

UTILIZACIÓN DEL RAZONAMIENTO BASADO EN CASOS EN LA TOMA DE DECISIONES. APLICACIÓN EN UN PROBLEMA DE SECUENCIACIÓN

PAOLO PRIORE
DAVID DE LA FUENTE
RAÚL PINO
JAVIER PUENTE

DPTO. DE ADMINISTRACIÓN DE EMPRESAS Y CONTABILIDAD.
UNIVERSIDAD DE OVIEDO

Resumen: *El razonamiento basado en casos (CBR, de Case-Based Reasoning), 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 CBR, 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 CBR. 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 CBR que permite usar, en cada momento, la regla de secuenciación más adecuada.*

Palabras clave: *Secuenciación, CBR, FMS, Simulación*

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 CBR. 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, Montazeri y Van Wassenhove, 1990; 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.

Sin embargo, a pesar de haber obtenido muy buenos resultados con el CBR en otros campos de aplicación y de ser muy sencillo en cuanto a su concepción, apenas se ha aplicado para diseñar sistemas de secuenciación que modifican de forma dinámica las reglas de secuenciación. Por lo tanto, en

este trabajo, se propone una metodología de secuenciación para FMSs que utiliza CBR.

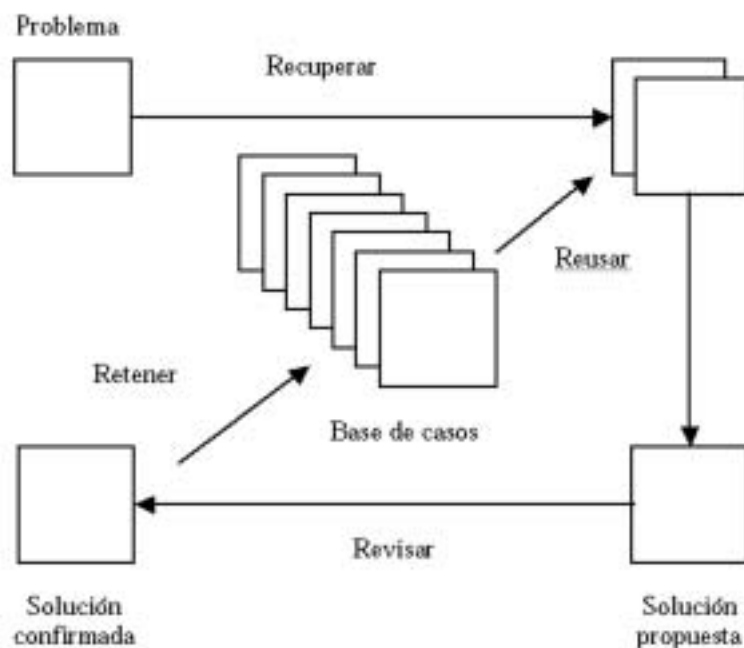
Este trabajo está organizado de la siguiente forma. En primer lugar, se describe el CBR y una clasificación de las aplicaciones en las que se usa esta técnica; asimismo, se enumeran las principales ventajas e inconvenientes del CBR. A continuación, se presenta el algoritmo de los vecinos más próximos ya que es uno de los algoritmos de CBR más utilizados. Posteriormente, se presenta una metodología para secuenciar trabajos utilizando CBR y se efectúa el estudio experimental. Finalmente, se describe un resumen de los resultados del trabajo.

2. Razonamiento basado en casos

En la figura 1, se muestra el esquema general de un sistema de CBR que consta de cuatro etapas fundamentales (Watson, 1997):

1. Recuperar el caso (o casos) más similar al problema planteado.
2. Reusar el caso (o casos) para intentar resolver el problema.
3. Revisar la solución propuesta.
4. Retener la nueva solución para que forme parte de la base de casos.

Figura 1
Esquema general de un sistema de CBR



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 el CBR (o cualquier otra técnica 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.

