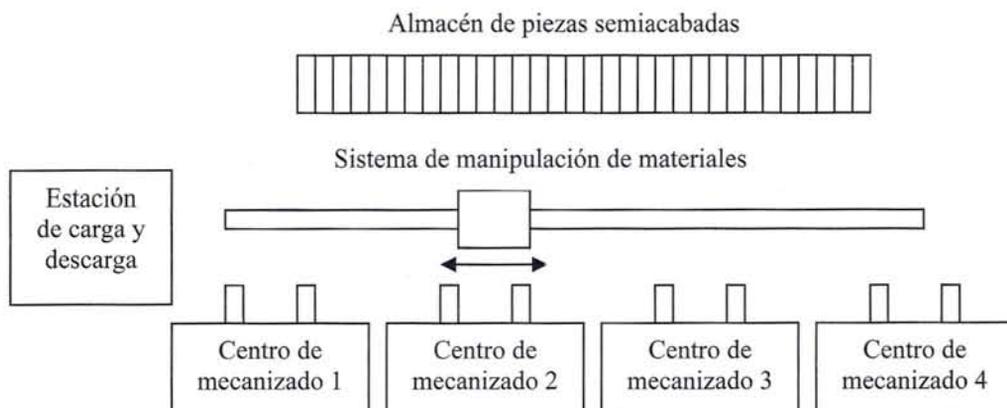
SPT:	$p_{ij}$
EDD:	$d_i$
MDD:	$\max \{t + P_{ij}, d_i\}$
SRPT:	$P_{ij}$

donde  $d_i$  es la fecha de entrega del trabajo  $i$ ;  $p_{ij}$  es el tiempo de procesamiento de la operación  $j$  del trabajo  $i$ ;  $P_{ij}$  es el tiempo de procesamiento que queda del trabajo  $i$  al comienzo de la operación  $j$  y  $t$  es el momento en que se toma la decisión de secuenciación. La fecha de entrega del trabajo  $i$  ( $d_i$ ) se calcula, de acuerdo con Baker (1984), mediante la siguiente expresión:

$$d_i = t_i + p_i * F$$

donde  $F$  es un parámetro que determina la holgura de la fecha de entrega del trabajo  $i$ ;  $t_i$  es el momento en que llega el trabajo  $i$  al sistema y  $p_i$  es el tiempo de procesamiento total del trabajo  $i$ .

**Figura 4. Sistema de fabricación flexible propuesto**



El segundo tipo de decisión, corresponde a la selección de las máquinas por parte de las piezas, debido a que una operación se puede realizar en diferentes máquinas. Para ello, las reglas de secuenciación utilizadas en esta configuración son (Kim et al., 1998; Min et al., 1998; O'keefe y Kasirajan, 1992):

1. SPT (Shortest Processing Time): elige la máquina que ejecuta la operación en el menor tiempo.
2. NINQ (Number in Queue): selecciona la máquina que tiene el menor número de trabajos pendientes de realizar.
3. WINQ (Work in Queue): elige la máquina que tiene la menor cantidad de trabajo por ejecutar.
4. LUS (Lowest Utilized Station): selecciona la máquina que posee el menor grado de utilización.

### 6.1. Generación de los ejemplos de entrenamiento y de test

Los ejemplos necesarios para la etapa de aprendizaje se obtienen, mediante simulación, utilizando el programa Witness (Witness, 1996). Para ello, se realizan las siguientes suposiciones:

1. Los trabajos, o piezas, llegan al sistema siguiendo una distribución de Poisson.
2. Los tiempos de procesamiento de cada operación se ajustan a una distribución exponencial con un valor medio de uno.
3. El número de operaciones que se asigna a cada trabajo sigue una distribución uniforme entera, siendo los valores extremos del intervalo uno y cuatro.
4. La probabilidad de asignar una operación a una máquina depende de los parámetros  $PO_i$  (porcentaje de operaciones asignadas a la máquina  $i$ ). Estos porcentajes fluctúan entre el 10% y el 40%. Asimismo, se supone que las dos primeras máquinas son las que soportan más carga de trabajo.

5. El número de máquinas alternativas para realizar una operación varía entre uno y dos.
6. El ritmo de llegada de los trabajos se modifica de forma que la utilización global del sistema oscile entre el 55% y el 95%.
7. El valor del factor  $F$  está comprendido entre uno y diez.

