

# ***Greedy Randomized Adaptive Search Procedure (GRASP), una alternativa valiosa en la minimización de la tardanza total ponderada en una máquina<sup>1</sup>***

**Greedy Randomized Adaptive Search Procedure (GRASP):  
A Valuable Alternative for Minimizing Machine Total Weighted  
Tardiness<sup>2</sup>**

***Greedy Randomized Adaptive Search Procedure (GRASP),  
uma alternativa valiosa na minimização da tardança total  
ponderada em uma máquina<sup>3</sup>***

*Juan Pablo Caballero-Villalobos<sup>4</sup>  
Jorge Andrés Alvarado-Valencia<sup>5</sup>*

---

<sup>1</sup> Fecha de recepción: 25 de enero de 2010. Fecha de aceptación para publicación: 16 de julio de 2010. Este artículo se deriva de un proyecto de investigación denominado *Funciones de estimación de costos y programación lineal, enfoque iterativo para abordar el problema de secuenciación en una máquina para el mejoramiento de la tardanza ponderada*, financiado por la Pontificia Universidad Javeriana con número de registro 2079.

<sup>2</sup> Submitted on: January 25, 2010. Accepted on: July 15, 2010. This article is the result of the research project *Cost Estimation Functions and Linear Programming: A Recurrent Approach to Study Machine Problems in order to improve Weighted Tardiness*, funded by the Pontificia Universidad Javeriana with registration number 2079.

<sup>3</sup> Data de recepção: 25 de janeiro de 2010. Data de aceitação para publicação: 16 de julho de 2010. Este artigo deriva-se de um projeto de pesquisa denominado *Funções de estimativa de custos e programação linear, enfoque iterativo para abordar o problema de sequenciamento em uma máquina para o melhoramento da tardança ponderada*, financiado pela Pontificia Universidade Javeriana com número de registro 2079.

<sup>4</sup> Ingeniero Industrial, Pontificia Universidad Javeriana, Bogotá, Colombia. Magíster en Ingeniería Industrial de la Universidad de los Andes, Bogotá, Colombia. Profesor asistente, Departamento de Ingeniería Industrial, Pontificia Universidad Javeriana. Correo electrónico: [juan.caballero@javeriana.edu.co](mailto:juan.caballero@javeriana.edu.co).

<sup>5</sup> Ingeniero Industrial, Pontificia Universidad Javeriana, Bogotá, Colombia. Magíster in Analytics, North Carolina State University, Estados Unidos. Profesor asistente, Departamento de Ingeniería Industrial, Pontificia Universidad Javeriana. Correo electrónico: [jorge.alvarado@javeriana.edu.co](mailto:jorge.alvarado@javeriana.edu.co).

### Resumen

Este artículo presenta los resultados experimentales obtenidos de secuenciar trabajos en una máquina, a fin de minimizar la tardanza total ponderada mediante un algoritmo GRASP. Los resultados se compararon con los valores óptimos o mejores valores reportados hasta el momento para cada una de las instancias de *OR-Library* y se encontró una excelente relación entre la calidad de los resultados (93% de las instancias se solucionaron con una desviación máxima del 1% respecto a estos valores) y el esfuerzo computacional y de implementación requerido. El algoritmo se implementó usando macros en una hoja de cálculo. La fase de postoptimización se realizó mediante una estrategia de Búsqueda Local que utilizó reglas de dominancia que, aun cuando sencillas, permitieron mejorar sustancialmente la tardanza total ponderada de las secuencias obtenidas en la fase constructiva del algoritmo.

### Palabras clave

Programación de la producción, metaheurísticas, tardanza ponderada, GRASP.

### Abstract

A GRASP algorithm was implemented in a common spreadsheet for single machine scheduling total weighted tardiness problem, and was tested with *OR-Library* instances. Results were compared with optimum or best known schedules for each instance, yielding less than 1% of difference in 93% of the cases, which results in an excellent tradeoff among results quality, computational effort and implementation easiness. Local search was performed on the post-optimization phase, based on dominance rules, which yielded even better results with little implementation effort.

### Key words

Production programming, metaheuristics, weighted tardiness problem, GRASP.

### Resumo

Este artigo apresenta os resultados experimentais obtidos ao sequenciar trabalhos em uma máquina, a fim de minimizar a tardança total ponderada mediante um algoritmo GRASP. Os resultados compararam-se com os valores ótimos ou os melhores valores conhecidos até o momento para cada uma das instâncias de *OR-Library* e encontrou-se uma excelente relação entre a qualidade dos resultados (93% das instâncias solucionaram-se com um desvio máximo de 1% com respeito a estes valores) e o esforço computacional e de implementação requerido. O algoritmo implementou-se usando macros em uma planilha de cálculo. A fase de pós-otimização realizou-se mediante uma estratégia de Busca Local que utilizou regras de dominância que, ainda que simples, permitiram melhorar substancialmente a tardança total ponderada das seqüências obtidas na fase construtiva do algoritmo.

### Palavras chave

Programação da produção, metaheurísticas, tardança ponderada, GRASP.

## Introducción

El problema aquí considerado se conoce como el problema de secuenciación en una máquina para la tardanza total ponderada, el cual se define así: sean  $n$  trabajos  $J_1, J_2, \dots, J_n$  y una sola máquina que puede procesar un solo trabajo a la vez. Cada trabajo  $J_j$  tiene un tiempo de procesamiento  $p_j$ , una fecha de entrega  $d_j$  y un costo de penalización por cada unidad de tiempo de retraso  $w_j$ . Todos los trabajos están disponibles en el tiempo a efectos de ser procesados. Para cada trabajo, dada una secuencia propuesta de procesamiento  $\pi$ , se obtiene un tiempo  $C_j(\pi)$  en el que dicha tarea será completada. La tardanza de cada trabajo se define como  $T_j(\pi) = \max(0, C_j(\pi) - d_j)$  y la tardanza ponderada de la secuencia completa de trabajos  $f(\pi)$  está dada por  $f(\pi) = \sum_{j=1}^n w_j T_j(\pi)$ . El objetivo es minimizar  $f(\pi)$  (Cheng *et al.*, 2005; Bilge, Kurtulan y Kirac, 2007).

El problema se conoce como  $n/1/\sum w_j T_j$  en la notación de Conway (Conway, Maxwell y Miller, 1967) o como  $1||\sum_{j=1}^n w_j T_j$  en la notación de Graham (Graham *et al.*, 1979), y será mencionado a lo largo del artículo como el problema SMTWT (por su sigla en inglés). Debido a la facilidad de implementación de los algoritmos GRASP (Ribeiro, Martins y Rosseti, 2007), la investigación se orientó a probar la pertinencia y calidad de los resultados obtenidos mediante esta técnica para el problema específico SMTWT, los cuales podrían ser de importancia en pequeñas y medianas empresas con presupuestos limitados que sí podrían utilizar una hoja de cálculo y una metaheurística de fácil implementación para lograr resultados competitivos y así apropiarse el GRASP como alternativa valiosa en la solución del problema SMTWT.

