X Congreso de Ingeniería de Organización
Valencia, 7 y 8 de septiembre de 2006

Aplicación de EAGH al problema de equilibrado de líneas de montaje y al de *flow-shop* permutativo[*]

Albert Corominas[1], Rafael Pastor[1]

[1] Dpto. de Organización de Empresas. Instituto de Organización y Control (IOC). Universidad Politécnica de Cataluña. ETSEIB, Av. Diagonal, 647, planta 11, 08028 Barcelona. {albert.corominas/rafael.pastor}@upc.edu

Resumen

Una heurística greedy, utilizada para resolver un problema de optimización combinatoria dado, puede ser vista como un elemento de un conjunto de infinitas heurísticas, H, el cual es definido mediante una función que depende de varios parámetros. En Corominas (2005) se propone EAGH (Empirically Adjusted Greedy Algorithms), un procedimiento para determinar el mejor elemento de H para un conjunto de ejemplares de calibración del problema de optimización combinatoria a resolver. EAGH consiste, básicamente, en aplicar un algoritmo directo de optimización no lineal a una función de los parámetros que caracterizan H. EAGH es aplicado al problema de equilibrado de líneas de montaje y al de flow-shop permutativo.

Palabras clave: algoritmos *greedy*, equilibrado de líneas, *flow-shop*

1. Introducción a EAGH (Empirically Adjusted Greedy Algorithms),

A pesar de los espectaculares avances realizados en los últimos años en cuanto a la resolución exacta de problemas de optimización combinatoria, particularmente aquellos procedimientos basados en programación lineal mixta (Bixby, 2002), las técnicas heurísticas siguen desempeñando un papel muy importante. Y, dentro de éstas, los procedimientos greedy son unos de los más populares.

Las heurísticas greedy se basan en tomar, en cada iteración k, una decisión irreversible, la cual está basada en un *indicador* asociado a cada una de las posibles decisiones. Dicho indicador es calculado utilizando los datos del ejemplar a resolver y si el algoritmo es adaptativo también considera las consecuencias de las decisiones tomadas previamente.

Generalmente, un indicador para una heurística greedy puede ser una función, h, de los datos (o de los atributos asociados con las decisiones a tomar en cada iteración) y de varios parámetros adicionales. De esta forma, para un problema de optimización combinatoria, potencialmente hay un conjunto, H, de infinitas heurísticas greedy que se corresponden con los infinitos indicadores que pueden ser utilizados para tomar las decisiones. La suma (o tal vez otra función) de los valores de la función objetivo f, correspondiente a las soluciones obtenidas aplicando cada una de las heurísticas de H a un conjunto de ejemplares dado, depende de la heurística específica aplicada y, de esta forma, es una función, φ, de los parámetros de los que h depende.

[*] Financiado por los proyectos del MCYT DPI2004-03472 y DPI2004-05797, cofinanciado por FEDER.

En Corominas (2005) se propone EAGH (Empirically Adjusted Greedy Algorithms), un procedimiento para determinar el mejor elemento de H para un conjunto de ejemplares de calibración del problema de optimización combinatoria a resolver. EAGH consiste, básicamente, en aplicar un algoritmo directo de optimización no lineal, con variables reales (los parámetros Π), a la función φ de los parámetros que caracterizan H.

Sea un conjunto de ejemplares, I, de un problema P y uno de sus subconjuntos, $T \subset I$ (donde $|T| = N$), el cual es denominado conjunto de entrenamiento. Sea $X_{i,\bar{\Pi}}$ la solución del ejemplar $i \in T$ cuando el algoritmo greedy definido por la función heurística $h\left(a_i^k, \bar{\Pi}\right)$ es aplicado (siendo a_i^k los atributos a considerar para el ejemplar i en la decisión (iteración) k). Se define la función $\varphi(\Pi)$ como sigue: $\varphi(\Pi) = \sum_{i \in T} f\left(X_{i,\Pi}\right)$.

