

**COMPARACIÓN DE LA EFICIENCIA DE LAS METAHEURÍSTICAS, RECOCIDO
SIMULADO Y BÚSQUEDA TABÚ EN LA SOLUCIÓN DEL PROBLEMA DE
PLANEACIÓN DE PRODUCCIÓN TEMPRANA / TARDANZA
(*EARLINESS/TARDINESS PRODUCTION*)**

RUBÉN DARÍO OSMA GARCIA

**UNIVERSIDAD AUTONOMA DE BUCARAMANGA
FACULTAD DE INGENIERIA DE SISTEMAS
BUCARAMANGA**

2006

**COMPARACIÓN DE LA EFICIENCIA DE LAS METAHEURÍSTICAS, RECOCIDO
SIMULADO Y BÚSQUEDA TABÚ EN LA SOLUCIÓN DEL PROBLEMA DE
PLANEACIÓN DE PRODUCCIÓN TEMPRANA / TARDANZA
(*EARLINESS/TARDINESS PRODUCTION*)**

RUBÉN DARÍO OSMA GARCIA

PHD. ALINA FEDOSSOVA

Directora

ING. JULIAN SANTOYO DIAZ

Asesor

UNIVERSIDAD AUTONOMA DE BUCARAMANGA

FACULTAD DE INGENIERIA DE SISTEMAS

BUCARAMANGA

2006

Nota de Aceptación

Presidente del jurado

Jurado

Jurado

Bucaramanga, 25 de febrero de 2008

CONTENIDO

	pág.
INTRODUCCIÓN	9
1. MODELO MATEMÁTICO DEL PROBLEMA DE PLANEACIÓN DE PRODUCCIÓN TEMPRANA / TARDANZA (EARLINESS/TARDINESS PRODUCTION.	11
2. RECOCIDO SIMULADO	16
3. BÚSQUEDA TABÚ	17
4. SOLUCIÓN INICIAL POR HEURÍSTICA	18
5. AJUSTE DE SECUENCIA DE X_i ($X_i = 1, 2, \dots, N$) POR RECOCIDO SIMULADO	20
6. APLICACIÓN	22

7. CONCLUSIONES

25

REFERENCIAS

27

LISTA DE TABLAS

	Pág.
Tabla 1. Valores Iniciales	22
Tabla 2. Resultados (Primera corrida)	23
Tabla 3. Resultados (Segunda Corrida)	24

LISTA DE FIGURAS

pág

Figura 1. Grafica de comparación de resultados obtenidos por las metaheurísticas donde $0 < L_i \leq X_i \leq T, i = 1, 2, \dots, n,$

25

RESUMEN

Este artículo considera la búsqueda de soluciones apropiadas, aplicadas al modelo de planeación de producción temprana/tardanza (*Earliness/Tardiness Production*), tomando como referencia el uso de las metaheurísticas Recocido Simulado y Búsqueda Tabú, lo que convierte este trabajo en innovador y de importante aporte para las ciencias.

Palabras Clave: Metaheurísticas, Optimización Combinatoria, Programación Semi-infinita, Recocido Simulado, Complejidad Computacional, Búsqueda Tabú.

INTRODUCCIÓN

Desde que el JIT (*Just In Time*) fue desarrollado en Japón en los años 70s, esta filosofía ha sido aceptada por la comunidad industrial, porque significa una reducción en costos de producción e inventario. Desde entonces los requerimientos de recursos del mundo real son dependientes del tiempo, no lineales y no convexos. Por esta razón se plantea la implementación del modelo de planeación de "Producción Temprana / Tardanza" (*Earliness / Tardiness Production*) como un problema de optimización semi-infinita, sujeto a un conjunto infinito de restricciones¹.