Las tareas de clasificación y de síntesis son los dos

Tabla 1
Tabla atributo-valor de los ejemplos de entrenamiento

Ejemplos	Atributo 1	Atributo 2	Atributo m	Clase
Ejemplo 1	A11	A12	A1m	C1
Ejemplo 2	A21	A22	A2m	C2
.....
Ejemplo n	An1	An2	Anm	Cn

tipos fundamentales de aplicaciones en las que se utiliza CBR (Althoff et al., 1995). Las tareas de clasificación cubren un rango amplio de aplicaciones entre las que destacan:

1. Previsión. Por ejemplo, la predicción de un fallo en un equipo o el comportamiento de la bolsa de valores.
2. Planificación. La reutilización de la planificación de un viaje o de la programación de trabajos.
3. Valoración. Por ejemplo, el análisis de riesgos en el sector bancario o de seguros, o la estimación de los costes de un proyecto.
4. Diagnóstico. La realización de diagnósticos médicos o de fallos de un equipo.
5. Control de procesos. Por ejemplo, el control de un sistema de fabricación flexible.

En las tareas de clasificación, se debe determinar aquellos casos de la base que más se parezcan al nuevo caso que se pretende clasificar y reutilizar directamente sus soluciones sin efectuar ningún tipo de modificación. Sin embargo, en las tareas de síntesis, se intenta crear una nueva solución mediante la combinación de soluciones previas almacenadas en la base de casos. Las tareas de síntesis son inherentemente complejas debido a las restricciones entre los elementos de las soluciones que se utilizan. Dentro de las tareas de síntesis destacan:

1. Diseño. La creación de una nueva máquina mediante la adaptación de elementos de otras máquinas.
2. Planificación. La creación de nuevos programas de trabajo a través de elementos de otros programas ya utilizados.

Entre las innumerables ventajas que posee el CBR, se señalan las siguientes:

1. No es necesario realizar ningún tipo de entrenamiento como en otros algoritmos, con el consiguiente ahorro de 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. Posee una gran eficacia de clasificación, a pesar de su sencillez. Así, Rachlin et al. (1994) realizan una serie de experimentos sobre los ejemplos de la "UCI repository" (Murphy y Aha, 1994). Los resultados demuestran una eficacia similar a otros algoritmos de aprendizaje automático.

Los principales problemas que surgen cuando se aplica el CBR son los siguientes:

1. En algunas aplicaciones es difícil medir la similitud entre los casos. Si todos los atributos son continuos, se puede calcular la distancia entre dos casos como la raíz cuadrada de la suma de los cuadrados de las diferencias de los atributos; asimismo, se deben escalar los atributos, de modo que su contribución sea comparable. Sin embargo, cuando existen atributos no ordinales, la interpretación de la distancia se vuelve más problemática. Además, si hay muchos atributos irrelevantes, dos casos similares pueden parecer bastante distintos ya que tienen valores diferentes en atributos poco importantes (Stanfill y Waltz, 1986).
2. El tiempo requerido para realizar la clasificación de un nuevo caso es mucho mayor que en otros algoritmos, sobre todo si el número de casos almacenados es muy grande.

3. Existe cierta dificultad para saber qué casos de entrenamiento se deben recordar, ya que si se guardan todos los casos, el clasificador puede volverse lento y difícil de manejar. Por lo tanto, lo ideal es retener casos prototipo que, juntos, resuman toda la información importante (Aha et al., 1991)

3. El algoritmo de los vecinos más próximos

Uno de los métodos más populares y utilizados dentro del CBR es el algoritmo de los vecinos más próximos. La formulación de éste, denominado NN (Nearest Neighbor), o k-NN en la versión más sofisticada, es relativamente simple. Se parte de una métrica d entre ejemplos y un conjunto de casos de entrenamiento, E . A partir de este par, (E, d) , cuando se presenta un nuevo caso x , del cual no se sabe su clase, se determina ésta de la siguiente manera:

$$\text{Clase}(x) = \text{Clase}(\text{vmp})$$

donde:

$$\begin{aligned} \text{vmp: vecino más próximo} \\ d(x, \text{vmp}) = \text{Mínimo} \{d(x, e) : e \in E\} \end{aligned}$$

Dicho de otra forma, la clase del caso es la del ejemplo o vecino más próximo (vmp). Este esquema inicial se puede sofisticar, introduciendo un valor entero k ($k \geq 1$), de tal forma que se calculen, para x , los k vecinos más próximos en E . Entonces, se decide la clase de x en función de la clase mayoritaria en los k vecinos. Para calcular la distancia ($d(x, e)$) entre un nuevo caso (x) y un ejemplo de entrenamiento (e) se emplea la siguiente formulación:

$$d(x, e) = w_i \sqrt{\sum_{i=1}^n (a_{ix} - a_{ie})^2}$$

donde n es el número de atributos considerados; a_{ix} es el valor del atributo i en el caso x ; a_{ie} es el valor del atributo i en el ejemplo e y w_i es el peso asignado al atributo i en función de su importancia.

