

Secuenciación de sistemas de fabricación flexible mediante aprendizaje automático

Paolo Priore, Javier Puente, Raúl Pino

Dpto. de Admón. de Empresas y Contabilidad. Escuela Politécnica Superior de Ingeniería de Gijón. Campus de Viesques, 33204 Gijón (Asturias). priore@uniovi.es, pedro.lopez@uniovi.es

Resumen

La forma habitual de secuenciar dinámicamente los trabajos en los sistemas de fabricación flexible (FMS) 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 automático que permite usar, en cada momento, la regla de secuenciación más adecuada. Mediante esta técnica, analizando el comportamiento previo del sistema, se obtiene un “conocimiento de secuenciación” que permite determinar la regla de secuenciación más apropiada en cada momento. En este trabajo, se emplearán y compararán tres tipos distintos de algoritmos de aprendizaje automático para obtener el “conocimiento de secuenciación”: aprendizaje inductivo, redes neuronales backpropagation y razonamiento basado en casos.

Palabras clave: Secuenciación, Aprendizaje Automático, FMS

1. Introducción

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. 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). En el segundo enfoque, se emplean un conjunto de simulaciones previas del sistema de fabricación (para obtener los ejemplos de entrenamiento y de test) para determinar, mediante un algoritmo de aprendizaje automático, cuál es la mejor de las reglas en cada posible estado del sistema (ver por ejemplo, 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 emplearán y compararán tres tipos distintos de algoritmos de aprendizaje automático para obtener el “conocimiento de secuenciación”: el algoritmo de aprendizaje inductivo *C4.5* (Quinlan, 1993), las redes neuronales

backpropagation (Rumelhart et al., 1986) y el algoritmo de los vecinos más próximos (Aha, 1992).

2. Secuenciación dinámica mediante aprendizaje automático

La metodología que se propone en este trabajo, para secuenciar tareas utilizando aprendizaje automático, se puede dividir en cuatro etapas (ver figura 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 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.
- Determinación del “conocimiento de secuenciación” mediante un algoritmo de aprendizaje automático.

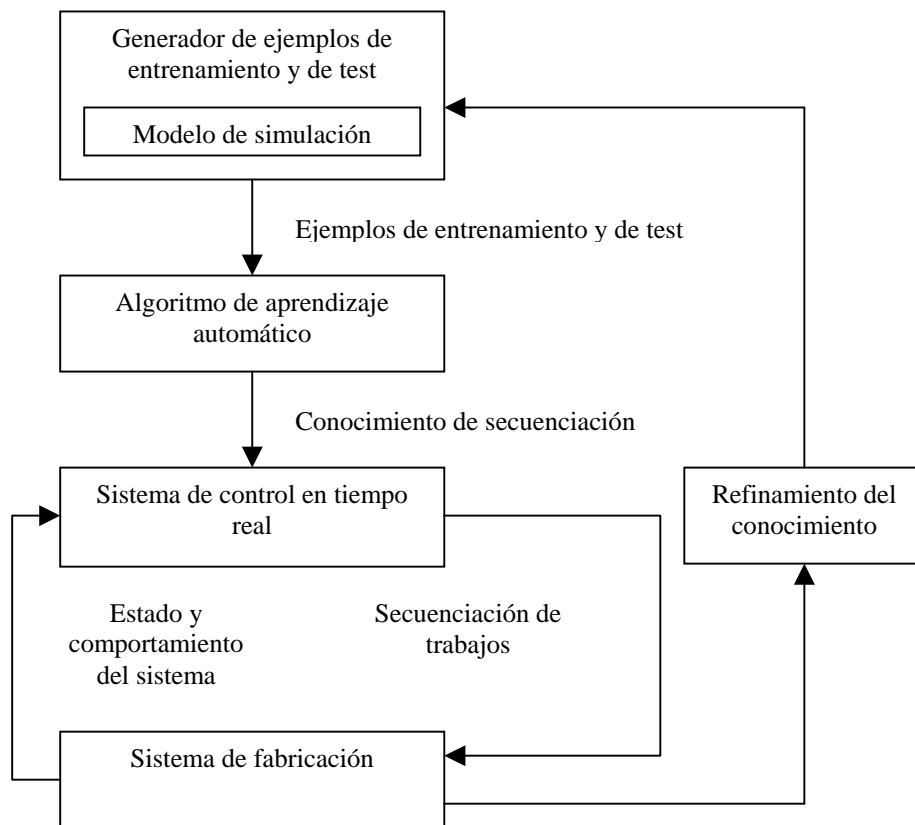


Figura 1. Esquema general de un sistema de secuenciación basado en el conocimiento

- 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.
- Comparación del comportamiento del sistema de fabricación utilizando aprendizaje automático 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.

