

## **Programación de la Producción Predictiva, Dinámica y Estocástica: Estrategia y Políticas**

**Pedro Gómez<sup>1</sup>**

<sup>1</sup> Centro de Investigación de Gestión e Ingeniería de la Producción (CIGIP). Escuela Técnica Superior de Ingenieros Industriales, Universidad Politécnica de Valencia, Campus de Vera, 46022 Valencia. pgomez@cigip.upv.es

### **Resumen**

*El mayor reto de las organizaciones industriales actuales es como responder de forma ágil y económicamente eficiente a los eventos en un sistema dinámico y estocástico. Los cambios rápidos en las necesidades y la demanda de los clientes, los fallos de los proveedores o los fallos en las máquinas deben ser tenidos en cuenta en la realidad. Debido a que los sistemas productivos son dinámicos y a que los eventos inesperados ocurren habitualmente, la reprogramación o “rescheduling” es necesaria para actualizar el programa de producción cuando el estado del sistema se hace infactible. Haciendo uso de una extensa base de datos de más de 300 contribuciones sobre este campo, el autor intenta considerar diferentes aspectos que deberían ser tenidos en cuenta en el establecimiento Estrategias y Políticas de la programación distribuida para un sistema dinámico y estocástico.*

**Palabras clave:** Programación, Dinámica, Estocástica.

### **1. Introducción**

Esta ponencia realiza una revisión de la literatura en algunos aspectos importantes (estrategias, medias de eficacia, políticas, métodos, etc.) en los problemas de programación de la producción dinámica y estocástica en sistemas productivos distribuidos. Este trabajo está organizado como sigue: En la sección 2 se definen los conceptos básicos del problema. En la sección 3 se analizan diferentes estrategias usadas en los problemas de programación dinámica y estocástica. En la sección 4 se introduce la visión de las diferentes políticas de reprogramación, en concreto para el caso predictivo-reactivo. Finalmente en la sección 5, se realizan las conclusiones y se apuntan líneas futuras de trabajo.

### **2. Conceptos básicos**

La mayor parte de las herramientas tradicionales usadas para modelar problemas de toma de decisiones son deterministas. Por determinista se entiende que el modelo se conoce con precisión y certidumbre (Mula, 04;Maccarthy and Liu, 93) . La precisión asume que los parámetros de un modelo representan exactamente la percepción del fenómeno o las características del sistema modelado. Según (Galbraith, 73) la diferencia entre la cantidad de información requerida para ejecutar una tarea y la cantidad de información que realmente posee una organización se entiende como incertidumbre. Este tipo de incertidumbre de carácter estocástico se ha gestionado

tradicionalmente mediante teoría de probabilidad y la estadística (Koopman, 40;Kolmogoroff, 50).

Existen al menos tres aspectos que se deben considerar a la hora de abordar un problema de programación de producción en lo que se refiere al aspecto temporal, que son: El modelo de la realidad que se desea reflejar, el momento que se toman las decisiones y el entorno sobre el cual se actúa.

Por lo que se refiere al modelo que se desea considerar, los problemas pueden ser divididos en estáticos o dinámicos. Si se atiende al momento en el que se toman las decisiones o estrategia los programas pueden estar en línea o fuera de línea. Por lo que se refiere al sistema que gestiona el programa de la producción se entiende que el programa se realiza en tiempo real o no.

### 3. Estrategias de Programación de la Producción en Problemas Dinámicos y Estocásticos

En esta sección se mostrarán las estrategias de programación de la producción más extendidas que afrontan el problema dinámico y estocástico. La estrategia debe ser entendida atendiendo al momento en el que se toman las decisiones: on-line o off-line. Las tres alternativas principales son: La reactiva, la predictivo-reactiva o la mixta.

Tabla 1. Resumen de las principales estrategias de programación de la producción en un entorno dinámico

<i>Autores</i>	<i>Estrategia En Línea</i>	<i>Estrategia Mixta</i>
(Vieira, Herrmann, and Lin, 00a)	- En línea	- Reactiva
(Vieira, Herrmann, and Lin, 03)	- Dinámica (reactiva)	- Predictivo- reactiva
(Mehta and Uzsoy, 99b) (O'Donovan, Uzsoy, and McKay, 99)	- Completamente reactiva	- Predictivo-reactiva - Robusta - Basada en el conocimiento
(Shafaei and Brunn, 99d;Shafaei and Brunn, 99h)	- Teoría de colas	- Reprogramación - Predictivo-robusto
(Raheja and Subramaniam, 02)	- Completamente reactiva	- Reactiva
(Ouelhadj, 03)	- En línea	- Predictivo-reactiva - Robusta
(Aytug, Lawley, Mckay,	- Completamente reactiva	- Predictivo-reactiva