4. Secuenciación dinámica mediante CBR

La metodología que se propone en este trabajo, para secuenciar tareas utilizando CBR, se puede dividir en tres etapas, tal como se muestra en la figura 2:

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. Los atributos seleccionados se denomi-

nan 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.

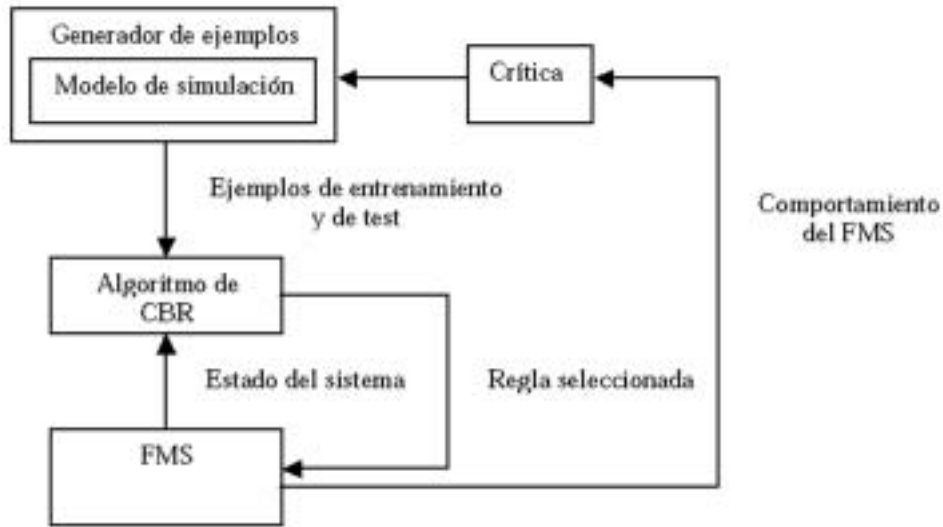
2. Determinación de la regla de secuenciación más adecuada (o combinación de ellas, si existe más de un tipo de decisión), mediante CBR, dependiendo de los valores que presenten los atributos (estado del sistema de fabricación) en cada momento.

3. Comparación del comportamiento del sistema de fabricación utilizando CBR 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.

5. 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 3, 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:

Figura 2
Secuenciación dinámica mediante CBR



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:

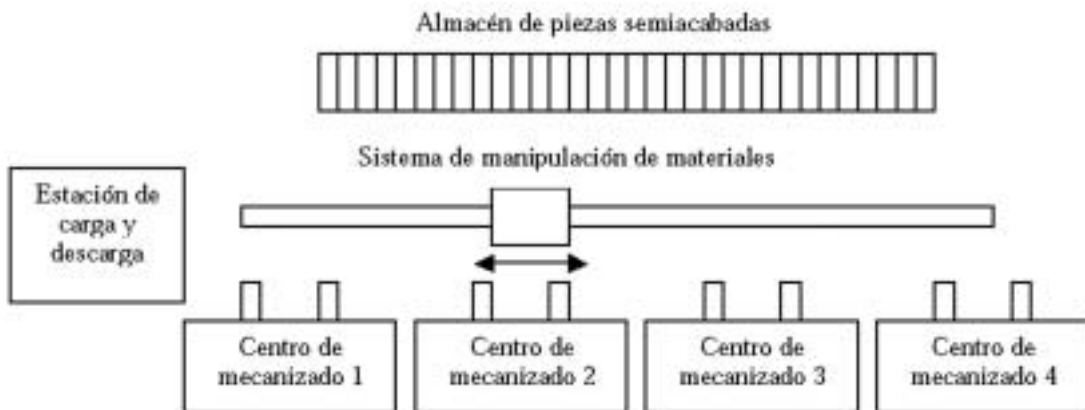
$$d_i = t_i + p_i * F$$

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 . 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.