Este artículo está organizado de la siguiente forma: en la primera sección se muestra una revisión de las diferentes técnicas empleadas para abordar el problema SMTWT. La segunda describe los elementos teóricos componentes de los algoritmos que implementan técnicas GRASP. La tercera sección presenta la metodología utilizada, incluidas las características diferenciadoras del algoritmo implementado GRASP\_1TWT respecto a los elementos usualmente

recomendados para los algoritmos basados en GRASP, la metodología propuesta de análisis y la descripción del conjunto de problemas con los que se trabaja. La cuarta presenta los resultados experimentales obtenidos con las instancias de *benchmarking*. Por último, se concluye a partir de los resultados y se plantean las recomendaciones de los autores para futuras investigaciones.

### 1. Revisión de técnicas usadas en $1 \parallel \sum_{i=1}^n w_i T_i$

El problema SMTWT ha sido clasificado como *NP-duro*, en sentido fuerte (Lawler, 1977; Lenstra, Rinnooy Kan y Brucker, 1977), de acuerdo con la teoría de la complejidad computacional de los problemas de optimización combinatoria (Lenstra y Rinnooy Kan, 1979), lo que hace computacionalmente ineficiente el uso de algoritmos exactos de optimización para solucionarlo (Abdul-Razaq, Potts y Van Wassenhove, 1990).

Algoritmos como el de ramificación y acotamiento o la programación dinámica tienen complejidad de tiempo  $O(2^n)$  o superior para este problema (Cheng *et al.*, 2005). Sin embargo, en la generación de estos algoritmos se han encontrado reglas de dominancia útiles que facilitan el hallazgo de soluciones óptimas para el problema SMTWT, y uno de los primeros en plantear reglas de dominancia fue Emmons (1969), quien exploró el problema de la tardanza en una máquina sin ponderación planteando un conjunto de reglas de dominancia, algunas de las cuales fueron extensibles al problema SMTWT, que nos ocupa (en particular el teorema 1, los corolarios 2.1, 2.2 y el teorema 3).

Posteriormente (Rinnooy Kan, Lageweg y Lenstra, 1975) realizaron algunas extensiones al trabajo de Emmons. Baker y Bertrand (1982) plantearon una regla de prioridad dinámica perfeccionada por Rachamadugu (1987). Akturk y Yildirim (1998) plantearon una regla de dominancia que es condición suficiente para la detección de óptimos locales.

La opción alternativa al uso de algoritmos exactos de optimización es el empleo de técnicas heurísticas y metaheurísticas que pueden llevar a encontrar buenas soluciones con un menor esfuerzo computacional. En cuanto al problema SMTWT, no existe una regla de despacho que funcione bien en todos los tipos de problemas. Por ejemplo, la regla de la fecha de entrega más temprana (EDD, por su sigla en inglés) es óptima si solamente un trabajo estará tarde; entre tanto, el tiempo más corto de proceso ponderado (SWPT, por su sigla en inglés) dará la secuencia óptima si todos los trabajos están necesariamente tarde.

Generalmente, la EDD funciona mejor cuando la máquina tiene trabajo liviano, mientras que el SWPT funciona mejor con máquinas bastante carga-

das, donde muchos trabajos se entregarán tarde. Basados en esta idea, se han desarrollado diversas heurísticas como la de Carroll (1965), Rachamadugu y Morton (1982) y Panwalkar (Panwalkar, Smith y Koulamas, 1993). Se revisaron estos algoritmos (Chen, Potts y Woeginger, 1998) y se encontró que es difícil encontrar un desarrollo heurístico que dé garantías de desempeño.

En el campo de las metaheurísticas, Matsuo (Matsuo, Suh y Sullivan, 1989) y Potts y Van Waswihove (1991) usaron recocido simulado. Crauwels y su equipo (1998) compararon la búsqueda tabú, recocido simulado, *descent*, búsqueda umbral y algoritmos genéticos usando representaciones binarias o de permutación de secuencias y encontraron que la representación de permutaciones tenía mayores probabilidades de producir soluciones óptimas, pero la representación binaria daba consistentemente soluciones de buena calidad. La búsqueda tabú fue superior a las demás técnicas en este estudio.

Más adelante, Congram, Potts y Van de Velde (2002) abordaron el problema mediante una búsqueda dinámica iterada que usa programación dinámica para hallar el mejor movimiento, consistente en un conjunto de movimientos de intercambio independientes, seguido de una búsqueda de un vecindario de tamaño exponencial en tiempo polinomial. Bilge y su grupo (2007) desarrollaron un algoritmo basado en búsqueda tabú, cuyos resultados son mejores que la mayoría de los algoritmos conocidos. Dichos resultados son competitivos frente a la propuesta de Congram, Potts y Van de Velde (2002).

La revisión bibliográfica no evidenció el uso de GRASP de manera consistente para solucionar el problema SMTWT, salvo algunos trabajos relacionados de manera indirecta (Vega Mejía y Caballero-Villalobos, 2010).

## 2. GRASP

Un algoritmo de tipo *Greedy Randomized Adaptive Search Procedure* (GRASP) es una metaheurística iterativa multiarranque que en cada iteración realiza dos fases perfectamente definidas. La primera de ellas construye una solución factible al problema, adicionando uno a uno los elementos según un criterio de utilidad asignado a cada elemento. La segunda fase optimiza mediante la exploración del vecindario de soluciones hasta caer en un óptimo local.

Esta metaheurística ha sido aplicada a la resolución de múltiples problemas combinatorios de alta complejidad, por ejemplo: empaquetamiento de conjuntos (Delorme, Gandibleux y Rodríguez, 2004), *max-min diversity problem* (MMDP) (Resende *et al.*, 2010), programación de la producción en ambiente *Job Shop* (Aiex, Binato y Resende, 2003), corte y empaquetado (Álvarez-Valdés,

Parreno y Tamarit, 2008), programación de proyectos con recursos parcialmente renovables (Álvarez-Valdés *et al.*, 2008), cartero chino mixto (Corberan, Marti y Sanchis, 2002), recolección y distribución de un producto (Hernández-Pérez, Rodríguez-Martín y Salazar-González, 2009), entre otros.

### 2.1 GRASP: fase constructiva

La lógica de esta fase se ilustra en el siguiente pseudocódigo tomado de (Resende y Ribeiro, 2003), con el fin de facilitar su entendimiento:

```

Procedure Greedy_Randomized_Construction (Seed)
  Solution  $\leftarrow \emptyset$ ;
  Evaluate the incremental cost of the candidate elements;
  While Solution is not a complete solution do
    Build the restricted candidate list (RCL);
    Select an element  $s$  from the RCL at random;
    Solution  $\leftarrow$  Solution  $\cup \{ s \}$ ;
    Reevaluate the incremental cost;
  end;
  return Solution;
end Greedy_Randomized_Construction.

```

Donde la evaluación del costo o utilidad de cada uno de los elementos candidatos se calcula según una función que discrimine numéricamente y de manera adecuada la posibilidad de considerar cada elemento como el siguiente elemento que se va a incorporar en la solución. La construcción de la lista restringida de candidatos contempla el cálculo de un intervalo dentro del cual debe encontrarse el costo o utilidad de los posibles candidatos, de manera que se excluyan aquellos que no cumplan con esta condición.