En general no se espera que la función $\varphi(\Pi)$ presente ninguna propiedad especial. Pero al aplicar la heurística al conjunto de entrenamiento, T, se puede calcular el valor de la función para cada conjunto de valores de los parámetros. Así pues, para buscar buenos valores de los parámetros Π es posible utilizar un algoritmo de optimización directo, como, por ejemplo, el de Nelder y Mead, también llamado de los poliedros flexibles (Nelder and Mead, 1965, y Corominas *et al.*, 1997).

Esquemáticamente el procedimiento EAGH consiste en lo siguiente:

(i) Generar un conjunto de ejemplares I del problema P y un subconjunto de I, T (un conjunto de ejemplares de entrenamiento).
(ii) Seleccionar un conjunto inicial de heurísticas greedy elementales.
(iii) Aplicar las heurísticas elementales a los ejemplares de T y establecer la mejor de ellas, la cual es denominada función heurística $h^*\left(a_i^k\right)$.
(iv) Diseñar la función $h\left(a_i^k, \Pi\right)$ de forma que $h^*\left(a_i^k\right)$ esté representada, utilizando elementos de las otras funciones heurísticas elementales.
(v) Determinar los mejores valores de los parámetros Π utilizando un algoritmo directo (esto es, optimizando $\varphi(\Pi)$) con los ejemplares del conjunto de entrenamiento T.
(vi) Validar los resultados aplicando la heurística resultante en la etapa (v) al resto de ejemplares de la misma población, $I \setminus T$.

Como se ha comentado, el algoritmo a utilizar en la etapa (v) es el de Nelder y Mead (N&M), que se basa en $n+1$ puntos ($n = |\Pi|$), en el espacio n-dimensional de los parámetros, que deben formar un hipertetraedro, preferiblemente regular. En cada iteración, se genera uno o más puntos en este espacio y se calcula el valor de la función (φ en nuestro caso) de cada uno de dichos puntos. En EAGH este cálculo implica aplicar la heurística definida por las coordenadas del punto (es decir, los valores de los parámetros del conjunto Π) a todos los ejemplares del conjunto de entrenamiento T. Como vértice inicial del hipertetraedro se utilizan los valores correspondientes a la heurística greedy, entre las pertenecientes al conjunto H, que ha proporcionado un mejor valor para los ejemplares del conjunto T.

EAGH ya ha sido aplicado con éxito para la determinación de secuencias en una máquina multiproducto sujeta a fallos y con costes cuadráticos (ver Corominas et al. 2005, para más

detalles). En este trabajo se presenta la aplicación de EAGH al problema de equilibrado de líneas de montaje y al problema de *flow-shop* permutativo. Ambos son problemas muy estudiados, para los que se han publicado heurísticas muy efectivas y difíciles de mejorar; de esta forma, la aplicación de EAGH se presenta con el objetivo principal de mostrar ejemplos que faciliten la comprensión del funcionamiento y aplicabilidad de EAGH, y no para obtener una heurística que supere a la mejor de las publicadas (aunque EAGH mejora las heurísticas greedy elementales utilizadas para diseñar la función heurística $h\left(a_i^k, \Pi\right)$ a calibrar).

2. Aplicación de EAGH al problema de equilibrado de líneas de montaje

Se ha diseñado un procedimiento EAGH para resolver el problema de equilibrado de líneas de montaje SALBP-1: dados un tiempo ciclo y las tareas a realizar, así como sus tiempos de proceso y las relaciones de precedencia entre las mismas, se trata de asignar las tareas a las estaciones de trabajo de modo que se minimice el número de dichas estaciones.

Se han programado seis algoritmos greedy elementales de la literatura, expuestos en Talbot et al. (1986), correspondientes a las funciones heurísticas $h_1 = h(RPW_i) = RPW_i$, $h_2 = h(NS_i) = NS_i$, $h_3 = h(t_i) = t_i$, $h_4 = h(NIS_i) = NIS_i$, $h_5 = h(RPW_i, NS_i) = {RPW_i}\big/{(NS_i + 1)}$ y $h_6 = h(NS_i, UB_i, LB_i) = {NS_i}\big/{(UB_i - LB_i)}$. Los atributos correspondientes a la decisión i en cualquier iteración k, a_i^k, son: RPW_i, peso de Helgeson y Birnie; NS_i, número de tareas seguidoras; t_i, tiempo de la tarea i; NIS_i, número de tareas seguidoras inmediatas; y LB_i y UB_i, primera y última estación a la que puede ser asignada la tarea i.