En la **¡Error! No se encuentra el origen de la referencia.** se muestra un resumen de las diferentes denominaciones utilizadas por los autores consultados, clasificadas en dos grandes grupos. Según la revisión literaria realizada se puede decir que existen dos estrategias principales para enfocar la programación de la producción en un entorno dinámico: La programación en línea (en discordancia con otros autores que la denominan dinámica, pero en este trabajo usaremos preferentemente el término en línea) dominada por las reglas de despacho, y la programación mixta basada en un plan previo que se va modificando según se hacen patentes las distorsiones. En la programación de la producción mixta existe una base común denominada habitualmente estrategia predictivo-reactiva, a partir de la cual se distinguen dos mejoras importantes: la programación robusta y la programación basada en el conocimiento.

La programación en línea se puede entender como aquella que no crea programas de producción de forma anticipada, de manera que las decisiones se suelen tomar de forma local y en línea. Normalmente se utilizan reglas de despacho, que son aplicadas cuando una máquina queda libre, para seleccionar el siguiente trabajo. Con estas reglas de despacho o con heurísticas se priorizan los trabajos en cola de espera para establecer el orden de entrada. La ventaja de las reglas de despacho es su fácil aplicación y su rápido cálculo, pero adolecen de una visión global del problema y es difícil de establecer cual será su rendimiento en un entorno cambiante.

Una vez se han diferenciado las estrategias de programación de la producción en entornos dinámicos y estocásticos, y se han destacado las más extendidas, se presentan un conjunto de referencias complementarias. Los autores correspondientes no ofrecen diversas alternativas estratégicas, pero aportan aspectos interesantes que ayudan a entender la situación actual de cada una de ellas.

Dentro de las estrategias completamente reactiva destacamos: (Bhaskaran and Pinedo, 91;Chrétienne, Coffman, Lensrta, and Liu, 95;Dempster, Lensrta, and Rinnooy, 82) presentan una discusión sobre reglas de despacho aplicadas de forma local. (Wu and Wysk, 89) estudiaron el problema de las reglas de despacho en un sistema de fabricación flexible. Los autores dividieron el horizonte temporal en intervalos y al inicio de cada uno de ellos se simulaba el comportamiento de un conjunto de reglas y seleccionaba la mejor para aplicarla en dicho periodo. (Kim and Kim, 94;Jeong and Kim, 98) realizaron aplicaciones parecidas a las de (Wu et al. 89) introduciendo sensibles mejoras. (Aytug, Bhattacharyya, Koehler, and Snowdon, 94) realiza una profunda revisión literaria sobre la aplicación de la técnica de inteligencia artificial denominada máquina de aprendizaje (machina learning) para la selección de la regla de despacho adecuada en un problema de programación de la producción.(Piramuthu, Park, Raman, and Shaw, 91) se basó en la técnica de la máquina de aprendizaje para seleccionar la regla de despacho más adecuada según las condiciones del taller en base a diferentes eventos establecidos en un modelo de simulación. (Chen and Yih, 96) usan una red neuronal para predecir la regla de despacho en función del estado del sistema. (Ovacik and Uzsoy, 94) propone diversos horizontes de reprogramación para un sistema de máquina única con tiempo de cambio de partida dependientes de la secuencia. (Ovacik and Uzsoy, 95) desarrollaron un algoritmo similar para un problema de máquinas paralelas.

Dentro de las estrategias mixtas se encuentran aportaciones interesantes en el enfoque predictivo-reactivo puro o enfoques robustos y basados en el conocimiento.

El enfoque predictivo-reactivo presenta aportaciones como: (Raman, Rachamadugu, and Talbot, 89) que usan una reprogramación continua para la resolución del problema presentado. (Adam and Surkis, 80; Church and Uzsoy, 92) investigan la frecuencia de reprogramación en un taller de trabajo dinámico con fallos en máquinas y tiempos de proceso variables. (Muhlemann, Lockett, and Farn, 82) realiza un estudio de simulación para investigar el rendimiento de un conjunto de reglas heurísticas basadas en un algoritmo de generación de programas sin retraso. Aunque no se pudo concluir con la supremacía de ninguna reglas, si se detecto que la frecuencia de la reprogramación influía en el rendimiento del programa. (Shafaei and Brunn, 99c) realizan un estudio de simulación para investigar sobre el rendimiento de una serie de reglas de despacho en un entorno de reprogramación en un taller de trabajo dinámico. Los resultados muestran como la reglas SPT-C/R es la mas apropiada para minimizar el coste total y como se puede mostrar una relación entre el rendimiento y el intervalo de reprogramación. (Yamamoto and Nof, 85) realizan un estudio mediante reprogramación completa en un taller con fallos aleatorios en las máquinas. Los resultados mostrados por los autores concluyen con resultados mejores de dicho método frente a las reglas de despacho o modificaciones parciales del programa. (Church et al. 92) estudiaron el problema de la reprogramación con una única máquina con llegada dinámica de trabajos. Según su trabajo la política de reprogramación a intervalos fijos es preferible frente a la dirigida por evento excepto cuando se producen llegadas de trabajos urgentes al inicio del programa.

