

## RESOLUCIÓN DEL RESPONSE TIME VARIABILITY PROBLEM MEDIANTE TABU SEARCH\*

**Dr. Albert Corominas, Ing. Alberto García-Villoria y Dr. Rafael Pastor**

Profesor Catedrático, Ayudante y Titular, respectivamente  
Instituto de Organización y Control de Sistemas Industriales (IOC).  
Universidad Politécnica de Cataluña (UPC).  
Av. Diagonal, 647, 08028. Barcelona.  
{albert.corominas/alberto.garcia-villoria/rafael.pastor@upc.edu}

### Resumen

El Response Time Variability Problem (RTVP) es un problema combinatorio de scheduling publicado recientemente en la literatura. Dicho problema de optimización combinatoria es muy fácil de formular pero muy difícil de resolver de forma exacta (es NP-hard). El RTVP se presenta cuando productos, clientes o tareas se han de secuenciar minimizando la variabilidad entre los instantes de tiempo en los que reciben los recursos que ellos necesitan. Este problema tiene una gran cantidad de aplicaciones reales: secuenciación de modelos en líneas de montaje mixtas, asignación de recursos a sistemas multiprocesadores, mantenimiento continuo, recogida de basuras o la secuenciación de anuncios en televisión. La Inteligencia Artificial dispone de herramientas eficientes, tales como las metaheurísticas, para resolver problemas combinatorios de scheduling complejos. En trabajos previos, el RTVP ha sido resuelto mediante varios algoritmos metaheurísticos provenientes de la Inteligencia Artificial (entre otros, las metaheurísticas multi-start, PSO y GRASP). En este trabajo se propone un algoritmo de búsqueda tabu (tabu search), el cual mejora los mejores resultados referenciados en la literatura.

**Palabras clave:** response time variability, metaheurísticas, tabu search, scheduling

## SOLVING THE RESPONSE TIME VARIABILITY PROBLEM BY MEANS OF A TABU SEARCH APPROACH<sup>□</sup>

### Abstract

The Response Time Variability Problem (RTVP) is a combinatorial scheduling problem that has recently appeared in the literature. Although this combinatorial optimisation problem is easy to formulate, it is very difficult to solve (it is NP-hard). The RTVP occurs whenever products, clients or jobs need to be sequenced so as to minimize variability in the time between the instants at which they receive the necessary resources. This problem has a broad range of real-world applications: to sequence on mixed-model

---

\* Investigación subvencionada por el Ministerio de Educación y Ciencia de España, proyecto DPI2007-61905, cofinanciado por FEDER.

<sup>□</sup> Supported by the Spanish Ministry of Education and Science under project DPI2007-61905; co-funded by the ERDF.

assembly lines, to resource allocation in computer multi-threaded systems, in the periodic machine maintenance problem, in the collection of waste and in the schedule of commercial videotapes for television. The field of Artificial Intelligence has provided us with efficient tools such as metaheuristic techniques for solving complex combinatorial scheduling problems. In previous studies, several metaheuristic algorithms (among others, a multi-start, a PSO and a GRASP algorithm) were proposed to solve the RTVP. In this study we propose a tabu search algorithm for the RTVP which improves the best results reported in the literature.

**Keywords:** response time variability, metaheuristics, tabu search, scheduling

## 1. Introducción

El concepto de secuencia *fair* (secuencia justa, imparcial, buena, ideal, regular) ha surgido de forma independiente en problemas de *scheduling* de diversos entornos. El objetivo común de ese tipo de problemas de *scheduling*, como es definido en Kubiak (2004), consiste en construir una secuencia *fair* utilizando  $n$  símbolos, de forma que el símbolo  $i$  ( $i = 1, \dots, n$ ) se presente  $d_i$  veces en la secuencia. Una secuencia *fair* es aquella que asigna, repartiendo de forma *fair*, las posiciones a cada símbolo  $i$  en cualquier subsecuencia. Este reparto *fair* o ideal de posiciones asignadas al símbolo  $i$  en una subsecuencia de longitud  $k$  es proporcional a la importancia relativa del símbolo  $i$  ( $d_i$ ) respecto al total de copias de los diferentes símbolos a repartir (que, obviamente, es igual a  $\sum_{i=1..n} d_i$ ). No existe una definición universal de *fairness*, ya que se pueden definir diversas métricas razonables en función del problema específico que se está considerando.

