

Estado del arte del problema de flujo general flexible con costes en la función objetivo*

Albert Corominas, Néstor Andrés González, Rafael Pastor

Institut d'Organització i Control de Sistemes Industrials
Universitat Politècnica de Catalunya
Av. Diagonal 647 planta 11, 08028 Barcelona
{*albert.corominas, nestor.andres.gonzalez, rafael.pastor*}@upc.edu

Resumen: En el ámbito de la programación de operaciones en taller un importante número de investigaciones se han destinado, en las últimas dos décadas, al estudio del problema de flujo general flexible o fJSP. En el fJSP existen m máquinas y n jobs. Cada job j está compuesto por una secuencia de operaciones y la ejecución de la operación h del job j requiere una máquina de un conjunto de máquinas. Desde un punto de vista combinatorio, el fJSP es complejo ya que se deben tratar dos subproblemas: el de asignación de las operaciones a las máquinas y el de secuenciación de las operaciones en cada una de las máquinas. Aunque en el fJSP es común que se seleccione como criterio de optimización el *makespan*, en los talleres existe otro tipo de criterios que reflejan las necesidades de la industria moderna, como por ejemplo, el coste de producción y los costes asociados a los adelantos y retrasos que se generan con respecto a la fecha de entrega comprometida d_j . En este documento se presenta una descripción del fJSP, así como una exposición del estado del arte haciendo énfasis en la literatura del problema que considera costes en la función objetivo. Por último se realiza un análisis del estado del arte.

Palabras clave: Problema del taller mecánico de flujo general flexible

1. Introducción

En la mayoría de sistemas productivos es importante realizar una buena planificación y programación de operaciones y es necesario establecer un programa, es decir, establecer la secuencia de operaciones en cada máquina y el instante de inicio de las operaciones, con el objetivo de optimizar una función objetivo determinada. En la programación de operaciones en taller (problema del taller mecánico) n jobs (órdenes de trabajo) deben realizarse en m máquinas (recursos, secciones, puestos de trabajo). La realización de cada job implica la ejecución de una serie de operaciones prefijadas, y cada operación debe ser asignada a una de las m máquinas y tiene una duración (tiempo de proceso) determinada y conocida.

* Con el apoyo del Programa Alβan, Programa de Becas de Alto Nivel de la Unión Europea para América Latina, beca nº E06D100344CO

En un taller es muy común el problema de programación de operaciones de flujo general (*Job Shop*), que se caracteriza porque los *jobs* tienen diferentes rutas de proceso. Lo que significa, por ejemplo, que existen *jobs* en los que dos operaciones sucesivas se ejecutan la primera en la máquina m_1 y la siguiente en la máquina m_2 , y *jobs* en los que dos operaciones sucesivas se ejecutan la primera en la máquina m_2 y la siguiente en la máquina m_1 .

En Borne *et al.* (2002a) se asume que en el modelo de flujo general más simple un *job* debe ser procesado en una máquina en particular al menos una vez, en su ruta a través del taller. Sin embargo, en otro tipo de modelos de flujo general, el *job* puede estar obligado a visitar una máquina muchas veces a lo largo de la ruta que realiza a través del taller, por lo que en un taller de este tipo los *jobs* pueden recircular, lo que aumenta considerablemente la complejidad del problema a resolver (Pinedo, 2005).

Una generalización del problema de flujo general, es el problema de flujo general flexible (*flexible Job Shop Scheduling Problem*) o fJSP como se le conoce por sus siglas en inglés. En este problema existen m máquinas y n *jobs*. Cada *job* j está compuesto por una secuencia de operaciones y la ejecución de la operación h del *job* j requiere una máquina de un conjunto de máquinas M_{jh} y un tiempo de proceso determinado para cada una de las máquinas. Desde un punto de vista combinatorio, el fJSP es muy complejo ya que se deben tratar dos subproblemas: el de asignación de las operaciones a las máquinas que pueden ejecutarlas y el de secuenciación de las operaciones en cada una de las máquinas (Armentano *et al.*, 2004). En el primer subproblema se realiza la asignación de operaciones a las máquinas teniendo en cuenta que una operación de un *job* puede ser ejecutada por varias máquinas. En el subproblema de secuenciación, se secuencian las operaciones de los *jobs* que se realizan en una máquina.

