

# Comparación de cuatro métodos para minimizar el retraso total en una máquina con tiempos de preparación variables

Mayra D'Armas

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

*The present work can be located in the frame of production scheduling and pretends to evaluate, by means of computational tests, the behavior of the metaheuristics Annealing Simulated, Genetic Algorithms, Tabu Search and GRASP to solve a scheduling problem. The dealt problem considers a single machine with the objective to minimize the total tardiness in an atmosphere with products families and sequence-dependent setup times. The metaheuristics were codified in Visual Basic 6.0, and the computational experience was made with a collection of 1000 instances with a number of families from 4 to 6 and with a number of jobs from 15 to 25. The results demonstrate that Tabu Search can provide good solutions for the specific problem studied when considering the total tardiness objective with sequence dependent setup times, since it needs a short run time and the best solutions are achieved. GRASP can have an advantage on TS for problems with great amount of orders because it needs less time to find a good solution. These conclusions are based on these specific algorithms in a specific set of test problems.*

## INTRODUCCIÓN

El presente trabajo tiene por objetivo evaluar, mediante pruebas computacionales, el comportamiento de las metaheurísticas Recocido Simulado, Algoritmos Genéticos, Búsqueda Tabú y GRASP para resolver el problema de secuenciación de operaciones. El problema investigado es el de encontrar una secuencia, en un ambiente de una máquina, con familias de productos, y donde los tiempos de preparación de la máquina dependen de la secuencia de las familias de productos a fabricar, de forma tal que se cumplan con los tiempos de entrega, o en su defecto, se minimice la suma de retrasos. Se considera la dependencia de la secuencia, debido a la importancia de reducir los tiempos totales de preparación, de aumentar la productividad y de mejorar el servicio al cliente.

## FORMULACIÓN DEL PROBLEMA

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 (tiempo cero) 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 del orden de la secuencia, de tal manera que hay un tiempo de preparación que debe preceder al inicio del procesamiento de la pieza dada, que depende de la familia de la pieza a ser procesada y la familia de la pieza que la precede. Para cada pieza se conoce el tiempo de procesamiento ( $p_i$ ); el tiempo de entrega comprometida ( $d_i$ ) y la familia a la que pertenece dicha pieza ( $b_i$ ). Además, se conocen los tiempos de preparación de la máquina al pasar de una familia a otra ( $ST_{hi}$ ). Según la secuencia escogida, para cada pieza  $i$  el instante en que la pieza sale del taller ( $c_i$ ), viene dado  $c_i = r_i + w_i + p_i + ST_{hi}$ . Siendo  $r_i$  el instante de entrada al taller de la pieza  $i$  y  $w_i$  el tiempo de espera de la pieza  $i$ . 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:  $[MIN] Z = \sum T$ . Dado que el retraso medio de las piezas  $T_{med} = 1/n \sum T$ , minimizar la suma de los retrasos es equivalente a minimizar el retraso medio.

## SOLUCIONES INICIALES

**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).** Ordena las piezas por familias de acuerdo con el tiempo de preparación más corto, y, además, secuencia las piezas entre familia por EDD.

**Índice  $CR_1$ (Cociente).** Basado en De Castro et al (2003)  $CR_1 = d_i / (ST_{hi} + p_i)$ . Y se asignan las piezas en orden no decreciente del índice.

**Índice  $CR_2$  (Suma ponderada).** Basado en Companys y D'Armas (2005),  $CR_2 = 0.2 d_i + 0.8 (p_i + ST_{hi})$ . Y se asignan las piezas en orden no decreciente del índice.

## METAHEURÍSTICAS

**Recocido Simulado (SA).** SA parte de una solución inicial determinada por una heurística, y mediante la exploración de su entorno, trata de encontrar la solución óptima (Kirkpatrick et al, 1983). Para la implementación del recocido se tomó una temperatura inicial  $T = 0,4$  y la tasa de enfriamiento  $r = 0,95$ . 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).** TS es una metaheurística de alto nivel introducida y desarrollada 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.

**GRASP.** GRASP fue propuesto por Feo y Resende (1989). La técnica empleada para esta metaheurística es de tipo iterativo. Se aplicó un GRASP basado en Ribas y Companys (2004). En la primera Fase, se construye una solución inicial a partir de  $CR_2$ . Se determina un valor de referencia mediante un incremento del 20% del menor valor  $CR_2$ . Se elige al azar una pieza cuyo  $CR_2$  sea menor que el valor de referencia, y se asigna como primera pieza a procesar. Se recalculan los  $CR_2$  y se ordenan nuevamente las piezas, desde la segunda posición, de acuerdo con un orden no decreciente de los índices recalculados. En la segunda Fase, el post-procesamiento, se aplicó una variante de la heurística Algoritmo No Exhaustivo de Descenso ANED.