El enfoque de programación robusta presenta aportaciones como: (Graves, 81; Dooley and Mahmoodi, 92; Yellig and Mackulak, 97) definieron la robustez como un objetivo para evaluar el rendimiento real de un método de programación en un entorno estocástico. (Leon, Wu, and Storer, 90) considera la programación robusta de un taller de trabajo con fallos en las máquinas, variación de los tiempos de proceso y minimización de  $C_{max}$  y la desviación del  $C_{max}$  como medidas de rendimiento. (Wu, Byeon, and Storer, 99) propusieron una descomposición para alcanzar la robustez. Ellos consideraron el problema con variación en los tiempos de proceso y con la minimización del retraso ponderado total (weighted tardiness) como función objetivo. (Daniels and Kouvelis, 95; Kouvelis and Yu, 97; Daniels and Carrillo, 97) aportan una discusión interesante sobre la programación robusta. Definen la incertidumbre en términos de escenarios y redefinen el problema de programación como la búsqueda de una secuencia que minimice la máxima desviación entre el rendimiento de la secuencia y las secuencias optimas asociadas al conjunto de escenarios. (Mehta and Uzsoy, 88) desarrollaron un algoritmo que minimizaba el retraso (lateness) y la diferencia entre el  $C_{max}$  previsto y realizado. Este estudio muestra como los programas que son robustos a las distorsiones estocásticas pueden ser generados sin sacrificar demasiado rendimiento del mismo. (Mehta and Uzsoy, 99a) propone una programación predictiva, intentando minimizar  $L_{max}$  sobre una única máquina con llegada dinámica de trabajos y fallos la máquina. Para ello los autores insertan tiempos muertos mientras se recupera el fallo. La propuesta predictiva les supere mejoras importantes de la robustez a costa de una pequeña disminución de  $L_{max}$ . (Mehta and Uzsoy, 98) aplicaron una estrategia predictiva sobre un taller de trabajo con maquinas no fiables y obtuvieron resultados similares al anterior caso. (Shafaei and Brunn, 99g) investiga sobre la robustez de un

conjunto de reglas de despacho en un entorno dinámico y estocástico usando reprogramación cíclica. Las reglas evalúan sobre una función de coste. Se concluye estableciendo una relación entre la influencia de la carga y el balance del taller y la robustez de un programa. (Bollapragada and Sadeh, 96) desarrollan un programa predictivo para minimizar el retraso-adelanto total de un taller de trabajo teniendo en cuenta los posibles fallos de máquinas y la variación de los tiempos de proceso. (McKay, Morton, Ramnath, and Wang, 00) introdujeron la reprogramación dinámica que sub-optimiza en un determinado periodo de tiempo para permitir al sistema estabilizarse y progresivamente optimizarse. (Singer, 00) aplica la reprogramación dinámica en un taller de trabajo con incertidumbre en los tiempos de proceso para minimizar el retraso total.

El enfoque de programación basada en el conocimiento presenta aportaciones como: (Fox and Smith, 84) realizaron una de las primeras y más exitosas aportaciones de la IA aplicada al problema de la programación. (Sadeh, Otsuka, and Schnellbach, 93) muestra la propuesta de la Universidad de Carnegie Mellon sobre propagación de restricciones aplicada a la reprogramación y usada en la propuesta de Micro-Boos como sistema de control y secuenciación. (Smith, 92) discute sobre las posibilidades de reprogramación en OPIS (Opportunistic Intelligent Scheduler), un sistema basado en el conocimiento desarrollado por la Universidad Carnegie Mellon. En (Smith, 87; Smith, Keng, and Kempf, 90) se puede encontrar sobre OPIS. (McKay, Buzacott, and Safayeni, 89) presenta el sistema de programación de carga finita en un taller de trabajo, WATPASS. En este caso la reprogramación se realiza por actuación directa del programador. (Szelke and Kerr, 94) proporciona una extensa revisión de la programación reactiva basada en el conocimiento. (Dutta, 90) propone un mecanismo de control monitorizando el entorno y tomando decisiones correctivas en función del evento. (Baptiste and Favrel, 93) compara esquemas para representar programas alternativos mientras se desarrolla el programa predictivo.