Una alternativa de solución a este tipo de problemas fue la implementación del algoritmo usado para simular el enfriamiento de material, un proceso llamado Recocido Simulado. Los algoritmos tradicionales de búsqueda local parten de una solución inicial que de modo paulatino se transforma en otras y a su vez son mejoradas al introducirles pequeñas perturbaciones o cambios (tales como cambiar el valor de una variable o intercambiar los valores que tienen dos variables)².

Es la Búsqueda Tabú (BT), otra técnica para encontrar soluciones cercanas al óptimo en problemas de optimización combinatoria, que se basa en los principios

¹ WANG, D. y Fang, SC, A semi-infinite programming model for earliness/tardiness production planning with a Genetic Algorithm. 1997.

² DOWSLAND A. Kathryn. Y Daz, Belarmino Adenso, Diseño de Heurísticas y fundamentos del Recocido Simulado. 2001.

generales de Inteligencia Artificial (IA). La filosofía de la BT, maneja y explota una colección de principios para resolver problemas de manera inteligente. Como es el caso la memoria flexible, que envuelve el proceso dual de crear y explotar estructuras para tomar ventaja mediante la combinación de actividades de adquisición, evaluación y mejoramiento de la información de manera histórica³.

El objetivo es comparar la eficiencia de las metaheurísticas en la solución del problema de planeación de Producción Temprana / Tardanza (*Earliness / Tardiness Production*).

³ MELIAN, Belen, Moreno A. Jose y Moreno Vega J. Marcos, Metaheuristics: A global view. 2003.

1. MODELO MATEMÁTICO DEL PROBLEMA DE PLANEACIÓN DE PRODUCCIÓN TEMPRANA / TARDANZA (EARLINESS/TARDINESS PRODUCTION)^{4,5}.

Se reciben n ordenes en el periodo de planeación $[0, T]$. Para cada orden i , el ciclo de producción es L_i y una fecha de entrega acordada d_i . Los requerimientos de recursos para cada orden i se representan a través de una función del tiempo de finalización de la producción, permitiendo a x_i , que es el vector de ordenes i , ser cercano a t (tiempo de realización esperado para cada orden i). La función de requerimientos de recursos para cada orden i , es denotada por $R_i(t, x_i)$, para cada $t \in [0, T]$.

Se define $G(t)$ como la cantidad de recurso disponible en el tiempo t . En el ambiente JIT, una decisión es tomada deseando que el tiempo de finalización de la producción este cerca de la fecha de entrega acordada. Usando una función de penalización cuadrática para el modelo de planeación de Producción Temprana / Tardanza, se puede formular el problema como un problema de programación semi-infinita (SIP).

$$\min_x \sum_{i=1}^n a_i(x_i - d_i)^2, \quad (1)$$

⁴ WANG, D. y Fang, SC, A semi-infinite programming model for earliness/tardiness production planning with a Genetic Algorithm. 1997.

⁵ LI Y.; Wang D., A semi-infinite programming model for earliness/tardiness production planning with Simulated Annealing. 1997.

s.a

$$s.t \sum_{i=1}^n R_i(t, x_i) \leq G(t), \quad t \in [0, T] \quad (2)$$

$$0 < L_i \leq x_i \leq T, \quad i = 1, 2, \dots, n, \quad (3)$$

donde a_i es el peso para la orden i que es usualmente proporcional a la importancia de cada orden i . El modelo de programación semi-infinita tiene n variables y un conjunto infinito de restricciones. Esto permite describir el problema en un dominio de tiempo continuo, en cambio del caso de un modelo discreto.

