Combinación de Algoritmos Evolutivos y Técnicas Heurísticas para Problemas de Scheduling

Ramiro Varela , Javier Blanco, Camino Rodríguez, Jorge Puente y César Alonso

Centro de Inteligencia Artificial.
Universidad de Oviedo. Campus de Viesques. E-33271 Gijón.
Tel. +34-8-5182032. FAX +34-8-5182125.
e-mail: ramiro@aic.uniovi.es
http:\\www.aic.uniovi.es

Resumen. En este trabajo presentamos un método para resolver problemas de Scheduling mediante un Algoritmo Genético en el que los cromosomas de la población inicial se construyen a partir de la información heurística que proporcionan los heurísticos de ordenación de variables y valores. Mostramos resultados experimentales que ponen de manifiesto el ámbito de aplicación del método, y proponemos también algunas ideas para generalizarlo de modo que sea efectivo para una familia más amplia de problemas.

1 Introducción

En este trabajo presentamos una estrategia de resolución del problema *Job Shop Scheduling (JSS)* que consiste en combinar una familia de heurísticos basados en la probabilidad con algoritmos evolutivos. En esencia la propuesta consiste en partir de un Algoritmo Genético (AG) tomado de la literatura [1,2,4] que utiliza un esquema de codificación basado en permutaciones con repetición, y el algoritmo de planificación G&T propuesto en [3] para la evaluación de cromosomas. El objetivo es mejorar la eficiencia de este AG mediante la inyección de conocimiento sobre el dominio del problema en la fase de inicialización. Para ello proponemos el uso de los heurísticos de ordenación de valores y variables [5] que permiten predecir cuáles son los recursos más críticos así como medir la probabilidad de éxito de cada una de las asignaciones de tiempo de inicio a cada una de las tareas. La principal aportación del trabajo consiste en el diseño y análisis experimental de una estrategia que permite incorporar este conocimiento en la codificación de los cromosomas.

El contenido del resto del trabajo es el siguiente: en la sección 2 introducimos el problema JSS y un AG para resolverlo construido a partir de propuestas tomadas de la literatura. En la sección 3 describimos los heurísticos de ordenación de variables y valores. En la sección 4 presentamos el método desarrollado para construir los cromosomas iniciales, junto con algunos resultados experimentales que muestran el ámbito de aplicación. En la sección 5 proponemos algunas ideas para generalizar el

---

Este trabajo ha sido parcialmente financiado por el MCyT con el proyecto TIC2001-4936-E y por la FICYT con el proyecto PB-TBI01-04.

método con el objetivo de que sea aplicable a una familia más amplia de problemas. Y finalmente en la sección 6 presentamos las principales conclusiones del trabajo.

2 Algoritmos Genéticos y Problemas de Scheduling

En este trabajo consideramos el problema *Job Shop Scheduling* (JSS) con tiempos máximos de finalización de los trabajos y como criterio de optimización la obtención del mínimo makespan. Este es el problema conocido en la literatura como $J//C_{max}$. Se trata de planificar la ejecución de un conjunto de trabajos $\{J_1,...,J_n\}$ sobre un conjunto de recursos $\{R_1,...,R_m\}$. Cada trabajo *Ji* está compuesto por un conjunto de tareas $\{t_{i1},...,t_{im}\}$ que deben ser ejecutadas secuencialmente. Cada tarea t_{ij} requiere el uso exclusivo de uno de los recursos durante su tiempo de procesamiento du_{ij}. Suponemos además que para cada trabajo hay un tiempo mínimo de inicio y un tiempo máximo de fin entre los cuales deben ser planificadas todas sus tareas. El objetivo es conseguir un tiempo de inicio para cada una de las tareas de modo que se satisfagan todas las restricciones del problema y que además se minimice el tiempo de finalización de todas las tareas, es decir el makespan. La Figura 1 muestra el grafo de restricciones de un problema con 3 trabajos y 3 recursos.