Este intervalo se calcula con un parámetro  $\alpha$  que sirve para regular el grado de aleatoriedad que se le permitirá al algoritmo a través de la *lista restringida de candidatos* (RCL, por su sigla en inglés). Las expresiones usadas para calcular los intervalos en los casos de maximización y minimización son, respectivamente, las siguientes:

$[utilidad\_maxima - \alpha(utilidad\_maxima - utilidad\_minima), utilidad\_maxima]$   
y  $[utilidad\_minima, utilidad\_minima + \alpha(utilidad\_maxima - utilidad\_minima)]$ ,  
donde *utilidad\_maxima* y *utilidad\_minima* hacen referencia a la utilidad más

alta y más baja de alguno de los elementos que puede ser considerado siguiente elemento en la construcción de una solución factible.

## 2.2 GRASP: fase de optimización

La lógica de esta fase se basa en la exploración de soluciones adyacentes de forma que se detenga sólo cuando no se encuentren mejores soluciones en el vecindario definido. El funcionamiento de esta fase de búsqueda local se ilustra en el siguiente pseudocódigo tomado de (Resende y Ribeiro, 2003):

```

Procedure Local_Search (Solution)
    while Solution is not locally optimal do
        find  $s' \in N(\text{Solution})$  with  $f(s') < f(\text{Solution})$ ;
        Solution  $\leftarrow s'$ ;
    end;
    return Solution;
end Local_Search.

```

Donde el proceso de encontrar una solución adyacente que mejore la solución actual tiene decisiones importantes, como la definición de los posibles cambios que se pueden realizar entre los elementos implicando el tamaño del vecindario que se va a explorar y, por ende, los recursos que este proceso consumirá. Lo anterior teniendo siempre en mente que se debe conservar la factibilidad de las soluciones propuestas.

## 3. Metodología

### 3.1 Uso de GRASP\_1TWT

La implementación realizada para GRASP\_1TWT, esto es, GRASP *for Total Weighted Tardiness in a Single Machine*, se basó en la metaheurística GRASP e incluyó los siguientes aspectos diferenciadores respecto a las recomendaciones estándares encontradas en la literatura para los algoritmos basados en GRASP:

- Redefinió el factor  $a$  en la composición de la RCL como un número entero de mejores candidatos que pueden ser incluidos como próximo elemento en la construcción de la solución inicial, de forma que  $a$  ya no representa un porcentaje del rango asociado a las funciones de utilidad de los candidatos, sino que se convirtió en el tamaño de la RCL e incluyó sólo los mejores  $a$  candidatos factibles. La RCL se compone entonces de los  $a$  candidatos que presentan

los mayores valores de utilidad acorde con la valoración del retraso ponderado. La función de utilidad  $f$  evaluada para el  $j$ -ésimo candidato está expresada como:

$$f_j = w_j * (t + p_j - d_j)$$

Donde  $t$  es el tiempo actual del sistema o suma de los tiempos de procesamiento de los trabajos ya programados.

- Implementó un estrategia de *first-improve* en la fase de búsqueda local, analizando intercambios entre pares de trabajos  $i$  y  $j$  cuya condición es que al menos el trabajo  $j$  tenga retardo, que el trabajo  $i$  anteceda al trabajo  $j$  en la secuencia y que se cumpla la siguiente relación:

$$a + b + c < d \tag{1}$$

Dónde cada uno de los términos de la ecuación corresponden a:

- ✓  $\sum_{k=i+1}^{j-1} w_k * \max\{0, L_k + \Delta_{i,j}\}$
- ✓  $w_j * \max\{0, C_i + \Delta_{i,j} - d_j\}$
- ✓  $w_j * \max\{0, C_j - d_i\}$
- ✓  $\sum_{k=i}^j w_k * \max\{0, L_k\}$

Y los elementos básicos que conforman estas expresiones son:

$p_k$ : tiempo de procesamiento del trabajo  $k$ .

$d_k$ : fecha de entrega del trabajo  $k$ .

$w_k$ : peso ponderado del trabajo  $k$ .

$C_k$ : fecha de finalización del procesamiento del trabajo  $k$ .

$\Delta_{i,j}$ : adelanto/retardo de las fechas de finalización de los trabajos que se encuentran entre las posiciones de secuenciación  $i+1$  y  $j-1$ , se calcula como  $(p_j - p_i)$ .

$L_k$ : adelanto/retardo del trabajo  $k$ , se calcula como  $(C_k - d_k)$ .

Expresión que no es necesario evaluar exhaustivamente si se cumplen las relaciones

$$p_j < p_i, w_j \geq w_i \text{ y } d_j = d_i.$$

La relación expresada en la ecuación (1) valida la conveniencia de realizar el intercambio de un par de trabajos, ubicados en la  $i$ -ésima y  $j$ -ésima posición de la secuencia actual. El lado derecho y el lado izquierdo de la expresión corresponden, respectivamente, a las tardanzas ponderadas de los trabajos afectados antes y después del intercambio. A continuación se muestra la deducción de los términos que componen la ecuación (1).

En la secuenciación de trabajos en una máquina con fechas de *release* = 0 para todos los trabajos, se puede calcular el tiempo de finalización del trabajo en la  $i$ -ésima posición de la secuencia mediante  $C_i = C_{i-1} + p_i$ . Generalizando esta expresión, puede establecerse que para cualquier trabajo secuenciado  $n$  posiciones después del trabajo en la  $i$ -ésima posición se puede expresar su tiempo de finalización de procesamiento  $C_{i+n}$  como:

$$C_{i+n} = C_{i-1} + p_i + \sum_{k=i+1}^{i+n} p_k \Rightarrow C_{i+n} = C_i + \sum_{k=i+1}^{i+n} p_k \quad (i)$$

Si se intercambian los trabajos secuenciados en las posiciones  $i$ -ésima y  $j$ -ésima, el tiempo de finalización del nuevo trabajo en la  $i$ -ésima posición sería:  $C'_i = C_{i-1} + p_j$ , expresión que puede reescribirse como  $C'_i = C_{i-1} + p_j + p_i \Rightarrow C'_i = C_i + \Delta_{ij}$  donde  $\Delta_{ij} = p_j - p_i$ . De igual manera, el tiempo de finalización de cualquier trabajo secuenciado  $n$  posiciones después del trabajo secuenciado en la  $i$ -ésima posición cambiará acorde con la siguiente expresión:

$$C'_{i+n} = C'_i + \sum_{k=i+1}^{i+n} p_k \Rightarrow C'_{i+n} = C_i + \Delta_{ij} + \sum_{k=i+1}^{i+n} p_k \quad (ii)$$