3. Estudio experimental

El *FMS* considerado 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 2, 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).

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 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: *SPT* (Shortest Processing Time); *NINQ* (Number in Queue); *WINQ* (Work in Queue) y *LUS* (Lowest Utilized Station)

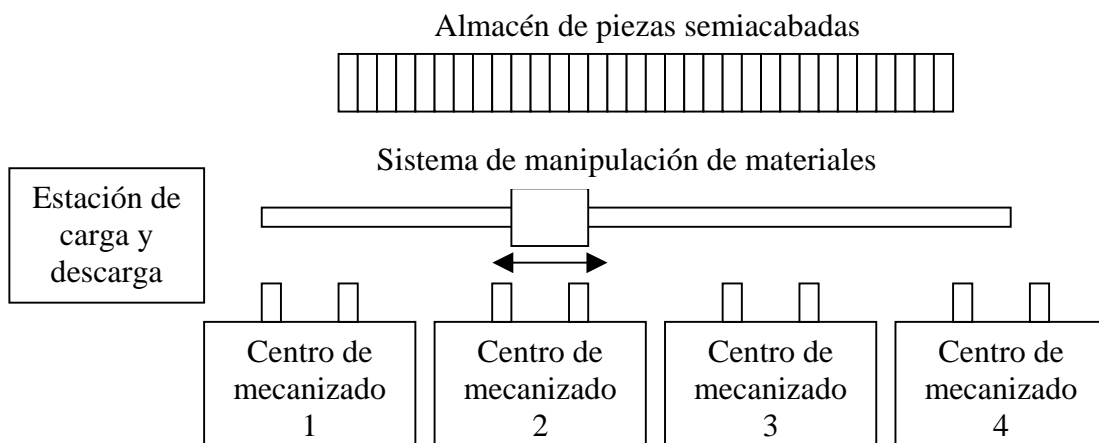


Figura 2. Sistema de fabricación flexible propuesto

3.1. 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:

- Los trabajos, o piezas, llegan al sistema siguiendo una distribución de Poisson.
- Los tiempos de procesamiento de cada operación se ajustan a una distribución exponencial con un valor medio de uno.
- El número de operaciones que se asigna a cada trabajo sigue una distribución uniforme entera, siendo los valores extremos del intervalo uno y cuatro.
- 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.
- El número de máquinas alternativas para realizar una operación varía entre uno y dos.
- El ritmo de llegada de los trabajos se modifica de forma que la utilización global del sistema oscile entre el 55% y el 95%.
- El valor del factor F (parámetro que determina la holgura de la fecha de entrega de un trabajo) 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. 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; Shaw et al., 1992): F ; $NMAO$, 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:

$$FDUM = \frac{10 * \sqrt{\frac{(U_1 - UM)^2 + (U_2 - UM)^2 + (U_3 - UM)^2 + (U_4 - UM)^2}{3}}}{UM} \quad (1)$$

$$FUMCB = \frac{10 * \max(U_1, U_2, U_3, U_4)}{UM} \quad (2)$$

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 1, 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 1. Comparación de las combinaciones de las reglas 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
<i>SPT+SPT</i>	23.27%	50%	<i>MDD+SPT</i>	25.64%	0%
<i>SPT+NINQ</i>	4.64%	22.45%	<i>MDD+NINQ</i>	11.27%	0%
<i>SPT+WINQ</i>	6.09%	25%	<i>MDD+WINQ</i>	3.82%	0.64%
<i>SPT+LUS</i>	0.36%	0%	<i>MDD+LUS</i>	1.27%	0%
<i>EDD+SPT</i>	1.09%	0%	<i>SRPT+SPT</i>	0%	0%
<i>EDD+NINQ</i>	8.73%	0%	<i>SRPT+NINQ</i>	0%	0%
<i>EDD+WINQ</i>	4.73%	0%	<i>SRPT+WINQ</i>	0%	1.91%
<i>EDD+LUS</i>	9.09%	0%	<i>SRPT+LUS</i>	0%	0%

