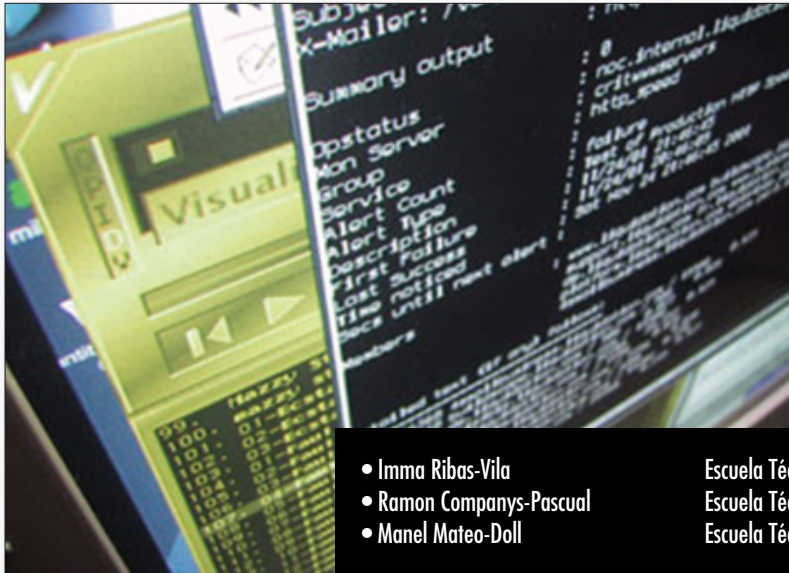


Programación bicriterio para máquinas en paralelo

Bicriteria scheduling problem on parallel machine



• Imma Ribas-Vila
• Ramon Companys-Pascual
• Manel Mateo-Doll

Escuela Técnica Superior de Ingenieros Industriales de Barcelona.
Escuela Técnica Superior de Ingenieros Industriales de Barcelona.
Escuela Técnica Superior de Ingenieros Industriales de Barcelona.

Recibido: 01/12/08 • Aceptado: 20/04/09

ABSTRACT

- This paper shows the parallel machine scheduling problem with sequence dependent set up times. The purpose is to schedule the jobs to minimize the weighted sum of average completion time and average tardiness of the jobs. The aim is to maintain a good service with minimum work in process. This research analyzes the behaviour of six simple heuristics easily to implement for practitioners which are the combination of three simple procedures to obtain an initial solution and two improvement procedures; one that acts on a neighbourhood formed from the sequences for each machine and another that acts on the neighbourhood established from the sequences in each machine. The results show that the second improvement procedure, which is closer to the matter of the problem, leads to better results. According to this result, the performance of the initial procedures, when the weighted of the two criteria are varied, has been studied. The results of the test show that, for more balanced criteria, the initial procedures which consider those criteria lead to better results.
- **Key words:** Scheduling, dependent setup times, bicriteria.

RESUMEN

En este artículo se estudia la problemática de programar la producción en un entorno de fabricación con máquinas idénticas en paralelo y tiempos de preparación dependientes de la secuencia. El objetivo fijado en la programación de las piezas es la minimización de la suma ponderada del plazo medio de fabricación y del retraso medio de las piezas, que equivale a intentar ofrecer un buen nivel de servicio al cliente manteniendo un nivel de inventario aceptable. El estudio analiza la eficiencia de seis procedimientos heurísticos de programación simples y fácilmente utilizables en la industria, resultantes de la combinación de tres reglas de secuenciación, para obtener una solución inicial, y dos procedimientos de mejora, uno que actúa sobre el vecindario de una secuencia de las piezas y otro que actúa sobre un vecindario establecido a partir de las secuencias en cada máquina. Los resultados obtenidos muestran que el segundo

procedimiento de mejora, más ligado a la estructura del problema, conduce a mejores soluciones. A partir de este resultado, se ha analizado el efecto que tiene la solución inicial empleada en la calidad de la solución obtenida. En particular se muestra que a medida que los valores de los criterios individuales están más equilibrados los procedimientos utilizados para encontrar una secuencia inicial enfocados a dichos criterios proporcionan soluciones finales de mayor calidad.

Palabras clave: secuenciación, tiempos de preparación, bicriterio.

1. INTRODUCCIÓN

Los sistemas productivos constituidos por máquinas en paralelo son frecuentes en la industria, y particularmente, en la de nuestro país. En este entorno, el director de producción no sólo debe decidir en qué máquina se va a hacer cada producto sino también la

secuencia de productos en cada una de las máquinas. La formulación clásica del problema consiste en considerar un conjunto de n piezas que deben recibir una operación en una máquina, de la que se disponen de m ejemplares idénticos. Para simplificar denominaremos máquinas a los distintos ejemplares. Cada máquina puede procesar un único trabajo a la vez y cada trabajo se procesa sin interrupción. Las piezas y las máquinas están disponibles en el instante inicial y cada pieza tiene un tiempo de proceso determinado. Se desea establecer la secuencia en cada una de las máquinas que optimice un cierto criterio de eficiencia. En la versión primitiva del problema no se consideraba el tiempo necesario para preparar la máquina al cambiar de producto, lo que significaba que dichos tiempos de preparación eran despreciables al lado de los tiempos operatorios, asunción sólo justificable en la realización de grandes series del mismo producto. Actualmente las circunstancias han cambiado y la competencia en un mercado cada vez más globalizado obliga a las empresas a ofrecer productos personalizados con el fin de obtener una mayor ventaja competitiva. Esta personalización de productos conlleva, en la mayoría de los casos, a series cortas con apreciables tiempos de preparación de máquina debido a limpieza, cambios del molde, ajustes, etc. para pasar de un tipo de producto a otro. Estos tiempos de cambio suelen no ser simétricos, por ejemplo el tiempo de limpieza requerido para pasar de producir colores claros a oscuros es inferior al requerido para pasar de oscuros a claros, por lo que existen unas secuencias de productos en las máquinas más eficientes que otras.

En la mayoría de los entornos industriales, la programación de la producción persigue un objetivo complejo, que considera, implícita o explícitamente, más de un criterio simple o factor de calidad. Con ello se pretende dar respuesta a las diferentes preocupaciones, a veces contrapuestas, que un director de producción tiene. Los responsables de las más diversas empresas frecuentemente nos han expresado que las dos obsesiones que les guían en el momento de programar son la de mantener y mejorar el nivel de servicio ofrecido a los clientes y la de reducir el nivel de stock. De manera más formal puede decirse que desean minimizar el stock (obra en curso) y que desean maximizar el nivel de servicio. Es evidente que si el coste de mantener el stock es muy pequeño y la penalización por tener retrasos muy alta se debe buscar, únicamente, cumplir con las fechas prometidas a los clientes, aunque ello implique tener más stock para hacer frente a problemas puntuales de capacidad. En cambio, si el coste del stock es elevado o si los artículos son perecederos lo importante es minimizar el stock aunque esto signifique no poder cumplir con algunos de los compromisos por falta de capacidad de fabricación. Por lo tanto, el planificador es el responsable de ponderar qué criterio tiene más peso, en cada momento, en función de

las necesidades concretas del periodo a planificar. Es difícil hallar una medida objetiva que permita apreciar si estas dos intenciones se alcanzan o no. Por nuestra parte hemos apreciado que el primer criterio puede medirse adecuadamente, en primera aproximación, mediante el plazo medio de fabricación, F , y el segundo, mediante el retraso medio, T de las piezas. En Arana, et al. (1995) ya se indica que al disminuir el plazo medio de fabricación aumenta la productividad de los recursos y flexibiliza la respuesta al mercado. La consideración de ambos criterios puede ser adecuada en muchos entornos industriales; por ello se ha utilizado la combinación ponderada de dichas medidas, como función objetivo, en la obtención de programas eficientes en un sistema con máquinas idénticas en paralelo y tiempos de preparación dependientes de la secuencia.