Los criterios que se utilizan para medir el comportamiento del sistema de fabricación son el retraso medio de los trabajos y el tiempo medio de un trabajo en el sistema, ya que son muy empleados en la literatura de secuenciación. El retraso medio se define como:

$$\frac{\sum T_i}{N} \quad T_i = \max \{0, L_i\}$$

donde  $N$  es el número de trabajos finalizados y  $L_i$  es la diferencia entre la fecha en que se termina el trabajo y la fecha de entrega comprometida ( $d_i$ ).

Para generar los ejemplos de entrenamiento y de test, es necesario, en primer lugar, definir los atributos de control que se utilizan para caracterizar el sistema de fabricación. Los atributos empleados en el FMS propuesto son los siguientes (Kim et al., 1998; Min et al., 1998; Shaw et al., 1992):  $F$ , parámetro que determina la holgura de la fecha de entrega de un trabajo; NMAO, número de máquinas alternativas para realizar una operación; UM, utilización media del sistema de fabricación;  $U_i$ , utilización de la máquina  $i$ ; WIP, número medio de piezas en el sistema; FUMCB, cociente entre la utilización de la máquina cuello de botella y la utilización media del sistema de fabricación; FDUM, cociente entre la desviación típica de las utilidades de las máquinas del sistema y la utilización media. Para calcular estos dos últimos atributos se emplean las siguientes expresiones:

$$FDUM = \frac{10 * \sqrt{\frac{(U_1 - UM)^2 + (U_2 - UM)^2 + (U_3 - UM)^2 + (U_4 - UM)^2}{3}}}{UM}$$

$$FUMCB = \frac{10 * \max (U_1, U_2, U_3, U_4)}{UM}$$

En total, se generan 1100 combinaciones diferentes de los atributos de control de forma aleatoria, utilizando 100 de ellas, como ejemplos de test. Para cada combinación de atributos, se determinan los valores de retraso medio y de tiempo medio en el sistema resultantes de emplear cada una de las reglas de secuenciación de forma aislada. En realidad, para generar un ejemplo de entrenamiento o de test, se necesitan realizar 16 simulaciones, pues existen cuatro reglas para cada una de las dos decisiones que se deben tomar.

En la tabla 2, se muestra el número de veces que se elige cada una de las combinaciones de las reglas de secuenciación empleadas (en tanto por ciento), para el criterio del retraso medio y del tiempo medio en el sistema. Se puede observar que la mayor parte de las combinaciones son elegidas, en mayor o menor medida, en cada uno de los 1100 escenarios propuestos. Este hecho constata la necesidad de modificar las reglas de secuenciación, en tiempo real, en función del estado que presente el sistema de fabricación en cada momento.

**Tabla 2. Comparación de las combinaciones de las reglas de secuenciación**

Combinación de reglas de secuenciación	Retraso medio	Tiempo medio en el sistema	Combinación de reglas de secuenciación	Retraso medio	Tiempo medio en el sistema
SPT+SPT	23.27%	50%	MDD+SPT	25.64%	0%
SPT+NINQ	4.64%	22.45%	MDD+NINQ	11.27%	0%
SPT+WINQ	6.09%	25%	MDD+WINQ	3.82%	0.64%
SPT+LUS	0.36%	0%	MDD+LUS	1.27%	0%
EDD+SPT	1.09%	0%	SRPT+SPT	0%	0%
EDD+NINQ	8.73%	0%	SRPT+NINQ	0%	0%
EDD+WINQ	4.73%	0%	SRPT+WINQ	0%	1.91%
EDD+LUS	9.09%	0%	SRPT+LUS	0%	0%

## 6.2. Aplicación del aprendizaje inductivo

Para obtener el “conocimiento de secuenciación” (formado en este caso por un árbol o un conjunto de reglas de decisión) se utiliza el algoritmo de aprendizaje inductivo C4.5. En la tabla 3, se muestra un resumen de los resultados logrados en la etapa de entrenamiento y de test, empleando conjuntos de ejemplos de entrenamiento de distinto tamaño para el criterio del retraso medio. En general, se puede observar que a medida que aumenta el número de ejemplos de entrenamiento utilizado, el tamaño del árbol de decisión, así como el número de reglas extraídas de éste, se incrementa de forma considerable. Sin embargo, el error de reescritura (error sobre los mismos ejemplos de entrenamiento) y, sobre todo, el error en los ejemplos de test (ejemplos que no han sido tratados anteriormente) se reduce si se incrementa el número de ejemplos de entrenamiento utilizado.