Considerando las diferentes definiciones de *fairness*, varios problemas de secuenciación *fair* han sido propuestos en la literatura, entre ellos el *Response Time Variability Problem* (RTVP). Este problema fue expuesto por primera vez en Waldspurger and Weihl (1994) y formalizado en Corominas et al. (2007). En el RTVP, la secuencia *fair* es aquella que minimiza la suma de la variabilidad de las distancias entre cualquier pareja de copias consecutivas de un mismo símbolo. En otras palabras, la distancia entre cualquier pareja de copias consecutivas de un mismo símbolo debería ser tan regular como sea posible (e, idealmente, constante).

El RTVP se presenta cuando productos, clientes o tareas se han de secuenciar minimizando la variabilidad entre los instantes de tiempo en los que reciben los recursos que ellos necesitan (Corominas et al., 2007). Este problema tiene una gran cantidad de aplicaciones reales: la secuenciación de modelos en líneas de montaje mixtas (Kubiak, 1993; Miltenburg, 1989); la asignación de recursos en sistemas multiprocesadores, tales como los servidores de redes o las aplicaciones de transmisión de videos (Dong et al., 1998; Waldspurger and Weihl, 1995); el mantenimiento continuo, cuando el tiempo entre servicios consecutivos en una misma máquina debe ser el mismo (Anily et al.,

1998; Wei and Liu, 1983); la recogida de basuras (Herrmann, 2007); o la secuenciación de anuncios en prensa o televisión (Bollapragada et al., 2004; Brusco, 2008).

El RTVP es un problema *NP-hard* (Corominas et al., 2007). Debido a que es un problema de optimización combinatoria difícil de resolver de forma óptima, en la literatura se han propuesto para su resolución diversos procedimientos heurísticos y metaheurísticos. Waldspurger and Weihl (1995) utilizan el método de distribución de escaños de Jefferson (Balinski and Young, 1982), un algoritmo heurístico *greedy* que ellos denominan como la técnica *stride scheduling*. Herrmann (2007) resuelve el RTVP aplicando un algoritmo heurístico basado en la técnica *stride scheduling*. Corominas et al. (2007) también proponen el método de Jefferson junto a otros cuatro algoritmos heurísticos. García et al. (2006) diseñan seis algoritmos metaheurísticos: un *multi-start*, un *greedy randomized adaptive search procedure* (GRASP) y cuatro variantes de un algoritmo basado en el *particle swarm optimization* (PSO) discreto. Un algoritmo *multi-start* mejorado, así como un nuevo algoritmo GRASP, también mejorado, son propuestos en Corominas et al. (2008); y otros diez algoritmos PSO discretos son propuestos en García-Villoria and Pastor (2009a). Un procedimiento *cross-entropy* es utilizado en García-Villoria et al. (2007). Un algoritmo basado en *Electromagnetism-like Mechanism* (EM) es propuesto en García-Villoria and Pastor (2009b). Finalmente, un algoritmo *Psychoclonal* es utilizado para resolver el RTVP en García-Villoria and Pastor (2008). Los mejores resultados reportados en la literatura hasta el momento se han conseguido con el algoritmo *psychoclonal* (García-Villoria and Pastor, 2008) y el algoritmo *multi-start* mejorado (Corominas et al., 2008).

Hasta la fecha, ningún algoritmo *tabu search* (TS) ha sido propuesto para resolver el RTVP. En este trabajo se propone un algoritmo TS para el RTVP que mejora los mejores resultados publicados hasta el momento en la literatura.

El resto del artículo está organizado como sigue: la Sección 2 presenta una definición formal del RTVP y describe, brevemente, los dos mejores procedimientos metaheurísticos publicados hasta el momento para resolver el problema de estudio; la Sección 3 propone un algoritmo TS para resolver el RTVP; la Sección 4 presenta el resultado del experimento computacional realizado; finalmente, varias conclusiones y sugerencias de trabajo futuro son expuestas en la Sección 5.