La problemática aquí expuesta es frecuente en las empresas manufactureras pero son pocos los artículos publicados que consideran la combinación de programación bicriterio, máquinas idénticas en paralelo y tiempos de preparación dependientes de la secuencia. Cao, et al., (2005) consideran que las máquinas son diferentes, por lo que el tiempo de proceso de una pieza puede variar en función de la máquina asignada, y proponen un procedimiento de programación con el objetivo de minimizar la suma del coste de la máquina y el coste de los retrasos incurridos. En Lin y Jeng (2004) consideran que los tiempos de preparación son independientes de la secuencia. En Eren (2009) consideran que el factor humano influye en los tiempos de preparación ya que si un trabajo se repite con cierta frecuencia el tiempo requerido para llevarlo a cabo va disminuyendo. Este hecho se conoce como “learning effect” en la literatura referente a la programación de trabajos. En Webster y Azizoglu (2001) y Azizoglu y Webster (2003) estudian el problema de programar las piezas en un sistema productivo con máquinas idénticas en paralelo y tiempos de preparación dependientes de la secuencia. En ambos artículos el objetivo perseguido es la minimización de la suma ponderada de los instantes de finalización de las piezas. Kim et al., (2006) consideran tiempos de preparación dependientes de la secuencia e instantes de disponibilidad de las piezas. En este caso el criterio de eficiencia utilizado es la suma de los retrasos. En cambio Pfund, et al., (2008) trata la minimización de la suma de los retrasos ponderados de las piezas. En el presente trabajo se analiza la problemática de la programación bicriterio de piezas en un sistema productivo que consta de máquinas idénticas en paralelo, en las que el tiempo de preparación requerido depende de la pieza fabricada y de la pieza que se va a fabricar. Los resultados obtenidos muestran que mediante procedimientos muy simples es posible alcanzar soluciones razonablemente eficientes en los casos de dimensión industrial.

2. DEFINICIÓN DEL PROBLEMA

Se consideran n piezas, con una única operación, que pueden procesarse en m máquinas idénticas. Las máquinas no pueden procesar más de una pieza a la vez y una vez iniciada la operación no se interrumpe hasta que haya finalizado. Cada pieza i tiene asociado un tiempo de proceso p_i , una fecha de vencimiento d_i , y está disponible en el instante r_i para iniciar su operación (la preparación de la misma, si la hubiera, puede haber empezado antes). $s_{h,i}$ es el tiempo de preparación necesario para pasar de la fabricación de la pieza h a la pieza i ; se supone que la matriz de tiempos de cambio cumple la desigualdad triangular. Se conoce además, el instante, τ_j , en que cada máquina j queda libre para realizar las operaciones sobre las n piezas y, por lo tanto, se debe considerar el estado inicial de cada máquina, φ_j , que corresponde a la última pieza procesada por ésta. Siendo c_i el instante en que la pieza i termina su proceso, el retraso viene dado por $T_i = \{0, c_i - d_i\}$ y el plazo de fabricación por $F_i = c_i - r_i$. Por tanto, los valores medios, F y T , se obtienen sumando los valores correspondientes a cada una de las piezas y dividiendo por el número de piezas. El objetivo es encontrar una secuencia de las piezas que minimice la suma ponderada del plazo medio de fabricación y del retraso medio de las piezas. Siguiendo la nomenclatura propuesta por T'Kindt y Billaut (2002) denotamos el problema como $P/s_{hi}, r_i/F_1 (C, T)$. En lo que sigue, consideraremos que tanto las piezas como las máquinas están disponibles en el instante inicial.

3. MODELO MATEMÁTICO

El problema considerado es NP-hard dado que el caso $1//\sum T$ ya lo es (Du y Leung, 1990). En consecuencia el uso de procedimientos de exploración exactos, en los que se analizan todas las soluciones, queda limitado a ejemplares de dimensión reducida.

Como base de comparación, hemos construido un modelo de Programación Lineal Mixta (PLM), que se inspira en el propuesto por Guinet y Solomon (1996) para el problema de flow shop híbrido. El modelo formulado utiliza una única variable binaria, x_{ijk} , que toma valor 1 si el trabajo j es el precedente del trabajo i en la máquina k .

$$\text{MIN}[z] = \alpha \cdot \sum_{i=1}^n T_i + (1 - \alpha) \cdot \sum_{i=1}^n c_i \quad (1)$$

$$\sum_{i=0}^n \sum_{k=1}^m x_{ijk} = 1 \quad j = 1 \dots n+1 \quad (2)$$

$$\sum_{j=1}^{n+1} x_{0jk} = 1 \quad k = 1 \dots m \quad (3)$$

$$\sum_{i=0}^n x_{ik+1k} = 1 \quad k = 1 \dots m \quad (4)$$

$$\sum_{i=0}^n x_{ihk} - \sum_{j=1}^{n+1} x_{jhk} = 0 \quad h = 1 \dots n, k = 1 \dots m \quad (5)$$

$$c_j \geq c_i + p_j + \sum_{k=1}^m s_{ij} \cdot x_{ijk} + \left(\sum_{k=1}^m x_{ijk} - 1 \right) \cdot M \quad \begin{matrix} \forall i \\ \forall j \\ i \neq j \end{matrix} \quad (6)$$

$$T_i \geq c_i - d_i \quad \forall i \quad (7)$$

$$\begin{matrix} T_i \geq 0 \\ c_i \geq 0 \\ x_{ijk} \in \{0,1\} \end{matrix} \quad \begin{matrix} \forall i \\ \forall i \\ \forall i, j, k \end{matrix} \quad (8)$$

A través de la función objetivo (1) se quiere minimizar la suma ponderada del retraso medio de las piezas y el tiempo de fabricación medio. El planificador, a través del parámetro α , puede adaptar la programación a las necesidades del momento, penalizando más o menos los retrasos respecto al plazo medio de fabricación.

La ecuación (2) obliga a asignar cada pieza a una máquina. Las ecuaciones (3) y (4) son ecuaciones de contorno. La (3) fija la primera pieza, j , asignada a la máquina k . La (4) fija la última pieza, i , asignada a la máquina k . La ecuación (5) secuencia las piezas en las máquinas. La (6) determina el instante de finalización de las piezas teniendo en cuenta el tiempo de preparación necesario para pasar de una pieza a otra. M es un parámetro que se utiliza para garantizar que la expresión (6) sólo actúe restringiendo los valores cuando la pieza j siga inmediatamente a la pieza i . En esta formulación se ha considerado que las máquinas, inicialmente, requieren un tiempo de preparación que depende de la primera pieza que se va a procesar. La ecuación (7) permite calcular el retraso asociado a las piezas y la (8) define las condiciones de contorno.