De los resultados presentados en la tabla 3, se puede notar que a partir de 600 ejemplos de entrenamiento, el error de test en las reglas oscila entre un 19% y un 16%, siendo con 750 y 1000 ejemplos cuando se logra el menor error de test. A continuación, a modo de ilustración, se presenta una muestra del conjunto de reglas obtenidas con 750 ejemplos de entrenamiento para el criterio del retraso medio:

Regla 1: IF  $F > 7$  AND  $WIP \leq 4.8$  AND  $U1 \leq 65.7$  AND  $U2 > 64.7$

THEN SPT+LUS [50.0%]

Regla 2: IF  $F > 7$  AND  $NMAO \leq 1$  AND  $FUMCB > 2.9$  AND  $U2 \leq 84$

THEN EDD+SPT [63.0%]

Regla 3: IF  $F > 9$  AND  $NMAO \leq 1$  AND  $U2 \leq 83.4$

THEN EDD+SPT [50.0%]

.....  
Regla 47: IF  $FDUM \leq 0$

THEN EDD+LUS [71.8%]

Regla 48: IF F > 2 AND F <= 5 AND NMAO > 1 AND U1 > 74.7 AND U2 > 74.1  
THEN MDD+NINQ [74.0%]

Regla 49: IF F > 2 AND F <= 6 AND NMAO > 1 AND WIP > 5.9 AND U2 > 74.1  
THEN MDD+NINQ [69.6%]

Regla por defecto: MDD+NINQ

**Tabla 3. Errores de entrenamiento y de test utilizando C4.5 para el criterio del retraso medio**

Número de ejemplos de entrenamiento	Tamaño del árbol	Número de reglas	Error de reescritura en el árbol	Error de reescritura en las reglas	Error de test en el árbol	Error de test en las reglas
50	17	7	12%	14%	20%	22%
100	27	10	11%	15%	27%	31%
150	33	17	10.7%	10%	28%	24%
200	39	19	10.5%	10.5%	20%	21%
250	49	21	10.4%	10%	18%	19%
300	55	23	9.7%	9%	17%	18%
350	69	24	8.3%	9.1%	16%	17%
400	83	30	8.5%	10%	15%	17%
450	73	31	9.3%	8.7%	20%	21%
500	81	32	10.6%	10%	20%	22%
550	93	36	8.4%	9.3%	20%	21%
600	99	39	8%	7.7%	17%	17%
650	111	37	7.8%	9.2%	16%	17%
700	119	39	8%	10.1%	18%	18%
750	103	49	9.9%	8.8%	14%	16%
800	117	41	9%	10%	17%	19%
850	129	47	8.6%	9.1%	17%	19%
900	135	49	8%	8.8%	16%	18%
950	137	47	8.2%	8.9%	16%	19%
1000	131	49	8.5%	8.5%	17%	16%

Cuando se presenta un nuevo caso para clasificar (estado del sistema de fabricación en un instante determinado), se prueba cada una de las 49 reglas en el orden establecido. La conclusión de la primera regla heurística que cumpla el caso, proporciona la combinación de las reglas de secuenciación que se debe aplicar. La clase o regla por defecto es MDD+NINQ, y será la regla que se utilice cuando no se elija, en el momento de la evaluación, ninguna de las restantes 49 reglas heurísticas. El número que aparece junto a la conclusión de cada regla, es la probabilidad que tiene dicha regla de clasificar correctamente un caso no visto con anterioridad.

En la tabla 4, se presentan los resultados logrados, en la etapa de entrenamiento y de test, con el criterio del tiempo medio de un trabajo en el sistema. De nuevo, se observa que, normalmente, a medida que se incrementa el número de ejemplos de entrenamiento empleado,

el tamaño del árbol de decisión, así como el número de reglas extraídas de éste, aumenta. De todas formas, este incremento es menor al obtenido si se utiliza el retraso medio como criterio de comportamiento. Asimismo, los errores son también bastante inferiores a los mostrados en la tabla 3; esto, es debido a que el número de clases, realmente utilizadas, es de cinco frente a las doce empleadas para el criterio del retraso medio (ver tabla 2). Por otro lado, se puede notar que a partir de 650 ejemplos de entrenamiento, el error de test en las reglas oscila entre un 1% y un 2%.

