VIII Congreso de Ingeniería de Organización
Leganés, 9 y 10 de septiembre de 2004

Un algoritmo genético multiobjetivo para resolver el problema de secuenciación en talleres de flujo con tiempos de cambio de partida

Carlos Andrés Romano[1], José Vicente Tomás Miquel[2], José Pedro García Sabater[1]

[1] Centro de investigación en Gestión e Ingeniería de Producción Universidad Politécnica de Valencia. Camino de Vera s/n. 46022 (Valencia). candres@omp.upv.es, jpgarcia@omp.upv.es
[2] Centro de investigación en Gestión e Ingeniería de Producción Universidad Politécnica de Valencia. Plaza Ferrándiz y Carbonell, 2. 03801 Alcoy (Alicante). jotomi@doctor.upv.es

Resumen

Este trabajo propone un algoritmo genético multiobjetivo y su aplicación a la resolución de problemas de secuenciación en talleres de flujo (flowshop) donde hay tiempos de cambio de partida en las máquinas. Este tipo de algoritmos ya han sido aplicados a problemas tipo flowshop aunque en ausencia de tiempos de cambio de partida dependientes de la secuencia. Nuestra propuesta usa los conceptos de población de individuos no dominados y un procedimiento para que los individuos de este conjunto estén distribuidos uniformemente en la frontera de pareto basado en el algoritmo de clustering SLC.

Palabras clave: Algoritmo genético multiobjetivo, flowshop, tiempos de cambio de partida

1. Introducción

Muchos problemas reales de gestión requieren la optimización simultánea de varios objetivos. Una forma de resolverlos es mediante su transformación en un problema monobjetivo mediante una función de agregación. En muchos casos los pesos que se usan en esta función de agregación son desconocidos a priori, por lo que este tipo de técnicas no son adecuadas. De esta forma, el problema completo debe ser tratado simultáneamente de tal manera que el resultado del proceso de resolución del mismo sea un conjunto de soluciones entre las que el decisor final debe de elegir en función de sus preferencias.

Los problemas multiobjetivo persiguen optimizar un vector solución formado por las funciones objetivo de un problema. Entre los diferentes vectores solución a un problema multiobjetivo existe una relación denominada dominancia de Pareto, de tal manera que el conjunto de soluciones no dominadas se constituye como el conjunto final de soluciones al problema entre las que el decisor debe elegir la que considere más adecuada.

Los Algoritmos Genéticos (Goldberg (1989) y Michalewics y Janikov (1992)), han sido reconocidos desde sus orígenes como una técnica muy adecuada para la resolución de problemas multiobjetivo. Una población de individuos puede explorar eficientemente un espacio de soluciones multifuncional sacando ventaja de su paralelismo intrínseco. Desde los primeros trabajos de Schaffer (Schaffer (1985)) se han ido refinando los planteamientos multiobjetivo añadiendo procedimientos de dominancia basados en rango (Fonseca y Fleming (1993)), estratificación del conjunto de soluciones usando la dominancia de Pareto (Srinivas y Deb) y muchos otros (tal y como se recoge en Coello *et al.* (2002)).

El problema de secuenciación multiobjetivo en flowshop ha sido tratado varias veces en la literatura.

Por ejemplo, en Murata *et al.* (1996) se propone un algoritmo genético multiobjetivo para optimizar el makespan, la tardanza total y el tiempo de flujo total mediante vectores de pesos generados aleatoriamente en cada iteración. Este algoritmo ha sido mejorado mediante la adición de procedimientos de búsqueda local en Ishibuchi y Murata (1998).

En Talbi *et al.* (2001) se propone un algoritmo evolutivo para resolver el problema de flowshop con dos objetivos: minización de makespan y tardanza total. En su propuesta se aplica un algoritmo genético para obtener una aproximación al frente de Pareto y se usa búsqueda local a dicho frente.

En Basseur *et al.* (2002) se presentan una aproximación denominada algoritmo genético de Pareto con mutación dinámica para optimizar un problema similar al tratado en Talbi *et al.* (2001), usando diferentes operadores genéticos de una manera adaptativa.

En Bagchi (1999) y Bagchi (2001) se extiende el algoritmo NSGA al contexto de secuenciación en flowshop entre diferentes problemas analizados incorporando elitismo.

En Brizuela (2001) se aplica NSGA al problema de secuenciación flowshop con tres objetivos: minimización del makespan, minimización del tiempo de flujo medio y de la tardanza media, estudiando el efecto de los operadores genéticos.

2. Modelado del problema de secuenciación

En este problema existen N trabajos o tareas que deben ser procesadas en cada una de las M máquinas disponibles (esto es, cada trabajo tiene m operaciones).