#### **4 Políticas de Reprogramación en la Estrategia Predictivo-reactiva**

Este apartado trata el problema relacionado con la forma de abordar la revisión de las decisiones tomadas por el programador y plasmadas en un programa predictivo. En definitiva se trata de saber cuando es conveniente revisar las decisiones anteriores y en caso de modificarlas cual es el mejor procedimiento en función de las circunstancias.

Según (Vieira et al. 03) la reprogramación es el proceso de actualizar un programa productivo existente en respuesta a un evento.

(Church et al. 92) proporciona una taxonomía de las diferentes aproximaciones que existían al problema de la reprogramación. Los autores consideran la programación continua, la periódica y la dirigida por eventos. (Sabuncuoglu and Bayiz, 00; Vieira et al. 03) proponen tres políticas de reprogramación alternativas: Periódica, dirigida por eventos o híbrida.

La política de reprogramación periódica consiste en la revisión del programa en intervalos regulares, de forma que antes de revisar el programa se suele realizar un proceso de actualización de la información que permita disponer el estado real del taller y de las circunstancias del mismo. Los programas con reprogramaciones periódicas pueden ser vistos como un conjunto sucesivo de programas estáticos encadenados. Una

de las ventajas de esta política es que las técnicas clásicas de programación en entornos estáticos pueden ser aplicadas en cada reprogramación. Así mismo los programas alcanzan cierta estabilidad y su grado de nerviosismo es bajo. El inconveniente más importante de esta política es que los programas pueden volverse inestables frente a eventos de gran impacto ya que su modificación no se realizaría hasta el inicio del nuevo periodo.

En la política de reprogramación dirigida por eventos el programador actúa cuando el rendimiento del programa se ve disminuido por causa de un evento inesperado. En (Church et al. 92) se denomina programación continua a aquella en la que se actualiza el programa cuando se produce cualquier evento sin determinar si este afecta al rendimiento del sistema.

En la política de reprogramación híbrida los programas se revisan en intervalos regulares, y cuando algún evento inesperado reduce el rendimiento del programa en curso. En las políticas híbridas es fundamental realizar una correcta categorización de los tipos de eventos para evitar que el sistema de revisión se active muy frecuentemente. Se debe establecer un equilibrio entre el posible beneficio de corregir el impacto del evento y el coste de reprogramación, teniendo en cuenta que existe una revisión cíclica.

Las reprogramaciones periódicas e híbridas han recibido especial atención bajo el concepto "Rolling Time Horizon" o periodificación del espacio temporal. (Muhlemann et al. 82; Adam et al. 80; Ovacik et al. 94; Shafaei and Brunn, 99f; Shafaei and Brunn, 99b; Sun and Lin, 94b; Singer, 00; Qi, Burns, and Harrison, 00; Chen and Chen, 03) son algunas referencias significativas en las que se aplica el Rolling Time Horizon.

(Adam et al. 80) realiza un estudio basado en un taller dinámico en el cual se aplican las reglas de despacho que necesitan el cálculo del estado actual de las prioridades de los trabajos remanentes en el taller. Los autores proponen un nuevo procedimiento mediante el cual la actualización de la información y la programación se realiza a intervalos.

(Muhlemann et al. 82) fue una de las primeras aplicaciones de este concepto a la programación en entornos dinámicos. Los autores realizan un estudio de simulación para investigar el rendimiento de un conjunto de reglas heurísticas basadas en un algoritmo de generación de programas sin retraso. Aunque no se pudo concluir con la supremacía de ninguna reglas, si se detecto que la frecuencia de la reprogramación influía en el rendimiento del programa. Este trabajo profundiza en el análisis de la frecuencia de reprogramación en un taller de trabajo dinámico con fallos en máquinas y tiempos de proceso variables.

(Ovacik et al. 94) propone diversos horizontes de reprogramación para un sistema de máquina única con tiempos de cambio de partida dependientes de la secuencia.

(Sun and Lin, 94a) plantean el problema de programación en un taller general. Los autores plantean un algoritmo estático de vuelta atrás. El algoritmo debe hacer frente a la disminución del rendimiento provocada por los fallos en máquinas. Para ello se utiliza una política de reprogramación dirigida por eventos. Se plantean varias medidas de rendimiento: tiempo de flujo medio, makespan, penalización por retraso de los trabajos. El programa se modifica completamente en cada revisión si es necesario.