**Tabla 4. Errores de entrenamiento y de test utilizando C4.5 para el criterio del tiempo medio en el sistema**

Número de ejemplos de entrenamiento	Tamaño del árbol	Número de reglas	Error de reescritura en el árbol	Error de reescritura en las reglas	Error de test en el árbol	Error de test en las reglas
50	11	6	4%	4%	22%	22%
100	15	8	1%	1%	12%	13%
150	17	8	2.7%	2.7%	11%	11%
200	19	9	2.5%	2.5%	12%	12%
250	21	11	2.8%	2.8%	13%	13%
300	21	11	3.3%	3.3%	13%	13%
350	33	11	1.7%	2.6%	9%	9%
400	31	13	2.2%	2.2%	7%	7%
450	35	16	1.3%	1.6%	4%	4%
500	41	20	1%	1%	3%	3%
550	39	17	1.1%	0.9%	3%	3%
600	43	18	0.8%	0.8%	3%	3%
650	45	21	0.8%	0.8%	2%	2%
700	47	21	0.6%	0.6%	2%	2%
750	51	20	0.5%	0.8%	2%	2%
800	49	20	0.5%	0.5%	2%	2%
850	49	20	0.5%	0.5%	2%	2%
900	47	22	0.4%	0.4%	2%	2%
950	47	20	0.4%	0.4%	1%	1%
1000	45	21	0.5%	0.5%	1%	1%

### 6.3. Generación de nuevos atributos de control

En ocasiones, para elegir las mejores reglas de secuenciación, es necesario chequear relaciones del tipo: la utilización de la máquina 1 es menor que la de la máquina 2. Sin embargo, como se puede observar del apartado anterior, con los atributos originales, estas relaciones no se pueden obtener mediante el algoritmo de aprendizaje inductivo C4.5. Para poder lograr estas relaciones, sería necesario definir combinaciones aritméticas de los atributos originales. Pero en muchos casos, 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.

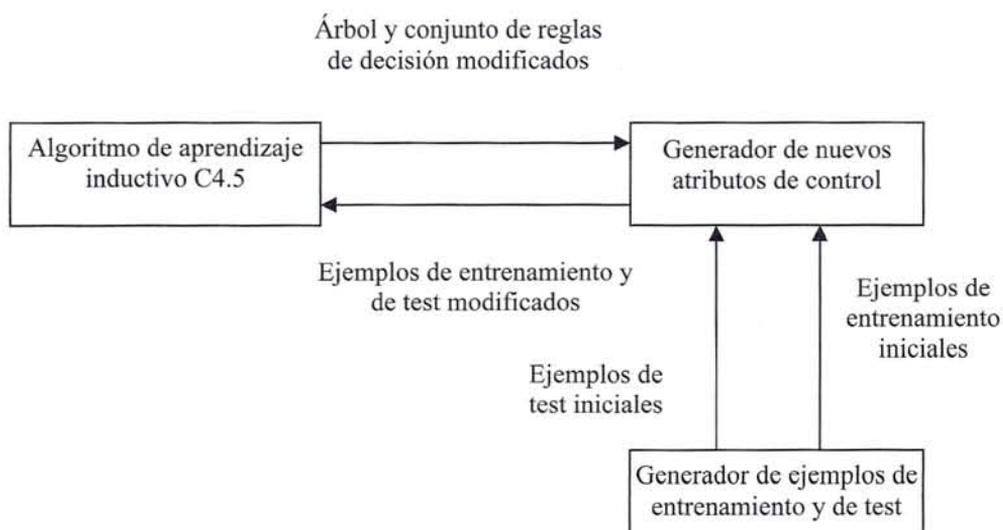
Por ello, se diseña un módulo que determina, de modo automático, las combinaciones “útiles” de los atributos originales, usando los datos de las simulaciones realizadas para obtener los ejemplos de entrenamiento y de test. Este módulo, junto con el algoritmo C4.5, genera las combinaciones de los atributos de control más adecuadas para el sistema de fabricación utilizado. Los operadores matemáticos básicos considerados son: la suma, la resta, la multiplicación y la división. Aunque se pueden emplear más operadores, por otra parte más complejos, se ha optado por los cuatro mencionados anteriormente por dos razones:

1. A medida que aumenta el número de operadores utilizados, se incrementa el tiempo de búsqueda necesario.
2. Como el procedimiento es recursivo, se pueden generar atributos bastante complejos a partir de los operadores básicos.

