

Programación de la Producción con tiempos de preparación variables mediante Recocido Simulado y Búsqueda Tabú

Mayra D'Armas

UNEXPO, Puerto Ordaz, Venezuela, mdarmas@bqto.unexpo.edu.ve

This work considers the problem of the scheduling in a single machine, with families of products, where the setup times are separated of the processing times, and are sequence depended, with the objective to minimize the total tardiness. The metaheuristics Simulated Annealing and Tabu Search were developed and its behavior was evaluated to solve the problem. The proposed algorithms were codified in Visual Basic 6.0. The computational experimentation was studied in a collection of 1000 instances with a number of families from 4 to 6 and with a number of jobs from 15 to 25 of the Laboratory of Industrial Organization of the Department of Management of the Technical University of Catalonia. Computational results reveal that Tabu Search is a procedure that can provide good solutions for the specific studied problem, when the total tardiness with sequence-dependent setup times is considered as the objective. The research presented can be taken as a base for possible extensions focused to consider other variations of the problem under study, incorporating elements which sometimes also appear in real industrial environments.

INTRODUCCIÓN

En los ambientes industriales, es común que se lleven a cabo diversos tipos de operaciones que requieren la preparación de las máquinas y los procesos. Sin embargo, muchas de las investigaciones en el área de la programación de operaciones, consideran que cuando una máquina cambia de un trabajo a otro no hay tiempos de preparación (D'Armas y Companys, 2005). Según (Allahverdi, 2000) esta suposición no es válida para el caso de tiempos de preparación dependientes de la secuencia, y en el caso de tiempos de preparación independientes de la secuencia puede conducir a un incremento de los tiempos muertos en algunas de las máquinas.

FORMULACIÓN DEL PROBLEMA

Este trabajo se enmarca dentro de la problemática de la programación de la producción, en un ambiente de una máquina, con familias de productos y tiempos de preparación dependientes de la secuencia. La notación usada es la siguiente:

n : número de piezas

i : índice de las piezas; $i \in I = \{1, 2, 3, \dots, n\}$

p_i : tiempo de operación de la pieza i

g_i : familia a la que pertenece la pieza i

ST_{hi} : tiempo de preparación de la pieza i cuando h ha sido la pieza anterior de la secuencia

d_i : instante comprometido de salida de la pieza i

r_i : instante de entrada de la pieza i en el taller

c_i : instante en que la pieza sale del taller

L_i : diferencia entre el instante de salida real y el previsto

T_i : retraso

Se tiene un conjunto de piezas que deben ser secuenciadas en una máquina. Se asume que la máquina puede procesar una sola pieza a la vez y que está

disponible en el instante cero. Las piezas están disponibles al inicio del proceso y tienen un tiempo límite de entrega. Se asume que las piezas están clasificadas en familias y que el tiempo de preparación se produce cuando se pasa de una familia a otra. El tiempo de preparación de la máquina depende de la familia de la pieza a ser procesada y la familia de la pieza que la precede. Para cada pieza i se conoce el tiempo de operación; el tiempo de entrega comprometida, la familia a la que pertenece dicha pieza y los tiempos de preparación de la máquina ST al pasar de una familia a otra. Según la secuencia escogida, para cada pieza i el instante en que la pieza sale del taller c_i , viene dado por: $c_i = r_i + w_i + p_i + ST_{hi}$ y su retraso T_i viene dado por: $T_i = \max\{0, c_i - d_i\}$

El objetivo es encontrar una secuencia de las piezas que minimice la suma del retraso de las piezas: $[MIN] Z = \sum T_i$

METODOLOGÍA

Las metaheurísticas se codificaron en Visual Basic 6.0 y en la experiencia computacional se probaron 1000 ejemplares con un número de familias entre 4-6 y con un número de piezas entre 15-25.

REGLAS DE SECUENCIACIÓN

Para resolver el problema se aplicaron cuatro reglas que permitieron generar soluciones iniciales, a saber:

EDD (earliest due date). Ordena las piezas, en orden creciente, de acuerdo con la fecha de vencimiento.

SST-EDD (shortest setup time-earliest due date). Ordenar las piezas por familias de acuerdo con el tiempo de preparación más corto cuando se cambia de una familia a otra, y, además, secuencia las piezas entre familia por orden creciente de fechas de vencimiento.

Índice CRI(Cociente). Asigna las piezas tomando en cuenta el índice de prioridad que se calcula por medio de la fecha de vencimiento dividida por la suma entre el

tiempo de preparación de la familia de la pieza i y el tiempo de procesamiento de la pieza i .

Índice CR2 (Suma ponderada). Asigna las piezas tomando en cuenta el índice de prioridad para cada pieza i , el cual pondera la fecha de vencimiento, el tiempo de preparación de la familia de la pieza i y el tiempo de procesamiento de la pieza i .

METAHEURÍSTICAS

Recocido Simulado (SA). Fue desarrollado por Kirkpatrick et al (1983). Para la implementación del recocido se tomaron los valores siguientes: (a) Temperatura inicial $T = 0,4$; (b) Tasa de disminución de la temperatura o tasa de enfriamiento $r = 0,95$; y (c) Size factor = 16. La generación de los vecinos se realizó generando aleatoriamente dos piezas a permutar en la secuencia dando lugar a una nueva secuencia vecina.

