

## **DOCUMENTO INTERNO DE TRABAJO**

# **PROCEDIMIENTOS DE EXPLORACIÓN DE ENTORNOS PARA LA PROGRAMACIÓN DE OPERACIONES EN UNA MÁQUINA CON TIEMPOS DE PREPARACIÓN**

Autores:

**MAYRA D'ARMAS**

**RAMÓN COMPANYS**

Barcelona, octubre de 2005

LABORATORIO DE ORGANIZACIÓN INDUSTRIAL  
DEPARTAMENTO DE ORGANIZACIÓN DE EMPRESAS  
UNIVERSIDAD POLITÉCNICA DE CATALUÑA

# PROCEDIMIENTOS DE EXPLORACIÓN DE ENTORNOS PARA LA PROGRAMACIÓN DE OPERACIONES EN UNA MÁQUINA CON TIEMPOS DE PREPARACIÓN

(Octubre, 2005)

M. D'Armas y R. Companys  
LOI-DOE-UPC

Este trabajo considera el problema de secuenciación en una máquina con el objetivo de minimizar el retraso total en un ambiente con preparaciones que dependen de la secuencia. Se comparan los resultados obtenidos mediante la aplicación de los procedimientos de exploración de entornos AED, ANED, Recocido Simulado, Algoritmos Genéticos, Búsqueda Tabú y GRASP al problema planteado. Los resultados sugieren que la Búsqueda Tabú es una técnica viable de solución que puede proporcionar buenas soluciones cuando se considera el objetivo retraso total con tiempos de preparación dependientes de la secuencia.

**Keywords:** Scheduling; Sequence-dependent setup times; Total tardiness; Metaheuristics; Local optimization algorithms.

## 1. INTRODUCCIÓN

El presente trabajo tiene por objetivo evaluar, mediante pruebas computacionales, el comportamiento de los procedimientos de exploración de entornos AED, ANED, 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, que 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.

La aplicación práctica del problema planteado (separar las tareas de preparación de las tareas de procesamiento, y considerar su dependencia) se encuentran con frecuencia en los ambientes industriales, tales como la industria química, farmacéutica, del metal, textil, del plástico, entre otros.

Muchas de las investigaciones en el área de la secuenciación de operaciones, consideran que cuando una máquina cambia de un trabajo a otro no hay tiempos de preparación, y si los hay, éstos se consideran independientes de la secuencia

(D'Armas, 2005). No obstante, la suposición de que los tiempos de preparación son insignificantes o forman parte del tiempo de procesamiento, 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 (Allahverdi, 2000).

## 2. FORMULACIÓN DEL PROBLEMA

En esta investigación se considera el problema de programación de familias de piezas en una máquina, donde se necesita un tiempo de preparación cuando se cambia de familia. 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 procesamiento de la pieza $i$ (processing time)
$b_i$	familia a la que pertenece la pieza $i$ (batch)
$ST_{hi}$	tiempo de preparación de la pieza $i$ cuando $h$ ha sido la pieza anterior de la secuencia (setup time)
$d_i$	instante comprometido de salida del taller de la pieza $i$ (due date)
$r_i$	instante de entrada al taller de la pieza $i$ (ready time)
$c_i$	instante de terminación o de salida de la pieza $i$ (completion time)
$w_i$	tiempo de espera de la pieza $i$ (waiting time)
$L_i$	diferencia entre el instante de salida real y el previsto (lateness)
$T_i$	retraso (tardiness)

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  $d_i$  límite de finalización o entrega. Se asume también que las piezas están clasificadas en varias familias y que el tiempo de preparación  $ST$  se produce cuando se pasa de una familia a otra. El tiempo de preparación de la máquina  $ST$  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. La matriz de tiempos de preparación no tiene por que ser simétrica. Para cada pieza  $i$  se conoce el tiempo de operación  $p_i$ ; 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  $ST$  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:

$$T_i = \text{máx. } \{0, c_i - d_i\}$$

El objetivo es encontrar una secuencia de las piezas que minimice la suma del retraso de las piezas:

$$[\text{MIN}] z = \sum_{i=1}^n T_i$$

Dado que el retraso medio de las piezas ( $T_{med}$ ) viene dado por:

$$\frac{1}{n} \sum_{i=1}^n T_i$$

minimizar la suma de los retrasos es equivalente a minimizar el retraso medio.

### 3. TÉCNICAS DE SOLUCIÓN

Se han utilizado seis métodos para resolver el problema estudiado, los algoritmos de optimización local: Algoritmo Exhaustivo de Descenso y Algoritmo No Exhaustivo de Descenso, y las metaheurísticas: Recocido Simulado, Búsqueda Tabú, GRASP y Algoritmo Genético.

#### 3.1 Solución inicial

Para obtener la solución inicial necesaria para aplicar los procedimientos: Algoritmo Exhaustivo de Descenso, Algoritmo No Exhaustivo de Descenso, Recocido Simulado y

Búsqueda Tabú, se utilizaron cuatro estrategias de secuenciación descritas a continuación:

- **EDD** (earliest due date): consiste en ordenar las piezas, en orden creciente, de acuerdo con la fecha de vencimiento. Para cada pieza  $i$  se considera la fecha de vencimiento  $d_i$  y el tiempo de preparación  $ST_{h,i}$  (dependiente de la familia de la pieza anterior  $h$ ).

Dada las características del problema tratado, esta regla favorece las piezas más prioritarias, sin embargo no aprovecha la ventaja de reducir los tiempos de preparación al procesar, sucesivamente, piezas de la misma familia.