En la figura 5, se representa un esquema que muestra el funcionamiento del módulo propuesto. A continuación, se detalla un pseudocódigo del generador de nuevos atributos de control:

1. Determinación de las combinaciones de los atributos actuales.
2. Generación de los nuevos ejemplos de entrenamiento y de test en función de las combinaciones anteriores.
3. Identificación de las combinaciones “útiles”. Éstas, aparecen en el árbol y en el conjunto de reglas de decisión generadas por C4.5.
4. Si el nuevo árbol y/o el conjunto de reglas de decisión poseen menores errores de clasificación, regresar al primer paso. En caso contrario, finalizar el algoritmo.

**Figura 5. Módulo generador de nuevos atributos de control**



En realidad, en el cuarto paso, se puede decidir continuar con el algoritmo ya que se pueden lograr menores errores de clasificación en la siguiente, o subsiguientes iteraciones, aunque en la actual no se haya conseguido mejorar el error. Aplicando el módulo propuesto, se obtienen las siguientes combinaciones “útiles” de los atributos de control para el criterio del retraso medio: U1+U2, U1+U4, U2+U3, U1-U2, U2-U4, U3-U4 y U3/U4.

En la tabla 5, se presentan los resultados obtenidos, para el criterio del retraso medio, cuando se aplica el módulo generador de nuevos atributos de control. De los resultados de la tabla, se puede observar que a partir de 700 ejemplos de entrenamiento, el error de test en las reglas oscila entre un 16% y un 13%, siendo con 900 y 1000 ejemplos cuando se logra el menor error de test. Si se comparan estos resultados con los de la tabla 3, se observa que se produce una mejora considerable.

Posteriormente, se aplica el módulo propuesto para el criterio del tiempo medio en el sistema y se determinan, como “útiles”, las siguientes combinaciones de atributos: U1-U2, U3-U4, U1/U2 y U2/U3. En la tabla 6, se muestran los resultados, para el criterio del tiempo medio en el sistema, cuando se usa el módulo generador de nuevos atributos de control. Se puede advertir que a partir de 700 ejemplos de entrenamiento, el error de test se reduce al 0%. Si se comparan estos resultados con los de la tabla 4, se puede notar que existe una mejora apreciable, sobre todo cuando el número de ejemplos de entrenamiento es pequeño.

#### 6.4 Secuenciación del FMS

El siguiente paso en este estudio es implementar en el modelo de simulación del FMS las reglas heurísticas obtenidas mediante el programa de aprendizaje inductivo C4.5 para seleccionar, en tiempo real, la combinación más apropiada de las reglas de secuenciación dependiendo del estado del FMS. Otro aspecto importante que se debe estudiar es la elección del período de supervisión, ya que la frecuencia que se utiliza para chequear los atributos de control, para decidir si se cambian, o no, las reglas de secuenciación, determina el comportamiento del sistema de fabricación. Para ello, se toman múltiplos del tiempo medio de procesamiento total de un trabajo (ver por ejemplo, Jeong y Kim, 1998; Kim y Kim, 1994; Wu y Wysk, 1989). En este caso, se emplean 2.5, 5, 10 y 20 unidades de tiempo.

**Tabla 5. Errores de entrenamiento y de test utilizando C4.5 y el módulo generador de nuevos atributos de control para el criterio del retraso medio**

Número de ejemplos de entrenamiento	Tamaño del árbol	Número de reglas	Error de reescritura en el árbol	Error de reescritura en las reglas	Error de test en el árbol	Error de test en las reglas
50	15	6	14%	18%	21%	29%
100	27	10	9%	12%	27%	26%
150	33	12	10%	16%	25%	28%
200	39	16	9%	10.5%	16%	16%
250	51	20	8%	8.4%	18%	18%
300	63	23	5.3%	8%	15%	16%
350	65	24	5.4%	7.7%	14%	16%
400	73	26	6%	7.8%	13%	15%