Figura 3
Sistema de fabricación flexible propuesto



5.1. 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 utilizaciones de las máquinas del sistema y la utilización media. Para calcular estos dos últimos atributos se emplean las siguientes expresiones:

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

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

En total, se generan 1100 combinaciones diferentes de los atributos de control de forma aleatoria, utilizando 100 de ellas, como ejemplos de test. En cada combinación de atributos, se determinan los valores de retraso medio y tiempo medio en el sistema resultante 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 en los ejemplos generados

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%

5.2. Aplicación del razonamiento basado en casos.

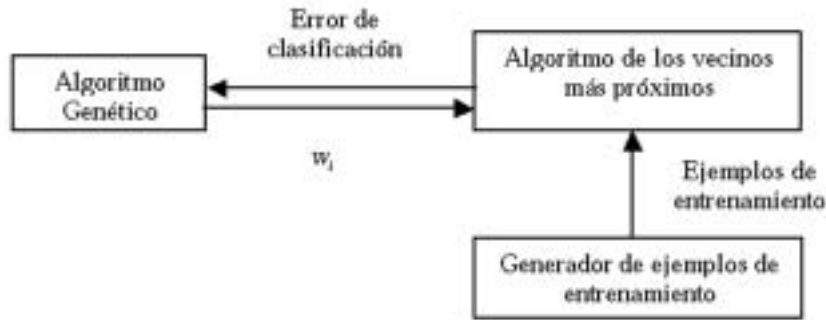
En este estudio se han empleado tres versiones del algoritmo de los vecinos más próximos. En la primera de ellas, se utiliza un valor de k igual a uno. En la segunda, se emplea un valor de k igual a tres, de tal forma que se calculan, para cada x , los tres vecinos más próximos. Entonces, se decide la clase de x en función de la clase mayoritaria en los tres vecinos. En la última versión, se utiliza un valor de k igual a cinco. En este caso, la clase de x se determina en función de la clase mayoritaria de los cinco vecinos más próximos. En el supuesto de que se produzca un empate (por ejemplo, si k es igual a cinco puede ocurrir que existan dos vecinos con la clase SPT+SPT y otros dos con la clase MDD+SPT) se determina la clase de x en función de las distancias obtenidas. Obviamente, se eligen aquellos vecinos que tengan las menores distancias para determinar la clase de x .

Uno de los mayores problemas que presenta el algoritmo de los vecinos más próximos es que su capacidad para clasificar nuevos casos depende de los pesos w_i elegidos. Cada uno de estos pesos se asigna en función de la importancia del atributo correspondiente. Sin embargo, el valor de cada w_i no se conoce a priori. Para solventar este problema, se diseña un algoritmo genético (Goldberg, 1989; Holland, 1975; Michalewicz, 1996) que permite determinar los valores óptimos de w_i , de tal forma que el error de clasificación sea el menor posible. Un esquema del sistema propuesto se muestra en la figura 4. En él, se puede observar que el algoritmo genético utiliza el método de los vecinos más próximos para calcular el error de clasificación para unos valores w_i dados. Este error es el "fitness" para

un conjunto dado de pesos w_i . Para determinar el error de clasificación, se utiliza un conjunto de 100 ejemplos de validación que se toman del conjunto de entrenamiento. Después de un número determinado de generaciones, el algoritmo genético identificará los valores de w_i óptimos. A continuación, se detalla un pseudocódigo del algoritmo genético propuesto:

1. Generación de una población de N individuos.
2. Selección del mejor individuo.
3. Cálculo de una distribución de probabilidad acumulada en función del "fitness" de cada individuo de la población.
4. Selección de N individuos generando N números aleatorios entre cero y uno, empleando la distribución de probabilidad calculada anteriormente.
5. Cruce simple de los individuos seleccionados en el cuarto paso, con una probabilidad de cruce (PC) predeterminada. Los progenitores se eliminan.
6. Mutación con una probabilidad de mutación (PM) predeterminada.
7. Selección del mejor individuo. Si tiene un "fitness" inferior al mejor individuo de la anterior generación, se guarda éste y se elimina el peor individuo de la actual generación.
8. Regresar al tercer paso si el número de generaciones es menor al número de generaciones máximo (NG_{max}).