Un ciclo de producción, usualmente inicia con una etapa de preparación de la producción, seguido por una etapa de procesos de manufactura y finalizada por una etapa de ensamble. Los requerimientos de recursos son usualmente altos para la etapa de procesamiento y bajos para las etapas de preparación y ensamble.

$$R_i(t, x_i) = a_i \exp \left[\frac{-(t - x_i + b_i)^2}{c_i} \right], \quad i = 1, 2, \dots, n, \quad (4)$$

donde

$$a_i = \frac{P_i}{\sqrt{2\pi L_i/4}}, \quad (5)$$

$$b_i = \frac{L_i}{2}, \quad (6)$$

$$c_i = \frac{L_i^2}{8}, \quad (7)$$

L_i es el ciclo de producción para cada orden i . Cuando el tiempo de realización $x_i = L_i$, $R_i(t, L_i)$ debe satisfacer la siguiente condición:

$$\int_0^{L_i} R_i(t, L_i) dt = \delta P_i, \quad i = 1, 2, \dots, n, \quad (8)$$

donde P_i es el total de requerimiento de recurso para la orden i y δ es el coeficiente de relajación.

Se asume que la cantidad de recurso disponible en el inicio es g_0 , el cual se incrementa con una tasa β en el tiempo. Suponemos que hay m actividades de mantenimiento programadas en el intervalo de tiempo $[0, T]$. Cada actividad requiere de algún recurso acorde a la siguiente formula de aproximación:

$$Q_j(t) = q_j \exp \left[\frac{-(t - g_j)^2}{h_j} \right], \quad t \in [0, T] \quad (9)$$

$$j = 1, 2, \dots, m,$$

donde los parámetros q_j , g_j , y h_j son especificados de la misma forma que a_i , b_i , y c_i .

La red de recursos disponibles será:

$$G(t) = g_o \exp(\beta t) - \sum_{j=1}^m Q_j(t), \quad t \in [0, T]. \quad (10)$$

El modelo de programación semi-infinita tiene un dominio factible no vacío cuando se cumplen las siguientes condiciones:

$$G(t) > \max \{a_i \mid i = 1, 2, \dots, n\}, \quad t \in [0, T] \quad (11)$$

$$\int_0^T G(t) dt \geq \sum_{i=1}^n P_i. \quad (12)$$

Una vez que la función de requerimiento de recursos $R_i(t, x_i)$, y el recurso disponible $G(t)$ son determinados, el trabajo restante es resolver el problema de programación semi-infinita presentado.

La función $R_i(t, x_i)$ ($i = 1, 2, \dots, n$) tiene forma de campana, prueba de que la mayoría de los n locales están situados al lado izquierdo de la ecuación (2) y los máximos están localizados en los rangos de $[x_i - L_i, x_i]$, $i = 1, 2, \dots, n$.

$$t_j^* = \arg \max_t \left\{ \sum_{i=1}^n R_i(t, x_i) - G(t) \mid x_j - L_j < t < x_j \right\} \quad (13)$$

y

$$\Phi(t_j^*, X) = \sum_{i=1}^n R_i(t_j^*, x_i) - G(t_j^*), \quad j = 1, 2, \dots, n. \quad (14)$$

Cuando $\Phi(t_j^*, X) > 0$, significa una posible violación de las restricciones de SIP en t_j^* . Si una tolerancia aceptable es $\epsilon > 0$ para cada restricción de recursos, el total de violaciones puede ser definido como:

$$V = \{t_j^* \mid j = 1, 2, \dots, n, \Phi(t_j^*, X) > \epsilon\} \quad (15)$$

Luego se obtiene una nueva función objetivo:

$$\min_x \sum_{i=1}^n a_i(x_i - d_i)^2 + \mu \sum_{t_j^* \in V} \Phi(t_j^*, X), \quad (16)$$

donde μ es un multiplicador muy grande.

2. RECOCIDO SIMULADO

Desde que Kirkpatrick et al. (1983) introdujeron el concepto de Recocido Simulado, esta metaheurística ha demostrado ser una herramienta muy exitosa para resolver una amplia gama de problemas de optimización combinatoria. El Recocido Simulado es una variante de la búsqueda local, que permite movimientos ascendentes para evitar quedar atrapado prematuramente en un óptimo local.