- **SST-EDD** (shortest setup time – earliest due date): consiste en ordenar las piezas por familias de acuerdo con el tiempo de preparación más corto cuando se cambia de una familia a otra (para favorecer un tiempo mínimo de cambios entre familias), y, además, secuenciar las piezas entre familia por orden creciente de fechas de vencimiento. Para cada pieza  $i$  se considera la fecha de vencimiento  $d_i$  y el tiempo de preparación  $ST_{h,i}$  (dependiente de la familia de la pieza anterior  $h$ ).

Una característica de la regla EDD (cuando no hay tiempos de preparación) es que se mantiene dentro de una familia dada, como lo demostrado por Monma y Potts (1989) de que existe un programa óptimo tal que las piezas dentro de cada familia se secuencien en orden creciente de sus fechas de vencimiento. Es decir, si se tienen familias de piezas, entonces las piezas dentro de la familia deberían secuenciarse de acuerdo con la regla EDD.

- **CR1** (Cociente). Índice crítico para cada pieza  $i$ , basada en De Castro et al (2003), que consiste en calcular el índice de prioridad por medio de la fecha de vencimiento dividida por la suma entre el tiempo de preparación de la familia de la pieza  $i$  (dependiente de la familia de la pieza anterior  $h$ ) y el tiempo de procesamiento de la pieza  $i$ . Las piezas que tienen el CR1 más pequeños se asignan primero. Para cada pieza  $i$  se considera la fecha de vencimiento  $d_i$ , el tiempo de preparación  $ST_{h,i}$  y el tiempo de procesamiento u operación  $p_i$ :

$$CR1 = d_i / (ST_{hi} + p_i)$$

- **CR2** (Suma ponderada). Basado en De Castro et al (2003), consiste en asignar 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  $i$  (dependiente de la familia de la pieza anterior  $h$ ) y el tiempo de procesamiento de la pieza.

$$CR2 = a d_i + (1-a) (p_i + ST_{hi}).$$

Se ha empleado un alfa igual a 0,2 de acuerdo con lo propuesto por Companys y D'Armas (2005).

En el siguiente ejemplo se aplican las estrategias de secuenciación descritas anteriormente, a un ejemplar de **12/1/STsd,b/ST** con cuatro familias, asumiendo que la familia anteriormente procesada es la número uno.

$i$	1	2	3	4	5	6	7	8	9	10	11	12
$p_i$	5	6	8	9	8	7	6	4	7	9	8	5
$d_i$	22	31	42	50	19	23	29	41	20	31	25	43
$g_i$	1	1	1	1	2	2	2	2	3	3	4	4

$ST_{hi}$	1	2	3	4
1	0	2	2	2
2	2	0	2	2
3	2	2	0	2
4	2	2	2	0

Si se ordena conforme a la fecha de vencimiento **EDD**, la secuencia es 5-9-1-6-11-7-2-10-8-3-12-4, y los resultados son los siguientes:

$i$	<b>5</b>	<b>9</b>	<b>1</b>	<b>6</b>	<b>11</b>	<b>7</b>	<b>2</b>	<b>10</b>	<b>8</b>	<b>3</b>	<b>12</b>	<b>4</b>
$p_i$	8	7	5	7	8	6	6	9	4	8	5	9
$b_i$	2	3	1	2	4	2	1	3	2	1	4	1
$ST_{hi}$	2	2	2	2	2	2	2	2	2	2	2	2
$c_i$	10	19	26	35	45	53	61	72	78	88	95	106
$d_i$	19	20	22	23	25	29	31	31	41	42	43	50
$L_i$	-9	-1	4	12	20	24	30	41	37	46	52	56
$T_i$	0	0	4	12	20	24	30	41	37	46	52	56

Si se ordenan por familia y fecha de vencimiento **SST-EDD**, la secuencia es 1-2-3-4-5-6-7-8-9-10-11-12, obteniéndose los resultados siguientes:

<i>i</i>	1	2	3	4	5	6	7	8	9	10	11	12
<b><i>p<sub>i</sub></i></b>	5	6	8	9	8	7	6	4	7	9	8	5
<b><i>b<sub>i</sub></i></b>	1	1	1	1	2	2	2	2	3	3	4	4
<b><i>ST<sub>hi</sub></i></b>	0	0	0	0	2	0	0	0	2	0	2	0
<b><i>c<sub>i</sub></i></b>	5	11	19	28	38	45	51	55	64	73	83	88
<b><i>d<sub>i</sub></i></b>	22	31	42	50	19	23	29	41	20	31	25	43
<b><i>L<sub>i</sub></i></b>	-17	-20	-23	-22	19	22	22	14	44	42	58	45
<b><i>T<sub>i</sub></i></b>	0	0	0	0	19	22	22	14	44	42	58	45

Si se ordenan por el índice crítico **CR1**, la secuencia es 5-9-11-6-10-1-7-2-3-4-12-8, y los resultados son:

<i>i</i>	5	9	11	6	10	1	7	2	3	4	12	8
<b>CR1</b>	1,90	2,22	2,50	2,56	2,82	3,14	3,63	3,88	5,25	5,56	6,14	6,83
<b><i>p<sub>i</sub></i></b>	8	7	8	7	9	5	6	6	8	9	5	4
<b><i>b<sub>i</sub></i></b>	2	3	4	2	3	1	2	1	1	1	4	2
<b><i>ST<sub>hi</sub></i></b>	2	2	2	2	2	2	2	2	0	0	2	2
<b><i>c<sub>i</sub></i></b>	10	19	29	38	49	56	64	72	80	89	96	102
<b><i>d<sub>i</sub></i></b>	19	20	25	23	31	22	29	31	42	50	43	41
<b><i>L<sub>i</sub></i></b>	-9	-1	4	15	18	34	35	41	38	39	53	61
<b><i>T<sub>i</sub></i></b>	0	0	4	15	18	34	35	41	38	39	53	61