Búsqueda Tabú (TS). Es un procedimiento desarrollado en su forma actual por Fred Glover (1989). Para la implementación del TS se generaron los vecinos mediante la permutación entre dos piezas haciéndose todos los cambios posibles de dos posiciones, la lista tabú se tomó proporcional a la dimensión del vecindario y como nivel de aspiración se seleccionaron las secuencias cuyo valor de $\sum T <$ mejor solución hallada hasta el momento.

RESULTADOS

La tabla 1 resume los resultados experimentales de los 1000 ejemplares en términos de cantidad de mejores y peores soluciones. Se puede observar que, para todos los casos estudiados, la mayoría de las mejores soluciones se obtienen con TS.

Tabla 1: Cantidad de mejores y peores soluciones.

Ejemplares:	n = 15, b = 4		n = 20, b = 4	
Cantidad de soluciones:	peores	mejores*	peores	mejores*
SA	196	4	199	1
TS	2	198	0	200
	* 2 empates binarios		* 1 empate binario	
Ejemplares:	n = 25, b = 4		n = 20, b = 5	
Cantidad de soluciones:	peores	mejores	peores	mejores
SA	200	0	100	0
TS	0	200	0	100
Ejemplares:	n = 25, b = 5		n = 25, b = 6	
Cantidad de soluciones:	peores	mejores*	peores	mejores*
SA	100	0	200	0
TS	0	100	0	200

En la figura 1 se muestran gráficamente los valores del retraso medio de SA y TS en cada una de las colecciones de problemas estudiados.

Para juzgar la eficacia de los algoritmos se computó el tiempo promedio para resolver un ejemplar mediante SA y TS. Los resultados se presentan en la tabla 2, donde se puede observar que el tiempo promedio de ejecución del SA es superior al de TS, sin embargo las diferencias no son muy grandes, con un tiempo promedio máximo que no supera los 5 segundos.

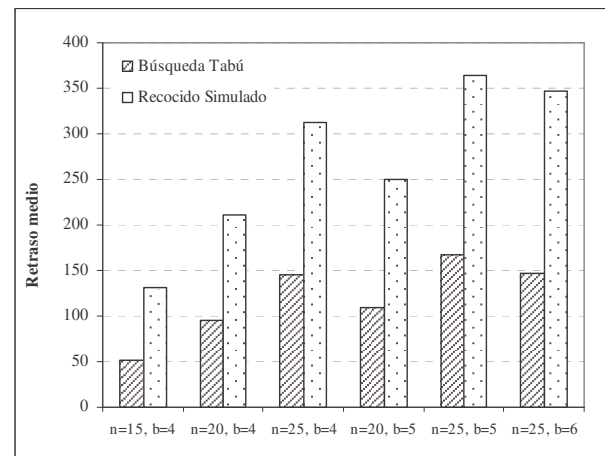


Figura 1: Retraso promedio.

Tabla 2: Tiempo promedio de ejecución en segundos.

	n=15, b=4	n=20, b=4	n=25, b=4	n=20, b=5	n=25, b=5	n=25, b=6
SA	1,0648	2,4884	4,9190	2,6042	4,9190	4,9190
TS	0,9549	2,3148	4,3403	2,0255	4,3403	4,3403

Para juzgar la calidad la solución heurística se comparó en términos del porcentaje de error relativo, de acuerdo con lo propuesto (Rajendran y Ziegler, 2003). De los resultados presentados en la tabla 3, se deduce que TS es la que ofrece menor dispersión de los resultados.

Tabla 3: Promedio de error relativo

	n=15, b=4	n=20, b=4	n=25, b=4	n=20, b=5	n=25, b=5	n=25, b=6
SA	2371,0	1748,4	1370,0	2118,3	1335,6	3007,0
TS	0,1	0,0	0,0	0,0	0,0	0,0

Finalmente, se concluye que TS es el más adecuado para el problema específico estudiado ya que necesita poco tiempo de procesamiento y logra las mejores soluciones

REFERENCIAS

- Allahverdi A (2000). "Minimizing mean flowtime in a two-machine flowshop with sequence-independent setup times", *Computers & Operations Research*, Vol. 27, No. 2, pp 111-127.
- D'Armas M. y Companys R. (2005). "Programación de operaciones con tiempo de preparación mediante algoritmos de optimización local". *Universidad, Ciencia y Tecnología*, Vol. 9, No. 35, pp 155-162.
- Glover F. (1989). "Tabu Search, Part I". *ORSA Journal on Computing*. Vol. 1, pp. 190-206.
- Kirkpatrick S. et al (1983). "Optimization by simulated annealing", *Science*, Vol. 220, pp 661-680.
- Rajendran C y Ziegler H. (2003). "Scheduling to minimize the sum of weighted flowtime and weighted tardiness of jobs in a flowshop with sequence-dependent setup times". *European Journal of Operational Research*. Vol. 149, pp. 513-522.