Mediante este modelo hemos podido resolver, únicamente, ejemplares de dimensión muy reducida (6 piezas y dos máquinas) y, por tanto, no se ha podido utilizar para contrastar los resultados obtenidos por los procedimientos heurísticos implementados.

3.1 EJEMPLO

A continuación se presenta un ejemplo sencillo formado por 6 piezas que pueden ser procesadas en una de las 2 máquinas disponibles. Los datos asociados al ejemplo consisten en S , la matriz de tiempos de cambio, y en los vectores de tiempo de proceso y fechas de vencimiento p y d .

$$S = \begin{pmatrix} 3 & 2 & 1 & 2 & 3 & 2 \\ 0 & 3 & 2 & 1 & 4 & 3 \\ 2 & 0 & 3 & 2 & 1 & 2 \\ 1 & 2 & 0 & 3 & 2 & 1 \\ 2 & 1 & 2 & 0 & 3 & 2 \\ 4 & 2 & 3 & 1 & 0 & 3 \\ 3 & 4 & 2 & 3 & 1 & 0 \end{pmatrix} \quad p = (2 \ 3 \ 5 \ 4 \ 2 \ 3) \quad d = (7 \ 5 \ 6 \ 3 \ 7 \ 10)$$

Si $\alpha=0,1$ el planificador prioriza programar las piezas para minimizar la suma de los instantes de finalización frente a la minimización de la suma de los retrasos. A la solución obtenida. Figura 1, le corresponde un tiempo medio de finalización de $F=55/6$ y un retraso medio de $T=22/6$

exploración de entornos parten de una solución inicial que se va modificando para obtener una solución distinta a la que se suele llamar solución vecina. El conjunto de vecinos de una solución constituye su entorno o vecindario. En función de cómo se represente la solución y de cómo se proceda a modificarla se accede a un vecindario o a otro. En

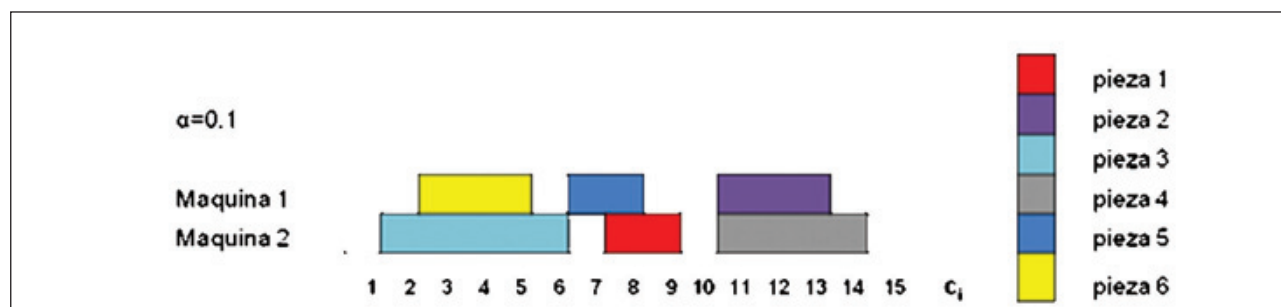


Figura 1. Programación de las piezas para $\alpha=0,1$.

En cambio, si $\alpha=0,9$ el planificador prioriza la minimización de la suma de los retrasos de las piezas frente a la suma de los tiempos de finalización. La programación obtenida en este caso, figura 2, muestra una solución menos compacta, con un tiempo medio de finalización ligeramente superior, $F = 56/6$ pero con un retraso medio, $T = 18/6$ inferior al obtenido en la solución representada en la figura 1.

este caso, cada uno de los procedimientos implementados actúa sobre un vecindario diferente, tal y como se explica en los apartados siguientes.

4.1 PROCEDIMIENTO SPT

Se calcula de forma dinámica, para cada combinación de pieza y máquina, un índice que pretende valorar el

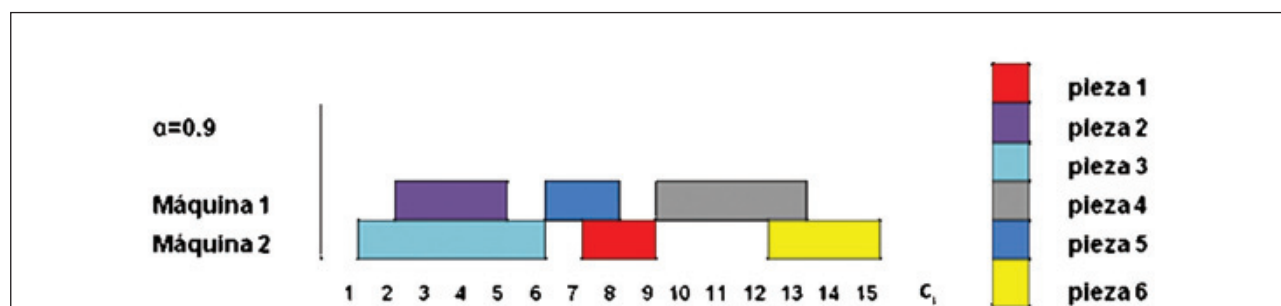


Figura 2. Programación de las piezas para $\alpha=0,9$.

4. PROCEDIMIENTOS HEURÍSTICOS

El problema de programación multicriterio puede abordarse de diferentes formas tal y como se explica en T'kindt y Billaut (2002). Aquí se ha optado, siguiendo la denominación de Hwang y Masud (1979), por un procedimiento *a priori* en el que el planificador, en función de las necesidades del periodo a programar, proporciona el peso que quiere dar a cada uno de los criterios.

Se han implementado seis heurísticas que resultan de combinar tres formas sencillas de obtener una solución inicial (*SPT*, *pseudo-SPT* y *Random*) y dos procedimientos de mejora por exploración de entornos (*SSA* y vecindario máquina) que son una adaptación, al caso bicriterio, de los propuestos en Ribas (2007) para el caso con un único criterio. Los procedimientos o heurísticas de mejora por

instante de finalización de cada pieza en cada máquina. Este índice es una adaptación de la regla *Shortest Processing Time* (SPT), que secuenciar las piezas según orden no decreciente de su tiempo de proceso, al problema con tiempos de preparación dependientes de la secuencia. El índice se utiliza como criterio de asignación de las piezas a las máquinas. Se asigna la pieza a la máquina que obtenga el índice de menor valor, es decir, se asigna la pieza a la máquina que la termine antes.

Para cada pieza y máquina se calcula $CR_{j,i}$, según la expresión (9).

$$CR_{j,i}(t) = (\max\{\tau_j + s_{hi}, r_{ij}\} + P_i) \quad (9)$$

donde P_i es el tiempo de proceso de la pieza i , s_{hi} es el tiempo de preparación necesario para realizar la pieza i cuando anteriormente se ha procesado la pieza h , τ_j es el

instante en que la máquina queda libre y r_i es el instante de disponibilidad de la pieza.

$$CR_i = \min\{CR_{j,i}\} \quad (10)$$

Para cada pieza se retiene el valor de CR_i calculado según (10) y se asigna la pieza que tiene el menor valor a la máquina que proporciona el mismo, actualizando su instante de disponibilidad.