El nombre viene de la idea de un algoritmo diseñado en los años 50 para simular el enfriamiento de materiales (un proceso llamado "recocido") [3]. El algoritmo de Recocido Simulado generalmente comienza con una solución inicial y una temperatura alta que descenderá gradualmente. En cada temperatura, algunas transiciones son hechas logrando un equilibrio. En cada temperatura, la mayoría de los movimientos son aceptados, porque el algoritmo puede aceptar un estado de solución con un valor deteriorado de la función objetivo con probabilidad $\exp(-\Delta f / T k)$, si $\Delta f > 0$, esto facilita salir de un óptimo local. Cuando la temperatura inicial es bastante alta, la temperatura de parada es bastante baja y el proceso de descenso es bastante lento, el algoritmo converge al óptimo local, con una probabilidad de 1.

La solución del modelo de programación semi-infinita por Recocido Simulado se hace en tres etapas. En la primera se obtiene una solución inicial por heurística, y luego una secuencia óptima de competición del tiempo x_i como la solución inicial en la siguiente etapa, en la última, se obtiene una solución cercana al óptimo y factible⁶.

⁶ DOWSLAND A. Kathryn. Y Daz, Belarmino Adenso, Diseño de Heurísticas y fundamentos del Recocido Simulado. 2001..

3. BÚSQUEDA TABÚ

En términos generales el método de Búsqueda Tabú se describe de la siguiente forma: se basa en un procedimiento iterativo "método de búsqueda local" para encontrar dentro de un conjunto finito X de soluciones factibles una solución S perteneciente a X , la cual minimice la función objetivo f . Para evitar que la búsqueda quede atrapada en ciclos, el procedimiento cuenta con una estructura, lista tabú L de longitud l (fija ó variable) que almacena los últimos movimientos realizados con el fin de prevenir que se visiten nuevamente soluciones encontradas en las ultimas l iteraciones. La diferencia con las técnicas de mejoramiento local radica en el hecho de que se pueden realizar movimientos desde la solución S hacia una peor S^* ⁷.

⁷ MELIAN, Belen, Moreno A. Jose y Moreno Vega J. Marcos, Metaheuristics: A global view. 2003.

4. SOLUCIÓN INICIAL POR HEURÍSTICA

Si una solución inicial es generada aleatoriamente, el proceso de recocido en temperaturas altas es largo. Para disminuir el tiempo computacional, en vez del recocido a temperaturas altas, en el se propone obtener una mejor solución inicial⁸. De acuerdo a la experiencia pasada, para disminuir la penalización general, el modelo de producción temprana / tardanza deberá reducir el tiempo tanto como sea posible para las ordenes con una prioridad a_i muy grande y un menor P_i . Se obtiene una secuencia de los valores de a_i / P_i ($i = 1, 2, \dots, n$). Las ordenes con un gran a_i / P_i son consideradas de primero.

El algoritmo es el siguiente⁹:

⁸ LI Y.; Wang D., A semi-infinite programming model for earliness/tardiness production planning with Simulated Annealing. 1997.

⁹ ibid. referencia 8.

PASO 1. Calcular el horizonte de planeación T .

$$T = \max\left\{\frac{\eta(\sum_{i=1}^n P_i + \sum_{j=1}^m q_j)}{g_o}, \max[d_i \mid i = 1, 2, \dots, n]\right\},$$

donde η es un coeficiente de relajación, para instancia, $\eta = 1,30$.

PASO 2. Calcular $u_i = a_i/P_i$, $i = 1, 2, \dots, n$, para la secuencia u_i de grande a pequeño: u_1, u_2, \dots, u_n .

PASO 3. Según la secuencia u_1, u_2, \dots, u_n , calcular los tiempos de completación x_j , ($j = 1, 2, \dots, m$) de cada orden.

3.1 Hacer $x_j = d_j$, si $\Phi(t_j^*, X) > 0$ ($t_j^* \in [d_j - L_j, d_j]$).

3.2 Calcular la penalización sobre tiempo temprano.

Para $k = 1, 2, \dots, d_j - L_j$, hacer $y_j = d_j - k$ hasta $\Phi(t_j^*, Y) < 0$.