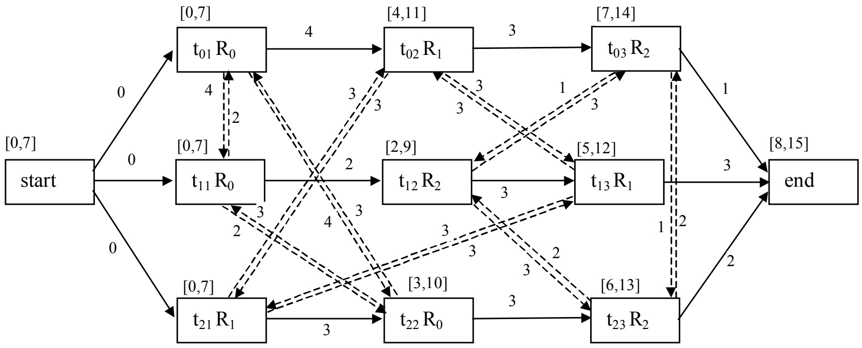

Figura 1. Representación gráfica de un problema con 3 trabajos de 3 tareas cada uno. El tiempo mínimo de inicio es 0 para todos ellos y el tiempo máximo de fin es 15. Los arcos están etiquetados con el coste de la tarea origen

En la literatura existen numerosas propuestas que resuelven el problema JSS mediante el uso de Algoritmos Evolutivos. Por ejemplo en [1,4] Bierwirth y Mattfeld proponen varios Algoritmos Genéticos con distintas estrategias de evolución y un esquema de codificación denominado *permutaciones con repetición*. De acuerdo con este esquema un cromosoma es en principio una permutación del conjunto de tareas en el que cada tarea viene representada simplemente por el número del trabajo al que pertenece. Por ejemplo si tenemos un problema con 3 trabajos de 3 tareas cada uno, un cromosoma puede ser el siguiente *(1 3 2 2 3 1 1 3 2)*. En esta codificación el primer *1* representa la primera tarea del primer trabajo, el segundo *1* la segunda y así sucesivamente. De este modo las tareas de cada trabajo aparecen representadas en el orden de ejecución secuencial, mientras que las tareas que requieren al mismo recurso pueden aparecer en cualquier orden. La ventaja principal que presenta este esquema

Tabla I. Resultados obtenidos con el AG descrito sobre una selección de problemas no triviales, con poblaciones iniciales aleatoria y heurística. En cada caso se muestra la mejor solución encontrada en la realización de 50 experimentos con cada problema, así como el error medio porcentual EM con respecto a la mejor solución conocida, y la desviación estándar porcentual.

Problema	Tamaño		Mejor Solución Conocida	Mejor Solución Encontrada	Error Medio Porcentual	Desviación Estándar Porcentual
	N	M				
FT10	10	10	930	935	4.0	1.5
FT20	20	5	1165	1183	4.1	1.6
abz7	20	15	665	693	3.3	1.1
abz8	20	15	670	708	6.6	1.2
abz9	20	15	686	724	2.9	1.2
la21	15	10	1046	1073	5.0	1.2
la24	15	10	935	965	5.9	1.2
la25	15	10	977	1000	4.2	1.0
la27	20	10	1235	1276	5.5	0.8
la29	20	10	1153	1212	7.3	1.1
la38	15	15	1196	1243	7.9	1.9
la40	15	15	1222	1254	5.5	1.1

de codificación es que permite diseñar operadores genéticos eficientes, de cruce y mutación, que producen cromosomas factibles.