Aunque en el problema de flujo general flexible es muy común que se seleccione como criterio de optimización el C_{\max} (*makespan*), en Corominas *et al.* (2008) se propone minimizar la suma de los costes asociados a los adelantos y retrasos que se generan con respecto a la fecha de entrega comprometida y unos costes dependientes del intervalo de realización de las operaciones. Un ejemplo de estos últimos es el coste de la energía, dado que existen talleres en los que se producen *jobs* durante diferentes intervalos temporales, como por ejemplo el día o la noche, y la energía que consumen las máquinas está sujeta a la tarifa (coste) de la empresa eléctrica. Habitualmente dicho coste es elevado durante el día y disminuye durante la noche.

Este documento está organizado de la siguiente forma. En la sección 2 se presenta una descripción del fJSP y se realiza una exposición del estado del arte, haciendo especial énfasis en la literatura del problema que considera costes en la función objetivo. En la sección 3 se realiza un análisis del estado del arte. Por último, en la sección 4, se presentan las conclusiones.

2. Estado del arte

En este apartado se describe el estado del arte del fJSP, cuyos orígenes se remontan a la década de los noventa, Bruker y Schlie (1990) y Brandimarte (1993). Pero ante todo es importante describir con precisión el fJSP; así, a continuación es formulado tomando

como base la nomenclatura que fue empleada en Fattahi *et al.* (2007). La terminología completa se relaciona en el anexo 1 de este documento.

2.1 Descripción del problema de flujo general flexible

En el fJSP existen m máquinas y n jobs. Cada job j está compuesto por una secuencia de operaciones O_{jh} , $h = 1, \dots, h_j$, donde O_{jh} y h_j denotan la h -ésima operación del job j y el número total de operaciones requerido para el job j , respectivamente. El conjunto de todas las máquinas se denota como M . El índice i denota la máquina, el índice j denota el job y el índice h denota la operación. La ejecución de cada operación h del job j (O_{jh}) requiere una máquina del subconjunto de máquinas $M_{jh} \subseteq M$ (es decir, para realizar cada operación h de cada job j existe un conjunto de máquinas M_{jh} donde ésta puede ser realizada y se debe seleccionar una de ellas) y un tiempo de proceso determinado p_{ijh} para cada una de las máquinas. Un ejemplo de un problema de flujo general flexible de dos jobs y tres máquinas se presenta en la figura 1. La secuencia de operaciones, las máquinas que pueden procesarlas, y los tiempos de proceso se pueden apreciar en dicha figura.

Operaciones:	1	2	3
job $j=1$	$p_{111} = 10$ o	$p_{212} = 12$ o	
	$p_{211} = 15$	$p_{312} = 18$	
job $j=2$	$p_{121} = 20$ o	$p_{122} = 25$	$p_{123} = 30$ o
	$p_{321} = 25$		$p_{223} = 15$ o
			$p_{323} = 25$

Figura 1. Ejemplo de un fJSP de dos jobs y tres máquinas.

Tras realizar la anterior descripción es importante recordar que en el fJSP se deben resolver dos subproblemas, el de asignación de cada operación a una de las máquinas que la puede realizar y el de secuenciación de las operaciones en cada una de éstas. Por otro lado es muy común que para el fJSP se seleccione, como criterio de optimización, el C_{\max} (*makespan*), véase por ejemplo Fattahi y Saidi-Mehrabad (2007).

A modo de resumen a continuación se enumeran las hipótesis que se consideran comúnmente en la mayoría de variantes del problema.

2.2. Hipótesis

1. El horizonte de producción H del taller se considera que está compuesto por un número finito de períodos (Mati *et al.*, 2011).
2. La secuencia de operaciones para cada job es fija y no puede ser modificada. Se considera que los jobs tienen unas precedencias lineales (establecidas) entre sus operaciones, lo que significa que hay una única operación que precede y una única

operación que es sucesora inmediata. Téngase en cuenta que la primera y la última operación de cada *job* son las únicas que no tienen una operación que precede y una que es sucesora inmediata, respectivamente (Gao *et al.*, 2006).