Se procede de esta forma hasta haber programado todas las piezas calculando su situación temporal y el retraso asociado.

4.2 PROCEDIMIENTO PSEUDO-SPT

Se calcula para cada pieza su CR_i asociado, según (10). A continuación se calcula el valor $CR_{min} = \min CR_i$ y se toma como valor umbral el resultado de (11).

$$Umbral = (1 + \beta) \cdot CR_{min} \quad (11)$$

En la implementación realizada $\beta=0,2$. Posteriormente se elige al azar una de las piezas cuyo valor $CR_i \leq umbral$ y se asigna a la máquina que ha proporcionado dicho valor. El procedimiento continúa hasta que se han programado todas las piezas.

4.3 PROCEDIMIENTO RANDOM

Se genera una permutación de las n piezas al azar y se asigna a las máquinas mediante el algoritmo *grinder* propuesto en Ribas (2007).

El algoritmo *grinder* partiendo de una secuencia de piezas, procede a la asignación, secuenciación y temporización de las piezas progresivamente, una después de otra. El criterio de asignación utilizado es la asignación de la pieza a la máquina que la termine antes. Sin tiempos de preparación dependientes de la secuencia, siendo todas las máquinas idénticas, las piezas se podrían asignar a la primera máquina disponible, pero cuando existen tiempos de preparación dependientes de la secuencia, el tiempo requerido para finalizar la pieza en una máquina depende de la pieza acabada de fabricar y, por lo tanto, el criterio más eficiente es asignar la pieza a la máquina que la termine antes.

4.4 PROCEDIMIENTO DE MEJORA SSA

En este procedimiento de mejora se explora el vecindario de una solución representada por una secuencia de piezas. Para valorar la solución se asigna las piezas a las máquinas a través del algoritmo *grinder* explicado anteriormente. El procedimiento implementado es una variante del algoritmo no exhaustivo de descenso (ANED) que, partiendo de una solución en curso, genera y evalúa en cierto orden sus vecinos. Un vecino se genera por intercambio de las piezas situadas en posiciones contiguas. Si uno de los vecinos es mejor que la solución en curso se toma como nueva solución y se prosigue aplicando el procedimiento a los vecinos de la nueva solución. El procedimiento termina cuando se han generado todos los vecinos de la solución en curso sin que ninguno sea mejor.

En esta variante hemos incorporado dos herramientas que permiten hacer frente a dos problemas existentes en muchos procedimientos de exploración: la existencia, en el espacio de las soluciones, de zonas en las que el valor de la función objetivo es la misma y en las cuales la exploración se encalla, y el orden en que se explora el vecindario que suele ser prefijado. Para superar el primer inconveniente se aceptan soluciones de igual valor con cierta probabilidad y para el segundo, se incorpora un vector de posiciones (que llamamos técnica revolver) que permite recorrer el vecindario de forma aleatoria.

4.5 PROCEDIMIENTO DE MEJORA VECINDARIO MÁQUINA

El procedimiento de mejora que hemos llamado *vecindario máquina* parte de una solución inicial representada por la secuencia de piezas a fabricar en cada máquina y genera y analiza su vecindario mediante tres movimientos. Cada movimiento permite explorar una zona del vecindario. El primer movimiento modifica la secuencia de fabricación en cada máquina, intercambiando piezas de posiciones contiguas mediante la aplicación del procedimiento SSA, definido anteriormente. El segundo, intercambia piezas de máquinas y, si se produce alguna mejoría, se aplica de nuevo la mejora SSA sobre cada máquina. El tercer movimiento extrae una pieza de la secuencia de una máquina y la inserta en otra máquina y, nuevamente si hay mejoría, se aplica la mejora SSA sobre cada máquina.

5. EXPERIENCIA COMPUTACIONAL

Para ejemplares de pequeña dimensión como el mostrado en el apartado 3.1 cualquiera de las heurísticas proporciona con facilidad la solución óptima. Por ello hemos desarrollado una experiencia computacional con ejemplares de mayor dimensión. Se ha construido un generador de ejemplares que sigue la filosofía propuesta en Potts y Van Wassenhove (1982) para el problema de secuenciar piezas en una máquina con tiempos de preparación dependientes de la secuencia. Está basado en cuatro parámetros: P_{max} , λ , T y R (además de n y m). Para cada ejemplar los tiempos de proceso se distribuyen uniformemente entre 1 y P_{max} , los tiempos de preparación entre 1 y S_{max} , siendo $S_{max} = \lambda \cdot P_{max}$, efectuando las correcciones necesarias para garantizar la desigualdad triangular. Los tiempos de preparación iniciales, en las máquinas, se distribuyen uniformemente entre 1 y $S_{max}/2$. Las fechas de vencimiento se distribuyen uniformemente entre $C_{max} \cdot (1 - T/R/2)$ y $C_{max} \cdot (1 + T/R/2)$ donde

$$C_{max} = \frac{n}{m} \cdot \frac{2 + P_{max} + S_{max}}{2} - \frac{1 + S_{max}}{4}$$

es una estimación de la ocupación del taller. Las colecciones generadas están formadas por 10 ejemplares cada una y son combinación de tres valores de $(0.5, 0.8, 1.2)$, 4 valores de $T (0.2, 0.4, 0.6, 0.8)$ y 4 valores de $R (0.2, 0.4, 0.6, 0.8)$. Un total de 48 colecciones para cada combinación de número de piezas y máquinas. Se han generado ejemplares de 15 piezas con 2 máquinas, 20 piezas con 2 y 3 máquinas y 50 piezas con 3 y 4 máquinas. Las colecciones se han codificado con 4 dígitos. El primero indica el número de máquinas, el segundo el nivel de λ , representado por (5, 8 y 1) respectivamente y el tercer y cuarto dígitos corresponden al nivel de T y R codificado como 2, 4, 6 y 8.

Se han realizado dos tests en un Pentium IV con 512 MB RAM y un procesador de 2.8 GHz. El primer test tiene como objetivo analizar la influencia de la solución inicial sobre el resultado final, en función del vecindario implementado. En el segundo test se analiza el comportamiento de los algoritmos cuando se varía el peso asignado a cada criterio.

Para comparar los procedimientos se utiliza el índice I_j definido según (12):

$$I_j = \frac{alg_j - \min}{\max - \min} \quad (12)$$

Donde alg_j indica el valor obtenido por el procedimiento j , $\min$ es el valor mínimo obtenido por cualquiera de los procedimientos y $\max$ es el mayor valor obtenido por cualquiera de los procedimientos.

5.1 ANÁLISIS DE LOS RESULTADOS DEL PRIMER TEST

En este test se quiere comprobar si la calidad de la solución inicial determina la calidad de la solución final tanto si se utiliza el procedimiento de mejora SSA o el que hemos llamado *vecindario máquina*. Además se quiere

analizar si el número de máquinas afecta a la eficacia o a las conclusiones. Para ello hemos representado, gráficamente, el valor promedio de I_j por colección, separando las colecciones según el número de piezas y máquinas consideradas.

Los resultados obtenidos por cada uno de los procedimientos, en las colecciones de 15 piezas y 2 máquinas, se pueden ver en la figura 3. En cada gráfico se han representado los resultados obtenidos según el procedimiento de mejora utilizado con el fin de comparar la influencia de la solución inicial sobre dichos procedimientos.