450	89	31	4.2%	7.3%	18%	18%
500	89	37	5.2%	6.6%	16%	18%
550	93	36	5.5%	7.8%	14%	16%
600	101	32	5%	8.8%	14%	18%
650	115	32	4.6%	8.2%	17%	17%
700	111	35	6.1%	8.6%	14%	15%
750	123	40	5.7%	9.9%	12%	15%
800	105	36	6.8%	10.6%	12%	15%
850	105	38	6.9%	8.2%	14%	16%
900	113	37	6.4%	8.2%	13%	13%
950	117	38	6.1%	7.3%	14%	14%
1000	119	38	6%	8.1%	9%	13%

**Tabla 6. Errores de entrenamiento y de test utilizando C4.5 y el módulo generador de nuevos atributos de control para el criterio del tiempo medio en el sistema**

Número de ejemplos de entrenamiento	Tamaño del árbol	Número de reglas	Error de reescritura en el árbol	Error de reescritura en las reglas	Error de test en el árbol	Error de test en las reglas
50	11	6	2%	2%	9%	9%
100	9	5	1%	1%	1%	1%
150	11	6	2%	2%	1%	1%
200	15	8	3%	1.5%	4%	6%
250	21	10	0.4%	0.4%	3%	3%
300	23	11	0.3%	0.3%	2%	2%
350	23	11	0.3%	0.3%	2%	2%
400	15	7	1.5%	1.5%	2%	2%
450	17	7	1.6%	1.8%	1%	1%
500	19	10	1.4%	1.4%	1%	2%
550	25	8	0.9%	1.5%	0%	1%
600	27	12	1%	1%	0%	1%
650	23	12	1.2%	0.6%	1%	1%
700	31	11	0.6%	0.6%	0%	0%
750	35	12	0.4%	0.4%	0%	0%
800	35	13	0.5%	0.4%	0%	0%
850	35	13	0.5%	0.4%	0%	0%
900	35	14	0.4%	0.3%	0%	0%
950	35	14	0.4%	0.3%	0%	0%
1000	35	14	0.5%	0.4%	0%	0%

Asimismo, se proponen dos escenarios diferentes en el sistema de fabricación. En el primero, se generan cambios en el FMS, en determinados períodos de tiempo, en función de una variable aleatoria que varía uniformemente entre 50 y 500 unidades de tiempo. En el segundo escenario, esta variable fluctúa entre 2.5 y 250 unidades de tiempo. Por lo tanto, en este caso, el FMS sufrirá mayor número de cambios. Por último, señalar que se realizan cinco réplicas independientes durante 100000 unidades de tiempo y que, de acuerdo con los resultados anteriores, se opta por utilizar 900 ejemplos de entrenamiento en ambos criterios de comportamiento.

Un resumen de los resultados obtenidos se muestra en la tabla 7. Los valores de retraso medio y de tiempo medio, que aparecen en ésta, corresponden a la media de las cinco réplicas. Asimismo, para aumentar la legibilidad, los valores que se muestran son relativos al menor retraso medio y tiempo medio obtenido (a éstos, se les asigna el valor de uno). El período de supervisión elegido fue de 2.5 unidades de tiempo. En la tabla 7, se puede observar que la metodología de secuenciación basada en el algoritmo de aprendizaje inductivo C4.5 obtiene los mejores resultados. Dentro de los métodos que emplean una combinación de las reglas de secuenciación constante, destacan MDD+NINQ y MDD+WINQ. Sin embargo, los valores de retraso medio son superiores a los de la alternativa que utiliza C4.5, en unos porcentajes que varían entre un 10.63% y un 11.92%.

**Tabla 7. Retraso medio y tiempo medio en el sistema para los métodos propuestos**

Método utilizado	Retraso medio en el primer escenario	Retraso medio en el segundo escenario	Tiempo medio en el sistema para el primer escenario	Tiempo medio en el sistema para el segundo escenario
SPT+SPT	4.0156	5.3141	2.1113	2.4128
SPT+NINQ	1.1818	1.1825	1.0401	1.0476
SPT+WINQ	1.1754	1.1622	1.0427	1.0452
SPT+LUS	2.4500	2.5082	1.5129	1.5242
EDD+SPT	3.4398	4.5681	2.2084	2.6200
EDD+NINQ	1.4926	1.6259	1.3292	1.3959
EDD+WINQ	1.4874	1.6339	1.3289	1.3998
EDD+LUS	2.8121	3.1892	1.8670	2.0567
MDD+SPT	3.4514	4.6080	2.3021	2.6846
MDD+NINQ	1.1063	1.1080	1.2292	1.2540
MDD+WINQ	1.1136	1.1192	1.2330	1.2591
MDD+LUS	2.3330	2.4507	1.7756	1.8603
SRPT+SPT	4.4035	5.9465	2.2828	2.6645
SRPT+NINQ	1.3464	1.3716	1.1333	1.1451
SRPT+WINQ	1.3500	1.3633	1.1389	1.1460
SRPT+LUS	2.7709	2.9269	1.6728	1.7156
C4.5	1.0000	1.0000	1.0000	1.0000