Si se ordenan de acuerdo con el índice crítico **CR2** (con  $\alpha=0,5$ ), la secuencia es 1-5-9-6-7-11-2-10-8-12-3-4, obteniéndose los resultados siguientes:

<i>i</i>	1	5	9	6	7	11	2	10	8	12	3	4
<b>CR2</b>	13,5	14,5	14,5	16	17,5	17,5	19,5	21	23,5	25	26	29,5
<b><i>p<sub>i</sub></i></b>	5	8	7	7	6	8	6	9	4	5	8	9
<b><i>b<sub>i</sub></i></b>	1	2	3	2	2	4	1	3	2	4	1	1
<b><i>ST<sub>hi</sub></i></b>	0	2	2	2	0	2	2	2	2	2	2	0
<b><i>c<sub>i</sub></i></b>	5	15	24	33	39	49	57	68	74	81	91	100
<b><i>d<sub>i</sub></i></b>	22	19	20	23	29	25	31	31	41	43	42	50
<b><i>L<sub>i</sub></i></b>	-17	-4	4	10	10	24	26	37	33	38	49	50
<b><i>T<sub>i</sub></i></b>	0	0	4	10	10	24	26	37	33	38	49	50

Si se comparan los resultados de las tres heurísticas, se puede observar que la secuencia **SST-EDD** ofrece el menor retraso total; esta disminución puede ser debido a que se aprovecha la ventaja de reducir los tiempos de preparación al agrupar las piezas de la misma familia.

Índice de eficacia	CR1	EDD	CR2	SST-EDD
$T_{med}$	28,17	26,83	23,42	22,17
$T_{max}$	61	56	50	58
$ST$	338	322	281	266

### 3.2 Algoritmo Exhaustivo de Descenso (AED)

A partir de la solución en curso se generan y evalúan todos los vecinos, si el mejor de ellos (en caso de búsqueda de un mínimo el de menor valor) es mejor que la solución en curso se toma como nueva solución en curso y se reitera el procedimiento; en caso contrario, si el mejor vecino es peor o igual a la solución en curso el procedimiento se da por terminado.

### 3.3 Algoritmo No Exhaustivo de Descenso (ANED)

A partir de la solución en curso se generan y evalúan en cierto orden sus vecinos, si uno de ellos es mejor que la solución en curso se toma como nueva solución en curso (sin terminar la generación de los vecinos de la solución primitiva) y se prosigue aplicando el procedimiento a los vecinos de la nueva solución en curso; cuando se han generado todos los vecinos de una determinada solución en curso sin que ninguno sea mejor el procedimiento se da por terminado.

### 3.4 Recocido Simulado (SA, Simulated Annealing)

La aproximación SA fue desarrollada por Kirkpatrick, Gelatt y Vecchi (1983) y se le conoce como Recocido Simulado dado la analogía entre la simulación del recocido de sólidos y el problema de la resolución de los grandes problemas de optimización combinatoria.

SA parte de una solución inicial determinada por una heurística, y mediante la exploración de su entorno, trata de encontrar la solución óptima. El método consiste de



iteraciones, donde se compara la solución en curso con la solución vecina generada aleatoriamente. Si la solución vecina es mejor que la solución en curso, entonces la solución vecina pasa a ser la solución en curso. En el caso de que la solución vecina sea peor que la solución en curso se puede tomar esta solución vecina como la solución en curso, con una probabilidad que depende de la diferencia entre la solución vecina y la solución en curso y la temperatura  $T$ .

Para el problema estudiado, la regla de generación de la probabilidad de aceptación de la solución vecina que es peor que la solución en curso sigue la distribución Boltzmann:

$$p(X_{vec}) = e^{-[f(X_{vec}) - f(X_{cur})]/T}$$

Donde la función tomará valores entre  $[0,1]$ , y dependerá de la solución vecina, la solución en curso y la temperatura  $T$  en la iteración  $n$ .

Para la implementación del recocido se han tomado los valores siguientes: (a) Temperatura inicial  $T = 0,4$ ; (b) Tasa de disminución de la temperatura o rata de enfriamiento  $r = 0,95$ ; y (c) Size factor = 16. La generación de los vecinos se ha realizado al azar. Para ello se generan, aleatoriamente, dos piezas a permutar en la secuencia dando lugar a una nueva secuencia vecina. La secuencia vecina se diferencia de la secuencia madre en el orden de realización de dos piezas.

### **3.5 Búsqueda Tabú (TS, Tabu Search)**

TS es un procedimiento metaheurístico de alto nivel introducido y desarrollado en su forma actual por Fred Glover (1989). Su filosofía se basa en la explotación de diversas estrategias inteligentes para la resolución de problemas, basadas en procedimientos de aprendizaje. (Glover y Melián, 2003). A diferencia de otros algoritmos basados en técnicas aleatorias de búsqueda de soluciones cercanas, TS es determinista ya que elimina el azar en sus decisiones y la búsqueda del óptimo está guiada por una estrategia basada en el uso de estructuras de memoria.

El aspecto fundamental del procedimiento consiste en comenzar desde una solución inicial y generar un conjunto de soluciones de su vecindario, de estas elegir la mejor de las vecinas aún cuando ésta sea peor que la solución en curso. La característica importante de TS es esencialmente la construcción de una lista tabú de movimientos:

aquellos movimientos que no son permitidos (movimientos tabú) en la presente iteración. La razón de esta lista es la de excluir los movimientos que pueden hacer regresar a algún punto de una iteración anterior, es decir evitar regresar al mismo óptimo local.