Las operaciones sobre cada trabajo se realizan en el mismo orden a través de las diferentes etapas productivas. En cada máquina del sistema, existe una matriz de tiempos de cambio de partida entre los diferentes trabajos que identifica el tiempo necesario para pasar de realizar un tipo de trabajo a otro diferente. Además se asume que hay capacidad infinita de almacén entre las máquinas, así como que los trabajos no se pueden interrumpir y no se pueden realizar más de un trabajo a la vez en cada máquina.

El problema es encontrar una secuencia de paso por todas las máquinas (se asume que es la misma) de tal forma que se alcancen los objetivos. El problema de secuenciación monobjetivo ha sido planteado por varios autores como en Srikar y Ghosh (1986) y Stafford y Tseng (1990) En este artículo se propone una versión mejorada de los mismos que incorpora la optimización multiobjetivo.

2.1. Índices

i, i1, i2: índice de los trabajos.(i,i1,i2=1...N)
j: índice de las máquinas (j=1,...,M)

2.2. Parámetros

p(i,j) : Tiempo de proceso del trabajo i en la máquina j.
ST(i1,i2, j) : Tiempo de ajuste de la máquina j para pasar de realizar el trabajo i1 al trabajo i2.
d(i) : Fecha de finalización del trabajo i.
M : Un número arbitrariamente grande.

2.3. Variables

x(i1,i2) vale 1 si el trabajo i1 es procesado inmediatamente antes que el trabajo i2 en todas las máquinas y 0 en otro caso.
c(i,j) fecha máxima de inicio de las operaciones sobre el trabajo i en la máquina j. Las operaciones incluyen el tiempo de ajuste y cambio, y el tiempo de proceso del lote.
CM(i,M) fecha de finalización del trabajo i en la última máquina.
CMAX fecha de finalización del último trabajo i en la última máquina o makespan.
Lmed, tardanza media entre los valores L(i)=CM(i,M)-d(i)

2.4. Modelo

Min:

F1=Cmax
F2=Lmed

Sujeto a:

$$\sum_{\substack{i1=0 \\ i1\neq i2}}^{N} \mathrm{x}(\mathrm{i}1,\mathrm{i}2) = 0 \qquad (1)$$

$\forall$i2=1…N
$\forall$j=1…M

$$\sum_{\substack{i1=1 \\ i1\neq i2}}^{N+1} \mathrm{x}(\mathrm{i}2,\mathrm{i}1) = 0 \qquad (2)$$

$\forall$i2=1…N
$\forall$j=1…M

$$\sum_{i=1}^{N+1} \mathrm{x}(0,\mathrm{i}) = 1 \qquad (3)$$

$\forall$i2=1…N
$\forall$j=1…M

$$\sum_{i=0}^{N} \mathrm{x}(\mathrm{i},\mathrm{n}+1) = 1 \qquad (4)$$

$\forall$i2=1…N
$\forall$j=1…M

.

$$c(i2,j) \geq c(i1,j) + \sum_{j=1}^{M} \sum_{\substack{i3=0 \\ i3 \neq i1 \\ i3 \neq i2}}^{N} \left(p(i1,j) + \left(ST(i3,i1,j) \cdot x(i3,i1)\right)\right) + M * \left(\sum_{j=1}^{M} x(i1,i2) - 1 \right) \tag{5}$$

$\forall$i1=1…N
$\forall$i2=1…N
$\forall$r=1…R

$$c(i2,j) \geq c(i2,j-1) + \left(\sum_{\substack{i1=0 \\ i1 \neq i2}}^{N} \sum_{j=1}^{M} \left(p(i2,j-1) + (ST(i1,i2,j-1) \cdot x(i1,i2))\right) \right) \tag{6}$$

$\forall$i2=1…N

$$CM(i2) \geq c(i2,M) + \sum_{\substack{i1=0 \\ i1 \neq i2}}^{N} \left(p(i2,M) + (ST(i1,i2,M) \cdot x(i1,i2)\right) \tag{7}$$

$\forall$i2=1…N

$$CMAX \geq CM(i2) \tag{8}$$

$\forall$i2=1…N

$$\text{Lmed} = \sum_{i=1}^{N} \left(CM(i,M) - d(i) \right) \tag{9}$$

$\forall$i=1…N

3. Planteamiento del algoritmo genético multiobjetivo

El algoritmo genético planteado para resolver el problema descrito en la sección anterior se ha denominado MOGASP (Multiobjective Genetic Algorithm for scheduling problems) y tiene la siguiente estructura:

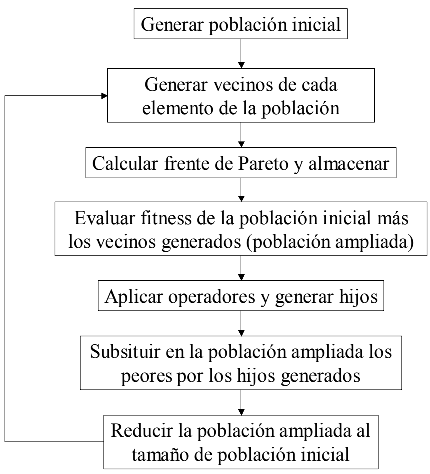

Figura 1. Estructura del algoritmo

A continuación se describen los bloques constructivos más significativos del algoritmo.

3.1. Generación de la población inicial

Se realiza de forma aleatoria y se genera el vecindario de cada individuo. La población original más el vecindario generado se denomina Npop*. A partir de esa etapa se calcula el conjunto de individuos no dominados y se almacena. Se hace un procedimiento de mejora exhaustiva sobre cada uno de los individuos del frente de Pareto, de tal forma que un individuo del frente es sustituido por los individuos generados a partir de ellos si estos últimos los dominan.

3.2. Evaluación de fitness

Para evaluar el fitness de cada individuo se usa la siguiente expresión:

$$\text{Fitness}=\frac{1}{rango \cdot (ind\_nich+1)} \tag{10}$$

Donde:

rango se calcula mediante la aproximación tradicional de Fonseca y P. J. Fleming (1993).
Ind_nich se evalúa a nivel de fenotipo usando la conocida función de sharing de Goldberg y Richardson (1987) según:

$$Ind\_nich=\frac{\sum_j \phi(d_{i,j})}{Npop^*} \tag{11}$$

donde Npop* es el numero de individuos de la población original más los generados en la primera fase.

3.3. Selección de individuos y aplicación de operadores

El operador de cruce usado se el conocido operador PMX. Sin embargo en el proceso de selección se usa un novedoso enfoque basado en el uso de procedimientos estratificados. La idea es que en la selección se pude influir sobre los procesos de exploración y explotación del entorno, intentando además mejorar el frente de Pareto.

Para ello se plantea una selección doble: basada en el fenotipo y basada en el genotipo. En ambos casos existen a su vez, dos y tres posibilidades.

En la selección basada en el fenotipo se eligen un número de determinado de individuos dominados y se cruzan con individuos también dominados teniendo en cuenta su fitness. Este procedimiento se usa para acercar los individuos más dominados a los menos dominados y más dispersos.

También se plantea elegir un número de individuos dominados teniendo en cuenta su fitness para cruzarlos con aquellos individuos del frente de Pareto. Esto se hace para acercar individuos dominados a zonas del frente de Pareto más dispersas.

En lo que respecta a la selección basada en el genotipo, un número de individuos dominados se eligen y se cruzan con otros individuos dominados que estén mas alejados a ellos a nivel de genotipo. Para ellos se calcula una distancia basada en la similitud en los genotipos. Esto se hace para asegurar diversidad en la población.

También se plantea elegir un número de individuos dominados se eligen y se cruzan con aquellos individuos del frente de Pareto que estén mas cercanos a ellos a nivel de genotipo. Esto se hace para asegurar una tendencia de los individuos dominados a aproximarse a la frontera de Pareto.

Por último, entre los individuos del frente de Pareto también se realizan una serie de cruces seleccionándolos mediante su genotipo. En este caso la prioridad de selección es entre aquellos más alejados entre si, con el objetivo de generar individuos de bajo rango en zonas poco pobladas del frente.

La manera de evaluar la distancia entre dos individuos basándose en sus genotipos es similar a la planteada por Talbi (2001). Así, para calcular la distancia entre dos individuos se genera una matriz NxN (siendo N el número de trabajos que se secuencian). Cada posición (i,j); se calcula de esta forma:

(i,j)= 1 si el trabajo i va antes que el j en ambos individuos y 0 en caso contrario y d(xy) es la suma de todas las posiciones de la matriz.

Al final de cada iteración, el tamaño de la población debe ser constante, es por ello que se ha diseñado un procedimiento para restablecerlo. Así, los individuos pertenecientes al frente de Pareto son transferidos íntegramente, mientras que sobre los restantes se eliminan los sobrantes hasta tener el tamaño de población inicial. Para eliminarlos, se eligen entre el subconjunto de individuos con mayor índice de nicho. Si hay varios individuos con el mismo índice de nicho y el número de ellos es menor que los que hay que suprimir se elimina todo el conjunto. Si es mayor, se eligen aleatoriamente hasta que se eliminan los que tocan.