## 2. El Response Time Variability Problem

El RTVP puede ser formulado como sigue. Sea  $n$  el número de símbolos,  $d_i$  el número de copias a ser secuenciadas del símbolo  $i$  ( $i = 1, \dots, n$ ) y  $D$  en número total de copias a secuenciar ( $\sum_{i=1..n} d_i$ ). Sea  $s$  una solución de un ejemplar del RTVP, que consiste en una secuencia circular de copias ( $s = s_1 s_2 \dots s_D$ ), donde  $s_j$  es la copia secuenciada en la posición  $j$  de la secuencia  $s$ . Para cada símbolo  $i$  con  $d_i \geq 2$ , sea  $t_k^i$  la distancia entre las posiciones en las que se encuentran las copias  $k + 1$  y  $k$  del símbolo  $i$  (considerando que la distancia entre dos posiciones consecutivas de la secuencia es igual a 1). Como

se ha introducido, la secuencia es circular, de esta forma la posición 1 viene inmediatamente después de la posición  $D$ , y  $t_{d_i}^i$  es la distancia entre la primera copia del símbolo  $i$  en un ciclo y la última copia de ese mismo símbolo en el ciclo precedente. Sea  $\bar{t}_i$  la distancia media deseada entre dos copias consecutivas del símbolo  $i$  ( $\bar{t}_i = D/d_i$ ). El objetivo es minimizar la métrica llamada *response time variability* (RTV), que se define como la suma de los cuadrados de los errores respecto a las distancias  $\bar{t}_i$ . Como los símbolos  $i$  con  $d_i = 1$  no intervienen en el cálculo del RTV, se asume que para cada uno de esos símbolo  $t_1^i$  es igual a  $\bar{t}_i$ . De esta forma, el RTV se obtiene con la expresión  $RTV = \sum_{i=1}^n \sum_{k=1}^{d_i} (t_k^i - \bar{t}_i)^2$ .

Por ejemplo, sea  $n=3$  y los símbolos A, B y C; considérese, adicionalmente, que  $d_A = 2$ ,  $d_B = 2$  y  $d_C = 4$ ; así,  $D = 8$ ,  $\bar{t}_A = 4$ ,  $\bar{t}_B = 4$  y  $\bar{t}_C = 2$ . Cualquier secuencia que contenga el símbolo  $i$  ( $\forall i$ ) exactamente  $d_i$  veces es una solución factible. Por ejemplo, la secuencia (C, A, C, B, C, B, A, C) es una solución factible con el siguiente valor:  $RTV = \left( (5-4)^2 + (3-4)^2 \right) + \left( (2-4)^2 + (6-4)^2 \right) + \left( (2-2)^2 + (2-2)^2 + (3-2)^2 + (1-2)^2 \right) = 12$ .

Como se ha introducido en la Sección 1, el algoritmo *psychoclonal* propuesto en García-Villoria and Pastor (2008) y el procedimiento *multi-start* presentado en Corominas et al. (2008) son los mejores procedimientos publicados hasta la fecha para resolver el RTVP.

El procedimiento *psychoclonal* es una metaheurística evolutiva propuesta por primera vez en Tiwari et al. (2005). De acuerdo con los autores, esta metaheurística obtiene sus fundamentos de la teoría de las necesidades jerárquicas de Maslow (1954) y del principio de selección por clonación (Gaspar and Collard, 2000). El esquema básico de la metaheurística *psychoclonal* es el siguiente: 1) Se genera una población inicial de soluciones y se dispone de una función que evalúa el *fitness* (la adecuación) de cualquier solución; 2) Las mejores soluciones son seleccionadas y clonadas un número de veces que es proporcional a su propio *fitness*; 3) Los clones generados son hipermutados (la hipermutación es un operador que modifica la solución con un ratio inversamente proporcional a su *fitness*); 4) Se forma una nueva población seleccionando los mejores clones e incorporando nuevas soluciones generadas de forma aleatoria; 5) Se repiten las etapas 2 a 4 hasta que se cumple un criterio de parada. Esta metaheurística fue adaptada y probada en la resolución del RTVP (para un mayor detalle, se recomienda García-Villoria and Pastor, 2008).

El esquema general de la metaheurística *multi-start* consta de dos fases. En la primera fase se genera una solución inicial, la cual es mejorada en la segunda fase del procedimiento. Dichas dos fases se aplican de forma iterativa hasta que se cumple un criterio de parada. En el algoritmo *multi-start* propuesto en Corominas et al. (2008) para resolver el RTVP, la generación de la solución inicial se realiza de forma aleatoria y la fase de mejora con un procedimiento de optimización local; por otro lado, el criterio de

parada consiste en alcanzar un tiempo de computación preestablecido (para un mayor detalle, se recomienda Corominas et al., 2008).