Debido a que la lista tabú puede prohibir soluciones que tengan una calidad superior a la de las soluciones ya conocida, se hizo uso del concepto nivel de aspiración. El criterio que se ha tomado en este trabajo es seleccionar aquellas secuencias de operaciones cuyo valor de  $\Sigma T$ , retraso total, sea inferior al de la mejor solución hallada hasta el momento. Primero se comprueba si la solución vecina actual satisface el nivel de aspiración, si es así, la solución vecina sustituye a la solución en curso y a la mejor. En caso contrario, si dicha solución vecina es igual o peor que la mejor, entonces se compara con los atributos de la lista tabú.

La generación de vecinas se ha realizado mediante la exploración de todo el entorno de la solución en curso, es decir, dada una secuencia de operaciones se generan todos los vecinos mediante la permutación entre dos piezas, haciéndose todos los cambios posibles de dos posiciones (que podrán ser contiguas o no).

### **3.6 GRASP**

Fue desarrollado originalmente por Feo y Resende (1989) al estudiar un problema de cobertura de alta complejidad combinatoria. La técnica empleada para esta metaheurística es de tipo iterativo. Cada iteración en GRASP consta generalmente de dos pasos: la fase de construcción y el procedimiento de búsqueda local. En el primero se construye una solución tentativa, que luego es mejorada mediante un procedimiento de intercambio hasta que se llega a un óptimo local. En la fase constructiva, GRASP toma en cuenta la función objetivo con la intención de que al término de la iteración se cuente con una solución de alta calidad, sobre la cual se efectúa una mejora o Fase II.

En este trabajo se ha aplicado una heurística GRASP basada en Ribas y Companys (2004). En la primera Fase, para disponer de una solución inicial se construye una solución inicial a partir de Índice Crítico CR2, secuenciándose las piezas en orden no decreciente de su índice. Se toma el menor valor del índice CR2, y se determina un valor de referencia mediante un incremento del veinte por ciento (20%) del menor

valor. Se elige al azar una de las piezas cuyo índice CR2 sea menor que el valor de referencia, y se asigna como primera pieza a procesar. Se recalculan los índices CR2 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 ha aplicado un procedimiento de intercambio de 2intercambio, y una búsqueda a  $k$ vecindades. El procedimiento de mejora aplica una variante de la heurística ANED, donde cada iteración recorre el conjunto de soluciones de forma distinta pudiendo acceder a vecindarios no explorados.

Para lograr esto, se define un nuevo vector de posiciones (*rev*), que permite codificar los punteros de posición que se usan durante la exploración del vecindario, con el fin de evitar recorrer dicho vecindario siempre de la misma forma. Inicialmente, el vector *rev* contiene de forma ordenada las diferentes posiciones que una pieza puede ocupar en la secuencia. No obstante, posteriormente se mezclan al azar las posiciones y, durante la aplicación del procedimiento de mejora de la heurística ANED, se codifican, a través del vector *rev*, los punteros de posición utilizados para explorar el vecindario de una solución. Es decir, para una posición  $k1$ ,  $k2$  se toma su equivalente  $k1rev$ ,  $k2rev$  del vector *rev*, siendo  $k1rev = rev(k1)$  y  $k2rev = rev(k2)$ , y se aplica el procedimiento con estos nuevos punteros.

### **3.7 Algoritmo Genético (GA, Genetic Algorithm)**

GA es un algoritmo de búsqueda que explora un espacio de solución que simula procesos en un sistema natural hacia la evolución, específicamente aquellos que siguen el principio de la supervivencia en función de la adaptabilidad. (Díaz et al, 1996). Fue desarrollado por Holland (1975), y se distingue muy claramente de todos los anteriores, básicamente por el hecho de que en cada iteración se tiene un conjunto de soluciones, o población en curso y no una única solución en curso. Las soluciones sucesoras se obtienen a partir de parejas constituidas con los elementos de la población y no mediante la transformación de la solución en curso.

*3.7.1 Población inicial.* En este trabajo se ha aplicado una combinación de perturbaciones aleatorias con reglas de secuenciación, que permiten garantizar que la población inicial sea heterogénea y que se inicie con soluciones que tengan una

calidad aceptable. Así mismo, la dimensión de la población se ha tomado igual al número de operaciones de tal manera que la población tiene una relación directamente proporcional con la dimensión del problema a tratar. En cuanto a su composición, la población inicial se ha dividido en dos partes iguales, las cuales se diferencian entre sí por el esquema de confección de los cromosomas que conforman dicha población. La mitad de la población se forma secuenciando todas las operaciones, en orden creciente, de acuerdo con la fecha de vencimiento. Posteriormente, 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 el Índice Crítico CR2 en orden no decreciente de su índice. Posteriormente, se generan aleatoriamente, dos genes a permutar en la secuencia dando lugar a una nueva secuencia.

**3.7.2 Selección.** Mediante la selección se determinan los individuos de la población que tendrán descendencia. En este trabajo el problema estudiado tiene como objetivo minimizar los retrasos en las entregas de los pedidos, por lo tanto la medida de la calidad de un individuo (solución) que se ha considerado es el valor de  $ST$ . El proceso de selección se ha realizado mediante la combinación de las mejores secuencias y de forma aleatoria, es decir un 50% de las secuencias a cruzar lo conforman las mejores soluciones de la población actual y el otro 50% se elige aleatoriamente.