3. Cada máquina sólo puede procesar un *job* a la vez (Rajkumar *et al.*, 2011).
4. Las operaciones no pueden ser interrumpidas (Abraham *et al.*, 2006).
5. Cada *job* j debe ser completado idealmente en su fecha de entrega comprometida d_j (Benrejeb *et al.*, 2008).
6. Los tiempos de preparación (*setup times*) son independientes de la secuencia y están incluidos en los tiempos de proceso de los *jobs*.

En el siguiente apartado se describen los dos principales enfoques de resolución del fJSP.

2.3. Enfoques de resolución y complejidad del problema de flujo general flexible

Una detallada revisión de procedimientos de resolución del fJSP puede encontrarse en Fattahi *et al.* (2007) y Correa *et al.* (2008). Bruker y Schlie (1990) fueron los primeros autores que abordaron este problema. Ellos desarrollaron un algoritmo polinomial para resolver el fJSP con dos *jobs*. Para resolver el caso con más de dos *jobs* se han utilizado principalmente dos enfoques: el jerárquico y el integrado.

En el enfoque jerárquico los dos subproblemas, el de asignación de operaciones a las máquinas y la secuenciación de operaciones en las máquinas, se tratan separadamente, lo que significa que la asignación y secuenciación son consideradas independientemente. Por su parte, en los enfoques integrados la asignación y secuenciación se consideran simultáneamente.

Los enfoques jerárquicos se basan en la idea de descomponer el problema original con el fin de reducir su complejidad. Brandimarte (1993) fue el primero que utilizó esta descomposición para el fJSP. En Brandimarte (1993) se resolvió el subproblema de asignación de las operaciones a las máquinas empleando reglas de *dispatching* y luego el subproblema de secuenciación empleando una búsqueda tabú. En Fattahi y Saidi-Mehrabad (2007) se presentó un modelo matemático y un algoritmo de búsqueda tabú para resolver el fJSP con tiempos de preparación dependientes de la secuencia (*sequence-dependent setup times*). En este trabajo los autores usaron un enfoque jerárquico con dos heurísticas para resolver el problema. La primera heurística se empleó para la asignación de cada operación a una máquina del conjunto de máquinas disponibles y la segunda para secuenciar las operaciones asignadas en todas las máquinas, con el fin de obtener una secuencia factible que minimizara el C_{\max} (*makespan*). En Gao *et al.* (2009) se resolvió el subproblema de asignación de las operaciones a las máquinas empleando el método de optimización por enjambre de partículas (*particle swarm optimization*) y luego se resolvió el subproblema de secuenciación empleando una búsqueda tabú. En Arkat *et al.* (2009) se utilizó un algoritmo de recocido simulado para, por separado, resolver el subproblema de asignación de las operaciones a las máquinas y el subproblema de secuenciación. Para

validar el algoritmo de recocido simulado se compararon los resultados con la solución óptima obtenida mediante *Branch and Bound*. En De Giovanni y Pezzella (2010) se utilizó un procedimiento de optimización jerárquico que combina un algoritmo genético con procedimientos de búsqueda local, para resolver el subproblema de asignación de las operaciones a las máquinas y el subproblema de secuenciación. Literatura adicional sobre este enfoque fue presentada en Borne *et al.* (2002b), Wu y Xia (2005) y Low *et al.* (2006).

Con referencia al enfoque integrado, en Hurink *et al.* (1994) se propuso una búsqueda tabú. El enfoque integrado que presentaron Dauzère-Pérès y Paulli (1997) fue definido como una estructura de optimización local basada en vecindarios. Gambardella y Mastrololli (2002) presentaron dos procedimientos de optimización local basados en vecindarios y mejoraron la técnica de optimización de Dauzère-Pérès y Paulli (1997).

La complejidad del problema de flujo general, en el que sólo se trata el problema de secuenciación, se ha determinado como NP-hard (Garey *et al.*, 1976). Como el fJSP es una extensión del problema de flujo general en el que se deben tratar dos subproblemas, el de asignación de las operaciones a las máquinas y el de secuenciación de las operaciones en cada una de las máquinas, se ha establecido que el fJSP es fuertemente NP-hard, lo que justifica por algunos autores el uso de heurísticas o algoritmos de aproximación para obtener una solución satisfactoria y en el mejor de los casos una solución óptima, Jain y Meeran (1999) y Blazewicz *et al.* (1996).