Como estrategia de decodificación suele utilizarse un esquema que produzca una *planificación activa*. Una planificación es activa si para adelantar la ejecución de una cualquiera de las tareas es preciso retrasar al menos la ejecución de otra. El conjunto de las planificaciones activas es un subconjunto de las planificaciones factibles de un problema que tiene la propiedad de que incluye al menos una planificación óptima; de ahí su interés. La forma más extendida de calcular planificaciones activas es el conocido algoritmo G&T propuesto por Giffler y Thomson en [3]. Se trata de un algoritmo voraz que en cada iteración calcula un conjunto de tareas candidatas a ser planificadas de modo que cualquiera que sea la tarea elegida dentro de este conjunto, el algoritmo garantiza que la planificación resultante es activa. Cuando se utiliza como esquema de decodificación, el no determinismo que se produce en cada iteración a la hora de seleccionar una tarea se resuelve eligiendo la tarea que aparece más a la izquierda en el cromosoma. De este modo se consigue una planificación activa respetando en la medida de lo posible el orden de las tareas indicado en el cromosoma. En [2], Bierwirth y Mattfeld proponen un método de reducción del espacio de búsqueda que se controla mediante un parámetro $\delta \in [0,1]$. De modo que cuando $\delta=1$, no hay ningún tipo de reducción y cuando $\delta<1$ se produce una reducción de modo que la búsqueda se realiza solamente en un subconjunto de las planificaciones activas, con la consiguiente posibilidad de eliminar todas las soluciones óptimas del espacio de búsqueda. En particular cuando $\delta=0$ se restringe la búsqueda a las planificaciones del tipo “non-delay”. Este tipo de planificaciones se caracterizan porque nunca hay un recurso inactivo cuando una tarea está en condiciones de utilizarlo. Las planificaciones non-delay son en general bastante buenas pero no hay garantía de que entre ellas se encuentre una planificación óptima. Mediante un estudio experimental en [2] se prueba que un valor de $\delta=0.5$ produce mejores resultados que los valores extremos, 0 y 1, para problemas de tamaño medio. Nuestra experiencia concuerda con estos resultados, por lo que en los experimentos de este trabajo se toma este valor como referencia.

La Tabla I muestra los resultados experimentales obtenidos por un AG convencional, en concreto hemos utilizado el esquema de permutaciones con repetición, una estrategia de planificación basada en el algoritmo G&T y una estrategia evolutiva basada en selección proporcional al fitness (inverso del makespan), cruce y mutación, en el que los hijos reemplazan siempre a los padres. Los problemas han sido tomados de la OR-library y constituyen un conjunto pequeño pero significativo por ser reconocidos como problemas no triviales. Como cabe esperar al tratarse de un AG construido a partir de propuestas estándar tomadas de la literatura los resultados son similares a los presentados por otros autores.

3 Los heurísticos de Ordenación de Variables y Valores

Estos heurísticos fueron propuestos por N. Sadeh en [5] y se basan en un modelo probabilista del espacio de búsqueda que permite hacer estimaciones sobre lo adecuado que es un tiempo de inicio para una tarea y sobre la probabilidad de que las asignaciones de dos o más tareas distintas entren en conflicto. Estas estimaciones se realizan a partir de los perfiles de demanda de las tareas sobre los recursos: la demanda individual y la demanda agregada. La demanda individual $D_{ij}(R_p,\tau)$ de la tarea t_{ij} sobre el recurso R_p durante el intervalo de tiempo $T\leq\tau<T+1$ es la probabilidad de que la tarea requiera el uso del recurso durante el intervalo de tiempo y se calcula como la suma de las probabilidades $\sigma_{ij}(\rho)$ de que el tiempo de inicio de la tarea t_{ij} sea un instante ρ del intervalo $(T-du_{ij},T]$. En principio debemos asumir que todas las asignaciones posibles son igualmente probables. Así, por ejemplo, en el intervalo $6\leq\tau<7$ tenemos que $D_{02}(R_1,\tau) = \sigma_{02}(4)+\sigma_{02}(5)+\sigma_{02}(6) = 3/8$. La demanda individual es una medida de la dependencia de la tarea sobre la disponibilidad de un recurso.

La demanda agregada $D_{aggr}(R,\tau)$ sobre un recurso se obtiene sumando a lo largo del tiempo, las demandas individuales de las tareas que requieren al recurso. A partir de la demanda agregada se identifica el pico de disputa del recurso como el intervalo, cuya duración es igual a la duración media de las tareas que requieren al recurso, en el que la demanda es máxima. La demanda agregada de un recurso en un intervalo es una medida de la disputa por el recurso durante ese intervalo por parte de las tareas que lo requieren.