(Shafaei and Brunn, 99a) realizan un estudio de simulación para investigar sobre el rendimiento de una serie de reglas de despacho en un entorno de reprogramación en un taller de trabajo dinámico. Los resultados muestran como la reglas SPT-C/R es la mas apropiada para minimizar el coste total y como se puede mostrar una relación entre el rendimiento y el intervalo de reprogramación. (Shafaei and Brunn, 99e) investiga sobre la robustez de un conjunto de reglas de despacho en un entorno dinámico y estocástico usando reprogramación cíclica Las reglas evalúan sobre una función de coste. Se concluye estableciendo una relación entre la influencia de la carga y el balance del taller y la robustez de un programa.

(Singer, 00) usa un procedimiento heurístico de descomposición basado en un procedimiento centrado en el cuello de botella desarrollado por (Pinedo and Singer, 99).

(Qi et al. 00) propone un procedimiento heurístico de descomposición basado en un algoritmo genético multi-población.

(Chen et al. 03) realizan una propuesta de algoritmo dividido en dos etapas: asignación de un trabajo a una máquina y asignación de una herramienta a una operación, en un taller flexible. El algoritmo debe hacer frente a los diversos eventos taller como los fallos en máquinas. Para ello se utiliza una política de reprogramación dirigida por eventos. Se plantean tres medidas de rendimiento: tiempo de flujo medio, retrasos de los trabajos y porcentaje de trabajos retrasados. El programa se modifica completamente en cada revisión si es necesario.

Por lo que se refiere a la programación por eventos la mayoría de los autores estudiados están de acuerdo en que ésta suele ser más eficiente que la reprogramación periódica. Algunas referencias interesantes son (Yamamoto et al. 85;Church et al. 92;Vieira, Herrmann, and Lin, 00b;Vieira, Herrmann, and Lin, 00c;Cowling and Johansson, 02;Rossi and Dini, 00b).

(Yamamoto et al. 85) realizan un estudio mediante reprogramación completa en un taller con fallos aleatorios en las máquinas. Los resultados mostrados por los autores concluyen con resultados mejores de dicho método frente a las reglas de despacho o modificaciones parciales del programa.

(Church et al. 92) estudiaron el problema de la reprogramación con una única máquina con llegada dinámica de trabajos. Según su trabajo la política de reprogramación a intervalos fijos es preferible frente a la dirigida por evento excepto cuando se producen llegadas de trabajos urgentes al inicio del programa.

(Vieira, Herrmann, and Lin, 00d;Cowling et al. 02;Rossi and Dini, 00a) presentan un estudio para la búsqueda de un modelo que pueda predecir el rendimiento de un sistema de una máquina con reprogramación dirigida por eventos y periódica en un entorno donde diferentes tipos de trabajos llegan dinámicamente existiendo un tiempo de cambio de partida entre trabajos.

(Vieira et al. 00b) extiende el estudio anterior a un sistema de máquinas paralelas. Este trabajo demuestra que la frecuencia de reprogramación puede afectar de forma

determinante al rendimiento del sistema, dado que influye sobre el número de cambios de partida del sistema.

Como conclusión a este apartado se debe remarcar toda política de reprogramación está condicionada por los eventos que se producen en el sistema, ya que son éstos los que determinan el grado de degradación del programa original a través de diferentes impactos. La política elegida debe, en todo caso, intentar evitar la degradación del rendimiento del sistema. En la medida que la política seleccionada alcance este objetivo será o no adecuada para el problema en el cual se aplica.

## **5 Conclusiones y Líneas de Futuro**

Hasta la fecha se ha realizado un gran esfuerzo en el desarrollo de métodos que generan programas de producción óptimos. Tradicionalmente, dichas aportaciones generan una optimización combinatoria para problemas de programación de la producción. Dichos problemas se han abstraído hasta la obtención del modelo deseado para afrontar una u otra parte de un problema. En muchas ocasiones no se ha reflexionado suficientemente sobre la aplicabilidad de los resultados obtenidos.

El objetivo de las comunicaciones sobre reprogramación, parte fundamental de la gestión de sistemas fabriles dinámicos y estocásticos, es abordar problemas cercanos a la realidad. Este trabajo revisa algunos conceptos básicos necesarios en los sistemas dinámicos y estocásticos y algunas de las mayores contribuciones entre cientos de aportaciones.

Otros aspectos importantes que no se han tratado en este trabajo son aspectos como: Las medidas de eficiencia, los métodos y algoritmos de resolución, etc. En general, las aportaciones sobre casos dinámicos y estocásticos aplicados a la programación de sistemas productivos son bastante escasas, por lo que queda un gran trabajo por realizar.