Figura 4
Método de cálculo de los pesos w_i óptimos



Un paso previo a la aplicación del método propuesto, sobre los distintos conjuntos de ejemplos de entrenamiento, es la normalización de los atributos de forma que su valor máximo y mínimo sean, respectivamente, uno y cero. Cada uno de los N individuos de la población está formado por un vector del siguiente tipo:

$$(W_1, W_2, W_3, \dots, W_9, W_{10})$$

siendo w_i el peso del atributo i . Estos pesos, al igual que los atributos, tienen un valor comprendido entre cero y uno. Asimismo, de acuerdo con el apartado anterior, el número de atributos considerado es diez. Por otro lado, se estudiaron los valores más apropiados del tamaño de la población (N), la probabilidad de cruce (PC) y mutación (PM) y el número de generaciones máximo (NG_{max}). Los valores determinados son $N=50$, $PC=0.7$, $PM=0.025$ y

$NG_{max}=100$. En la figura 5, se muestra cómo funcionan los operadores de cruce y de mutación del algoritmo genético.

En la tabla 3, se presenta un resumen de los resultados obtenidos con el algoritmo de los vecinos más próximos y el algoritmo genético (para determinar los pesos w_i óptimos), cuando el criterio empleado es el retraso medio. En general, se puede observar que a medida que aumenta el número de ejemplos de entrenamiento utilizado, el error en los ejemplos de test (ejemplos que no han sido tratados anteriormente) disminuye de forma considerable. Además, el error también se reduce si se utiliza un valor de k mayor que uno. Sin embargo, a medida que aumenta el número de ejemplos de entrenamiento, se incrementa el tiempo necesario para que el algoritmo genético determine los pesos óptimos w_i que se deben utilizar. De los re-

Figura 5
Operadores de cruce y de mutación

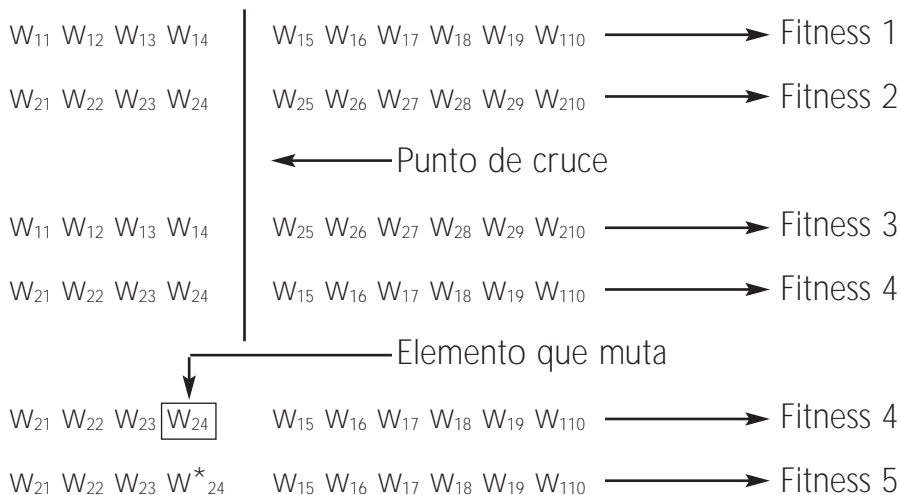


Tabla 3
Errores de test utilizando CBR para el criterio del retraso medio

Número de ejemplos de entrenamiento	Error de test (k=1)	Error de test (k=3)	Error de test (k=5)	Número de ejemplos de entrenamiento	Error de test (k=1)	Error de test (k=3)	Error de test (k=5)
150	17%	19%	22%	600	15%	13%	12%
200	17%	18%	18%	650	16%	13%	12%
250	17%	18%	18%	700	15%	13%	12%
300	16%	15%	16%	750	17%	13%	11%
350	14%	15%	17%	800	18%	14%	12%
400	14%	14%	13%	850	17%	14%	12%
450	14%	14%	13%	900	16%	12%	11%
500	15%	16%	16%	950	17%	12%	12%
550	14%	14%	13%	1000	16%	11%	11%

Tabla 4
Errores de test utilizando CBR para el criterio del tiempo medio en el sistema

Número de ejemplos de entrenamiento	Error de test (k=1)	Error de test (k=3)	Error de test (k=5)	Número de ejemplos de entrenamiento	Error de test (k=1)	Error de test (k=3)	Error de test (k=5)
150	8%	10%	9%	600	5%	3%	3%
200	7%	10%	10%	650	3%	2%	2%
250	6%	8%	9%	700	2%	2%	3%
300	5%	7%	7%	750	2%	3%	3%
350	5%	5%	6%	800	2%	2%	3%
400	5%	4%	4%	850	2%	2%	3%
450	5%	4%	4%	900	1%	2%	3%
500	5%	4%	4%	950	1%	2%	2%
550	5%	3%	4%	1000	1%	2%	3%

sultados presentados en la tabla 3, se puede observar que a partir de 600 ejemplos de entrenamiento, el error de test oscila entre un 11% y un 12% (k=5).