Se observa que la adaptación de la regla SPT no resulta eficiente para ninguno de los dos vecindarios, los valores promedio del índice I_j , por colección, son los que más se alejan del 0. También se observa que, para el procedimiento *vecindario máquina*, la solución inicial que lleva a mejores resultados es la obtenida de forma aleatoria (*Random*). Observamos que los valores promedio del índice I_j menores se obtienen con la solución inicial aleatoria y el procedimiento de mejora *vecindario máquina* por lo que esta combinación es la que resulta más eficiente en estas colecciones.

En la Figura 4 se han representado los valores obtenidos, en las colecciones de 20 piezas, con los procedimientos que utilizan la mejora SSA. El gráfico de la izquierda representa los valores para las colecciones con 2 máquinas y el de la derecha con 3 máquinas. En estos gráficos no se aprecia ninguna diferencia significativa. Los valores promedio del índice I_j presentan mucha dispersión sin que se pueda concluir que una de las soluciones iniciales es más aconsejable que otra.

Los valores obtenidos mediante los procedimientos que utilizan el *vecindario máquina* se han representado en la figura 5. En este gráfico sí se pueden observar diferencias entre el comportamiento de los algoritmos en las colecciones de 2 y 3 máquinas. En el gráfico de la

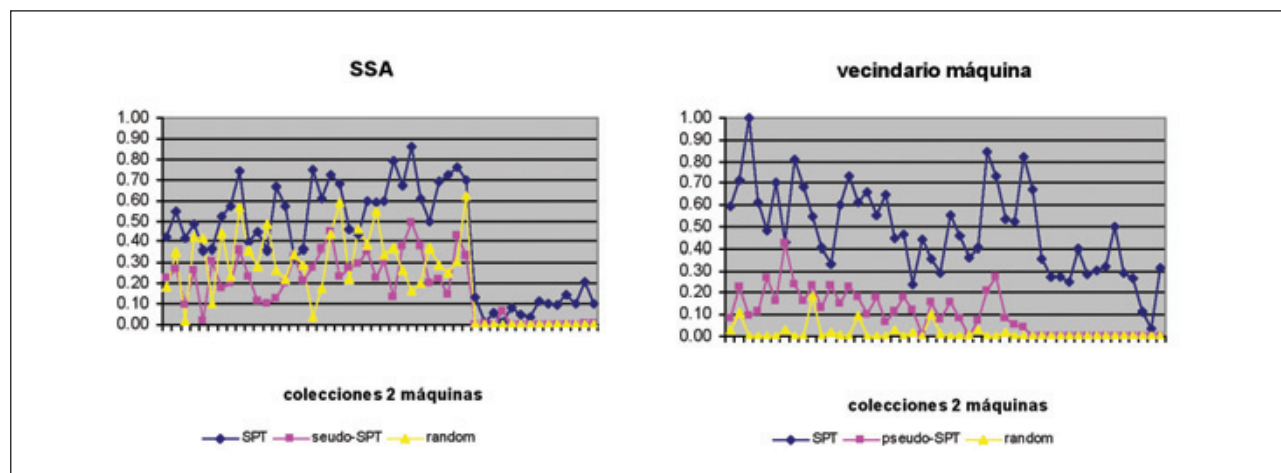


Figura 3. Valor de promedio de I_j para las colecciones de 15 piezas según vecindario utilizado.

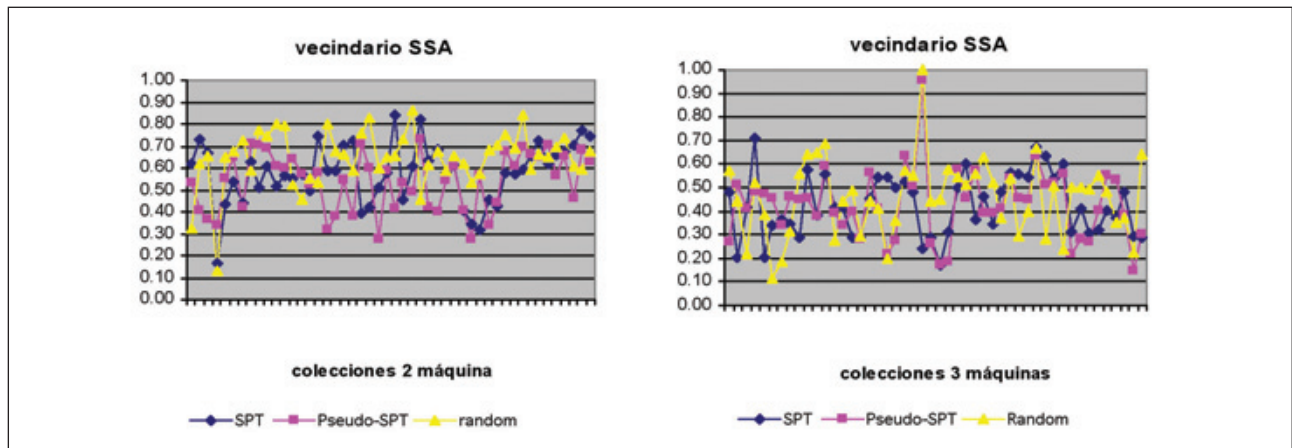


Figura 4. Valor promedio de I para las colecciones de 20 piezas, con 2 y 3 máquinas, para el vecindario. SSA

izquierda, colecciones con 20 piezas y 2 máquina, se observa que los mejores resultados se obtienen con la solución inicial *pseudo_SPT* o *random*, aunque ésta última presenta una gran dispersión ya que en algunas colecciones obtiene el mejor resultado y en otras casi el peor. En cambio, en las colecciones con 3 máquinas la solución inicial aleatoria (*Random*) es la que lleva a soluciones finales mejores, tal y como sucedía en las colecciones de 15 piezas.

comparando los gráficos de la Figura 7 que representan los valores obtenidos con los algoritmos que utilizan el procedimiento *vecindario máquina*.

En cambio, en la Figura 6 se observa que la solución inicial aleatoria, combinada con SSA no lleva a buenos resultados, mientras que para el *vecindario máquina*, Figura 7, no se puede concluir que un procedimiento inicial sea mejor que otro, ya que los valores promedio obtenidos, en cada colección, son muy similares.

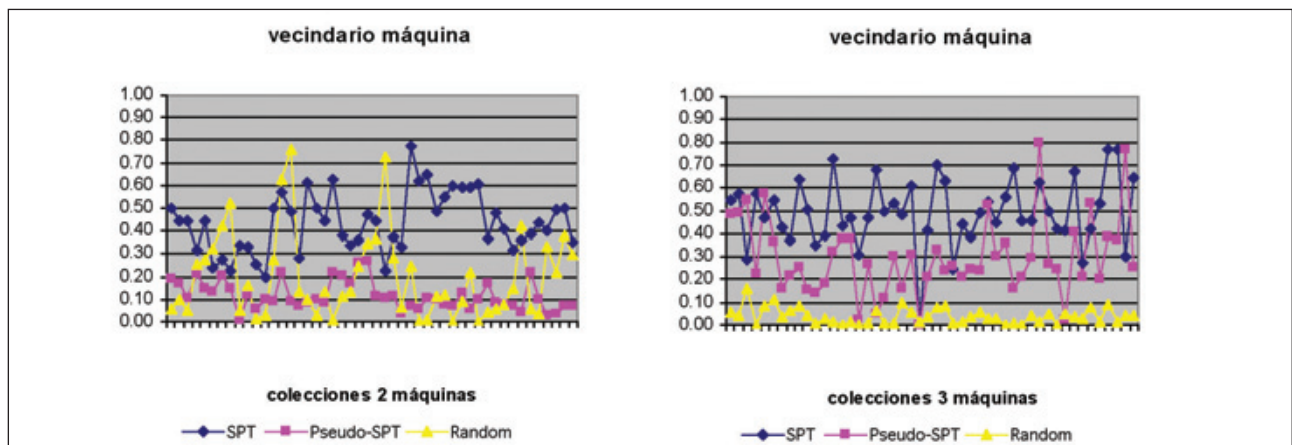


Figura 5. Valor promedio de I , en colecciones de 20 piezas, con 2 y 3 máquinas, para el vecindario máquina.