## **Referencias**

- Adam,N.R. and Surkis,J. 1980. Priority update intervals and anomalies in dynamic ratio job shop scheduling rules. *Management Science*, 26: 1127-1137.
- Aytug,H., Bhattacharyya,S., Koehler,G.J., and Snowdon,J.L. 1994. A review of machine learning in scheduling. *IEEE Transactions on Engineering Management*, 41(2): 165-171.
- Aytug,H., Lawley,M.A., Mckay,K., Mohan,S., and Uzsoy,R. 2005. Executing production schedules in the face of uncertainties: A review and some future directions. *European Journal of Operational Research*, 161(1): 86-110.
- Baptiste,P. and Favrel,J. 1993. Taking into account the rescheduling problem during the scheduling phase. *Production Planning & Control*, 4: 349-360.
- Bhaskaran,K. and Pinedo,M. 1991. Dispatching In G.Salvendy, (Ed.), *Handbook of Industrial Engineering*: New York: Wiley.
- Bollapragada,R. and Sadeh,N.M. 1996. Cost-based approaches to stochastic job shop scheduling problems. Research Report, Robotics Institute, Carnegie Mellon University.
- Chen,C. and Yih,Y. 1996. Identifying attributes for knowledge base development in dynamic scheduling environments. *International Journal of Production Research*, 34(6): 1739-1755.

- Chen,J. and Chen,F.F. 2003. Adaptive scheduling in random flexible manufacturing systems subject to machine breakdowns. *International Journal of Production Research*, 41(9): 1927-1951.
- Chrétienne,P., Coffman,J.E., Lenstra,J.K., and Liu,Z. 1995. *Scheduling theory and its applications*. Chichester: Wiley.
- Church,L.K. and Uzsoy,R. 1992. Analysis of Periodic and Event-Driven Rescheduling Policies in Dynamic Shops. *International Journal of Computer Integrated Manufacturing*, 5(3): 153-163.
- Cowling,P.I. and Johansson,M. 2002. Using real time information for effective dynamic scheduling. *International Journal Of Operational Research*, 139(2): 230-244.
- Daniels,R.L. and Carrillo,J.E. 1997. Beta-Robust scheduling for single machine systems with uncertain processing times. *IIE Transactions on Scheduling and Logistics*, 29: 977-985.
- Daniels,R.L. and Kouvelis,P. 1995. Robust Scheduling to Hedge Against Processing Time Uncertainty in Single-Stage Production. *Management Science*, 41(2): 363-376.
- Dempster,M.A., Lenstra,J.K., and Rinnooy,K. 1982. *Deterministic and stochastic scheduling*. 6-17 Durham (UK):
- Dooley,K.J. and Mahmoodi,F. 1992. Identification of robust scheduling heuristics: application of Taguchi methods in simulation studies. *Computers and Industrial Engineering*, 22: 359-368.
- Dutta,A. 1990. Reacting to Scheduling Exceptions in Fms Environments. *Iie Transactions*, 22(4): 300-314.
- Fox,M.S. and Smith,S.F. 1984. ISIS - Knowledge based system for factory scheduling. *Expert Systems*, 1: 300-314.
- Galbraith,J. 1973. *Designing complex organizations*. Massachusetts: Addison-Wesley Pub. Co.
- Graves,C.S. 1981. Review of production scheduling. *Operations Research*, 29: 646-675.
- Jeong,K.C. and 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: 2609-2626.
- Kim,M.H. and Kim,Y.D. Simulation-based real-time scheduling in a flexible manufacturing system. 13, 85-93. 1994. *J. Manufact. Syst. (GENERIC)*  
Ref Type: Generic
- Kolmogoroff,A. 1950. *Foundations of probability*. New York: Chelsea Pub Co; 2nd edition (June 1, 1960).
- Koopman,B.O. 1940. The axioms and algebra of intuitive probability. *Annals of Mathematics*, 41: 269-292.
- Kouvelis,P. and Yu,G. 1997. *Robust discrete optimization and its applications*. Kluwer Academic Publishers.
- Leon,V.J., Wu,S.D., and Storer,R.H. 1990. Robustness measures and robust scheduling for job shops. Department of Industrial Engineering, Lehigh University:
- Maccarthy,B.L. and Liu,J.Y. 1993. Addressing the Gap in Scheduling Research - A Review of Optimization and Heuristic Methods in Production Scheduling. *International Journal of Production Research*, 31(1): 59-79.
- Mckay,K., Mortn,T.E., Ramnath,P., and Wang,J. 2000. Aversion dynamics-scheduling when the system changes. *Journal of Scheduling*, 3(2): 71-88.
- McKay,K.N., Buzacott,J.A., and Safayeni,F.R. 1989. *The Scheduler's knowledge of uncertainty: The missing link In Anonymous, Knowledge based production management systems*: Amsterdam: Elsevier Science Publishers B. V.