En la tabla 4, se muestran los resultados para el criterio del tiempo medio en el sistema. En general, se observa que a medida que se incrementa el número de ejemplos de entrenamiento utilizados, el error en los ejemplos de test decrece de forma considerable. Asimismo, señalar que los errores de la tabla 4 son 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).

5.3. Aplicación de C4.5

En la tabla 5, se presentan los resultados obtenidos cuando se aplica el algoritmo de aprendizaje inductivo C4.5 (Quinlan, 1993) a los mismos ejemplos de test y de entrenamiento. Este algoritmo, muy utilizado en el campo de aprendizaje automático, se ha aplicado en diversos sistemas de fabricación para secuenciar tareas (ver por ejemplo, Shaw et al., 1992). Este algoritmo genera un árbol de decisión y un conjunto de reglas que se utilizan para determinar la regla de secuenciación que se debe de emplear en cada momento. Se puede observar que los errores obtenidos con el CBR son menores a los logrados con el algoritmo de aprendizaje inductivo C4.5. Esto confirma la idoneidad del CBR para resolver problemas de secuenciación en FMSs.

Tabla 5
 Errores de test utilizando C4.5 para los criterios del retraso medio (RM)
 y del tiempo medio en el sistema (TMS)

Número de ejemplos de entrenamiento	Error de test en las reglas (RM)	Error de test en las reglas (TMS)	Número de ejemplos de entrenamiento	Error de test en las reglas (RM)	Error de test en las reglas (TMS)
150	24%	11%	600	17%	3%
200	21%	12%	650	17%	2%
250	19%	13%	700	18%	2%
300	18%	13%	750	16%	2%
350	17%	9%	800	19%	2%
400	17%	7%	850	19%	2%
450	21%	4%	900	18%	2%
500	22%	3%	950	19%	1%
550	21%	3%	1000	16%	1%

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 <= 4.8 AND U1 <= 65.7 AND U2 > 64.7
 THEN SPT+LUS [50.0%]
 Regla 2: IF F > 7 AND NMAO <= 1 AND FUMCB > 2.9 AND U2 <= 84
 THEN EDD+SPT [63.0%]

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

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.

5.4. Secuenciación del FMS

El siguiente paso en este estudio es implementar el algoritmo de los vecinos más próximos en el modelo de simulación del FMS para seleccionar, en tiempo real, la combinación más apropiada de las

reglas de secuenciación dependiendo del estado del FMS. Asimismo, también se implementan en el modelo de simulación las reglas heurísticas obtenidas mediante el programa de aprendizaje inductivo C4.5. 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. 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 1000 ejemplos de entrenamiento en ambos criterios de comportamiento.

Un resumen de los resultados obtenidos se muestra en la tabla 6. 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 6, se puede observar que el sistema de secuenciación basado en el algoritmo de los vecinos más próximos obtiene los mejores resultados. Con el algoritmo C4.5, se logran valores de retraso medio mayores a los obtenidos con el algoritmo de los vecinos más próximos. Dentro de las estrategias 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 CBR, en unos porcentajes que varían entre un 12.44% y un 13.99%.

Asimismo, para el criterio del tiempo medio en el sistema, los sistemas de secuenciación basados en el

Tabla 6
Retraso medio y tiempo medio en el sistema para las estrategias propuestas

Estrategia utilizada	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	4.0814	5.4122	2.1174	2.4147
SPT+NINQ	1.2013	1.2043	1.0431	1.0484
SPT+WINQ	1.1947	1.1837	1.0457	1.0460
SPT+LUS	2.4902	2.5545	1.5173	1.5254
EDD+SPT	3.4962	4.6524	2.2148	2.6221
EDD+NINQ	1.5171	1.6559	1.3330	1.3970
EDD+WINQ	1.5118	1.6640	1.3327	1.4009
EDD+LUS	2.8582	3.2481	1.8724	2.0583
MDD+SPT	3.5080	4.6931	2.3088	2.6868
MDD+NINQ	1.1244	1.1284	1.2328	1.2550
MDD+WINQ	1.1318	1.1399	1.2366	1.2601
MDD+LUS	2.3712	2.4960	1.7807	1.8618
SRPT+SPT	4.4757	6.0563	2.2894	2.6666
SRPT+NINQ	1.3685	1.3969	1.1366	1.1460
SRPT+WINQ	1.3721	1.3885	1.1422	1.1469
SRPT+LUS	2.8164	2.9809	1.6776	1.7169
k-NN	1.0000	1.0000	1.0008	1.0012
C4.5	1.0361	1.0393	1.0000	1.0000