Si se comparan los gráficos de las figuras 4 y 5 concluimos que el procedimiento de mejora *vecindario máquina* también es el más aconsejable para las colecciones con 20 piezas.

Los valores obtenidos con los algoritmos que utilizan el procedimiento SSA, en las colecciones de 50 piezas con 3 y 4 máquinas, se muestran en la Figura 6. Comparando los dos gráficos entre sí no se observa un comportamiento diferente, en los algoritmos, atribuible al número de máquinas consideradas. Se llega a la misma conclusión

Los mejores resultados se han obtenido con los algoritmos que utilizan el procedimiento vecindario máquina

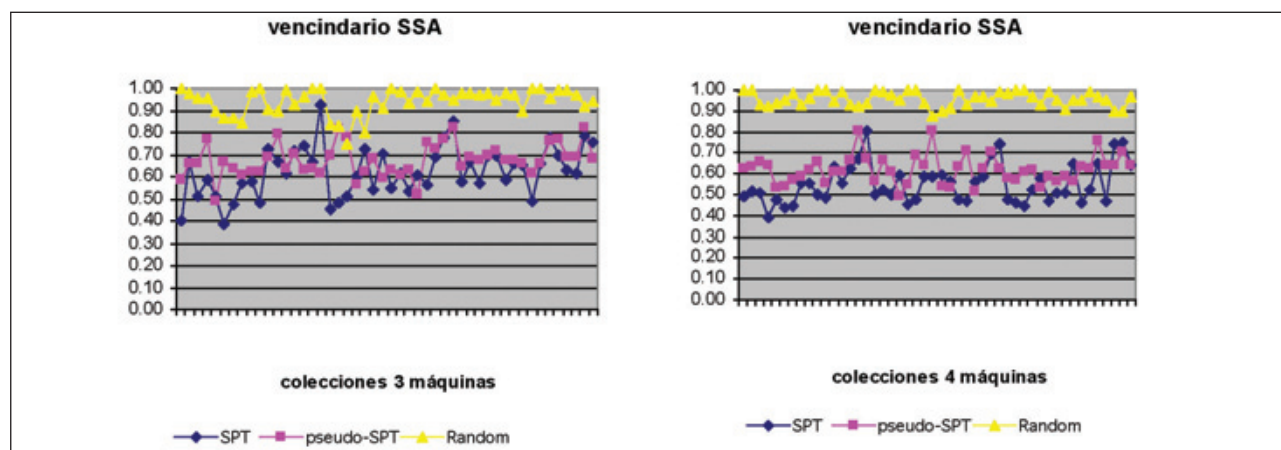


Figura 6. Valor promedio de I_j en las colecciones de 50 piezas, con 3 y 4 máquinas, para el vecindario SSA.

Comparando la Figura 6 y 7 se concluye que el procedimiento de mejora más eficiente es, también, el *vecindario máquina* ya que es con el que se obtienen valores del índice I_j menores.

5.2 ANÁLISIS DE LOS RESULTADOS DEL SEGUNDO TEST.

El segundo test se ha realizado con el fin de analizar si las conclusiones anteriores son válidas cuando el

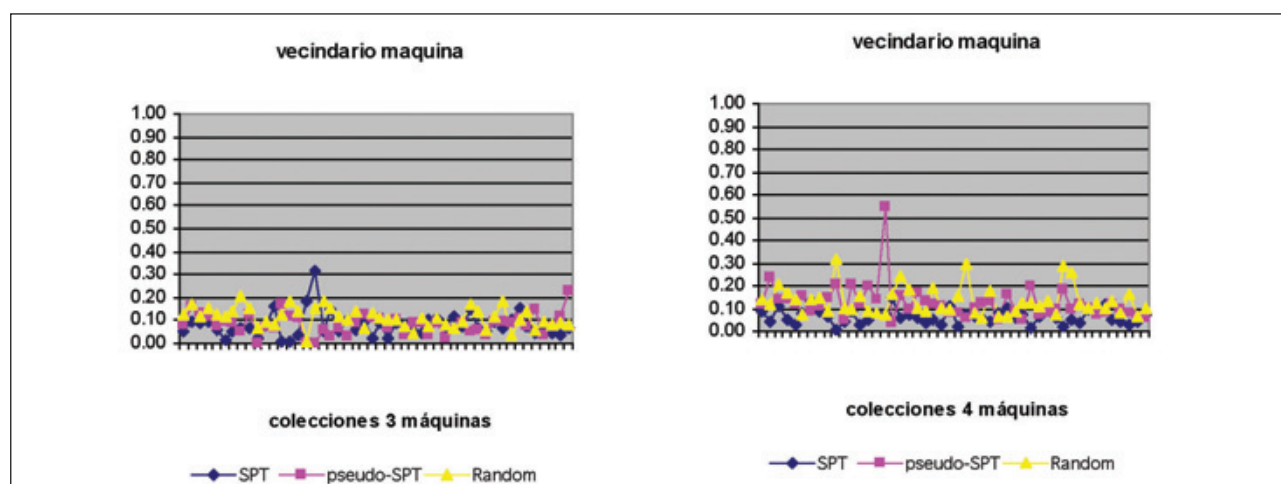


Figura 7. Valor promedio de I_j en las colecciones de 50 piezas, con 3 y 4 máquinas, para el vecindario máquina.

Por lo tanto, con los resultados obtenidos en el primer test podemos concluir que el procedimiento de mejora vecindario máquina es más eficaz que el procedimiento de mejora SSA. Cabe decir que el primero explora un vecindario más amplio que el segundo ya que realiza 3 movimientos que le permiten explorar tanto el entorno de cada máquina como el vecindario entre máquinas. También podemos concluir que la solución inicial obtenida de forma aleatoria es la que ha llevado a soluciones en la mayoría de las situaciones analizadas. En el siguiente test se analiza la validez de esta conclusión si se modifica el peso asignado a cada uno de los criterios de la función objetivo.

planificador varía el peso asignado a cada criterio. En este test se han elegido 3 colecciones de 50 piezas y 3 máquinas que tienen un retraso considerable, ya que en muchos ejemplares utilizados el retraso es nulo o de orden muy inferior al plazo medio de finalización. Las 3 colecciones elegidas tienen un retraso del orden del 10% del plazo medio de fabricación. Por esta razón, el test realizado analiza los resultados cuando el peso asignado al retraso es del 90, 95 y 97 %.