La penalización de tiempo temprano es $a = a_j (d_j - y_j)^2$, sino, hacer $a = M$ (M es un número muy grande).

3.3 Calcular la penalización de tiempo tardío.

Para $k = 1, 2, \dots, T - d_j$, hacer $z_j = d_j + k$ hasta $\Phi(t_j^*, Z) < 0$.

La penalización de tiempo tardío es $b = a_j (z_j - d_j)^2$, sino, hacer $b = M$ (M es un número muy grande).

3.4 Obtener x_j acorde a los valores de a y b :

Si $a < b$, $x_j = y_j$, sino $x_j = z_j$.

Si $a = b = M$, ajustar k o η .

Del procedimiento anterior se obtiene $X = (x_1, x_2, \dots, x_n)$. Estos funcionan como la solución inicial en la segunda etapa del Recocido Simulado.

5. AJUSTE DE SECUENCIA DE X_i ($X_i = 1, 2, \dots, N$) POR RECOCIDO SIMULADO

Inicialmente se hace una simple mejora por Recocido Simulado y esta se almacena como mejor solución. Puesto que el Recocido Simulado acepta las soluciones que deterioran el valor de la función objetivo, es posible que la solución final sea peor que la solución inicial. Las reglas de descenso en temperatura y en las condiciones de equilibrio implicado en el Recocido Simulado se definen como:

- **La regla de descenso en temperatura:** Multiplicar T_k por un número γ cerca a 1 para actualizar T .
- **Las condiciones de equilibrio:** En cada temperatura, el número de transiciones es igual al tamaño del vecindario. En el Recocido Simulado, la solución en el vecindario de la última solución es el resultado entrante del intercambio de x_i y x_j . El intercambio se debería hacer C_n^2 veces en el último para lograr un equilibrio en cada temperatura, $M = C_n^2 = n(n - 1)/2$.

El algoritmo de Recocido Simulado es el siguiente:

PASO 1. Obtener la solución inicial $X = (x_1, x_2, \dots, x_n)$ de la heurística.

Calcular la función objetivo fX_0 .

X_0 es guardada como la mejor solución.

PASO 2. Asignar una temperatura inicial T_0 y una temperatura de parada T_f .

Hacer $k = 0, T_k = T_0$.

PASO 3. Generar una nueva solución X_1 (intercambiar X_i y X_j aleatoriamente).

Hacer $k = k + 1$.

Calcular $\Delta f = f(X_1) - f(X_0)$.

PASO 4. SI $\Delta f < 0$, hacer $X_0 = X_1$.

Actualizar la mejor solución.

Ir al paso 6.

PASO 5. Generar $X_i \in U(0, 1)$. Si $\exp(-\Delta f/T_k) > \xi$, hacer $X_0 = X_1$.

PASO 6. SI $k = M$, hacer $k = 0$ e ir al paso 7, sino, ir al paso 3.

PASO 7. Hacer $T_k = \Upsilon T_k$.

PASO 8. Si $T_k \leq T_f$ parar, sino, ir al paso 3.

En el paso 3, después del intercambio de X_i y X_j , la función objetivo deberá ser muy grande cuando la secuencia es la mejor (pero no la mejor solución).

6. APLICACIÓN

Para el ejemplo numérico se consideró el caso de una compañía de construcción que recibe 10 órdenes de trabajo. La restricción de recurso disponible está en función de la mano de obra. El nivel actual de recurso es 100 kilo-horas/semana, con una tasa de incremento de $\beta = 0,005$ pronosticado. En la tabla 1 se observan los valores iniciales.