**3.7.3 Cruces.** El procedimiento de cruce consiste en combinar dos secuencias padres según ciertos criterios. Inicialmente, las soluciones progenitoras del grupo seleccionado, se han emparejado por orden de bondad de la solución, es decir, la primera solución se empareja con la segunda, la tercera con la cuarta, etc. dentro de la lista de soluciones elegidas ordenadas de acuerdo con el valor de  $ST$ . A continuación, una vez que se tienen las parejas, se procede a realizar el cruce entre ambas, las cuales generan dos soluciones descendientes. El procedimiento de cruce que se ha aplicado es el Cruce PMX (Partially-Matched Crossover). Dado dos soluciones progenitoras, se eligen dos puntos de cruce al azar, luego, el procedimiento copia los genes de uno de los padres directamente a las mismas posiciones en el hijo. Las posiciones restantes se llenan con los valores que aún no han sido utilizados en el mismo orden en que se encuentran en uno de los padres.

**3.7.4 Mutaciones.** La mutación es un mecanismo que, a imitación de lo que sucede en la naturaleza con los seres vivos, fomenta la diversidad. Para el problema, se ha fijado

la probabilidad en un treinta por ciento, de acuerdo con los resultados de Fernández-Baños (2003). Es decir, el procedimiento generará un número aleatorio entre 0 y 1, si este valor es inferior al parámetro de probabilidad fijado (0,3) se realiza la mutación en la secuencia. En el supuesto que lo haga, la mutación consiste en intercambiar un gen, seleccionado al azar, con su consecutivo en la secuencia.

3.7.5 Regeneración. Una vez incrementado la población con los descendientes, mutados o no, hay que reducirla al tamaño conveniente y este proceso es el que se denomina regeneración. Se trata, por lo tanto, de elegir entre las soluciones actuales, un cierto número de ellas. En este proceso de regeneración se han seleccionado las mejores secuencias de las dos poblaciones de soluciones, progenitoras e hijas. Es decir, la nueva población generada para la siguiente iteración está formada por la mitad del total de las soluciones (padres + hijos).

#### **4. Experiencia Computacional**

En esta sección se evalúa la efectividad de los algoritmos propuestos. Para llevar a cabo este propósito, los algoritmos se han codificado en lenguaje Visual Basic 6.0 y las pruebas se han realizado en un computador personal Pentium IV de 3,00 GHz y 1,00 GB de Ram. La viabilidad de los algoritmos se ha estudiado inicialmente en 1000 ejemplares, con un número de familias desde 4 hasta 6 y con un número de piezas desde 15 hasta 25. Los ejemplares se clasificaron en problemas de: (a) 15 piezas de 4 familias; (b) 20 piezas de 4 familias; (c) 25 piezas de 4 familias; (d) 20 piezas de 5 familias; (e) 25 piezas de 5 familias; y (f) 25 piezas de 6 familias.

La tabla I resume los resultados experimentales de los 1000 ejemplares en términos de cantidad de mejores y peores soluciones. Los algoritmos AED, ANED, SA y TS se han aplicado a las cuatro soluciones iniciales, SST-EDD, EDD, CR1 y CR2, seleccionándose de las cuatro soluciones mejoradas la que arrojaba mejor resultado. De la tabla se deduce que para todos los casos estudiados, la mayoría de las peores soluciones se obtienen con el Recocido Simulado, y que la mayoría de las mejores soluciones se obtienen con la Búsqueda Tabú. Con GRASP, AED y ANED se logran buenos resultados, aunque con un porcentaje muy inferior al obtenido con TS. Con SA y GA no se lograron buenos resultados para el problema estudiado.

**Tabla I. Cantidad de mejores y peores resultados.**

Ejemplares:	n = 15, b= 4		n = 20, b= 4		n = 25, b= 4	
Cantidad de soluciones:	peores	mejores*	peores	mejores*	peores	mejores*
<b>AED</b>	1	31	2	40	1	22
<b>ANED</b>	1	33	2	38	1	24
<b>SA</b>	169	2	197	0	197	0
<b>TS</b>	0	171	1	155	0	138
<b>GRASP</b>	5	36	4	53	1	53
<b>GA</b>	41	12	11	8	11	0
	*(7 empates binarios, 10 ternarios, 8 cuaternarios, 6 quintuplos, 3 séxtuples)		*(12 empates binarios, 15 ternarios, 12 cuaternarios, 4 quintuplos)		*(13 empates binarios, 6 ternarios, 4 cuaternarios)	
Ejemplares:	n = 20, b= 5		n = 25, b= 5		n = 25, b= 6	
Cantidad de soluciones:	peores	mejores*	peores	mejores*	peores	mejores*
<b>AED</b>	0	11	1	18	0	27
<b>ANED</b>	0	14	1	20	0	28
<b>SA</b>	97	0	99	0	197	0
<b>TS</b>	0	79	0	59	0	133
<b>GRASP</b>	1	17	0	29	1	52
<b>GA</b>	6	0	6	1	7	0
	*(7 empates binarios, 4 ternarios, 2 cuaternarios)		*(12 empates binarios, 6 ternarios, 1 cuaternarios)		*(22 empates binarios, 6 ternarios, 2 cuaternarios)	

En la tabla II, se pueden observar los valores del retraso medio de cada uno de los procedimientos de exploración de entornos en cada una de las colecciones de problemas estudiados. Los resultados demuestran que TS es la que arroja menores valores del retraso medio, y que los mayores retrasos se obtienen con SA. La heurística AED arrojó un retraso promedio inferior a ANED, sin embargo las diferencias son muy pequeñas. Los segundos menores retrasos se lograron con la aplicación de la metaheurística GRASP.