Del mismo modo, para el criterio del tiempo medio en el sistema, la metodología de secuenciación basada en C4.5 logra los mejores resultados. Por otro lado, se puede observar que las combinaciones de reglas SPT+NINQ y SPT+WINQ son las que presentan menor tiempo medio en el sistema, dentro de los métodos que emplean una combinación de las reglas de modo constante. Sin embargo, los valores de tiempo medio en el sistema son mayores que los de la alternativa que utiliza C4.5, en unos porcentajes que fluctúan entre un 4.01% y un 4.76%. Por último, se compara la metodología de secuenciación basada en C4.5 con el resto de los métodos, empleando el análisis de la varianza. Se concluye que esta metodología de secuenciación es superior al resto con un nivel de significación menor que 0.05.

## 7. CONCLUSIONES

En este trabajo se describe, en primer lugar, el aprendizaje inductivo y una clasificación de las aplicaciones en las que se usa esta técnica; asimismo, se presenta el algoritmo de aprendizaje inductivo C4.5 y se enumeran sus principales ventajas e inconvenientes. A continuación, como ejemplo de aplicación, se ha propuesto una nueva metodología para secuenciar tareas empleando aprendizaje inductivo y se ha diseñado un módulo para obtener nuevos atributos de control a partir de los originales. De este modo, aplicando este módulo, se obtienen unos errores de clasificación menores y, por lo tanto, un “conocimiento de secuenciación” más fiable. Uno de los inconvenientes de emplear esta metodología es la necesidad de ejecutar un elevado número de simulaciones para generar los ejemplos de entrenamiento y de test; sin embargo, estas simulaciones se deben realizar sólo una vez. Posteriormente, se compara el comportamiento del FMS que utiliza la metodología basada en aprendizaje inductivo con la alternativa de emplear una combinación de reglas de modo constante, y se comprueba que la metodología propuesta obtiene menores valores de retraso medio y de tiempo medio en el sistema en el FMS utilizado.

## BIBLIOGRAFÍA

- AYTUG, H., BHATTACHARYYA, S., KOEHLER, G.J. y SNOWDON, J.L. (1994): “A review of machine learning in scheduling”, *IEEE Transactions on Engineering Management*, 41, 2, 165-171.
- BAKER, K.R. (1984): “Sequencing rules and due-date assignments in a job shop”, *Management Science*, 30, 9, 1093-1103.
- BREIMAN, L., FRIEDMAN, J.H., OLSHEN, R.A. y STONE, C.J. (1984): *Classification and Regression Trees*. Belmont, CA: Wadsworth.
- BUNTINE, W. (1991): “Learning classification trees”, Technical Report FIA-90-12-19-01, NASA Ames Research Center, Moffett Field, CA.
- CHOI, R.H. y 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. y BOE, W.J. (1987): “Experimental investigation of flexible manufacturing system scheduling rules”, *International Journal of Production Research*, 25, 7, 979-994.
- EGBELU, P.J. y TANCHOCO, J.M.A. (1984): “Characterization of automated guided vehicle dispatching rules”, *International Journal of Production Research*, 22, 3, 359-374.
- FEIGENBAUM, E.A., MCCORDUCK, P. y NII, H.P. (1988): *The Rise of the Expert Company*. New York: Times Books.
- FISHER, D.H., PAZZANI, M.J. y LANGLEY, P. (1991): *Concept Formation: Knowledge and Experience in Unsupervised Learning*. San Mateo, CA: Morgan Kaufmann Publishers.
- FRIEDMAN, J.H. (1988): “Multivariate adaptive regression splines”, Technical Report 102, Laboratory for Computational Statistics, Stanford University, Stanford, CA.
- HENNEKE, M.J. y CHOI, R.H. (1990): “Evaluation of FMS parameters on overall system performance”, *Computer Industrial Engineering*, 18, 1, 105-110.
- HUNT, E.B., MARIN, J. y STONE, P. J. (1966): *Experiments in Induction*. New York: Academic Press.
- ISHII, N. y TALAVAGE, J. (1991): “A transient-based real-time scheduling algorithm in FMS”, *International Journal of Production Research*, 29, 12, 2501-2520.
- JEONG, K.-C. y 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, C.-O., MIN, H.-S. y YIH, Y. (1998): “Integration of inductive learning and neural networks for multi-objective FMS scheduling”, *International Journal of Production Research*, 36, 9, 2497-2509.
- KIM, M.H. y KIM, Y.-D. (1994): “Simulation-based real-time scheduling in a flexible manufacturing system”, *Journal of Manufacturing Systems*, 13, 2, 85-93.
- LANGLEY, P. y SIMON, H.A. (1995): “Applications of machine learning and rule induction”, *Communications of the ACM*, 38, 11, 55-64.
- MICHALSKI, R.S., CARBONELL, J.G. y MITCHELL, T.M. (1983): *Machine Learning. An Artificial Intelligence Approach*. Palo Alto, CA: Tioga Press.