**Algoritmo Genético (GA).** Fue desarrollado por Holland (1975), y se distingue de los anteriores, por que en cada iteración se tiene un conjunto de soluciones, o población en curso y no una única solución en curso. La población inicial se tomó igual al número de piezas y su composición, se dividió en dos partes iguales. La mitad de la población se forma secuenciando todas las operaciones, por EDD. Luego, se generan aleatoriamente, dos genes a permutar en la secuencia dando lugar a una nueva secuencia. La otra mitad, se forma secuenciando todas las operaciones de acuerdo con  $CR_2$ . Posteriormente, se generan aleatoriamente, dos genes a permutar en la secuencia dando lugar a una nueva secuencia. Se consideró como medida de la calidad de un individuo el valor de  $\Sigma T$ . El proceso de selección se realizó mediante la combinación de las mejores secuencias y de forma aleatoria. Las soluciones progenitoras del grupo seleccionado, se emparejaron por orden de bondad de la solución. El procedimiento de cruce que se aplicó es el Cruce PMX (Partially-Matched Crossover). La mutación se fijó la probabilidad en 30%. En el proceso de regeneración se seleccionaron las mejores secuencias de las dos poblaciones de soluciones, progenitoras e hijas.

## RESULTADOS

Los algoritmos se codificaron en lenguaje Visual Basic 6.0. La viabilidad de los algoritmos se estudió en 1000 ejemplares, con un número de familias de 4-6 y con un número de piezas de 15-25. Los algoritmos SA y TS se aplicaron a las soluciones iniciales SST-EDD, EDD,  $CR_1$  y  $CR_2$ , seleccionándose de las cuatro soluciones mejoradas la que arrojaba mejor resultado. En la Figura 1, se pueden observar los valores del retraso promedio de cada uno de los procedimientos en las colecciones de problemas estudiados. Los resultados demuestran que TS es la que arroja menores valores del retraso medio, que los mayores retrasos se obtienen con SA y que los segundos menores retrasos se lograron con GRASP.

Debido a que en el problema estudiado no se tiene la solución óptima, para juzgar la calidad se determinó la diferencia entre la solución heurística y la mejor solución. Porcentaje de error relativo:  $[100x (\text{Heurística} - \text{Mejor Solución}) / \text{Mejor Solución}]$ . En la Figura 2 se muestra el promedio del porcentaje de error relativo de cada uno de

los métodos. Se observa que TS es la que ofrece menor dispersión de los resultados, tiene un porcentaje de error relativo muy bajo, lo que indica que la mayoría de las mejores soluciones se obtienen con esta metaheurística.

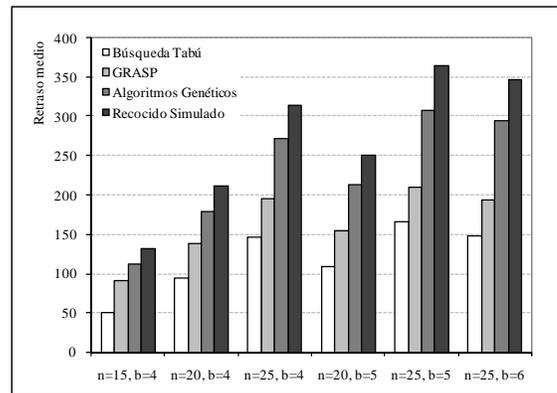


Figura 1: Comparación de las metaheurísticas.

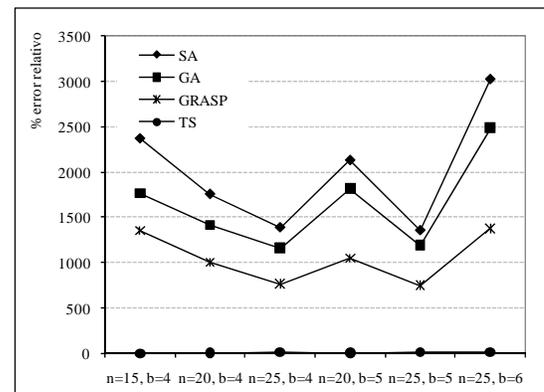


Figura 2: Promedio del error relativo

## REFERENCIAS

- 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.
- Companys R. y D'Armas M. (2005), "Programación de operaciones con tiempo de preparación mediante algoritmos de optimización local". *Universidad, Ciencia y Tecnología*, Vol. 9, No. 35, pp.155-162
- De Castro R., Companys R. y Mateo M. (2003), "Programación de las órdenes de fabricación en una sola máquina con tiempos de preparación", *V Congreso de Ingeniería de Organización*, Valladolid-Burgos, pp. 59-60.
- Feo T. y Resende. (1989), "A probabilistic heuristic for a computationally difficult set covering problem", *Operations Research Letters*. Vol. 8, pp.67-71
- Holland J. (1975), *Adaptation in Natural and Artificial Systems*, Ann Arbor: University of Michigan Press.
- Rivas I. y Companys R. (2004), "Secuenciación de pedidos en una máquina con tiempos de preparación variables". DIT. Dpto. de Organización de Empresas. Universidad Politécnica de Cataluña. España