Cambiando también el retardo y tardanza ponderada, para cualquier trabajo secuenciado  $n$  posiciones después del trabajo secuenciado en la  $i$ -ésima posición, conforme a:

$$L'_{i+n} = C'_{i+n} - d_{i+n} \Rightarrow L'_{i+n} = C_i + \Delta_{ij} + \sum_{k=i+1}^{i+n} p_k - d_{i+n}$$

$$T'_{i+n} = w_{i+n} * \max \{0, L'_{i+n}\} \Rightarrow T'_{i+n} = w_{i+n} * \max \left\{ 0, C_i + \Delta_{ij} + \sum_{k=i+1}^{i+n} p_k - d_{i+n} \right\} \quad (iii)$$

$$T'_{i+n} = w_{i+n} * \frac{\max}{\forall k > i, k < j} \{0, L_k + \Delta_{ij}\}$$

Esta última expresión permite contabilizar los cambios generados en la tardanza ponderada de todos los trabajos secuenciados entre las posiciones  $i$ -ésima y  $j$ -ésima de la secuencia original, al efectuar el intercambio de este par de trabajos, que originan el primer término de la ecuación (1).

$$a = \sum_{k=i+1}^{j-1} w_k * \max \{0, L_k + \Delta_{ij}\}$$

Las ecuaciones (ii) y (iii) pueden reformularse como:

$$C'_{i+n} = C'_i + \sum_{k=i}^{i+n} p_k - p_j \Rightarrow C'_{i+n} = C_i + \Delta_{ij} + \sum_{k=i}^{i+n} p_k - p_j$$

$$T'_{i+n} = w_{i+n} * \max \left\{ 0, C_i + \Delta_{ij} + \sum_{k=i}^{i+n} p_k - p_j - d_{i+n} \right\}$$

La última de las cuales, para el caso particular cuando  $n=0$ , tomará la forma del segundo término de la ecuación (1).

$$b = w_j * \max \{0, C_i + \Delta_{ij} - d_j\}$$

Se sabe que la suma de los tiempos de procesamiento de los trabajos secuenciados entre las posiciones  $i$ -ésima y  $j$ -ésima es una constante que no se modifica al efectuar alguna variación en el orden de procesamiento entre los trabajos. Lo anterior permite afirmar que el tiempo de finalización de procesamiento del último trabajo también es constante y que al realizar el intercambio de los trabajos en las posiciones  $i$ -ésima y  $j$ -ésima, es posible escribir la tardanza ponderada del último trabajo que se va a procesar según el tercer término de la ecuación (1):

$$c = w_i * \max \{0, C_j - d_i\}$$

Como corolario de la ecuación (1), se tiene que si  $p_j < p_i$  entonces  $\Delta_{ij} < 0$  y que, por lo tanto, los términos  $a$  y  $b$  decrecerán en valor. Adicionalmente, ante un posible incremento en la tardanza ponderada ocasionada por el término  $c$ , este sería igual o menor que el decremento obtenido en el término  $b$ , dadas las condiciones  $w_j \geq w_i$  y  $d_j = d_i$ .

### 3.2 Metodología de análisis de resultados

El algoritmo GRASP\_1TWT fue probado en un conjunto de instancias del problema SMTWT de 40, 50 y 100 trabajos, desarrollado y probado por (Crauwels, Potts y Van Wassenhove, 1998). Hay 125 instancias por cada uno de los tres tipos de problema. Estas se generaron aleatoriamente basándose en distribuciones uniformes en el rango [1-100] para los tiempos de procesamiento  $p_i$ , en el rango [1-10] para los pesos  $w_i$  y con una combinación de dos distribuciones uniformes para crear fechas de entrega  $d_i$  que generaran diferentes grados de dificultad (Beasley, 2001), de acuerdo con el esquema propuesto por (Potts y Van Wassenhove, 1985).

Las instancias están disponibles en la biblioteca de problemas de investigación de operaciones mantenida por (Beasley, 2001), que sirve de referencia mundial para la comparación de algoritmos y técnicas de solución en diferentes problemas de investigación de operaciones. En lo referente al problema SMTWT, esta biblioteca actualmente cuenta con soluciones óptimas para 124 problemas de los 125 de 40 trabajos y 115 de los 125 problemas de 50 trabajos, además de las mejores soluciones conocidas y reportadas en la literatura a la fecha para las 375 instancias.

En lo referente a la calidad de las soluciones obtenidas, estas se compararon contra los óptimos o las mejores soluciones conocidas para el conjunto de problemas, calculando la desviación porcentual frente a esos óptimos y el porcentaje o número total de veces que se alcanzó el valor óptimo o mejor conocido, tal como es usual en la literatura (Bilge, Kurtulan y Kirac, 2007; Congram, Potts y Van de Velde, 2002).

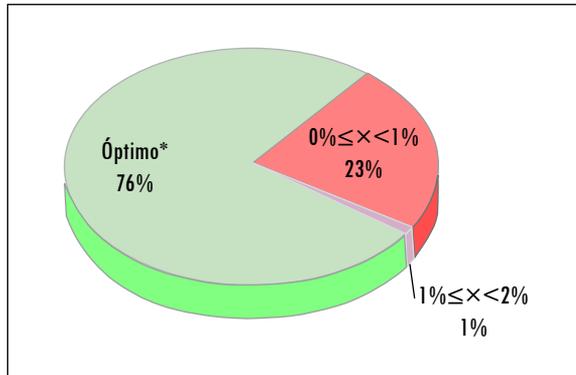
Respecto al tiempo de ejecución computacional, y puesto que se trata de un conjunto de instancias experimental generado aleatoriamente, se analizó el efecto del parámetro  $\alpha$  en dos niveles (3 y 5) y del número de trabajos en la instancia en tres niveles (40, 50 y 100) sobre los tiempos de ejecución mediante un análisis de varianza (Anova) factorial completo, lo que dio lugar a seis diferentes tratamientos. Al no verificarse las suposiciones del Anova, se utilizaron una prueba de Kruskal-Wallis y comparaciones múltiples de Tukey para detectar diferencias entre los tratamientos de cada factor.

#### 4. Resultados

Los resultados obtenidos mediante GRASP\_1TWT se encuentran disponibles en la página web del grupo de investigación Zentech, del Departamento de Ingeniería Industrial de la Pontificia Universidad Javeriana. Ahí están las secuencias de los trabajos encontradas por el algoritmo GRASP\_1TWT para cada una de las instancias, así como la información de las instancias de la biblioteca empleada y una tabla resumen de los resultados.

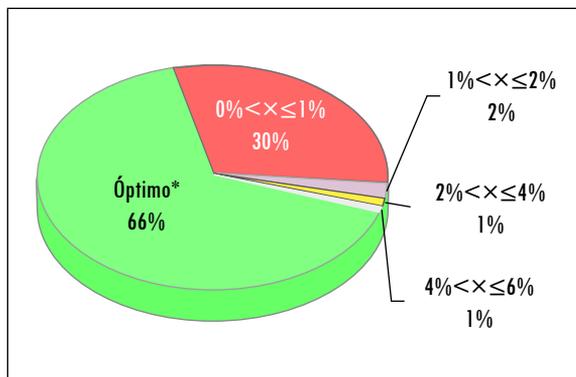
La comparación de los resultados en las 375 instancias de la biblioteca *OR-Library* respecto a las mejores soluciones reportadas para cada una de estos problemas se muestran en las figuras 1, 2 y 3, que individualizan los resultados acorde con el tamaño de las instancias. Por su parte, la Figura 4 compendia los resultados del conjunto total de instancias sin discriminar por su tamaño.