A partir de los perfiles de demanda se calculan las supervivencias de las asignaciones. La supervivencia de una asignación $\langle st_{ij}=T\rangle$ es una estimación de la probabilidad de que una determinada asignación para una tarea no entre en conflicto con las asignaciones del resto de las tareas que requieren el mismo recurso. En [5] las supervivencias se calculan mediante la expresión

$$\left(1-\frac{AVG\left(D^{aggr}\left(R_p,\tau\right)-D_{ij}\left(R_p,\tau\right)\right)}{AVG\left(n_p\left(\tau\right)-1\right)}\right)^{AVG\left(n_p(\tau)-1\right)*du_{ij}*\left(AVG(du)^{-1}\right)}, \qquad (1)$$

donde du es la duración media de las tareas que requieren al recurso R_p, $n_p(\tau)$ es el número de tareas que pueden demandar al recurso R_p en el instante τ y $AVG(f(\tau))$ representa el valor medio de la función $f(\tau)$ en el intervalo $[T,T+du_{ij})$. Cuando $AVG(n_p(\tau)-1)$ es 0, el valor de la expresión (1) se toma como 1.

Por último, a partir de las supervivencias de las tareas y de las relaciones de precedencia de las tareas del mismo trabajo, se calculan las bondades de las asignaciones. En el cálculo de las bondades se tiene en cuenta no solamente la supervivencia de la asignación, sino también el espacio que una determinada asignación deja para el resto de las tareas no planificadas del mismo trabajo. De este modo las bondades de las asignaciones nos dan una información más precisa de lo adecuada que puede ser a a priori la asignación de un tiempo de inicio para una tarea.

Una asignación ρ para una la θ_{il} es buena no solamente si sobrevive a la disputa de

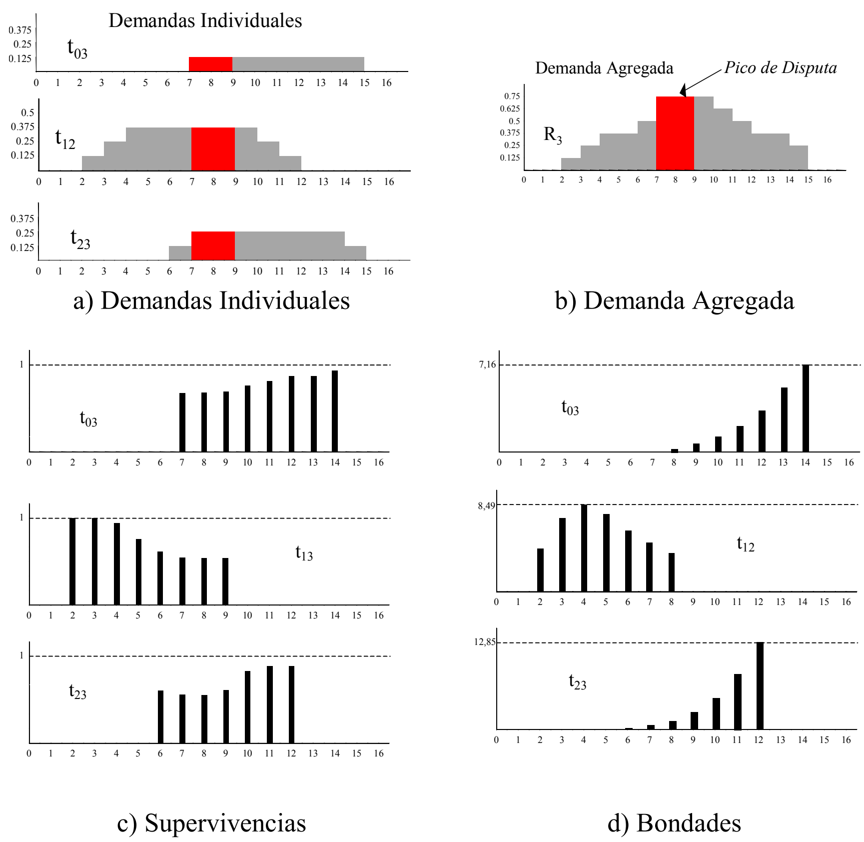

Figura 2. Perfiles de demanda, supervivencias y bondades para las tareas que requieren el recurso R_3 en el problema de la Figura 1