### 3. Un algoritmo Tabu Search para resolver el RTVP

Los métodos de búsqueda local presentan la gran desventaja que una vez se alcanza el óptimo local, éste suele ser una solución mediocre (Gendreau, 2003) y, además, el método finaliza la búsqueda y no permite evolucionar a nuevos óptimos locales. Para superar estas limitaciones, Glover (1986) propone la metaheurística Tabu Search (TS). TS se basa en aplicar una búsqueda local en la cual se permiten movimientos de no-mejora (es decir, que proporcionan soluciones con un valor peor que el valor de la solución de partida). Para evitar que el procedimiento entre en un ciclo infinito entre soluciones ya generadas, la historia más reciente de la búsqueda es guardada en una lista de soluciones tabu (prohibidas). En la lista tabu se podrían guardar las soluciones tabú completas (toda la secuencia que, en el RTVP, las identifica), pero esta estrategia podría necesitar mucha memoria de ordenador, podría hacer lento el comprobar si una solución es o no tabu y, sobretodo, podría no diversificar suficientemente la búsqueda. De esta forma, es común guardar únicamente los últimos movimientos (transformaciones) realizados en la solución de partida y prohibir las transformaciones inversas que, nuevamente, llevarían a (generarían) la solución de partida (Gendreau, 2003). La lista tabu usualmente es implementada como una lista de longitud fija con una política FIFO (*First In, First Out*). Una solución tabu puede ser utilizada (es decir, se cancela la prohibición de ser considerada) si satisface un criterio de aspiración (que, habitualmente, suele ser que proporcione un valor de la función de evaluación mejor que el valor de la mejor solución generada hasta el momento en todo el proceso de búsqueda). La Figura 1 muestra el esquema general del procedimiento TS.

<p>Sea <math>S</math> una solución inicial Definir la estructura de vecindario <math>N</math> Mientras no se cumpla la condición de parada hacer:     Sea <math>S'</math> la mejor solución de <math>N</math> que cumple el criterio de aspiración o no tabu     Si <math>S'</math> es mejor que la mejor generada hasta el momento <math>S^*</math>, hacer <math>S^* := S'</math>     Hacer <math>S := S'</math>     Añadir el movimiento actual a la lista tabu (y, tal vez, borrar el último) Fin mientras Devolver <math>S^*</math></p>
---

Figura 1. Esquema general del procedimiento TS

En este trabajo se propone un algoritmo basado en el esquema anterior de TS para resolver el RTVP. A continuación se especifican los elementos del TS propuesto:

- *Solución inicial.* Una solución es representada por la secuencia de copias de los símbolos a ser secuenciados. Se toma como solución inicial la mejor de las proporcionadas por los cinco procedimientos heurísticos propuestos en Corominas et al. (2007).

- *Vecindario*. El vecindario de una solución se obtiene insertando el símbolo asignado a cada posición entre el resto de posiciones de la secuencia que representa a dicha solución.
- *Movimientos tabu*. Un movimiento prohibido de la lista tabu consiste en una pareja “posición/símbolo”. Por ejemplo, el movimiento (3, A) significa que se consideran tabu todas las soluciones con el símbolo A secuenciado en la posición 3 de la secuencia.
- *Criterio de aspiración*. El criterio de aspiración consiste en que el movimiento proporcione una solución que sea mejor que la mejor solución generada hasta el momento en todo el proceso de búsqueda.
- *Condición de parada*. El algoritmo TS finaliza la búsqueda cuando se alcanza un tiempo de computación preestablecido (como en los algoritmos *psychoclinal* y *multi-start* propuestos en García-Villoria and Pastor (2008) y Corominas et al. (2008), respectivamente).

El algoritmo TS presenta un único parámetro cuyo valor ha de ser fijado de partida: el tamaño de la lista tabu. Aunque el valor de este tipo de parámetros es extremadamente importante, ya que los resultados de las metaheurísticas para cada problema son muy sensibles a ellos, la decisión de cómo determinarlos es comúnmente justificada de una de las siguiente maneras (Eiben et al., 1999; Adenso-Díaz and Laguna, 2006): 1) "a mano", en base a un pequeño número de experimentos que no son específicamente referenciados; 2) utilizando valores generales, recomendados para un amplio rango de diferentes tipos de problemas; 3) utilizando los valores que han sido efectivos en otros problema similares; o 4) seleccionando dichos valores sin ninguna explicación.

Adenso-Díaz and Laguna (2006) proponen una nueva técnica, llamada CALIBRA, diseñada específicamente para ajustar el valor de los parámetros de algoritmos heurísticos y metaheurísticos. CALIBRA es utilizada en García-Villoria and Pastor (2008) y en Corominas et al. (2008) para determinar los valores de los parámetros de los algoritmos *psychoclinal* y *multi-start*, respectivamente. En este trabajo también se ha utilizado CALIBRA para decidir el valor de la lista tabu del algoritmo TS propuesto: CALIBRA fija el valor de la lista tabu en 38.