3.2. Comparación de los algoritmos de aprendizaje automático

En la figura 3, se muestran los errores de test alcanzados, con cada uno de los algoritmos de aprendizaje automático, para el criterio del retraso medio. Se puede observar que el algoritmo de los vecinos más próximos (*k-NN*) obtiene los errores de test más bajos (9%). Asimismo, la fluctuación en los errores, a partir de 600 ejemplos de entrenamiento, es mínima. La red neuronal “backpropagation” (*BPN*) ocupa el segundo lugar con un error de test del 12%, siendo el algoritmo de aprendizaje inductivo *C4.5* el que genera mayores errores de test (16%). Los pesos más apropiados de los atributos en el algoritmo de los vecinos más próximos se determinan mediante un algoritmo genético (Goldberg, 1989).

Análogamente, en la figura 4, se presenta un resumen de los resultados obtenidos, con cada uno de los algoritmos de aprendizaje automático, para el criterio del tiempo medio en el sistema. Se observa que el algoritmo de los vecinos más próximos obtiene un error de test nulo. El algoritmo de aprendizaje inductivo *C4.5* ocupa el segundo lugar con un error de test del 1%, siendo la red neuronal “backpropagation” la que genera mayores errores de test (4%). Asimismo, los errores son también bastante inferiores a los mostrados en la figura 2; 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 I).

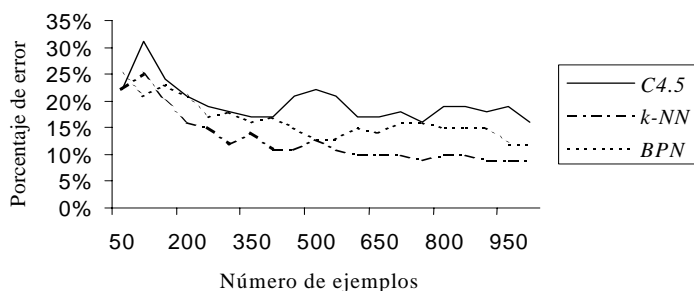


Figura 3. Comparación de los algoritmos de aprendizaje para el criterio del retraso medio

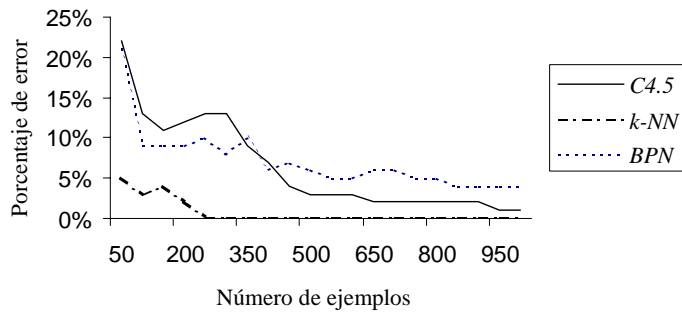


Figura 4. Comparación de los algoritmos de aprendizaje para el criterio del tiempo medio en el sistema

3.3. Secuenciación del *FMS*

El siguiente paso en este estudio es implementar en el modelo de simulación del *FMS* el “conocimiento de secuenciación” obtenido para seleccionar, en tiempo real, la combinación más apropiada de las reglas de secuenciación dependiendo del estado del *FMS*. 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. Asimismo, 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.