Tabla 1. Valores Iniciales

Número de Orden i	Ciclo de Producción L_i	Requerimientos de Recursos P_i	Precio del Contrato A_i	Fecha Deseada de Entrega D_i
1	20	400	10	25
2	20	900	18	30
3	30	800	12	35
4	40	800	15	40
5	25	1000	28	40
6	20	1200	20	40
7	50	2000	30	50
8	10	300	18	15
9	20	400	9	50
10	60	1500	30	60

Fuente: Autor del proyecto.

Para el ejemplo anterior los resultados se muestran en las tablas 2 y 3. Para cada orden i con una fecha de entrega deseada D_i , el tiempo de completación estimado según las metaheurísticas de Búsqueda Tabú, Recocido Simulado y Algoritmos Genéticos será x_i , respectivamente.

Tabla 2. Resultados (Primera corrida)

Número de Orden i	Fecha de Entrega Deseada D_i	Tiempo de Completación (X_i)		
		Búsqueda Tabú	Recocido Simulado	Algoritmos Genéticos*
1	25	24.3	23.6	30.74
2	30	28.0	28.8	20.76
3	35	32.8	33.2	41.88
4	40	38.7	38.6	40.49
5	40	38.3	38.9	52.50
6	40	38.4	38.6	86.77
7	50	48.7	49.2	84.38
8	15	13.4	14.2	10.00
9	50	48.5	49.3	75.75
10	60	58.7	58.7	87.28
Valor de la función objetivo		4.9000	4.4100	9.3615
* Wang,D. y Fang, SC, A semi-infinite programming model for earliness/tardiness production planning with a Genetic Algorithm. 1997.				

Fuente: Autor del proyecto.

Tabla 3. Resultados (Segunda Corrida)

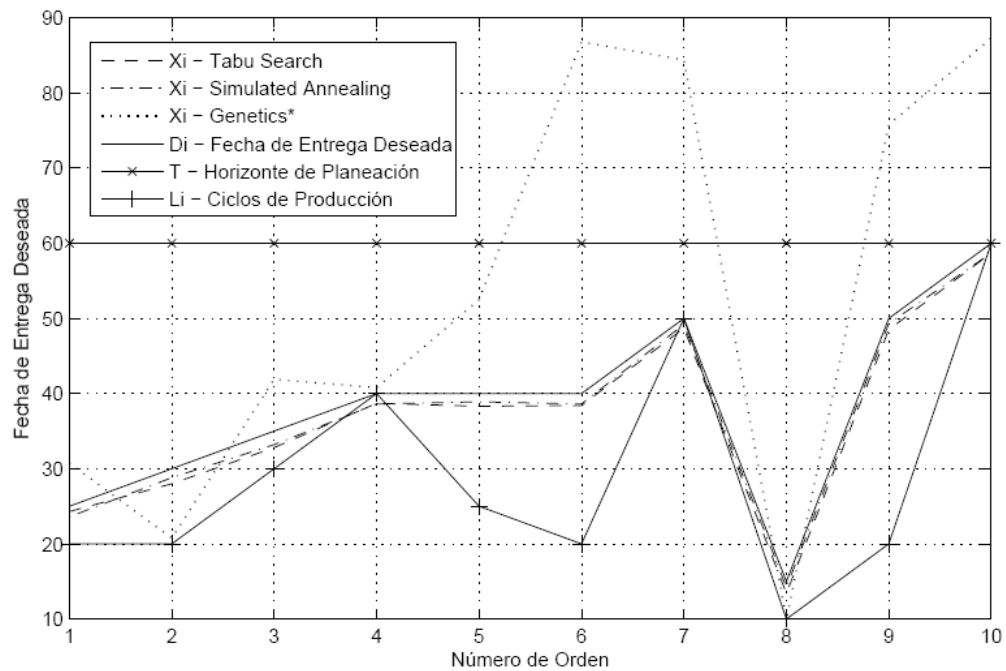
Número de Orden <i>i</i>	Fecha de Entrega Deseada <i>D_i</i>	Tiempo de Completación (<i>X_i</i>)	
		Búsqueda Tabú	Recocido Simulado
1	25	24.3	23.6
2	30	28.0	28.8
3	35	32.8	33.2
4	40	38.7	38.6
5	40	38.3	38.9
6	40	38.4	38.6
7	50	48.7	49.2
8	15	13.4	14.2
9	50	48.5	49.3
10	60	58.7	58.7
Valor de la función objetivo		3.2400	5.4000

Fuente: Autor del proyecto.