#### **4. Experimento computacional**

Los algoritmos *psychoclinal* y *multi-start*, propuestos en García-Villoria and Pastor (2008) y en Corominas et al. (2008), son los algoritmos más eficientes publicados en la literatura hasta el momento para resolver el RTVP. De esta forma, se compara la calidad del algoritmo TS propuesto con la de dichos dos algoritmos. En lo que resta de esta sección, el algoritmo TS será referido como *TS*, el algoritmo *psychoclinal* como *Psycho* y el algoritmo *multi-start* como *MS*.

Los tres algoritmos han sido codificados en Java y han sido ejecutados en un ordenador Pentium IV de 3.4 GHz con 1.5 GB de memoria RAM. En este trabajo se utilizan los mismos 60 ejemplares de entrenamiento y los 740 ejemplares de prueba que son utilizados en García-Villoria and Pastor (2008) y Corominas et al. (2008). Los 60 ejemplares de entrenamiento son utilizados por CALIBRA para fijar el valor de los parámetros de los algoritmos. Todos los ejemplares pueden ser agrupados en cuatro clases (de la *CAT1* a la *CAT4*, con 15 ejemplares de entrenamiento y 185 de prueba en cada clase) de acuerdo a su tamaño. Los ejemplares fueron generados utilizando las distribuciones uniformes mostradas en la Tabla 1 para generar, de forma aleatoria, los valores de  $D$  (número total de copias) y  $n$  (número de símbolos). Para todos los ejemplares y para cada símbolo  $i = 1, \dots, n$ , se generó un valor aleatorio de  $d_i$  (número de copias del símbolo  $i$  a ser secuenciadas) entre 1 y  $\lfloor (D-n+1)/2.5 \rfloor$ , cumpliendo, obviamente, que  $\sum_{i=1..n} d_i = D$ .

	<b>CAT1</b>	<b>CAT2</b>	<b>CAT3</b>	<b>CAT4</b>
<b><math>D</math></b>	U(25, 50)	U(50, 100)	U(100, 200)	U(200, 500)
<b><math>n</math></b>	U(3, 15)	U(3, 30)	U(3, 65)	U(3, 150)

**Tabla 1.** Distribución uniforme para generar los valores de  $D$  y de  $n$

Los tres algoritmos fueron ejecutados, para cada ejemplar, durante 50 y 1000 segundos. Las Tablas 2 y 3 muestran los valores promedio del RTV, obtenidos con cada uno de los tres algoritmos, para el total de los 740 ejemplares de prueba (*Global*) y para cada clase de ejemplares (*CAT1* a *CAT4*)

	<b>Global</b>	<b>CAT1</b>	<b>CAT2</b>	<b>CAT3</b>	<b>CAT4</b>
<b>TS</b>	210.47	10.26	22.56	73.26	735.78
<b>Psycho</b>	235.68	14.92	44.25	137.07	746.50
<b>MS</b>	2106.01	11.56	38.02	154.82	8219.65