otras tareas por el mismo recurso; sino que será preciso además que deje suficiente espacio para el resto de tareas no planificadas del mismo trabajo. Para tener esto en cuenta los heurísticos consideran también las operaciones no planificadas que son accesibles desde la tarea θ_{il} a través de restricciones de precedencia, este conjunto o relajación del problema se denota como $RELAX_{il}$. El espacio que una reserva $\rho=\langle st_{il}=t\rangle$ para la operación θ_{il} deja para el resto de las tareas del conjunto $RELAX_{il}$ viene dada por el número de soluciones de la relajación (compatibles con la asignación ρ para la tarea θ_{il}) que se espera que sobrevivan a la disputa de recursos. Esta cantidad se denota por $compsurv_{il}(t)$ y se combina con la supervivencia de la reserva para obtener la bondad final de la misma como

$$good_{il}(t) = surv_{il}(t) * compsurv_{il}(t). \qquad (2)$$

El valor de $compsurv_{il}(t)$ se calcula del siguiente modo. El conjunto $RELAX_{il}$ se puede obtener como la unión siguiente de conjuntos disjuntos:

$$RELAX_{il} = BEFORE_{il} \cup \{\theta_{il}\} \cup AFTER_{il}, \qquad (3)$$

donde $BEFORE_{il}$ es el conjunto de operaciones que preceden a θ_{il} en la relajación y $AFTER_{il}$ contiene a las operaciones posteriores. Es claro que una vez hecha la reserva $\rho=\langle st_{il}=t\rangle$ los subproblemas definidos por cada uno de los subconjuntos $BEFORE_{il}$ y $AFTER_{il}$ son independientes y en consecuencia se puede obtener el número de soluciones que se espera que sobrevivan a la disputa, compatibles con la asignación $st_{il}=t$ para la tarea θ_{il} como

$$compsurv_{ij}(t) = BEF_{ij}(t) * AFT_{ij}(t), \qquad (4)$$

donde a su vez $BEF_{il}(t)$ y $AFT_{il}(t)$ son respectivamente el número de soluciones de $BEFORE_{il}$ y $AFTER_{il}$ compatibles con la asignación $st_{il}=t$. $BEF_{il}(t)$ se obtiene acumulando las soluciones esperadas para $BEFORE_{i(l-1)}$ compatibles con cada una de las soluciones de la operación $\theta_{i(l-1)}$ como

$$BEF_{il}(t) = \sum_{\tau \le t - du_{i(l-1)}} \left[surv_{i(l-1)}(\tau) * BEF_{i(l-1)}(\tau) \right]. \qquad (5)$$

Y de forma análoga para $AFT_{il}(t)$ tendremos

$$AFT_{il}(t) = \sum_{\tau \ge t + du_{il}} \left[surv_{i(l+1)}(\tau) * AFT_{i(l+1)}(\tau) \right]. \qquad (6)$$

No obstante, los cálculos de estas recurrencias se pueden acelerar utilizando sumas parciales

$$BEF_{il}(t) = BEF_{il}(t-1) + BEF_{i(l-1)}\left(t - du_{i(l-1)}\right) * surv_{i(l-1)}\left(t - du_{i(l-1)}\right), \qquad (7)$$

$$AFT_{il}(t) = AFT_{il}(t+1) + AFT_{i(l+1)}\left(t + du_{il}\right) * surv_{i(l+1)}\left(t + du_{il}\right).$$