En este caso, se ha modificado el cálculo del índice $CR_{j,i}$, utilizado en el procedimiento SPT, para que también considere la fecha de vencimiento asociada a la pieza, la

ecuación que define ahora al índice $CR_{j,i}$ es la descrita en (13).

$$CR_{j,i}(t) = \alpha \cdot d_i + (1 - \alpha) \cdot (\max\{\tau_j + s_{h,g(i)}, r_i\} + p_i) \quad (13)$$

A la solución inicial que utiliza este índice le llamaremos, de ahora en adelante, CR. Al haber modificado el índice también queda modificada la solución inicial que hasta ahora llamábamos *pseudo-SPT*; aunque mantiene la misma estructura, trabaja sobre el nuevo índice y, siendo coherentes con la nomenclatura, pasaremos a llamar *pseudo-CR*. De esta forma, los 6 algoritmos ahora implementados son combinación de los procedimientos de obtención de una solución inicial CR, *pseudo-CR* y *Random* y los procedimientos de mejora SSA y *vecindario máquina*.

procedimiento *vecindario máquina* está entre el 0.1 y 0.24, los valores promedio de las heurísticas constituidas por el procedimiento SSA está entre el 0.69 y 0.83. Sin embargo, no resulta evidente sacar ninguna conclusión respecto a la eficacia en el uso de un procedimiento u otro para obtener la solución inicial.

Los procedimientos descritos son fácilmente adaptables a la mayoría de los contextos industriales con una problemática similar a la descrita

	CR SSA	CR vec. máq	Pseudo-CR SSA	Pseudo-CR vec. máq	Random SSA	Random vec.máq
$\alpha = 0.9$	0.77	0.13	0.74	0.25	0.73	0.22
	0.65	0.15	0.67	0.11	0.85	0.11
	0.64	0.13	0.70	0.09	0.70	0.13
promedio	0.69	0.14	0.70	0.15	0.76	0.15
$\alpha = 0.95$	0.77	0.05	0.70	0.12	0.85	0.17
	0.68	0.18	0.77	0.06	0.72	0.19
	0.73	0.24	0.66	0.12	0.91	0.14
promedio	0.73	0.16	0.71	0.10	0.83	0.17
$\alpha = 0.97$	0.69	0.16	0.80	0.09	0.83	0.23
	0.74	0.06	0.80	0.13	0.73	0.26
	0.75	0.17	0.63	0.09	0.86	0.22
promedio	0.73	0.13	0.74	0.10	0.81	0.24

Tabla 1. Valor promedio de I_j por colección y procedimiento.

En la Tabla 1 se muestran los resultados obtenidos con cada procedimiento, para cada uno de los valores de α considerados. Comparando los valores promedio del índice I_j de los procedimientos que utilizan la misma forma de obtener la solución inicial con cada uno de los procedimientos de mejora, (CR_SSA respecto CR_vec.máq, Pseudo-CR_SSA respecto Pseudo-CR_vec.máq y Random_SSA respecto Random_vec.máq) observamos que el procedimiento *vecindario máquina* es con el que se obtiene los valores promedio menores. En las filas *promedio* hemos calculado el promedio de todas las colecciones para cada valor de α fijado. Observando estos valores vemos que mientras el valor promedio de todas las colecciones obtenidos con las heurísticas constituidas por el

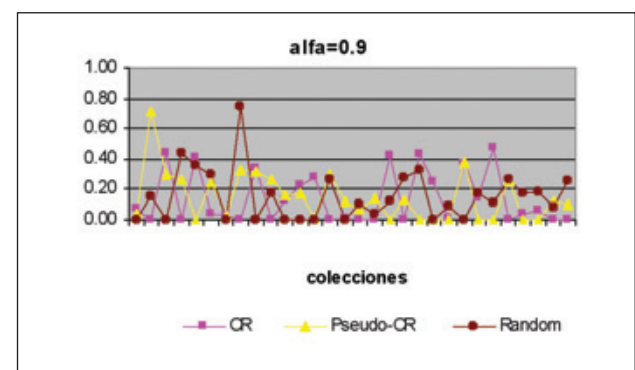


Figura 8. Valor de I_j para los ejemplares de las 3 colecciones de 50 piezas y 3 máquinas con $\alpha = 0.9$.

A continuación hemos analizado el valor del índice I_j obtenido por los algoritmos que utilizan el procedimiento de mejora vecindario máquina para cada uno de los 30 ejemplares que forman las 3 colecciones seleccionadas 3588, 3888 y 3188. Los valores obtenidos se representan en las Figuras 8 y 9.

Observando la Figura 8 vemos que, para un valor de α de 0.9, el comportamiento de los 3 procedimientos es similar. Hay mucha dispersión en los resultados y no se puede afirmar que uno sea mejor que el otro.

Sin embargo, comparando los gráficos de la Figura 9, observamos que a medida que los valores ponderados de ambos criterios están más equilibrados la utilización de una solución inicial más dirigida a los objetivos perseguidos permite obtener mejores soluciones.

La solución inicial obtenida de forma aleatoria es la que ha llevado a soluciones en la mayoría de las situaciones analizadas

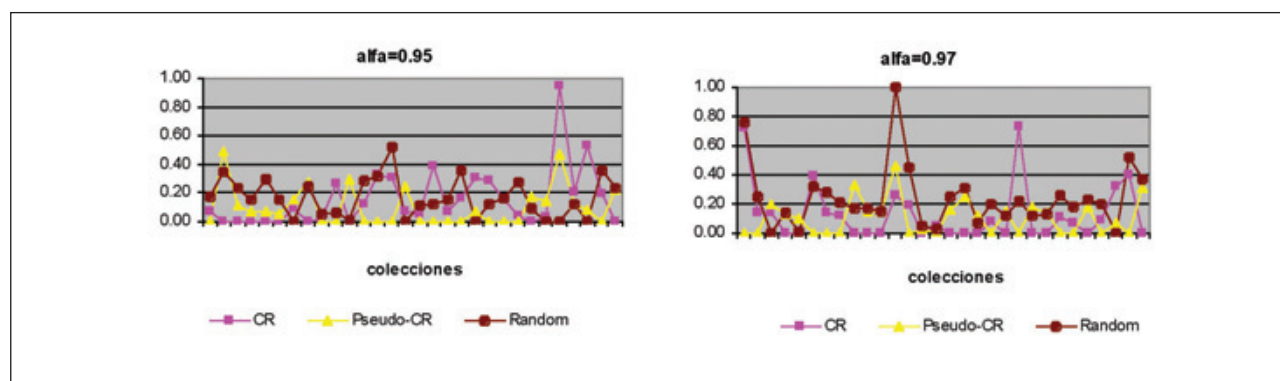


Figura 9. Valor de I_j para los ejemplares de las 3 colecciones de 50 piezas y 3 máquinas con $\alpha=0.95$ y $\alpha=0.97$.

Por lo tanto, con los resultados obtenidos en el segundo test podemos confirmar que el procedimiento de mejora que hemos llamado *vecindario máquina* es más adecuado para el problema de programación de piezas en máquinas idénticas en paralelo. En cambio, no podemos confirmar que la solución obtenida de forma aleatoria es siempre recomendable ya que hemos observado que el uso de una solución inicial dirigida a los criterios considerados es más eficaz cuando ambos criterios están equilibrados.