Figura 1. Calidad de secuencias encontradas por GRASP\_1TWT (40 trabajos)



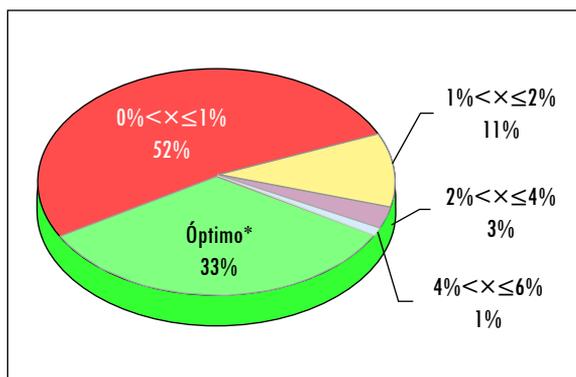
Fuente: presentación propia de los autores.

Figura 2. Calidad de secuencias encontradas por GRASP\_1TWT (50 trabajos)



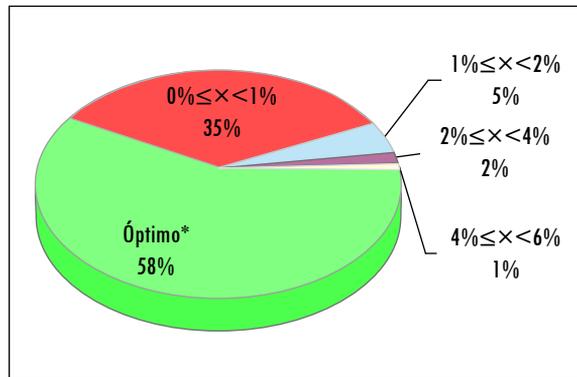
Fuente: presentación propia de los autores.

Figura 3. Calidad de secuencias encontradas por GRASP\_1TWT (100 trabajos)



Fuente: presentación propia de los autores.

Figura 4. Calidad de secuencias encontradas por GRASP\_1TWT (total instancias)



Fuente: presentación propia de los autores.

Las figuras 1, 2 y 3 agrupan las secuencias obtenidas acordes con un criterio de calidad —la desviación porcentual respecto a los valores óptimos o mejores conocidos de tardanzas ponderadas para estas instancias— y muestran que para las 125 instancias de 40 trabajos, el método GRASP empleado obtuvo en el 76% de los casos el mismo valor reportado como mejor y en un 23% de los casos la desviación se ubicó entre el 0% y el 1%. Ello permitió calcular que el 99% de las instancias de 40 trabajos se solucionaron con una desviación inferior al 1%. Para las 125 instancias de 50 trabajos las secuencias tienen el mismo valor de los mejores resultados conocidos en un 66% de los casos y 30% de estos sólo se desvían en 1%, lo cual implica que el 96% de las instancias se solucionaron con una desviación menor al 1%.

Finalmente, los resultados obtenidos para las 125 instancias más complejas (100 trabajos) se encontraron secuencias con el mismo valor de las mejores conocidas en el 33% de los casos y con una desviación máxima de 1% en el 52% de los casos; así mismo, en el 11% de los casos las soluciones obtenidas tuvieron desviación entre el 1% y 2%, lo que evidenció que, aun para estas instancias grandes, el método usado obtuvo soluciones con desviación inferior al 1% en el 85% de los casos y con desviación inferior al 2% en el 96% de los casos.

Al analizar los resultados, sin tener en cuenta las dimensiones de los problemas, la Figura 4 muestra que para el 58% de las 375 instancias de la biblioteca se igualaron los mejores valores reportados en la literatura, así como que en el 93% de los casos se logró encontrar una solución con desviación máxima del 1% y que en el 98% de las instancias se hallaron soluciones con desviación menor al 4%.

Los resultados ya mencionados de manera compendiada pueden verse individualizados en las tablas 1, 2 y 3, donde se hace una referencia a cada instancia con el valor óptimo o mejor valor reportado hasta el momento como solución, con el resultado obtenido por GRASP\_1TWT y con la desviación porcentual de este resultado respecto al mejor conocido. Las anteriores comparaciones evidencian la calidad del método propuesto y la pertinencia de su uso en este tipo de problemas. Estos resultados mostrados en las tablas 1, 2 y 3 se obtuvieron usando  $\alpha$  con valor de 3 y 5, ajuste hecho dependiendo de los resultados obtenidos para cada instancia, lo cual no presenta impedimento fuerte en tiempos de ejecución, acorde con los análisis presentados en la Figura 5.

**Tabla 1. Comparación resultados obtenidos instancias 40 trabajos *OR-Library***

Inst.	M. V.	GRASP	Desv. (%)	Inst.	M. V.	GRASP	Desv. (%)	Inst.	M.V.	GRASP	Desv. (%)
1	913	913	...	43	65.134	65.134	...	85	776	776	...
2	1.225	1.225	...	44	78.139	78.139	...	86	10.262	10.276	0,14
3	537	537	...	45	66.579	66.579	...	87	18.646	18.649	0,02
4	2.094	2.094	...	46	64.451	64.451	...	88	10.021	10.021	...
5	990	990	...	47	113.999	114.019	0,02	89	25.881	25.912	0,12
6	6.955	6.955	...	48	74.323	74.323	...	90	8.159	8.186	0,33
7	6.324	6.324	...	49	110.295	110.295	...	91	47.683	47.823	0,29
8	6.865	6.865	...	50	95.616	95.616	...	92	43.004	43.172	0,39
9	16.225	16.225	...	51	0	0	...	93	55.730	55.760	0,05
10	9.737	9.737	...	52	0	0	...	94	59.494	59.580	0,14
11	17.465	17.465	...	53	0	0	...	95	42.688	42.688	...
12	19.312	19.312	...	54	0	0	...	96	126.048	126.056	0,01
13	29.256	29.279	0,08	55	0	0	...	97	114.686	114.686	...
14	14.377	14.377	...	56	2.099	2.099	...	98	112.102	112.102	...
15	26.914	26.914	...	57	2.260	2.260	...	99	98.206	98.295	0,09
16	72.317	72.378	0,08	58	4.936	4.936	...	100	157.296	157.296	...
17	78.623	78.623	...	59	3.784	3.784	...	101	0	0	...
18	74.310	74.310	...	60	3.289	3.289	...	102	0	0	...
19	77.122	77.432	0,40	61	20.281	20.314	0,16	103	0	0	...
20	63.229	63.229	...	62	13.403	13.460	0,43	104	0	0	...
21	77.774	77.774	...	63	19.771	19.771	...	105	0	0	...
22	100.484	100.484	...	64	24.346	24.498	0,62	106	0	0	...
23	135.618	135.618	...	65	14.905	14.905	...	107	516	516	...
24	119.947	119.947	...	66	65.386	65.420	0,05	108	3.354	3.354	...
25	128.747	128.747	...	67	65.756	65.935	0,27	109	0	0	...