Cuyos casos base se calculan como

$$BEF_{il}(t)= AFT_{il}(t)=0 \text{ si } t\notin [est_{il},lst_{il}],$$

$$BEF_{il}(t)=1 \text{ si } t\in [est_{il},lst_{il}] \text{ y } BEFORE_{il} = \emptyset, \tag{8}$$

$$AFT_{il}(t)=1 \text{ si } t\in [est_{il},lst_{il}] \text{ y } AFTER_{il} = \emptyset.$$

Donde est_{il} y lst_{il} son respectivamente el menor y el mayor de los valores del dominio de tiempos de inicio actual $DOM(\theta_{il})$ para la tarea θ_{il}. En el estado inicial $DOM(\theta_{il}) = [est_{il},lst_{il}]$; sin embargo a medida que la búsqueda progresa tendremos que $DOM(\theta_{il})\subseteq[est_{il},lst_{il}]$ debido a que alguna de las asignaciones a otras tareas que requieren el mismo recurso se traduce en una reducción de $DOM(\theta_{il})$ de forma que algunos de los valores no extremos son eliminados.

Las Figuras 2a y 2b muestran los perfiles de demanda correspondientes al recurso R_3 del problema de la Figura 1, las zonas sombreadas representan el pico de disputa y la contribución de cada una de las tareas. La Figura 2c muestra las supervivencias de todas las asignaciones posibles inicialmente para las tareas que utilizan el recurso R_3. Y la Figura 2d muestra las correspondientes bondades. Como se puede observar las supervivencias siempre tomas valores comprendidos en el intervalo [0,1] ya que representan probabilidades, mientras que las bondades pueden tomar valores mayores que 1. Se puede observar también que las bondades en general discriminan más entre las distintas asignaciones y que al tener en cuenta información relativa tanto al recurso requerido por una tarea como al trabajo al que pertenece la tarea, en ocasiones producen estimaciones contradictorias con las estimaciones de las supervivencias. Esto se puede observar, por ejemplo, en las estimaciones sobre los instantes de inicio 2, 3 y 4 para la tarea t_{12}.

4 Incorporación de la Información Heurística en el AG

La primera idea que hemos propuesto para transformar la información que proporcionan los heurísticos en cromosomas prometedores se describe en [6]. De forma resumida consiste en tener en cuenta los recursos más críticos, es decir aquellos que tienen un mayor pico de disputa, y tratar de situar las tareas que requieren a estos recursos en una posición adecuada. Asumimos que una posición es adecuada para una tarea si ofrece a la tarea la oportunidad de obtener como tiempo de inicio un valor próximo al instante con mayor supervivencia (obviamente puede considerarse también la bondad en este caso, pero teniendo en cuenta que el cálculo es más costoso). Para conseguir esta distribución el algoritmo que hemos utilizado es el siguiente

1. *En primer lugar caracterizar a los recursos críticos como aquellos cuyo pico de disputa es superior a la media de los picos de disputa de todos los recursos.*
2. *Para cada recurso crítico*
    2.1. *Considerar las tareas que lo requieren y ordenarlas según sus instantes de mayor supervivencia, resolviendo los empates aleatoriamente.*

2.2. Generar un conjunto de cromosomas tales que contengan las tareas críticas en el orden anterior. El resto de las tareas se insertan de modo que se satisfagan las restricciones de precedencia y, en caso de que haya varias posibilidades, teniendo en cuenta de nuevo los instantes de mayor supervivencia, es decir si en una posición pueden colocarse varias tareas que requieren al mismo recurso, elegir aquella en el que su instante de máxima supervivencia es anterior.

La intuición que hay detrás de esta estrategia es obviamente asumir que los ordenes parciales entre tareas críticas constituyen bloques básicos a partir de los cuales se pueden construir cromosomas. Así con esta estrategia obtendremos cromosomas con esquemas prometedores que el AG se encargará de combinar durante el proceso evolutivo.