2.4 Estado del arte del fJSP con costes en la función objetivo

La literatura sobre el fJSP con costes en la función objetivo es considerablemente escasa. Aunque en los problemas de flujo general flexible es muy común que se seleccione como criterio de optimización el C_{\max} (*makespan*), Ham *et al.*, (2011), en los talleres existe otro tipo de criterios que son de gran utilidad y que reflejan las necesidades de la industria moderna.

- Por ejemplo, en Fatemi Ghomi *et al.*, (2005) se minimiza la suma del coste de setup (*setup cost*), del trabajo en proceso (*work-in-process*) y del coste de inventario por unidad de tiempo del producto terminado (*end product inventory holding cost per time unit*). Los autores seleccionaron dicha función objetivo, porque es común encontrarla en entornos de cadena de suministro, en donde el proveedor que genera múltiples productos desde su taller se enfrenta al problema de flujo general flexible y al problema de minimizar costes, como los de inventario. En el caso estudiado, la tasa de demanda de los productos es determinista y constante, y se basa en la relación existente entre el proveedor y su capacidad de producción. Los autores desarrollaron un programa lineal entero mixto (MILP) para resolver el problema.
- En Alvarez-Valdes *et al.*, (2005) se minimiza la suma de los costes asociados a los adelantos y retrasos que se generan con respecto a la fecha de entrega comprometida d_j . Además se minimiza otro objetivo, considerado como secundario, que es el trabajo en proceso (*work-in-process*). Los autores diseñaron e implementaron un sistema de programación de operaciones para una fábrica de vidrio, en el que resultar particular el hecho de que los trabajadores que poseen ciertas habilidades y realizan trabajos manuales, fueron modelados como máquinas especiales. Para resolver el

problema desarrollaron un procedimiento heurístico en el que primero se obtiene una solución inicial factible, que luego se intenta mejorar, teniendo en cuenta los criterios de la función objetivo.

- En el trabajo de Ho y Tay (2008) se minimiza el C_{\max} (*makespan*), el coste de los retrasos que se generan con respecto a la fecha de entrega comprometida d_j y el tiempo de flujo (*flow time*). Los autores asignaron la misma prioridad a los criterios de la función objetivo y resolvieron el problema con reglas de *dispatching* compuestas que obtuvieron a través de un enfoque de programación genética (*genetic programming*).
- Por su parte en Benrejeb *et al.* (2008) se minimizan cinco criterios que son: el C_{\max} (*makespan*), la carga de trabajo de la máquina crítica (*workload of the critical machine*), la carga de trabajo total de las máquinas (*total workload of machines*), los adelantos/retrasos (*penalties of earliness/tardiness*) y el coste de producción (*production cost*). Con referencia al criterio de minimizar los adelantos/retrasos, en Benrejeb *et al.* (2008) se define como la suma, para todos los *jobs*, de los adelantos e_j y los retrasos g_j , como se muestra a continuación:

$$\sum_{j=1}^n (X_j e_j + Y_j g_j)$$

con:

$$e_j = \max (0, d_j - t_j^{\text{FIN}})$$

$$g_j = \max (0, t_j^{\text{FIN}} - d_j)$$

y donde X_j y Y_j son las penalizaciones por adelanto y retraso, respectivamente, para el *job j*, d_j es la fecha de entrega comprometida y t_j^{FIN} es el instante en que se finalizan todas las operaciones del *job j*.

En el trabajo de Benrejeb *et al.* (2008) se desarrolló un algoritmo genético para resolver el problema con los cinco criterios de optimización expuestos.