**Tabla 2.** Valor promedio del RTV con 50 segundos de tiempo de computación

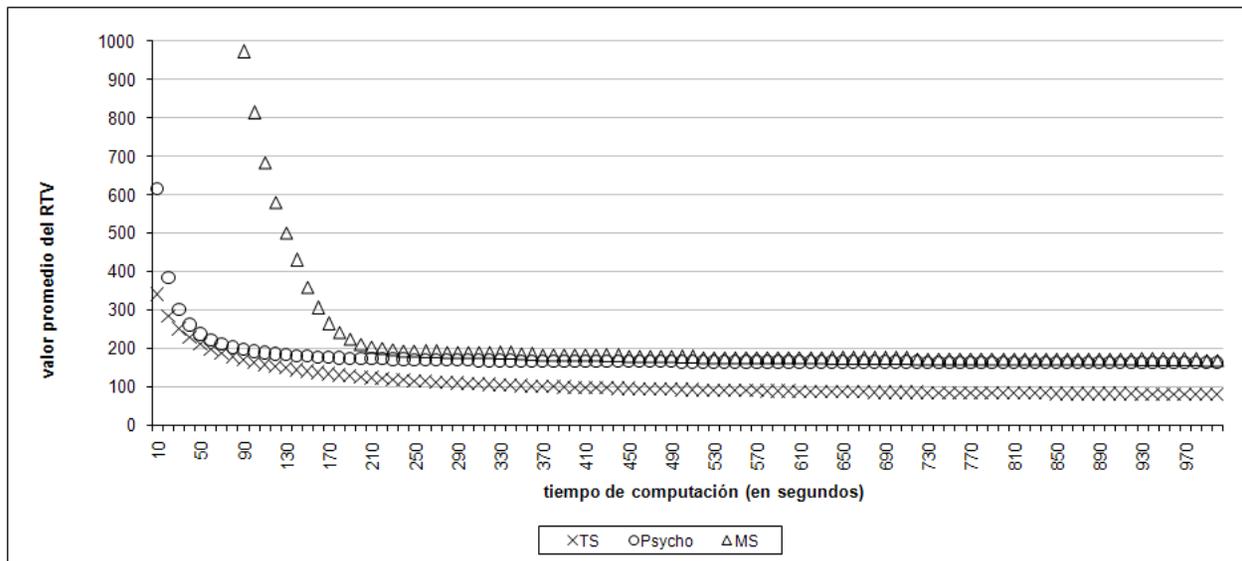
	<b>Global</b>	<b>CAT1</b>	<b>CAT2</b>	<b>CAT3</b>	<b>CAT4</b>
<b>TS</b>	78.62	10.24	21.16	48.12	234.96
<b>Psycho</b>	161.60	14.90	39.90	122.38	469.23
<b>MS</b>	169.25	10.51	31.21	123.27	512.02

**Tabla 3.** Valor promedio del RTV con 1000 segundos de tiempo de computación

Las Tablas 2 y 3 muestran que el algoritmo multi-start converge más lentamente que los otros dos algoritmos, en global y cuando se resuelven ejemplares de gran tamaño, los de la categoría *CAT4* (la Figura 2 muestra la convergencia de los algoritmos en función del tiempo de computación). De esta manera, se analizan los resultados obtenidos por los algoritmos después de 1000 segundos de tiempo de computación.

El valor promedio del RTV para *TS*, con 1000 segundos de tiempo de cálculo y para el conjunto de los 740 ejemplares (*Global*), es un 51.35% y un 53.55% mejor que los

valores obtenidos utilizando los algoritmos *Psycho* y *MS*, respectivamente. Si se consideran los resultados por clases, *MS* presenta mejores resultados que *Psycho* para las clases de ejemplares de menor tamaño (*CAT1* y *CAT2*), resultados muy semejantes para los ejemplares de tamaño medio (*CAT3*) y peores resultados para la clase de ejemplares de mayor tamaño (*CAT4*). De todas formas, el algoritmo *TS* obtiene mejores resultados que los otros dos algoritmos en todas las clases de ejemplares: *TS* es un 2.57% y un 32.21% mejor que *MS* para las clases *CAT1* y *CAT2*, respectivamente, y un 60.68% y un 49.93% mejor que *Psycho* para las clases *CAT3* y *CAT4*, respectivamente.



**Figura 2.** Valor promedio del RTV en función del tiempo de computación

La Tabla 4 muestra el número de veces que cada algoritmo obtiene el mejor valor RTV para cada ejemplar, considerando los tres algoritmos. Los resultados se presentan para el total de los 740 ejemplares (Global) y para cada clase de ejemplares.

	<b>Global</b>	<b>CAT1</b>	<b>CAT2</b>	<b>CAT3</b>	<b>CAT4</b>
<b>TS</b>	678	185	185	185	123
<b>Psycho</b>	58	51	7	0	0
<b>MS</b>	244	163	17	1	63

**Tabla 4.** Número de veces en que se obtiene la mejor solución

Como se podría esperar de los resultados de la Tabla 3, la Tabla 4 muestra que *TS* obtiene la mejor solución el mayor número de veces (en un 91.62% de ejemplares en global). Además, si se analizan los resultados por clases, se observa que *TS* siempre obtiene las mejores soluciones en las clases *CAT1*, *CAT2* y *CAT3*. Por otro lado, *MS* obtiene más veces que *Psycho* la mejor solución, incluso para los ejemplares de la clase *CAT4*. Este resultado puede sorprender ya que, para esta clase de ejemplares, *MS* obtiene un valor promedio del RTV peor que el obtenido por el algoritmo *Psycho*.