La experiencia demostró que la estrategia anterior no arroja buenos resultados para cualquier tipo de problemas, sino solamente para problemas en los que hay recursos que destacan por su carácter crítico. Hemos identificado este tipo de recursos como aquellos tales que la duración media de las tareas que los requieren es superior a la duración media del resto de las tareas y que además el orden en el que aparece el recurso en la secuencia de recursos visitados por cada trabajo es el mismo, o al menos

Tabla II. Resultados obtenidos con el AG descrito sobre el banco de problemas con recursos críticos, con poblaciones iniciales aleatorias y heurísticas. En cada caso se muestra la mejor solución encontrada en la realización de 50 experimentos con cada problema, así como el error medio porcentual con respecto a la mejor solución conocida, y la desviación estándar porcentual. En cada uno de los experimentos se han evaluado un número aproximado de 10000 cromosomas.

Problema	Sol. Conoc.	PI Aleatoria			PI Heurística		
		Mejor	EM%	DE%	Mejor	EM%	DE%
P_10×10_2R	1926	1926	0.03	0.09	1926	0.02	0.08
P_10×10_3R	2039	2039	0.58	0.53	2039	0.25	0.24
P_10×10_4R	2348	2351	1.45	0.83	2348	0.85	0.57
P_20×20_3R	4057	4071	1.55	0.81	4057	0.67	0.43
P_20×20_5R	4272	4292	1.83	0.71	4272	0.70	0.36
P_20×20_7R	4701	4777	2.91	0.64	4701	1.23	0.58
P_30×30_3R	5822	5872	2.30	0.64	5822	1.29	0.59
P_30×30_6R	6639	6679	1.90	0.53	6639	1.25	0.53
P_30×30_9R	7021	7080	1.94	0.60	7021	1.22	0.60
P_40×40_5R	8302	8401	2.66	0.66	8302	0.95	0.46
P_40×40_10R	9317	9507	3.34	0.68	9317	1.47	0.47
P_40×40_14R	10084	10232	3.18	0.67	10084	1.33	0.56

muy similar. En la literatura solamente hemos encontrado un banco de ejemplos de este tipo: el propuesto por N. Sadeh en [5]. Se trata de un conjunto de 60 ejemplos en los que hay 1 ó 2 recursos críticos, o cuellos de botella. En estos ejemplos el AG mejora notablemente su eficiencia cuando se utiliza una población inicial heurística. No obstante se trata de ejemplos de tamaño muy pequeño (10×5, es decir 10 trabajos con 5 tareas cada uno) por lo que no los hemos considerado suficientemente significativos y ante la ausencia de ejemplos con estas características en la OR-library, hemos optado por proponer nosotros mismos una serie de ejemplos de tamaños 10×10, 20×20, 30×30 y 40×40 con distintos números de recursos críticos, tal y como indican los nombres de los problemas. La Tabla II resume los resultados de la experimentación llevada a cabo con estos problemas. Como se puede observar la mejora que introduce la población heurística es evidente, y esta mejora se hace más patente a medida que aumentan tanto el tamaño del problema como el número de recursos críticos.

5 Generalización de la Estrategia de Generación de Cromosomas Heurísticos

La razón por la cual la estrategia propuesta en la sección anterior para la generación de cromosomas iniciales no ofrece buenos resultados en general es que estamos utilizando una información parcial que solamente es significativa cuando el problema tiene recursos críticos, ya que en este caso es más importante considerar la disputa entre las tareas que requieren a un recurso crítico que la disputa entre las tareas que pertenecen al mismo trabajo. Así, una forma evidente para tratar de mejorar los resultados será considerar la información que proporcionan las bondades en el algoritmo propuesto en la sección 3. En este caso habrá que contrastar la mejora obtenida con el aumento en el tiempo de cálculo de los heurísticos.