- En Berkoune y Mesghouni (2008) se resuelve un fJSP multicriterio en el que la lista de demandas del taller esta dividida en *jobs* confirmados (o seguros) y una predicción de nuevos *jobs* a realizar. El problema consiste en programar la ejecución de n *jobs* ($S1$ *jobs* confirmados y $S2$ *jobs* que se predice que se van a realizar) en m máquinas. Se minimizan cinco criterios que son: el C_{\max} (*makespan*), el coste de manufactura del *job j* (*manufacturing cost*), el coste de las penalizaciones del *job j* (*cost of the penalties*), el coste de almacenamiento del *job j* confirmado (*storage cost of the firm job*) y el coste de almacenamiento de los *jobs j* predcidos (*storage cost of the predicted jobs*). El coste de las penalizaciones se define como el producto del coste de penalización del *job j* por unidad de tiempo Y_j y la duración del retraso g_j , como se muestra a continuación:

$$\sum_{j=1}^n Y_j g_j$$

con:

$$g_j = \max(0, t_j^{\text{FIN}} - d_j)$$

y donde d_j es la fecha de entrega comprometida y t_j^{FIN} es el instante en que se finalizan todas las operaciones del *job j*.

Los autores desarrollaron un algoritmo genético para resolver el problema.

- En Corominas *et al.* (2008) se planteó una variante del fJSP en la que se considera una función objetivo que minimiza la suma de los costes asociados a los adelantos E_j y retrasos T_j que se generan con respecto a la fecha de entrega comprometida d_j , y unos costes dependientes del intervalo de realización de las operaciones. De entre estos últimos se hace énfasis en el coste energético, dado que existen talleres que producen *jobs* durante intervalos temporales como el día y la noche (periodos), y la energía que consumen las máquinas está sujeta a la tarifa (coste) de la empresa eléctrica. Por lo general dicho coste es elevado durante el día y disminuye durante la noche; así, en un taller de este tipo puede llegar a ser conveniente, por ejemplo, producir la mayor cantidad de *jobs* durante la noche.

Con referencia al criterio en el que se minimiza la suma de los costes asociados a los adelantos E_j y retrasos T_j , en Corominas *et al.* (2008) se definen como una función lineal para los adelantos y como una función no lineal para los retrasos (que corresponde a un polinomio de segundo grado):

$$E_j = \begin{cases} \delta_j (d_j - t_j^{\text{FIN}}) & \text{si el } \textit{job j} \text{ se adelanta.} \\ 0 & \text{si no.} \end{cases}$$

donde δ_j es un coeficiente que penaliza la función lineal de adelanto del *job j*.

$$T_j = \begin{cases} \beta_j (t_j^{\text{FIN}} - d_j)^2 + \gamma_j (t_j^{\text{FIN}} - d_j) & \text{si hay retraso.} \\ 0 & \text{si no.} \end{cases}$$

donde β_j y γ_j son coeficientes que intervienen en la función cuadrática de retraso del *job j* ($\beta_j > 0$, $\gamma_j > 0$). Se considera que $\gamma_j \geq \delta_j$.

3. Análisis del estado del arte

Como se mencionó anteriormente, la literatura sobre el fJSP es escasa y más aún la literatura sobre problemas en los que se consideran costes en la función objetivo. Aunque en los trabajos de Fatemi Ghomi *et al.* (2005), Alvarez-Valdes *et al.* (2005), Ho

y Tay (2008), Benrejeb *et al.* (2008) y Berkoune y Mesghouni (2008), se tienen en cuenta algunos costes, en ninguno de ellos se considera la función objetivo de Corominas *et al.* (2008), en donde se minimiza la suma de los costes asociados a los adelantos E_j y retrasos T_j que se generan con respecto a la fecha de entrega comprometida d_j , y unos costes dependientes del intervalo de realización de las operaciones.

La función objetivo expuesta en Corominas *et al.* (2008) puede ser el punto de partida de futuras investigaciones con una aplicación en las empresas dedicadas a la manufactura de productos. Además, esta variante del fJSP puede conducir a casos de estudio muy interesantes dado que se encuentra en ámbitos que poco o nada tienen que ver con los talleres mecánicos. Este es el caso de las empresas dedicadas a la prestación de servicios en las que prima otra clase de costes, como el de la mano de obra.

4. Conclusiones

Se ha realizado una revisión y análisis de los artículos del fJSP hasta el año 2011, logrando ofrecer una idea sobre el panorama actual de las funciones objetivo que consideran costes. En el trabajo se ha resaltado la propuesta de función objetivo de Corominas *et al.* (2008), porque constituye un campo de investigación abierto e inexplorado.