Para completar el análisis de los resultados, se calcula su dispersión. Se define una medida de dispersión (aquí denominada  $\sigma$ ) del valor del RTV obtenido por cada

algoritmo,  $alg = \{TS, Psycho, MS\}$ , en cada ejemplar,  $ins$ , según la expresión

$$\sigma(alg, ins) = \left( \frac{RTV_{ins}^{(alg)} - RTV_{ins}^{(best)}}{RTV_{ins}^{(best)}} \right)^2; \text{ donde } RTV_{ins}^{(alg)} \text{ es el valor del RTV de la solución}$$

obtenida con el algoritmo  $alg$  para el ejemplar  $ins$ , y  $RTV_{ins}^{(best)}$  es el mejor valor del RTV, para el ejemplar  $ins$ , de las soluciones obtenidas con los tres algoritmos. La Tabla 5 expone la dispersión  $\sigma$  promedio para el total de ejemplares y por clases.

	<b>Global</b>	<b>CAT1</b>	<b>CAT2</b>	<b>CAT3</b>	<b>CAT4</b>
<b>TS</b>	0.10	0.00	0.00	0.00	0.39
<b>Psycho</b>	3.00	1.08	1.76	3.67	5.49
<b>MS</b>	1.77	0.02	0.47	3.47	3.11

**Tabla 5.** Dispersión  $\sigma$  promedio

La Tabla 5 muestra que *TS* presenta la menor dispersión promedio para el total de ejemplares y para cada una de las clases (~~excepto para *CAT3*, en la que también es baja pero ligeramente peor que la dispersión promedio de los otros dos algoritmos~~). Para las tres primeras clases (*CAT1*, *CAT2* y *CAT3*), la dispersión es nula ya que siempre obtiene las mejores soluciones. En cambio, para los ejemplares más grandes (*CAT4*) la dispersión ya no es nula aunque continúa siendo muy baja. Este resultado significa que cuando *TS* no obtiene el mejor valor del RTV para un ejemplar, obtiene un valor muy cercano. *MS* también obtiene una baja dispersión para las dos clases más pequeñas (*CAT1* y *CAT2*). Por otro lado, *Psycho* presenta una dispersión peor que *MS* para todas las clases, aunque los valores promedios del RTV obtenidos por *Psycho* para *CAT3* y *CAT4* es mejor que el obtenido por *MS*; esto indica que aunque *Psycho* obtiene un mejor funcionamiento para *CAT3* y *CAT4*, en promedio, *MS* es un algoritmo más robusto que *Psycho*. De todas formas, el algoritmo *TS* propuesto en este trabajo es el que obtiene, en promedio, las mejores soluciones con un comportamiento muy estable.

## 5. Conclusiones y futuras líneas de investigación

El RTVP es un problema de *scheduling* que se presenta en una gran cantidad de aplicaciones reales. Como el RTVP es un problema *NP-hard*, varios procedimientos heurísticos y metaheurísticos han sido propuestos para su resolución. Entre ellos, los dos algoritmos que han obtenido mejores resultados son el algoritmo psychoclonal propuesto en García-Villoria and Pastor (2008) y el algoritmo multi-start presentado en Corominas et al. (2008). En este trabajo se presenta una aplicación basada en la metaheurística *tabu search* (*TS*) para la resolución del RTVP. Los resultados del experimento computacional realizado muestran que el algoritmo *TS* propuesto mejora los mejores resultados publicados en la literatura hasta el momento. Además se comprueba que el algoritmo *TS* es muy estable: cuando no obtiene el mejor resultado para un ejemplar del RTVP, obtiene un valor muy cercano al mejor.

La definición del vecindario es una decisión crítica en el diseño de cualquier metaheurística, en general, y del algoritmo TS, en particular (Gendreau et al., 2003). En este trabajo se genera el vecindario de una solución insertando el símbolo asignado a cada posición entre el resto de posiciones de la secuencia que representa a dicha solución. Debido al buen funcionamiento del algoritmo TS, se pretende seguir investigando para intentar mejorarlo. Para ello se propone diseñar y probar nuevas definiciones del vecindario de una solución; por ejemplo intercambiado los símbolos asignados a cada pareja de posiciones consecutivas de la secuencia que representa a una solución o intercambiando los símbolos asignados a cada pareja de posiciones consecutivas y no consecutivas de la secuencia.