Además proponemos aquí otra estrategia que consiste en sustituir el criterio determinista de la anterior, según el cual se elige para cada recurso crítico una planificación parcial determinada por los instantes de inicio más prometedores para cada una de las tareas, por un criterio probabilista. La idea es generar para cada recurso planificaciones parciales tales que cada tarea aparezca en la mejor posición (según el criterio anterior) con una determinada probabilidad, pero que también pueda aparecer en otras posiciones (con mayor probabilidad en posiciones próximas y con menor probabilidad en posiciones lejanas). Además como en general las planificaciones parciales obtenidas de forma independiente para cada recurso pueden ser incompatibles entre si, a la hora de mezclarlas para obtener un cromosoma trataremos de preservar aquellas que pertenecen a los recursos más críticos, es decir que la probabilidad de que un recurso mantenga la planificación parcial en la representación del cromosoma sea directamente proporcional a su pico de disputa.

El método que proponemos consiste en definir una distribución de probabilidad sobre el espacio de planificaciones parciales de cada recurso R requerido por las tareas $\{t_0,...,t_{M-1}\}$ tal que a cada planificación $(t_{k0},...,t_{kM-1})$ se le asigna una probabilidad

$$\Pr\left(\left(t_{k0},\ldots,t_{kM-1}\right)\right)=\sum_{\tau_{k0}<\tau_{k1}<\ldots<\tau_{kM-1}}\left(\prod_{j=0}^{M-1}\Pr_{kj}\left(\tau_{kj}\right)\right) \tag{9}$$

es decir, la probabilidad de obtener asignaciones de la forma $\tau_{k0} < \tau_{k1} < \ldots < \tau_{kM-1}$, donde τ_{ki} es el tiempo de inicio de la tarea t_{ki} seleccionado a su vez a partir de la distribución de probabilidad según la cual cada asignación tiene una probabilidad directamente proporcional a su bondad, es decir

$$\Pr_j\left(\tau\right)=good_j\left(\tau\right)\Big/\sum_{\tau'}good_j\left(\tau'\right). \tag{10}$$

La intuición que hay detrás de esta estrategia es que una planificación parcial obtenida de este modo ofrece a cada una de las tareas la oportunidad de obtener una asignación próxima a su instante de mayor bondad, ya que, como hemos comentado, la estrategia de planificación basada en el algoritmo G&T trata de mantener en la medida de lo posible las planificaciones parciales expresadas en el cromosoma.

6 Conclusiones

En este trabajo hemos presentado un método de inicialización de un AG que resuelve problemas de la familia JSS. El método se basa en el uso parcial de la información que proporcionan los heurísticos de ordenación de variables y valores [5]. Debido a este hecho, el método solamente es aplicable con éxito a una subfamilia de problemas con una determinada estructura: los problemas con recursos críticos o cuellos de botella. Para generalizar la estrategia proponemos como trabajo futuro el uso de toda la información heurística disponible, así como un método de carácter probabilista, como alternativa al método determinista presentado.

Bibliografía

1. Bierwirth, Ch. A Generalized Permutation Approach to Jobshop Scheduling with Genetic Algorithms. OR Spectrum, vol. 17 (1995) 87-92.
2. Bierwirth, Ch. and Mattfeld D. C. Production Scheduling and Rescheduling with Genetic Algoritms. Evolutionary Computation 7 (1), 1-17 (1999).
3. Giffler, B. Thomson, G. L. Algorithms for Solving Production Scheduling Problems. Operations Reseach 8, 487-503 (1960).
4. Matfeld, D. C. Evolutionary Search and the Job Shop. Investigations on Genetic Algorithms for Production Scheduling. Springer-Verlag, November 1995.
5. Sadeh, N., Fox, M.S. Variable and Value Ordering Heuristics for the Job Shop Scheduling Constraint Satisfaction Problem. Artificial Intelligence 86, 1-41 (1996).
6. Varela, R., Puente, J., Vela, C. R, Gómez, A. A Knowledge-Based Evolutionary Strategy for Scheduling Problems with Bottlenecks. Aceptado en European Journal of Operational Research, (2002).