6. CONCLUSIONES

Se ha analizado el problema de programación bicriterio en un entorno productivo con máquinas idénticas en paralelo y tiempos de preparación dependientes de la secuencia. Los criterios de eficiencia considerados, válidos en muchos entornos industriales, son el retraso medio de las piezas y el plazo medio de fabricación. Se ha utilizado la suma ponderada de ambos criterios como función objetivo lo que permite que el factor de ponderación pueda ser modificado por el planificador según las necesidades concretas de cada momento. El estudio realizado ha analizado la eficiencia de 6 procedimientos que combinan

3 procedimientos simples para obtener una solución inicial con dos procedimientos de mejora que utilizan dos estructuras de vecindarios distintas, el utilizado normalmente para el caso de una máquina y un vecindario específico del problema en cuestión. El primer objetivo ha sido analizar cuál de los dos procedimientos de mejora implementados resulta más eficiente. El segundo objetivo ha sido determinar si los procedimientos de mejora utilizados son más eficaces con una solución inicial más o menos elaborada.

Los resultados obtenidos en los dos test realizados permiten concluir que con el procedimiento de mejora *vecindario máquina*, que utiliza una estructura de vecindario más adaptado al problema en cuestión, es más eficaz. En cambio las conclusiones obtenidas respecto a la eficiencia de la solución inicial no son tan evidentes. En el primer test, cuando el criterio predominante era minimizar el plazo medio de fabricación, concluimos que la combinación de la solución obtenida de forma aleatoria con el procedimiento de mejora vecindario máquina era la más eficiente. En este caso es posible que existan óptimos locales en el espacio de soluciones en el que las soluciones iniciales con significado queden atrapadas, mientras que la aleatoriedad de la solución obtenida al azar los permite

Los autores, además de estudiar la forma idónea de ajustar los procedimientos a casos concretos, están considerando diversas ampliaciones y prolongaciones del presente trabajo que pueden considerarse agrupables

evitar. En el segundo test, hemos observado que, a medida que los criterios de la función objetivo están más equilibrados, las soluciones iniciales obtenidas por reglas de despacho mixtas que tienen en cuenta dichos criterios tienen más impacto en la calidad de la solución.

Los procedimientos descritos son fácilmente adaptables a la mayoría de los contextos industriales con una problemática similar a la descrita. Tratándose de heurísticas es sencillo introducir restricciones adicionales, propias del caso concreto de que se trate, sin perjudicar excesivamente la eficiencia: la calidad de los resultados será similar aunque el tiempo de cálculo sea ligeramente superior.

Los procedimientos descritos no son tributarios de las condiciones iniciales, ya que no es una condición esencial que todas las piezas y todas las máquinas estén disponibles en el instante 0. De hecho las heurísticas constructivas que proporcionan la secuencia inicial tienen en cuenta el



cambio de instante de disponibilidad de las máquinas al ser las piezas asignadas a las mismas. Por consiguiente dichos procedimientos son utilizables en un contexto dinámico, en el cual existan ciclos periódicos de re-programación.

Los autores además de estudiar la forma idónea de ajustar los procedimientos a casos concretos (definiendo los valores más adecuados de los parámetros) están considerando diversas ampliaciones y prolongaciones del presente trabajo que pueden considerarse agrupables en tres líneas principales:

- La definición de nuevas variables medibles que reflejen mejor los objetivos de los directores de producción,
- La adaptación del algoritmo para la determinación de programas eficientes (situados sobre la frontera de Pareto) para que el responsable pueda elegir el programa más conveniente sin tener que fijar a priori el valor de α .
- El análisis de casos más complejos con mayor número de objetivos (procedimientos multi-objetivo).



7. AGRADECIMIENTOS

Este trabajo se enmarca en un proyecto de investigación financiado por el Ministerio de Educación y Ciencia con referencia DPI2007-61371, titulado “Programación de operaciones multicriterio con máquinas paralelas, en varias etapas, sin interrupciones ni almacenajes”.

8. BIBLIOGRAFÍA

- Arana P, Ochoa C, Zubillaga FJ. "Criterios para evaluar técnica y económicamente la aplicación de sistemas de mejora de gestión de la producción". *Dyna*, 1995. Vol. 5 p.6-12.
- Azizoğlu M, Webster S. "Scheduling Parallel Machines to Minimize Weighted Flowtime with Family Setup Times". *International Journal of Production Research*. 2003. Vol. 41 p.1199-1215.
- Cao D, Chen M, Wan G. "Parallel machine selection and job scheduling to minimize machine cost and job tardiness". *Computers and Operations Research*. 2005. Vol 32 p.1995-2012.
- Du J, Leung JYT. "Minimizing total tardiness on one machine is NP-hard". *Mathematics of Operations Research*. 1990. Vol.15 p. 483-495.
- Eren T. "A bicriteria parallel machine scheduling with a learning effect of setup and removal times". *Applied Mathematical Modelling*, 2009. Vol. 33 p.1141-1150.
- Lin BMT, Jeng AAK. "Parallel-machine batch scheduling to minimize the maximum lateness and the number of tardy jobs". *International Journal of Production Economics*. 2004. Vol. 91 p.121-134.
- Guinet AGP, Solomon MM. "Scheduling hybrid flowshop to minimize maximum tardiness or maximum completion time". *International Journal of Production Research*. 1996. Vol. 34-6 p. 1643-1654.
- Hwang CL, Massud ASM. *Multiple objective decision making-methods and applications*. 1ª edición. Heidelberg : Springer-Verlag, 1979. 351 p. Lecture Notes in Economics and Mathematical Systems; 164. ISBN: 3-540.09111-4.
- Kim SI, Choi HS, Lee DH. "Tabu search heuristics for parallel machine scheduling with sequence-dependent setup and ready times". In: Gavrilova,

M.; Gervasi, O.; Kumar, V.; Tan, C.J.K.; Taniar, D.; Laganà, A.; Mun, Y.; Choo, H. (Eds). . Berlin: Springer, 2006. Vol. 3982 p 728-737.

- Pfund M, Fowler JW, Amit GA, et al. "Scheduling jobs on parallel machines with setup times and ready times". *Computers & Industrial Engineering*. 2008. Vol 54 p. 764-782.
- Potts CN, Van Wassenhove LN. "A decomposition algorithm for the single machine tardiness problem". *Operations Research Letters*. 1982. Vol. 1 p. 177-181.
- Ribas-Vila, I. "Programación Multicriterio de un Sistema Productivo con Flujo Regular sin Esperas y Estaciones en Paralelo. Aplicación a una Fábrica de Helados". Directores: Companys-Pascual R, Lario-Esteban F. Universidad Politécnica de Valencia. Departamento de Organización de Empresas, 2007.
- T'kindt V, Billaut JC. "Multicriteria scheduling: theory, models and algorithms". 1ª edición. Heidelberg: Springer-Verlag, 2002. 303 p. ISBN: 3-540-43617-0.
- Webster S, Azizoğlu M. "Dynamic Programming Algorithms for Identical Parallel Machines with Family Setup Times". *Computers and Operations Research*. 2001, Vol. 28, p.127-137.