- Mehta,S.V. and Uzsoy,R. 1988. Predictable scheduling of a shop job subject to breakdowns. *IEEE Trans.on Robotics and Automation*, 14: 365-378.
- Mehta,S.V. and Uzsoy,R. 1999a. Predictable scheduling of a single machine subject to breakdowns. *International Journal of Computer Integrated Manufacturing*, 12(1): 15-38.
- Mehta,S.V. and Uzsoy,R. 1999b. Predictable scheduling of a single machine subject to breakdowns. *International Journal of Computer Integrated Manufacturing*, 12(1): 15-38.
- Mehta,S.V. and Uzsoy,R.M. 1998. Predictable scheduling of a job shop subject to breakdowns. *Ieee Transactions on Robotics and Automation*, 14(3): 365-378.
- Muhlemann,A.P., Lockett,A.G., and Farn,C.K. 1982. Job Shop Scheduling Heuristics and Frequency of Scheduling. *International Journal of Production Research*, 20(2): 227-241.
- Mula,J. 2004. Modelos para la planificación de la producción bajo incertidumbre. Aplicación en una empresa del sector del automóvil. Departamento de Organización de Empresas, Economía Financiera y Contabilidad. Universidad Politécnica de Valencia, Alcoy (Alicante)
- O'Donovan,R., Uzsoy,R., and McKay,K.N. 1999. Predictable scheduling of a single machine with breakdowns and sensitive jobs. *International Journal of Production Research*, 37(18): 4217-4233.
- Ouelhadj,D. 2003. A multi-agent system for the integrated dynamic scheduling of steel production. Nottingham University, School of Computer Science and Information Technology,
- Ovacik,I.M. and Uzsoy,R. 1994. Rolling horizon algorithms for a single machine dynamic scheduling problems with sequence-dependent set-up times. *International Journal of Production Research*, 32: 1243-1263.
- Ovacik,I.M. and Uzsoy,R. 1995. Rolling horizon procedures for dynamic parallel machine scheduling problems with sequence dependent set up times. *International Journal of Production Research*, 33: 3173-3192.
- Pinedo,M. and Singer,M. 1999. A shifting bottleneck heuristic for minimizing the total weighted tardiness in a job shop. *Naval Research Logistics*, 46: 1-17.
- Piramuthu,S., Park,S.C., Raman,N., and Shaw,M.J. 1991. Integration of simulation modelling and inductive learning in an adaptive decision support system. *IEEE Society Press*.
- Qi,J.G., Burns,G.R., and Harrison,D.K. 2000. The application of parallel multipopulation genetic algorithms to dynamic job shop scheduling. *International Journal of Advanced Manufacturing Technology*, 16: 609-615.
- Raheja,A.S. and Subramaniam,V. 2002. Reactive recovery of job shop schedules - A review. *International Journal of Advanced Manufacturing Technology*, 19: 756-763.
- Raman,N., Rachamadugu,R.V., and Talbot,F.B. 1989. Real-Time Scheduling of An Automated Manufacturing Center. *European Journal of Operational Research*, 40(2): 222-242.
- Rossi,A. and Dini,G. 2000a. Dynamic scheduling of FMS using a real-time genetic algorithm. *International Journal of Production Research*, 38(1): 1-20.
- Rossi,A. and Dini,G. 2000b. Dynamic scheduling of FMS using a real-time genetic algorithm. *International Journal of Production Research*, 38(1): 1-20.
- Sabuncuoglu,I. and Bayiz,M. 2000. Analysis of reactive scheduling problems in a job shop environment. *European Journal of Operational Research*, 126(3): 567-586.