En el Recocido Simulado, cuando los parámetros Υ y M son muy pequeños, el algoritmo converge a diferentes soluciones locales. Cuando Υ o M son suficientemente grandes, el algoritmo converge a una solución global.

7. CONCLUSIONES

Figura 1: Grafica de comparación de resultados obtenidos por las metaheurísticas donde $0 < L_i \leq X_i \leq T$, $i = 1, 2, \dots, n$.



Fuente: Autor del proyecto.

El tipo de metaheurística se establece en función del tipo de procedimientos a los que se refiere. Para problemas con un conjunto infinito de restricciones, estas necesitan ser combinadas con la función objetivo. Para esto se empleó una función de penalización.

Para resolver el modelo de programación semi-infinita, la metaheurística de Recocido Simulado y Búsqueda Tabú, deben combinarse con una heurística que genere la solución inicial.

Las metaheurísticas Recocido Simulado y Búsqueda Tabú, presentan una complejidad computacional de orden $O(n^2)$, donde n es el tamaño de la entrada. Para mejoras en los algoritmos se podrían implementar en la Búsqueda Tabú funciones *hash* que optimicen la búsqueda de nuevas soluciones. En el Recocido Simulado, implementar vecindarios, a fin de acotar los intervalos de búsqueda para los nuevos intercambios.

REFERENCIAS

BAASE, Sara. Computer Algorithms, Introduction to Design and Analysis. Addison-Wesley Series in Computer Science, 1998.

CERROLAZA Miguel y Annicchiarico William. Algoritmos de optimización estructural basados en simulación genética. Universidad Central de Venezuela, 1996.

DIMOPOULOS, C. and Zalzalá, A. M. A. Recent developments in evolutionary computation for manufacturing optimisation: problems, solutions and comparisons. IEEE Trans. Evolutionary Computation, 2000, 4(2), 93-113.

DOWSLAND A. Kathryn. Y Daz, Belarmino Adenso, Diseño de Heurísticas y fundamentos del Recocido Simulado. 2001

F. GLOVER, Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*. 1986.

HUNG, Y.-F., Shih, C.-C. and Chen, C.-P. Evolutionary algorithms for production planning problems with setup decisions. J. Opl Res. Soc., 1999, 50(8), 857-866.

LI, Y. and Man, K. F. Scheduling and planning problem in manufacturing systems with multiobjective genetic algorithm. In Proceedings of the 24th Annual Conference of IEEE (IECON '98), 1998, pp. 274-279 (Industrial Electronics Society).

LI Y.; Wang D., A semi-infinite programming model for earliness/tardiness production planning with Simulated Annealing. 1997.

Melian, Belen, Moreno A. Jose y Moreno Vega J. Marcos, Metaheuristics: A global view. 2003.

OGRADY, P.J. JUST IN TIME, Una estrategia fundamental para los jefes de producción, 1992.

THOMAS H. Cormen, Charles E. Leiserson and Ronald L. Rivest. Introduction to Algorithms. Faculty of the Electrical Engineering and Computer Science Department at the Massachusetts Institute of Technology 1996.

WANG, D. y Fang, SC, A semi-infinite programming model for earliness/tardiness production planning with a Genetic Algorithm. 1997.

WU, Y., Liu, M. and Wu, C. A genetic algorithm for optimising the MPS of a processing-assembly production line with identical machines. In Proceedings of the International Conference on Machine Learning and Cybernetics, Beijing, PR China, 2002, pp. 1172-1177.