Continúa

Inst.	M. V.	GRASP	Desv. (%)	Inst.	M. V.	GRASP	Desv. (%)	Inst.	M.V.	GRASP	Desv. (%)
26	108	108	...	68	78.451	78.451	...	110	0	0	...
27	64	64	...	69	81.627	81.627	...	111	31.478	31.478	...
28	15	15	...	70	68.242	68.257	0,02	112	21.169	21.169	...
29	47	47	...	71	90.486	90.486	...	113	27.077	27.077	...
30	98	98	...	72	115.249	115.249	...	114	19.648	19.648	...
31	6.575	6.659	1,28	73	68.529	68.529	...	115	13.774	13.774	...
32	4.098	4.098	...	74	79.006	79.006	...	116	46.770	46.782	0,03
33	5.468	5.468	...	75	98.110	98.110	...	117	50.364	50.384	0,04
34	2.648	2.648	...	76	0	0	...	118	25.460	25.460	...
35	5.290	5.341	0,96	77	0	0	...	119	66.707	66.723	0,02
36	19.732	19.732	...	78	0	0	...	120	69.019	69.092	0,11
37	17.349	17.349	...	79	0	0	...	121	122.266	122.370	0,09
38	24.499	24.499	...	80	0	0	...	122	82.456	82.566	0,13
39	19.008	19.008	...	81	684	684	...	123	75.118	75.118	...
40	19.611	19.611	...	82	172	172	...	124	73.041	73.527	0,67
41	57.640	57.640	...	83	798	803	0,63	125	104.531	104.531	...
42	81.462	81.720	0,32	84	617	617	...				

Inst: número de instancia; M. V.: mejor valor conocido de la función objetivo;  
GRASP: valor de la función objetivo encontrada con la metodología GRASP;  
Desv. desviación porcentual relativa entre el mejor valor conocido y el valor GRASP.

Fuente: presentación propia de los autores.

Tabla 2. Comparación resultados obtenidos instancias 50 trabajos *OR-Library*

Inst.	M. V.	GRASP	Desv. (%)	Inst.	M. V.	GRASP	Desv. (%)	Inst.	M. V.	GRASP	Desv. (%)
1	2.134	2.134	...	43	84.126	84.126	...	85	3.780	3.796	0,42
2	1.996	1.998	0,10	44	123.893	124.186	0,24	86	20.751	20.762	0,05
3	2.583	2.583	...	45	79.883	79.983	0,13	87	36.053	36.230	0,49
4	2.691	2.691	...	46	157.505	157.505	...	88	28.268	28.504	0,83
5	1.518	1.604	5,67	47	133.289	133.289	...	89	28.846	29.291	1,54
6	26.276	26.276	...	48	191.099	191.099	...	90	15.451	15.669	1,41
7	11.403	11.403	...	49	150.279	150.279	...	91	89.298	89.445	0,16
8	8.499	8.499	...	50	198.076	198.076	...	92	66.340	66.774	0,65
9	9.884	9.884	...	51	0	0	...	93	61.060	61.125	0,11
10	10.655	10.655	...	52	0	0	...	94	42.453	42.468	0,04
11	43.504	43.504	...	53	0	0	...	95	56.522	56.679	0,28
12	36.378	36.378	...	54	0	0	...	96	177.909	177.975	0,04
13	45.383	45.383	...	55	0	0	...	97	139.591	139.771	0,13

Continúa

Inst.	M. V.	GRASP	Desv. (%)	Inst.	M. V.	GRASP	Desv. (%)	Inst.	M. V.	GRASP	Desv. (%)
14	51.785	51.974	0,36	56	1.258	1.258	...	98	148.906	148.996	0,06
15	38.934	38.934	...	57	3.679	3.679	...	99	179.264	179.295	0,02
16	87.902	87.902	...	58	2.522	2.522	...	100	120.108	120.108	...
17	84.260	84.260	...	59	3.770	3.770	...	101	0	0	...
18	104.795	104.795	...	60	5.904	5.904	...	102	0	0	...
19	89.299	89.369	0,08	61	25.212	25.320	0,43	103	0	0	...
20	72.316	72.316	...	62	17.337	17.343	0,03	104	0	0	...
21	214.546	214.555	0,00	63	30.729	30.729	...	105	0	0	...
22	150.800	150.800	...	64	18.082	18.561	2,65	106	0	0	...
23	224.025	224.025	...	65	25.028	25.049	0,08	107	1.717	1.722	0,29
24	116.015	116.015	...	66	76.878	76.879	0,00	108	0	0	...
25	240.179	240.179	...	67	85.413	85.413	...	109	6.185	6.195	0,16
26	2	2	...	68	92.756	93.534	0,84	110	1.295	1.295	...
27	4	4	...	69	77.930	78.122	0,25	111	27.310	27.551	0,88
28	755	755	...	70	74.750	74.825	0,10	112	15.867	16.100	1,47
29	99	99	...	71	150.580	150.580	...	113	35.106	35.106	...
30	22	22	...	72	131.680	131.703	0,02	114	15.467	15.472	0,03
31	9.934	9.934	...	73	98.494	98.511	0,02	115	10.574	10.574	...
32	7.178	7.178	...	74	135.394	135.446	0,04	116	35.727	35.727	...
33	4.674	4.674	...	75	135.677	135.683	0,00	117	71.922	71.986	0,09
34	4.017	4.017	...	76	0	0	...	118	65.433	65.433	...
35	6.459	6.459	...	77	0	0	...	119	106.043	106.043	...
36	34.892	35.425	1,53	78	0	0	...	120	101.665	101.979	0,31
37	22.739	22.740	0,00	79	0	0	...	121	78.315	78.390	0,10
38	29.467	29.467	...	80	0	0	...	122	119.925	119.986	0,05
39	49.352	49.352	...	81	816	816	...	123	101.157	101.170	0,01
40	26.423	26.423	...	82	4.879	4.879	...	124	139.488	139.600	0,08
41	71.111	71.629	0,73	83	973	973	...	125	110.392	110.444	0,05
42	90.163	90.163	...	84	508	508	...				

Inst: número de instancia; M. V.: mejor valor conocido de la función objetivo;  
 GRASP: valor de la función objetivo encontrada con la metodología GRASP;  
 Desv. desviación porcentual relativa entre el mejor valor conocido y el valor GRASP.

Fuente: presentación propia de los autores.