**Tabla II. Retraso medio.**

Procedimientos	Retraso promedio					
	n=15, b=4	n=20, b=4	n=25, b=4	n=20, b=5	n=25, b=5	n=25, b=6
<b>AED</b>	94,63	142,20	205,415	162,55	221,59	199,80
<b>ANED</b>	94,64	143,29	209,13	164,09	222,09	201,20
<b>SA</b>	131,67	210,62	313,185	249,96	363,60	346,18
<b>TS</b>	51,12	94,68	145,68	109,35	166,44	147,19
<b>GRASP</b>	90,81	137,57	195,16	154,55	210,28	192,79
<b>GA</b>	111,94	178,58	272,165	213,01	306,77	293,70

De acuerdo con Bulfin y Sipper (1998) los algoritmos heurísticos se juzgan por su calidad y eficacia. Refiriéndose a la calidad como la diferencia entre una solución heurística y la óptima, y la eficacia como el esfuerzo realizado para obtener la solución.

En este trabajo, la justificación empírica sobre la calidad y eficacia de los algoritmos heurísticos probados, se logra mediante la resolución de cada uno de los 1000 ejemplares usando las cuatro heurísticas directas, y las heurísticas de mejora AED y ANED, y las metaheurísticas SA, TS, GA y GRASP. Para juzgar la eficacia de los algoritmos se ha computado el tiempo promedio para resolver un ejemplar, de cada uno de los procedimientos.

Los resultados se presentan en la tabla III. Los algoritmos heurísticos que necesitan más tiempo para resolver un ejemplar son el SA y la TS, sin embargo, el tiempo promedio máximo no supera los 5 segundos. El algoritmo con menor tiempo de ejecución es el ANED con un tiempo medio máximo de cuatro centésimas de segundos, seguido por GRASP con seis centésimas de segundo. El tiempo medio de ejecución de SA es, en promedio, unas 200 veces mayor que el tiempo medio de ejecución de la heurística ANED; y 80 veces mayor que el tiempo medio de AED y GRASP.

**Tabla III. Tiempo promedio de ejecución.**

Procedimientos	Tiempo promedio en segundos					
	<i>n=15, b=4</i>	<i>n=20, b=4</i>	<i>n=25, b=4</i>	<i>n=20, b=5</i>	<i>n=25, b=5</i>	<i>n=25, b=6</i>
<b>AED</b>	0,0135	0,0386	0,0926	0,0174	0,0482	0,0926
<b>ANED</b>	0,0077	0,0154	0,0347	0,0077	0,0174	0,0328
<b>SA</b>	1,0648	2,4884	4,9190	2,6042	4,9190	4,9190
<b>TS</b>	0,9549	2,3148	4,3403	2,0255	4,3403	4,3403
<b>GRASP</b>	0,0174	0,0289	0,0579	0,0347	0,0579	0,0579
<b>GA</b>	0,3704	0,6076	1,0648	0,6134	1,0648	1,0706

Para juzgar la calidad se ha determinado la diferencia entre la solución heurística y la mejor solución, debido a que en el problema estudiado no se tiene la solución óptima. Es decir, la solución heurística se comparó en términos del porcentaje de error relativo, de acuerdo con lo propuesto por varios autores como son Rajendran y Ziegler (2003),

Allahverdi y Aldowaisan (2000), Yang, Kreipl y Pinedo (2000), Raghu y Rajendran (1995).

Para cada uno de los procedimientos de exploración de entornos, se le ha determinado el porcentaje de error relativo de cada ejemplar de la colección, mediante la expresión:

$$[100x (\text{Heurística} - \text{Mejor Solución}) / \text{Mejor Solución}]$$

Posteriormente se le ha calculado el promedio de todos los ejemplares de la colección.

En la figura 1 se muestra el promedio de error relativo y la desviación estándar relativa, de cada uno de los procedimientos en las colecciones estudiadas. En la figura 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 iniciales se obtienen con esta metaheurística. También puede observarse que tiene una desviación estándar relativa muy baja, lo cual demuestra que el procedimiento es estable.

Además, se ha realizado un análisis de varianza de una vía para probar los efectos de los procedimientos de exploración de entornos en la solución final. Puesto que, para todas las colecciones estudiadas, el P-Valor del test F es inferior a 0,05, hay diferencia estadísticamente significativa entre las medias de los 6 procedimientos a un nivel de confianza del 95,0%. Los resultados del ANOVA se presentan en la tabla IV.

**Tabla IV. Análisis de varianza.**

<b>ANOVA</b>	<b>F-Ratio y (P-Valor)</b>
<i>n=15,b=4</i>	37,2448* (0,0000)
<i>n=20,b=4</i>	30,6131* (0,0000)
<i>n=25,b=4</i>	41,3572* (0,0000)
<i>n=20,b=5</i>	24,6915* (0,0067)
<i>n=25,b=5</i>	26,8466* (0,0010)
<i>n=25, b=6</i>	55,6488* (0,0000)