Después de revisar los trabajos publicados, se concluye que la mayoría considera funciones objetivo que optimizan el C_{\max} (*makespan*). Sin embargo al analizar las aportaciones recientes en revistas, se observa una tendencia a resolver problemas con funciones objetivo multicriterio, ya que éstas se ajustan de forma adecuada a las necesidades reales de las empresas dedicadas a la manufactura de productos y a la prestación de servicios.

Anexo 1. Terminología

A continuación se expone la nomenclatura utilizada en el documento.

d_j	fecha de entrega comprometida del <i>job j</i> .
e_j	adelanto del <i>job j</i> .
E_j	coste por adelanto del <i>job j</i> respecto a su fecha de entrega comprometida d_j .
g_j	retraso del <i>job j</i> .
h_j	número de operaciones del <i>job j</i> ($h = 1 \dots h_j$)
H	horizonte de producción del taller ($t = 1 \dots H$)
m	número de máquinas ($i = 1 \dots m$)
M	conjunto de todas las máquinas.

M_{jh}	subconjunto de máquinas en que se puede realizar la operación O_{jh} ; $M_{jh} \subseteq M$.
n	número de <i>jobs</i> ($j = 1 \dots n$)
O_{jh}	operación h del <i>job</i> j .
p_{ijh}	tiempo de proceso de la operación O_{jh} , si ésta es ejecutada en la máquina i ($i \in M_{jh}$)
t_j^{FIN}	instante en que se finalizan todas las operaciones del <i>job</i> j .
T_j	coste por retraso del <i>job</i> j respecto a su fecha de entrega comprometida d_j .
X_j	penalización por adelanto del <i>job</i> j .
Y_j	penalización por retraso del <i>job</i> j .
β_j	coeficiente que penaliza el termino no lineal de la función cuadrática de retraso del <i>job</i> j ($\beta_j > 0$)
γ_j	coeficiente que penaliza el termino lineal de la función cuadrática de retraso del <i>job</i> j ($\gamma_j \geq \delta_j$)
δ_j	coeficiente de penalización del adelanto del <i>job</i> j ($\delta_j \geq 0$)

Referencias

- Abraham, A., Choi, O., Liu, H., and Moon, S.H., (2006), “Variable neighborhood particle swarm optimization for multi-objective flexible job-shop scheduling problems”, *Lecture Notes in Computer Science*, 4247, 197–204.
- Alvarez-Valdes, R., Fuertes, A., Giménez, G., Ramos, R., and Tamarit, J.M., (2005), “A heuristic to schedule flexible job-shop in a glass factory”, *European Journal of Operational Research*, 165, 525–534.
- Arkat, J., Fattahi, P., and Jolai, F., (2009), “Flexible job shop scheduling with overlapping in operations”, *Applied Mathematical Modelling*, 33, 3076–3087.
- Armentano, V.A., Laguna, M., and Scrich, C.R., (2004), “Tardiness minimization in a flexible job shop: a tabu search approach”, *Journal of Intelligent Manufacturing*, 15 (1), 103–115.
- Benrejeb, M., Borne, P., Hammadi, S., and Saad, I., (2008), “Choquet integral for criteria aggregation in the flexible job-shop scheduling problems”, *Mathematics and Computers in Simulation*, 76, 447–462.

Berkoune, D., and Mesghouni, K., (2008), “Resolution approach for multi-objective problems with uncertain demands”, *European Journal of Operational Research*, 187 (2), 403–414.

Blazewicz, J., Domschke, W., and Pesch, E., (1996), “The job shop scheduling problem: Conventional and new solution techniques”, *European Journal of Operational Research*, 93, 1–33.

Borne, P., Hammadi, S., and Kacem, I., (2002a), “Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems”, *IEEE Transactions on Systems, Man, and Cybernetics - part c: applications and reviews*, 32 (1), 1–13.

Borne, P., Hammadi, S., and Kacem, I., (2002b), “Pareto-optimality approach for flexible job-shop scheduling problems: Hybridization of evolutionary algorithms and fuzzy logic”. *Mathematics and Computers in Simulation*, 60, 245–276.

Brandimarte, P., (1993), “Routing and scheduling in a flexible job shop by tabu search”. *Annals of Operations Research*, 41, 157–183.