- Sadeh,N.M., Otsuka,S., and Schnelbach,R. 1993. Redictive and reactive scheduling with micro-boss scheduling and control system. Centre for Integrated Manufacturing Decision Systems, Carnegie Melon University:
- Shafaei,R. and Brunn,P. 1999d. Workshop scheduling using practical (inaccurate) data Part 1: The performance of heuristic scheduling rules in a dynamic job shop environment using a rolling time horizon approach. *International Journal of Production Research*, 37(18): 3913-3925.
- Shafaei,R. and Brunn,P. 1999h. Workshop scheduling using practical (inaccurate) data Part 2: An investigation of the robustness of scheduling rules in a dynamic and stochastic environment. *International Journal of Production Research*, 37(18): 4105-4117.
- Shafaei,R. and Brunn,P. 1999g. Workshop scheduling using practical (inaccurate) data Part 2: An investigation of the robustness of scheduling rules in a dynamic and stochastic environment. *International Journal of Production Research*, 37(18): 4105-4117.
- Shafaei,R. and Brunn,P. 1999b. Workshop scheduling using practical (inaccurate) data Part 1: The performance of heuristic scheduling rules in a dynamic job shop environment using a rolling time horizon approach. *International Journal of Production Research*, 37(18): 3913-3925.
- Shafaei,R. and Brunn,P. 1999f. Workshop scheduling using practical (inaccurate) data Part 2: An investigation of the robustness of scheduling rules in a dynamic and stochastic environment. *International Journal of Production Research*, 37(18): 4105-4117.
- Shafaei,R. and Brunn,P. 1999a. Workshop scheduling using practical (inaccurate) data Part 1: The performance of heuristic scheduling rules in a dynamic job shop environment using a rolling time horizon approach. *International Journal of Production Research*, 37(18): 3913-3925.
- Shafaei,R. and Brunn,P. 1999e. Workshop scheduling using practical (inaccurate) data Part 2: An investigation of the robustness of scheduling rules in a dynamic and stochastic environment. *International Journal of Production Research*, 37(18): 4105-4117.
- Shafaei,R. and Brunn,P. 1999c. Workshop scheduling using practical (inaccurate) data Part 1: The performance of heuristic scheduling rules in a dynamic job shop environment using a rolling time horizon approach. *International Journal of Production Research*, 37(18): 3913-3925.
- Singer,M. 2000. Forecasting policies for scheduling a stochastic due date job shop. *Computer and Operations Research*, 28: 193-207.
- Smith,S.F. 1987. A constraint-based framework for reactive management of factory schedules. Hilton Head, South Carolina:
- Smith,S.F. 1992. A methodology and architecture for reactive scheduling. The Robotic Institute, Carnegie Mellon University:
- Smith,S.F., Keng,N., and Kempf,K. 1990. Exploiting local flexibility during execution of pre-computed schedules. The Robotic Institute, Carnegie Mellon University and Knowledge based Laboratory, Intel Corporation:
- Sun,D. and Lin,L. 1994a. A Dynamic Job-Shop Scheduling Framework - A Backward Approach. *International Journal of Production Research*, 32(4): 967-985.
- Sun,D. and Lin,L. 1994b. A Dynamic Job-Shop Scheduling Framework - A Backward Approach. *International Journal of Production Research*, 32(4): 967-985.
- Szelke,E. and Kerr,R.M. 1994. Knowledge-Based Reactive Scheduling. *Production Planning & Control*, 5(2): 124-145.

- Vieira,G.E., Herrmann,J.W., and Lin,E. 2000b. Predicting the performance of rescheduling strategies for parallel machine systems. *Journal of Manufacturing Systems*, 19(4): 256-266.
- Vieira,G.E., Herrmann,J.W., and Lin,E. 2000d. Analytical models to predict the performance of a single-machine system under periodic and event-driven rescheduling strategies. *International Journal of Production Research*, 38(8): 1899-1915.
- Vieira,G.E., Herrmann,J.W., and Lin,E. 2000c. Analytical models to predict the performance of a single-machine system under periodic and event-driven rescheduling strategies. *International Journal of Production Research*, 38(8): 1899-1915.
- Vieira,G.E., Herrmann,J.W., and Lin,E. 2000a. Analytical models to predict the performance of a single-machine system under periodic and event-driven rescheduling strategies. *International Journal of Production Research*, 38(8): 1899-1915.
- Vieira,G.E., Herrmann,J.W., and Lin,E. 2003. Rescheduling manufacturing systems: A framework of strategies, policies, and methods. *Journal of Scheduling*, 6(1): 39-62.
- Wu,D.S. and Wysk,R.A. 1989. An application of discrete-event simulation to on-line control and scheduling of flexible manufacturing. *International Journal of Production Research*, 27(9)
- Wu,S.D., Byeon,E.S., and Storer,R.H. 1999. A graph-theoretic decomposition of the job shop scheduling problem to achieve scheduling robustness. *Operations Research*, 47(1): 113-124.
- Yamamoto,M. and Nof,S.Y. 1985. Scheduling Rescheduling in the Manufacturing Operating System Environment. *International Journal of Production Research*, 23(4): 705-722.
- Yellig,E.J. and Mackulak,G.T. 1997. Robust deterministic scheduling in stochastic environments: the method of capacity hedge points. *International Journal of Production Research*, 35: 369-379.