Tabla 2. 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
<i>SPT+SPT</i>	4.1023	5.4431	2.1223	2.4183
<i>SPT+NINQ</i>	1.2074	1.2112	1.0455	1.0500
<i>SPT+WINQ</i>	1.2008	1.1904	1.0481	1.0476
<i>SPT+LUS</i>	2.5029	2.5691	1.5208	1.5277
<i>EDD+SPT</i>	3.5140	4.6790	2.2199	2.6260
<i>EDD+NINQ</i>	1.5249	1.6654	1.3361	1.3991
<i>EDD+WINQ</i>	1.5195	1.6735	1.3358	1.4030
<i>EDD+LUS</i>	2.8728	3.2667	1.8767	2.0614
<i>MDD+SPT</i>	3.5260	4.7199	2.3141	2.6908
<i>MDD+NINQ</i>	1.1301	1.1349	1.2356	1.2569
<i>MDD+WINQ</i>	1.1376	1.1464	1.2394	1.2620
<i>MDD+LUS</i>	2.3834	2.5102	1.7848	1.8646
<i>SRPT+SPT</i>	4.4986	6.0908	2.2947	2.6706
<i>SRPT+NINQ</i>	1.3755	1.4049	1.1392	1.1477
<i>SRPT+WINQ</i>	1.3791	1.3964	1.1448	1.1486
<i>SRPT+LUS</i>	2.8307	2.9979	1.6815	1.7195
<i>k-NN</i>	1.0000	1.0000	1.0000	1.0000
<i>BPN</i>	1.0170	1.0207	1.0230	1.0215
<i>C4.5</i>	1.0370	1.0397	1.0102	1.0073

Un resumen de los resultados obtenidos se muestra en la tabla 2. 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). En este caso, la metodología de secuenciación basada en el algoritmo de los vecinos más próximos obtiene los mejores resultados. Con el algoritmo *C4.5* y la red neuronal “backpropagation”, se logran valores de retraso medio no muy superiores a los obtenidos con el algoritmo de los vecinos más próximos. 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 razonamiento basado en casos, en unos porcentajes que varían entre un 13.01% y un 14.64%.

Del mismo modo, para el criterio del tiempo medio en el sistema, la metodología de secuenciación basada en el algoritmo de los vecinos más próximos 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 razonamiento basado en casos, en unos porcentajes que fluctúan entre un 4.55% y un 5%. Por último, se compara la metodología de secuenciación basada en el algoritmo de los vecinos más próximos 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.

4. Conclusiones

En este trabajo, se han empleado y comparado tres tipos distintos de algoritmos de aprendizaje automático para secuenciar tareas en un *FMS*. 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. Asimismo, se ha podido comprobar que el algoritmo de los vecinos más próximos es el que obtiene mejores resultados en el *FMS* propuesto, y que utilizando “conocimiento de secuenciación” siempre se supera a la alternativa de emplear una combinación de reglas de secuenciación de modo constante.

Referencias

- Aha, D.W. (1992). Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms. *International Journal of Man-Machine Studies*, Vol. 36, pp. 267-287.
- Aytug, H.; Bhattacharyya, S.; Koehler, G.J.; Snowdon, J.L. (1994). A review of machine learning in scheduling. *IEEE Transactions on Engineering Management*, Vol. 41, N° 2, pp. 165-171.
- Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison Wesley.
- Ishii, N.; Talavage, J. (1991). A transient-based real-time scheduling algorithm in FMS. *International Journal of Production Research*, Vol. 29, N° 12, pp. 2501-2520.
- Kim, C.-O.; Min, H.-S.; Yih, Y. (1998). Integration of inductive learning and neural networks for multi-objective FMS scheduling. *International Journal of Production Research*, Vol. 36, N° 9, pp. 2497-2509.

- Montazeri, M.; Wassenhove, L.N.V. (1990). Analysis of scheduling rules for an FMS. *International Journal of Production Research*, Vol. 28, N° 4, pp. 785-802.
- Priore, P.; De La Fuente, D.; Gómez, A.; Puente, J. (2001). A review of machine learning in dynamic scheduling of flexible manufacturing systems. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 15, pp. 251-263.
- Quinlan, J.R. (1993). *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann.
- Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. (1986). Learning representations by back-propagating errors. *Nature*, Vol. 323, pp. 533-536.
- Shaw, M.J.; Park, S.; Raman, N. (1992). Intelligent scheduling with machine learning capabilities: the induction of scheduling knowledge. *IIE Transactions*, Vol. 24, N° 2, pp. 156-168.
- Tang, L.-L.; Yih, Y.; Liu, C.-Y. (1993). A study on decision rules of a scheduling model in an FMS. *Computer in Industry*, Vol. 22, pp. 1-13.
- WITNESS (1996). *User Manual. Release 8.0*. AT&T ISTEEL Limited.