4. Estudio experimental

Para comprobar el funcionamiento del algoritmo se han hecho diversas experiencias computacionales. En la siguiente figura se muestran los resultados de un experimento sobre un problema de flowshop con 10 trabajos y 5 máquinas. El tamaño de la población se ha elegido bajo (50 individuos) para analizar la capacidad de alcanzar con rapidez un frente de Pareto eficiente. El número de iteraciones planteado ha sido de 100.

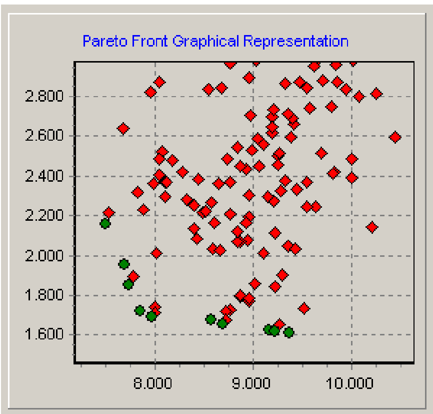

Figura 2. Frente de Pareto obtenido tras la experimentación

Se puede observar claramente que la propuesta realizada, genera un frente de Pareto continuo rápidamente.

Referencias

Bagchi Tapan, P. (1999) “Multiobjective Scheduling By Genetic Algorithms”, Kluwer Academic Publishers.

Bagchi Tapan, P. (2001) “Pareto-Optimal Solutions for Multi-objective Production Scheduling Problems”, Proceedings of the 1st International Conference on Evolutionary Multi-Criterion Optimization EMO 2001 Lecture Notes in Computer Science, Vol. 1993, Springer, pp. 458-471.

Basseur, M., Seynhaeve, F., Talbi, E.G. (2002) “Design of Multi-objective Evolutionary Algorithms to the Flow-shop Scheduling Problem”, Proceedings of the 2002 Congress on Evolutionary Computation CEC 2002, pp. 1151-1156.

Brizuela, C., Sannomiya, N., Zhao, Y. (2001) “Multi-objective Flow-Shop: Preliminary Results”, Proceedings of the 1st International Conference on Evolutionary Multi-Criterion Optimization EMO 2001 Lecture Notes in Computer Science, Vol. 1993, Springer, pp. 443-457.

Coello, C., Van Veldhuizen, D. y Lamont, G. (2002) “Evolutionary algorithms for solving multi-objective problems”, Kluwer.

Fonseca, C.M. y Fleming, P.J. (1993) “Genetic algorithms for multiobjective optimization: formulation, discussion and generalization”. Proceeding of the 5st ICGA. pp. 416-423.

Goldberg, D.E. (1989) “Genetic algorithms in search, optimization and machine learning”. Addison Wesley.

Goldberg, D.E. y Richardson J. (1987) Genetic algorithm with sharing for multimodal function optimization. Proceeding of the 2st ICGA. pp. 41-49.

Ishibuchi, H., Murata, T. (1998) A Multi-Objective Genetic Local Search Algorithm and its Application to Flowshop Scheduling, IEEE Transactions on Systems, Man and Cybernetics - Part C: Applications and Reviews, 28, pp. 392-403.

Michalewics, Z. y Janikow, C. (1992) “Genetic algorithms + Data structures =Evolution programs. Springer Verlag.

Murata, T., Ishibuchi, H., Tanaka, H., (1996) “Genetic Algorithms for Flowshop Scheduling Problems”, Computers and Industrial Engineering, Vol. 30, No. 4, pp. 1061-1071.

Schaffer, J.D. (1985) “Multiple objective optimization with vector evaluated genetic algorithms”. Proceeding of the 1st ICGA. pp. 93-100.

Srikar, B. N. y Ghosh, S. (1986) “A MILP Model for the n-job, M stage flowshop with sequence dependent setup times”, International Journal of Production Research, 24, 1459-1474.

Srinivas, N. y Deb, K. (1995) “Multiobjective optimization using nondominated sorting in Genetic Algorithms” Evolutionary Computation, 2, pp. 221-248.

Stafford, E. F. y Tseng, F. T. (1990) “On the Srikar-Ghosh MILP model for the NxM SDST flowshop problem”, International Journal of Production Research, 28, 1817-1830.

Talbi, E.G., Rahoudal, M., Mabed, M.H., Dhaenens, C. (2001) “A Hybrid Evolutionary Approach for Multicriteria Optimization Problems: Application to the Flow Shop”, Proceedings of the 1st International Conference on Evolutionary Multi-Criterion Optimization EMO 2001 Lecture Notes in Computer Science, Vol. 1993, Springer, pp. 416-428.