- MICHIE, D. (1987): "Current developments in expert systems", in J.R. Quinlan (eds.), *Applications of Expert Systems* (pp. 137-156). Wokingham, UK: Addison-Wesley.
- MICHIE, D. (1989): "Problems of computer-aided concept formation", in J.R. Quinlan (eds.), *Applications of Expert Systems* (Vol. 2) (pp. 310-333). Wokingham, UK: Addison-Wesley.
- MIN, H.-S., YIH, Y. 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. y WASSENHOVE, L.N.V. (1990): "Analysis of scheduling rules for an FMS", *International Journal of Production Research*, 28, 4, 785-802.
- NAKASUKA, S. y YOSHIDA, T. (1992): "Dynamic scheduling system utilizing machine learning as a knowledge acquisition tool", *International Journal of Production Research*, 30, 2, 411-431.
- O'KEEFE, R.M. y KASIRAJAN, T. (1992): "Interaction between dispatching and next station selection rules in a dedicated flexible manufacturing system", *International Journal of Production Research*, 30, 8, 1753-1772.
- PRIORE, P., DE LA FUENTE, D., GÓMEZ, A. y PUENTE, J. (2001): "A review of machine learning in dynamic scheduling of flexible manufacturing systems", *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 15, 3, 251-263.
- QUINLAN, J.R. (1979): "Discovering rules by induction from large collections of examples", in D. Michie (eds.), *Expert Systems in the Micro Electronic Age*. Edinburgh, UK: Edinburgh University Press.
- QUINLAN, J.R. (1983): "Learning efficient classification procedures", in R.S. Michalski, J.G. Carbonell and T.M. Mitchell (eds.), *Machine Learning: An Artificial Intelligence Approach*. Palo Alto, CA: Tioga Press.
- 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 Publishers, 1991. Reprinted in B.G. Buchanan and D. Wilkins (eds.), *Readings in Knowledge Acquisition and Learning*. San Mateo, CA: Morgan Kaufmann Publishers, 1992.
- QUINLAN, J.R. (1988): "Decision trees and multi-valued attributes", in J.E. Hayes, D. Michie and J. Richards (eds.), *Machine Intelligence* (pp. 305-318). Oxford, UK: Oxford University Press.
- QUINLAN, J.R. (1993): *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann.
- SHAW, M.J., PARK, S. y RAMAN, N. (1992): "Intelligent scheduling with machine learning capabilities: the induction of scheduling knowledge", *IIE Transactions*, 24, 2, 156-168.
- STECKE, K.E. y SOLBERG, J. (1981): "Loading and control policies for a flexible manufacturing system", *International Journal of Production Research*, 19, 5, 481-490.
- TANG, L.-L., YIH, Y. y LIU, C.-Y. (1993): "A study on decision rules of a scheduling model in an FMS", *Computer in Industry*, 22, 1-13.
- WHISTON, P.H. (1994): *Inteligencia Artificial*. Wilmington, Delaware: Addison-Wesley Iberoamericana.
- WITNESS (1996): *User Manual. Release 8.0*. AT&T ISTEEL Limited.
- WU, S.-Y.D. y 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.

La Revista *Investigaciones Europeas de Dirección y Economía de la Empresa* recibió este artículo el 28 de febrero de 2003 y fue aceptado para su publicación el 5 de febrero de 2004.