Tabla 3. Comparación resultados obtenidos instancias 100 trabajos *OR-Library*

Inst.	M. V.	GRASP	Desv. (%)	Inst.	M. V.	GRASP	Desv. (%)	Inst.	M. V.	GRASP	Desv. (%)
1	5.988	6.066	1,30	43	320.537	320.556	0,01	85	284	284	...
2	6.170	6.170	...	44	360.193	360.332	0,04	86	66.850	68.619	2,65
3	4.267	4.267	...	45	306.040	306.579	0,18	87	84.229	85.414	1,41
4	5.011	5.011	...	46	829.828	829.851	0,00	88	55.544	56.457	1,64
5	5.283	5.283	...	47	623.356	623.734	0,06	89	54.612	54.849	0,43
6	58.258	58.258	...	48	748.988	749.054	0,01	90	75.061	76.036	1,30
7	50.972	50.972	...	49	656.693	656.908	0,03	91	248.699	249.064	0,15
8	59.434	59.893	0,77	50	599.269	599.321	0,01	92	311.022	314.961	1,27
9	40.978	41.005	0,07	51	0	0	...	93	326.258	327.253	0,30
10	53.208	53.251	0,08	52	0	0	...	94	273.993	275.626	0,60
11	181.649	182.894	0,69	53	0	0	...	95	316.870	317.980	0,35
12	234.179	234.709	0,23	54	0	0	...	96	495.516	496.216	0,14
13	178.840	178.974	0,07	55	0	0	...	97	636.903	637.034	0,02
14	157.476	157.558	0,05	56	9.046	9.478	4,78	98	680.082	680.378	0,04
15	172.995	173.207	0,12	57	11.539	11.539	...	99	622.464	622.697	0,04
16	407.703	407.703	...	58	16.313	16.353	0,25	100	449.545	449.884	0,08
17	332.804	332.804	...	59	7.965	8.156	2,40	101	0	0	...
18	544.838	545.186	0,06	60	19.912	19.999	0,44	102	0	0	...
19	477.684	479.762	0,44	61	86.793	87.207	0,48	103	0	0	...
20	406.094	406.367	0,07	62	87.067	88.817	2,01	104	0	0	...
21	898.925	898.994	0,01	63	96.563	97.772	1,25	105	0	0	...
22	556.873	556.973	0,02	64	100.788	102.206	1,41	106	0	0	...
23	539.716	539.940	0,04	65	56.510	59.521	5,33	107	1.193	1.193	...
24	744.287	744.318	0,00	66	243.872	244.492	0,25	108	0	0	...
25	585.306	585.373	0,01	67	401.023	401.819	0,20	109	232	232	...
26	8	8	...	68	399.085	400.794	0,43	110	0	0	...
27	718	718	...	69	309.232	310.782	0,50	111	159.123	160.323	0,75
28	27	27	...	70	222.684	223.446	0,34	112	174.367	176.000	0,94
29	480	480	...	71	640.816	641.573	0,12	113	91.169	92.371	1,32
30	50	50	...	72	611.362	611.423	0,01	114	168.266	169.051	0,47
31	24.202	24.202	...	73	623.429	623.463	0,01	115	70.190	70.940	1,07
32	25.469	25.886	1,64	74	584.628	584.782	0,03	116	370.614	371.386	0,21
33	32.964	33.612	1,97	75	575.274	575.278	0,00	117	324.437	326.216	0,55
34	22.215	22.393	0,80	76	0	0	...	118	246.237	247.048	0,33
35	19.114	19.490	1,97	77	0	0	...	119	293.571	294.305	0,25
36	108.293	111.884	3,32	78	0	0	...	120	267.316	267.838	0,20
37	181.850	183.631	0,98	79	0	0	...	121	471.214	472.816	0,34
38	90.440	90.440	...	80	0	0	...	122	570.459	570.976	0,09

Continúa

Inst.	M. V.	GRASP	Desv. (%)	Inst.	M. V.	GRASP	Desv. (%)	Inst.	M. V.	GRASP	Desv. (%)
39	151.701	154.404	1,78	81	1.400	1.400	...	123	397.029	397.531	0,13
40	129.728	129.747	0,01	82	317	317	...	124	431.115	432.691	0,37
41	462.324	463.224	0,19	83	1.146	1.146	...	125	560.754	562.231	0,26
42	425.875	426.391	0,12	84	136	136	...				

Inst: número de instancia; M. V.: mejor valor conocido de la función objetivo;  
 GRASP: valor de la función objetivo encontrada con la metodología GRASP;  
 Desv. desviación porcentual relativa entre el mejor valor conocido y el valor GRASP.

Fuente: presentación propia de los autores.

La fase experimental se realizó en un computador Pentium IV de 1,8 MHz, con una giga de memoria RAM, sistema operativo Windows XP y Microsoft Excel 2003. Hubo un análisis de varianza factorial completo (Anova) con los tiempos de ejecución como variable de respuesta y los valores de  $\alpha$  y el número de trabajos como factores influyentes, teniéndose así en total dos factores con seis tratamientos. Puesto que los residuales no cumplieron los criterios de normalidad ni de homocedasticidad, se realizó una prueba de Kruskal-Wallis para cada factor, la cual resultó significativa para el número de trabajos (valor  $p < 0,001$ ) y no significativa para el valor de  $\alpha$ , lo que indica la existencia de diferencias estadísticamente apreciables en el tiempo de procesamiento únicamente debidas al número de trabajos.

Al llevar a cabo comparaciones múltiples por pares con el método de Tukey para el número de trabajos por instancias, se presentaron diferencias significativas en el tiempo de procesamiento entre los siguientes pares de números de trabajos: 100 trabajos *vs.* 50 trabajos (valor  $p < 0,001$ ) y 100 trabajos *vs.* 40 trabajos (valor  $p < 0,001$ ). Ello mostró que el tiempo de procesamiento fue significativamente superior para las instancias de 100 trabajos que para las instancias con un menor número de trabajos.

## 5. Conclusiones y recomendaciones

Este trabajo presenta la adaptación de una técnica metaheurística, de fácil implementación, que reporta soluciones de alta calidad para un problema complejo que aparece con frecuencia en entornos de producción en diversos sectores de la economía. La plataforma empleada apoya la idea de ampliar la utilización de *software* de fácil acceso, una hoja de cálculo como el caso de Calc de Open Office, KSpread de KOffice o Excel de Microsoft Office, que puede disminuir la necesidad

de recurrir a paquetes especializados en programación de la producción, con la consecuente disminución de costos en licenciamiento de *software*.

La calidad de los resultados sufre pequeñas disminuciones, explicadas por el tamaño de los problemas que se van a solucionar; sin embargo, las desviaciones de las tardanzas ponderadas obtenidas para la totalidad de las instancias consideradas muestran resultados de alta calidad: para el 58% de los problemas considerados (217) se consiguió igualar el mejor valor reportado en la literatura; para el 93% de los problemas (348) se logró obtener soluciones con desviación máxima de hasta el 1% y para el 98% de las instancias (367) se encontraron soluciones con desviación menor al 4%.

El comportamiento de los tiempos de ejecución del algoritmo se presenta creciente no lineal, como era de esperarse, con relación al número de trabajos de la instancia que se iba a solucionar; no obstante, su desempeño para instancias grandes de 100 trabajos no supera los dos minutos en promedio, lo que permite considerarlo una alternativa atractiva a la luz de la buena calidad de los resultados obtenidos.

Como trabajo futuro se debe investigar la posibilidad de aplicar esta técnica a problemas de programación de la producción en entornos más complejos, como máquinas paralelas, *flow shop* y *job shop*. Otra posible investigación futura estaría en comparar otros algoritmos de fácil implementación en hojas de cálculo para evaluar alternativas adicionales al GRASP para pymes.