algoritmo de los vecinos más próximos y en C4.5 logran resultados similares. 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 las estrategias 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 4.31% y un 4.84%.

Por último, se compara el sistema de secuenciación basado en el algoritmo de los vecinos más próximos con el resto de las estrategias, empleando el análisis

de la varianza. Se concluye que este sistema de secuenciación es superior al resto de estrategias con un nivel de significación menor que 0.05. La única excepción se produce entre los sistemas de secuenciación basados en el algoritmo k-NN y C4.5 para el criterio del tiempo medio en el sistema.

6. Conclusiones

En este trabajo se describe el CBR y una clasificación de las aplicaciones en las que se usa esta técnica; asimismo, se enumeran las principales ventajas e incon-

venientes del CBR. Posteriormente, se presenta el algoritmo de los vecinos más próximos ya que es uno de los algoritmos de CBR más utilizado. Asimismo, se ha propuesto una nueva metodología para secuenciar tareas empleando CBR y se ha diseñado un algoritmo genético para calcular los pesos w_i óptimos que necesita el algoritmo de los vecinos más próximos. 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 utilizando el sistema de secuenciación basado en el algoritmo de los vecinos más próximos con el resto de las estrategias y se comprueba que la metodología propuesta obtiene menores valores de retraso medio y de tiempo medio en el sistema en el FMS utilizado. La única excepción se produce entre los sistemas de secuenciación basados en el algoritmo k-NN y C4.5 para el criterio del tiempo medio en el sistema, debido a que los errores de test en ambos algoritmos son muy similares.

Bibliografía

- AHA, D.W.; KIBLER, D. y ALBERT, M.K. (1991): "Instance-based learning algorithms", *Machine Learning*, 6, 1, 37-66.
- ALTHOFF, K-D; AURIOL, E.; BARLETTA, R. y MANAGO, M. (1995): *A Review of Industrial Case-Based Reasoning Tools*. Oxford: AI Intelligence.
- 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.
- GOLDBERG, D.E. (1989): *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison Wesley.
- HOLLAND, J. (1975): *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan 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.
- MICHALEWICZ, Z. (1996): *Genetic Algorithms + Data Structures = Evolution Programs*. Berlin: Springer Verlag.
- MICHALSKI, R.S.; CARBONELL, J.G. y MITCHELL, T.M. (1983): *Machine Learning. An Artificial Intelligence Approach*. Palo Alto, CA: Tioga Press.
- 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.
- MURPHY, P. y AHA, D.W. (1994): *UCI Repository of Machine Learning Databases, a Machine-readable Data Repository*. Maintained at the Department of Information and Computer Science, University of California, Irvine. Anonymous ftp from ics.uci.edu in the directory pub/machine-learning-databases.
- 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* (pendiente de publicar).
- QUINLAN, J.R. (1993): *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann.
- RACHLIN, J.; KASIF, S.; SALZBERG, S. y AHA, D.W. (1994): "Towards a better understanding of memory-based reasoning systems", *Proceedings of the Eleventh International Conference* (pp. 242-250). New Brunswick, New Jersey.
- 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.
- STANFILL, C. y WALTZ, D. (1986): "Toward memory-based reasoning", *Communications of the ACM*, 29, 12, 1213-1228.
- 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.
- WATSON, I. (1997): *Applying Case-Based Reasoning: Techniques for Enterprise Systems*. San Francisco, CA: Morgan Kaufmann Publishers.

WITNESS (1996): User Manual. Release 8.0. AT&T ISTEEL Limited.
WU, S.-Y. D. y WYSK, R.A. (1989): "An application of discrete-

te-event simulation to on-line control and scheduling in flexible manufacturing", *International Journal of Production Research*, 27, 9, 1603-1623.