\*Estadísticamente significativo con  $\alpha=0,05$



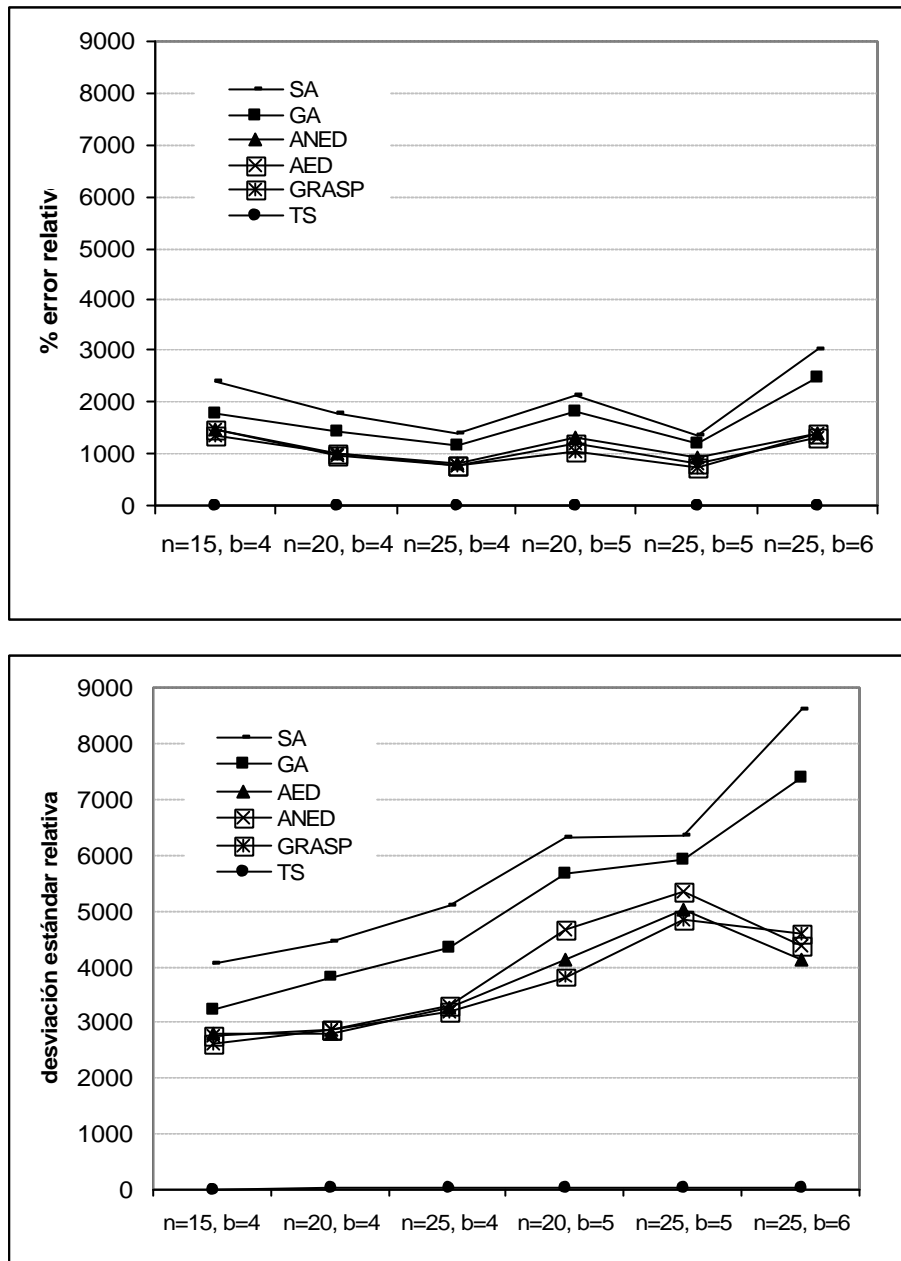


Figura 1. Promedio del error relativo y de la desviación estándar relativa

Para determinar las medias que son significativamente diferentes unas de otras, se ha realizado el procedimiento de las menores diferencias significativas de Fisher (LSD). En la tabla V se muestran los resultados de la comparación múltiple de medias.

De los resultados expuestos en la tabla V se deduce que TS es estadísticamente más efectiva que GRASP, AED, ANED, SA y GA. Las heurísticas AED y ANED son estadísticamente más efectivas que SA y GA. La metaheurística GRASP es

estadísticamente más efectiva que SA y GA. No existen diferencias estadísticamente significativas entre AED, ANED y GRASP.

**Tabla V. Comparación múltiple de medias**

PRUEBA:	CONTRASTE MÚLTIPLE DE RANGO					
	Diferencias (+/- Límites)					
Contraste	<i>n</i> =15, <i>b</i> =4	<i>n</i> =20, <i>b</i> =4	<i>n</i> =25, <i>b</i> =4	<i>n</i> =20, <i>b</i> =5	<i>n</i> =25, <i>b</i> =5	<i>n</i> =25, <i>b</i> =6
AED - ANED	-0,01 (12,1388)	-1,09 (19,793)	-3,715 (25,7197)	-1,54 (27,4558)	-0,5 (38,8476)	-1,4 (27,5765)
AED - SA	-37,04* (12,1388)	-68,425* (19,793)	-107,77* (25,7197)	-87,41* (27,4558)	-142,01* (38,8476)	-146,385* (27,5765)
AED - TS	43,515* (12,1388)	47,52* (19,793)	59,735* (25,7197)	53,2* (27,4558)	55,15* (38,8476)	52,61* (27,5765)
AED - GRASP	3,825 (12,1388)	4,625 (19,793)	10,255 (25,7197)	8,0 (27,4558)	11,31 (38,8476)	7,01 (27,5765)
AED - GA	-17,31* (12,1388)	-36,385* (19,793)	-66,75* (25,7197)	-50,46* (27,4558)	-85,18* (38,8476)	-93,905* (27,5765)
ANED - SA	-37,03* (12,1388)	-67,335* (19,793)	-104,055* (25,7197)	-85,87* (27,4558)	-141,51* (38,8476)	-144,985* (27,5765)
ANED - TS	43,525* (12,1388)	48,61* (19,793)	63,45* (25,7197)	54,74* (27,4558)	55,65* (38,8476)	54,01* (27,5765)
ANED - GRASP	3,835 (12,1388)	5,715 (19,793)	13,97 (25,7197)	9,54 (27,4558)	11,81 (22,6811)	8,41 (27,5765)
ANED - GA	-17,3* (12,1388)	-35,295* (19,793)	-63,035* (25,7197)	-48,92* (27,4558)	-84,68* (38,8476)	-92,505* (27,5765)
SA - TS	80,555* (12,1388)	115,945* (19,793)	167,505* (25,7197)	140,61* (27,4558)	197,16* (38,8476)	198,995* (27,5765)
SA - GRASP	40,865* (12,1388)	73,05* (19,793)	118,025* (25,7197)	95,41* (27,4558)	153,32* (38,8476)	153,395* (27,5765)
SA - GA	19,73* (12,1388)	32,04* (19,793)	41,02* (25,7197)	36,95* (27,4558)	56,83* (38,8476)	52,48* (27,5765)
TS - GRASP	-39,69* (12,1388)	-42,895* (19,793)	-49,48* (25,7197)	-45,2* (27,4558)	-43,84* (38,8476)	-45,6* (27,5765)
TS - GA	-60,825* (12,1388)	-83,905* (19,793)	-126,485* (25,7197)	-103,66* (27,4558)	-140,33* (38,8476)	-146,515* (27,5765)
GRASP - GA	-21,135* (12,1388)	-41,01* (19,793)	-77,005* (25,7197)	-58,46* (27,4558)	-96,49* (38,8476)	-100,915* (27,5765)