## Bibliografía

1. Kubiak, W. Fair Sequences. Chapter 19 in *Handbook of Scheduling: Algorithms, Models and Performance Analysis*, Chapman and Hall, 2004.
2. Waldspurger, C.A.; Weihl, W.E. Lottery Scheduling: Flexible Proportional-Share Resource Management. *First USENIX Symposium on Operating System Design and Implementation*, 1994.
3. Corominas, A.; Kubiak, W.; Moreno, N. Response time variability. *Journal of Scheduling* 10: 97-110, 2007.
4. Kubiak, W. Minimizing variation of production rates in just-in-time systems: A survey. *European Journal of Operational Research* 66: 259-271, 1993.
5. Miltenburg, J. Level schedules for mixed-model assembly lines in just-in-time production systems. *Management Science* 35: 192-207, 1989.
6. Dong, L.; Melhem, R.; Mosse, D. Time slot allocation for real-time messages with negotiable distance constrains requirements. *Fourth IEEE Real-Time Technology and Applications Symposium (RTAS'98)*, Denver, CO, 131-136, 1998.
7. Waldspurger, C.A.; Weihl, W.E. Stride Scheduling: Deterministic Proportional-Share Resource Management. *Technical Report MIT/LCS/TM-528*, Massachusetts Institute of Technology, 1995.
8. Anily, S.; Glass, C.A.; Hassin, R. The scheduling of maintenance service. *Discrete Applied Mathematics* 82: 27-42, 1998.
9. Wei, W.D.; Liu, C.L. On a periodic maintenance problem. *Operations Research Letters* 2: 90-93, 1983.
10. Herrmann, J.W. Generating Cyclic Fair Sequences using Aggregation and Stride Scheduling. *Technical Report TR 2007-12*, University of Maryland, USA, 2007.
11. Bollapragada, S.; Bussieck, M.R.; Mallik, S. Scheduling Commercial Videotapes in Broadcast Television. *Operations Research* 52: 679-689, 2004.
12. Brusco, M.J. Scheduling advertising slots for television. *Journal of the Operational Research Society* 59: 1363-1372, 2008.
13. Balinski, M.L.; Young, H.P. Fair Representation: meeting the ideal of one man, one vote. Yale University Press, New Haven CT, 1982.
14. García, A.; Pastor, R.; Corominas, A. Solving the Response Time Variability Problem by means of metaheuristics. *Frontiers in Artificial Intelligence and Applications* 146: 187-194, 2006.

15. Corominas, A.; García-Villoria, A.; Pastor, R. Solving the Response Time Variability Problem by means of Multi-start and GRASP metaheuristics. *Frontiers in Artificial Intelligence and Applications* 184: 128-137, 2008.
16. García-Villoria, A.; Pastor, R. Introducing dynamic diversity into a discrete particle swarm optimization. *Computers and Operations Research* 36: 951-966, 2009a.
17. García-Villoria, A.; Pastor, R.; Corominas, A. Solving the Response Time Variability Problem by means of the Cross-Entropy Method. *International Journal of Manufacturing Technology and Management*, to be published, 2007.
18. García-Villoria, A.; Pastor, R. Solving the Response Time Variability Problem by means of the Electromagnetism-like Mechanism. *International Journal of Production Research*, doi: 10.1080/00207540902862545, 2009b.
19. García-Villoria, A.; Pastor, R. Solving the Response Time Variability Problem by means of a psychoclonal approach. *Journal of Heuristics*, doi:10.1007/s10732-008-9082-2, 2008.
20. Tiwari, M.K.; Prakash, A.; Kumar, A.; Mileham, A.R. Determination of an optimal sequence using the psychoclonal algorithm. *ImechE, Part B: Journal of Engineering Manufacture* 219: 137-149, 2005.
21. Maslow, A.H. *Motivation and personality*. New York: Harper & Bros, 1954.
22. Gaspar, A.; Collard, P. Two models of immunization for time dependent optimization. *IEEE International Conference on Systems Manufacturing and Cybernetics*, 113-118, 2000.
23. Gendreau, M. An Introduction to Tabu Search. Chapter 2 in *Handbook of Metaheuristics*, Eds. Glover and Kochenberger, Kluwer Academic Publishers, 37-54, 2003.
24. Glover, F. Future paths for Integer Programming and Links to Artificial Intelligence. *Computers and Operations Research* 5: 533-549, 1986.
25. Eiben, A.E.; Hinterding, R.; Michalewicz, Z. Parameter control in evolutionary algorithms. *IEEE Transactions on evolutionary computation* 3: 124-141, 1999.
26. Adenso-Díaz, B.; Laguna, M. Fine-tuning of algorithms using fractional experimental designs and local search. *Operations Research* 54:, 99-114, 2006.