## Referencias

- ABDUL-RAZAQ, T.; POTTS, C. y VAN WASSENHOVE, L. A survey of algorithms for the single machine total weighted tardiness scheduling problem. *Discrete Applied Mathematics*, 1990, vol. 26, núms. 2-3, pp. 235-253.
- AIEX, R.; BINATO, S. y RESENDE, M. Parallel GRASP with path-relinking for job shop scheduling. *Parallel Computing in Numerical Optimization*, 2003, vol. 29, núm. 4, pp. 393-430.
- AKTURK, M. y YILDIRIM, M. A new lower bounding scheme for the total weighted tardiness problem. *Computers and Operations Research*, 1998, vol. 25, núm. 4, pp. 265-278.
- ÁLVAREZ-VALDÉS, R.; PARRENO, F. y TAMARIT, J. Reactive GRASP for the strip-packing problem. *Computers & Operations Research*, 2008, vol. 33, núm. 8, pp. 1065-1083.
- ÁLVAREZ-VALDÉS, R. *et al.* GRASP and path relinking for project scheduling under partially renewable resources. *European Journal of Operational Research*, 2008, vol. 189, núm. 3, pp. 1153-1170.
- BAKER, K. y BERTRAND, J. A dynamic priority rule for scheduling against due-dates. *Journal of Operations Management*, 1982, vol. 3, núm. 1, pp. 37-42.

- BEASLEY, J. *OR-Library*. 2001 [web en línea]. <<http://people.brunel.ac.uk/~mastjjb/jeb/orlib/wtinfo.html>>. [Consulta: 10-07-2007].
- BILGE, U.; KURTULAN, M. y KIRAC, F. A tabu search algorithm for the single machine total weighted tardiness problem. *European Journal of Operations Research*, 2007, vol. 176, núm. 3, pp. 1423-1435.
- CARROLL, D. *Heuristic sequencing of single and multiple components*. PhD Dissertation. Boston, MA: Massachusetts Institute of Technology, 1965.
- CHEN, B.; POTTS, C. y WOEGINGER, G. A review of machine scheduling: Complexity, algorithms and approximability. En *Handbook of combinatorial optimization*. Boston: Kluwer Academic Publishers, 1998, pp. 21-169.
- CHENG, T. *et al.* Single machine scheduling to minimize total weighted tardiness. *European Journal of Operations Research*, 2005, vol. 165, núm. 2, pp. 423-443.
- CONGRAM, R.; POTTS, C. y VAN DE VELDE, S. An iterated dynasearch algorithm for the single machine total weighted tardiness scheduling problem. *Inform Journal on Computing*, 2002, vol. 14, núm. 1, pp. 52-67.
- CONWAY, R.; MAXWELL, W. y MILLER, L. *Theory of scheduling*. Reading, MA: Addison-Wesley, 1967.
- CORBERAN, A.; MARTI, R. y SANCHIS, J. A GRASP heuristic for the mixed Chinese postman problem. *European Journal of Operational Research*, 2002, vol. 142, núm. 1, pp. 70-80.
- CRAUWELS, H.; POTTS, C. y VAN WASSENHOVE, L. Local search heuristics for single machine total weighted tardiness problem. *Inform Journal on Computing*, 1998, vol. 10, núm. 3, pp. 341-350.
- DELORME, X.; GANDIBLEUX, X. y RODRÍGUEZ, J. GRASP for set packing problems. *European Journal of Operational Research*, 2004, vol. 153, núm. 3, pp. 564-580.
- EMMONS, H. One machine sequencing to minimize certain functions of Job tardiness. *Operations Research*, 1969, vol. 17, núm. 4, pp. 701-715.
- GRAHAM, R. *et al.* Optimisation and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics*, 1979, vol. 5, pp. 287-326.
- HERNÁNDEZ-PÉREZ, H.; RODRÍGUEZ-MARTÍN, I. y SALAZAR-GONZÁLEZ, J. A hybrid GRASP/VND heuristic for the one-commodity pickup-and-delivery traveling salesman problem. *Computers & Operations Research*, 2009, vol. 36, núm. 5, pp. 1639-1645.
- LAWLER, E. A pseudopolynomial algorithm for sequencing jobs to minimize total tardiness. *Annals of Discrete Mathematics*, 1977, vol. 1, pp. 331-342.
- LENSTRA, J. y RINNOOY KAN, A. Computational complexity of discrete optimization problems. *Annals of Discrete Mathematics*, 1979, vol. 4, pp. 121-140.
- LENSTRA, J.; RINNOOY KAN, A. y BRUCKER, P. Complexity of machine scheduling problems. *Annals of Discrete mathematics I*, 1977, vol. 7, pp. 343-362.
- MATSUO, H.; SUH, C. y SULLIVAN, R. A controlled search simulated annealing method for the single machine weighted tardiness problem. *Annals of Operations Research*, 1989, vol. 21, pp. 85-108.

- PANWALKAR, S.; SMITH, M. y KOULAMAS, C. A heuristic for the single machine tardiness problem. *European Journal of Operations Research*, 1993, vol. 70, pp. 304-310.
- POTTS, C. y VAN WASSENHOVE, L. A branch and bound algorithm for the total weighted tardiness problem. *Operations Research*, 1985, vol. 33, núm. 2, pp. 363-377.
- POTTS, C. y VAN WASSENHOVE, L. Single machine tardiness sequencing heuristics. *IIE Transactions*, 1991, vol. 23, núm. 4, pp. 346-354.
- RACHAMADUGU, R. A note in the weighted tardiness problem. *Operations Research*, 1987, vol. 35 núm. 3, pp. 450-452.
- RACHAMADUGU, R. y MORTON, T. *Myopic heuristics for the single machine weighted total tardiness problem*. Working Paper 30-82-83. Pittsburgh, PA: Carnegie Mellon University, 1982.
- RESENDE, M. y RIBEIRO, C. Greedy randomized adaptative search procedure. En *Handbook of metaheuristics*. Boston, MA: Kluwer Academic Publishers, 2003.
- RESENDE, M.; MARTI, R.; GALLEGO, M. y DUARTE, A. GRASP and path relinking for the max-min diversity problem. *Computers & Operations Research*, 2010, vol. 37, núm. 3, pp. 498-508.
- RIBEIRO, C.; MARTINS, S., y ROSSETI, I. Metaheuristics for optimization problems in computer communications. *Computer communication*, 2007, vol. 30, núm. 4, pp. 656-669.
- RINNOOY KAN, A.; LAGEWEG, B. y LENSTRA, J. Minimizing total costs in one-machine scheduling. *Operations Research*, 1975, vol. 23, núm. 5, pp. 908-927.
- VEGA MEJÍA, C. A. y CABALLERO-VILLALOBOS, J. P. Uso combinado de GRASP y Path-Relinking en la programación de producción para minimizar la tardanza total ponderada en una máquina. *Ingeniería y Universidad*, 2010, vol. 14, núm. 1, pp. 79-96.