\* Estadísticamente significativo con  $\alpha=0,05$

## 5. CONCLUSIONES

En este trabajo se han adaptado los procedimientos heurísticos Algoritmo Exhaustivo de Descenso, Algoritmo No Exhaustivo de Descenso, Recocido Simulado, Búsqueda Tabú, GRASP y Algoritmo Genético a la programación de operaciones con familias de productos y tiempos de preparación dependientes de la secuencia.

La comparación de los diferentes procedimientos se ha realizado en términos de su calidad y eficacia. Para juzgar la eficacia de los algoritmos se ha computado el tiempo promedio para resolver un ejemplar, y para juzgar la calidad se ha determinado porcentaje de error relativo (diferencia entre la solución heurística y la mejor solución), debido a que en el problema estudiado no se tiene la solución óptima.

Los resultados demuestran que GRASP, AED y ANED proporcionan, al problema planteado, soluciones aceptables con un esfuerzo computacional muy bajo. Para las metaheurísticas SA, GA y TS, el número de iteraciones máximas recomendables es de 500, ya que a partir de este valor la calidad de las soluciones no mejora y el esfuerzo computacional es mucho mayor. Mediante la comparación de los resultados obtenidos con los diferentes algoritmos de exploración de entornos, se ha demostrado que las metaheurísticas GA y SA son los procedimientos menos recomendados para el problema estudiado ya que arrojaron los peores resultados, y que la Búsqueda Tabú es el mejor procedimiento, logrando las mejores soluciones en todos los casos estudiados y con un tiempo de ejecución bastante corto.

## 6. BIBLIOGRAFÍA

- Allahverdi A. y Aldowaisan T. (2000), "No-wait and separate setup three-machine flowshop with total completion time criterion", *International Transactions in Operational Research*, Vol. 7, No. 3, pp. 245-264
- 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
- Bulfin R. Jr. y Sipper D. (1998), "Planeación y Control de la Producción", Editorial Mc Graw Hill, México
- 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

- D'Armas M. (2005), "Estado del arte de la programación de operaciones con tiempos de preparación: tema para futuras investigaciones". Parte II. Universidad, Ciencia y Tecnología, Vol. 9, No. 34, pp. 96-102
- 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.
- Díaz A., Glover F., Ghasiri H., González JL., Laguna M., Moscazo P. y Tseng F. (1996), "Optimización Heurística y Redes Neuronales en Dirección de Operaciones e Ingeniería", Editorial Paraninfo, España
- Feo T. y Resende. (1989), "A probabilistic heuristic for a computationally difficult set covering problem", Operations Research Letters. Vol. 8, pp.67-71
- Fernández-Baños I. (2003), Programación de la secuencia de fabricación en una máquina, con tiempos de preparación variables, mediante la aplicación de Algoritmos Genéticos. Proyecto de Fin de Carrera. Escuela Técnica Superior de Ingeniería Industrial de Barcelona, España
- Glover F. (1989), "Tabu Search, Part I", ORSA Journal on Computing, Vol. 1, pp.190-206.
- Glover F. y Melián B. (2003), "Búsqueda Tabú", Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial, Vol. 19, pp. 29-48
- Holland J. (1975), Adaptation in Natural and Artificial Systems, Ann Arbor: University of Michigan Press.
- Kirkpatrick S., Gelatt CD. y Vecchi MP. (1983), "Optimization by simulated annealing", Science, Vol. 220, pp. 671-680.
- Raghu T.S. y Rajendran C. (1995), "Due-date setting methodologies based on simulated annealing-an experimental study in a real-life job shop", International Journal of Production Research, Vol. 33, No. 9, pp. 2535-2554
- 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
- Rivas I. y Companys R. (2004), "Secuenciación de pedidos en una máquina con tiempos de preparación variables". Documento Interno de Trabajo. Laboratorio de Organización Industrial. Departamento de Organización de Empresas. Universidad Politécnica de Cataluña. España
- Yang Y., Kreipl S. y Pinedo M. (2000), "Heuristics for minimizing total weighted tardiness in flexible flow shops", Journal of Scheduling, Vol. 3, pp.89-108