Bruker, P., and Schlie, R., (1990), “Job shop scheduling with multi-purpose machines”. *Computing*, 45, 369–375.

Corominas, A., Gonzalez, N., and Pastor, R., (2008), “Problema de programación de operaciones de flujo general flexible considerando diferentes criterios de optimización”, Instituto de organización y control de sistemas industriales, Working paper IOC-DT-I-2008-01.

Correa, A., Londoño, M., and Rodríguez, E., (2008), “Scheduling for shop settings types to flexible Job Shop: State of the art”. *Revista Avances en Sistemas e Informática*, 5 (3), 151–161.

Dauzère-Pérès, S., and Paulli, J., (1997), “An integrated approach for modeling and solving the general multiprocessor job-shop scheduling problem using tabu search”. *Annals of Operations Research*, 70, 281–306.

De Giovanni, L., and Pezzella, F., (2010), “An Improved Genetic Algorithm for the Distributed and Flexible Job-shop Scheduling problem”, *European Journal of Operational Research*, 200, 395–408.

Fatemi Ghomi, S.M.T., Karimi, B., and Torabi, S.A., (2005), “The common cycle economic lot scheduling in flexible job shops: The finite horizon case”, *International Journal of Production Economics*, 97, 52–65.

Fattahi, P., and Saidi-Mehrabad, M., (2007), “Flexible job shop scheduling with tabu search algorithms”. *International Journal of Advanced Manufacturing Technology*, 32, 563–570.

Fattahi, P., Jolai, F., and Saidi-Mehrabad, M., (2007), “Mathematical modeling and heuristic approaches to flexible job shop scheduling problems”, *Journal of Intelligent Manufacturing*, 18, 331–342.

Gambardella, L.M., and Mastrololli, M., (2002), “Effective neighborhood functions for the flexible job shop problem”, *Journal of Scheduling*, 3 (1), 3–20.

Gao, J., Gen, M., and Sun, L., (2006), “Scheduling jobs and maintenances in flexible job shop with a hybrid genetic algorithm”, *Journal of Intelligent Manufacturing*, 17 (4), 493–507.

Gao, L., Li, P., Shao, X., and Zhang, G., (2009), “An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem”, *Computers & Industrial Engineering*, 56, 1309–1318.

Garey, M.R., Johnson, D.S., and Sethi, R., (1976), “The complexity of flowshop and jobshop scheduling”, *Mathematics of Operations Research*, 1, 117–129.

Ham, M., Lee, Y.H., and Kim, S.H., (2011), “Real-time scheduling of multi-stage flexible job shop floor”, *International Journal of Production Research*, 49 (12), 3715–3730.

Ho, N.B., and Tay, J.C., (2008), “Evolving dispatching rules using genetic programming for solving multi-objective flexible job-shop problems”, *Computers & Industrial Engineering*, 54, 453–473

Hurink, E., Jurisch, B., and Thole, M., (1994), “Tabu search for the job shop scheduling problem with multi-purpose machines”, *Operations Research Spektrum*, 15, 205–215.

Jain, A.S., and Meeran, S., (1999), “Deterministic job-shop scheduling: Past, present and future”, *European Journal of Operational Research*, 113, 390–434.

Low, C., Wu, T., and Yip, Y., (2006), “Modelling and heuristics of FMS scheduling with multiple objectives”, *Computers & Operations Research*, 33, 674–694.

Mati, Y., Lahlou, C., and Dauzère-Pérès, S., (2011), “Modelling and solving a practical flexible job-shop scheduling problem with blocking constraints”, *International Journal of Production Research*, 49(8), 2169–2182.

Pinedo, M., (2005), *Planning and Scheduling in Manufacturing and Services*, Springer.

Rajkumar, M., Asokan, P., Anilkumar, N., and Page, T., (2011), “A GRASP algorithm for flexible job-shop scheduling problem with limited resource constraints”, *International Journal of Production Research*, 49(8), 2409–2423.

Wu, Z., and Xia, W., (2005), “An effective hybrid optimization approach for multi objective flexible job-shop scheduling problems”. *Computers & Industrial Engineering*, 48, 409–425.