La función heurística planteada que define el conjunto H de infinitas heurísticas y que depende de 12 parámetros, $\Pi = \{\alpha_1, \beta_1, \alpha_2, \beta_2, \alpha_3, \beta_3, \alpha_4, \beta_4, \alpha_5, \beta_5, \alpha_6, \beta_6)\}$, es la siguiente:

$$h\left(a_i^k, \Pi\right) = h(RPW_i, NS_i, t_i, NIS_i, LB_i, UB_i, \alpha_1, \beta_1, \alpha_2, \beta_2, \alpha_3, \beta_3, \alpha_4, \beta_4, \alpha_5, \beta_5, \alpha_6, \beta_6) =$$

$$\alpha_1 \cdot RPW_i^{\beta_1} + \alpha_2 \cdot NS_i^{\beta_2} + \alpha_3 \cdot t_i^{\beta_3} + \alpha_4 \cdot NIS_i^{\beta_4} + \alpha_5 \cdot \left({RPW_i}\big/{(NS_i + 1)}\right)^{\beta_5} + \alpha_6 \cdot \left({NS_i}\big/{(UB_i - LB_i)}\right)^{\beta_6}$$

Para realizar el experimento computacional se ha generado de forma aleatoria un conjunto I de 2,000 ejemplares, del cual se extrae un subconjunto $T \subset I$ de 1,000 ejemplares; los conjuntos T y $I \setminus T$ son utilizados como de entrenamiento y de validación, respectivamente. Los ejemplares de I presentan las siguientes características: entre 50 y 150 tareas, según una ley uniforme discreta; *order strength* del grafo de precedencias comprendido entre 0.5 y 0.9, según una ley uniforme continua; tiempos de proceso de las tareas como valores enteros distribuidos uniformemente entre 5 y 50; y tiempo ciclo igual a 1, 2 ó 3 veces el valor del tiempo de la tarea de mayor tiempo de proceso.

Para cada una de las seis funciones heurísticas elementales $h\left(a_i^k\right)$, se han resuelto los 1,000 ejemplares del conjunto de entrenamiento T. La heurística $h_3 = t_i$ es la que proporciona los mejores resultados, con 36,011 estaciones de trabajo para los ejemplares de T, y 36,267 para los ejemplares del conjunto de validación.

En la tabla 1 se muestran los resultados de calibrar la función $h\left(a_i^k,\Pi\right)$ con los ejemplares de entrenamiento y aplicarla, posteriormente, a los de validación. Como vértice inicial del hipertetraedro en el algoritmo N&M $\left(V_0\right)$ se ha tomado los valores de los parámetros que corresponden a h_3: $\alpha_3=1;\alpha_i=0,\ i=1,2,4,5,6;\beta_i=1,\ i=1,...,6$. Por otro lado se han utilizado varios valores de la longitud de la arista del hipertetraedro $\left(\delta\right)$. Se presenta la suma del número de estaciones para los ejemplares del conjunto T $\left(\varphi_T(\Pi)\right)$ y $I\setminus T$ $\left(\varphi_{I\setminus T}(\Pi)\right)$.

Tabla 1. Suma del número de estaciones

V_0	δ	$\varphi_T(\Pi)$	$\varphi_{I\setminus T}(\Pi)$
h_3	1	35,848	36,135
h_3	2	35,842	36,110
h_3	0.5	35,840	36,091
h_3	0.1	35,833	36,083

Como se puede observar, la mejor heurística resultante de aplicar EAGH, $h_3\_0.1$, mejora en 184 estaciones de trabajo los resultados obtenidos con la mejor de las heurísticas iniciales, h_3.

Cuando el tiempo de cálculo necesario para ejecutar una heurística puede ser considerado despreciable, como ocurre con las heurísticas planteadas, parece necesario presentar comparaciones adicionales. Si se considera, para cada ejemplar de $I\setminus T$, el resultado de la mejor de las heurísticas iniciales para dicho ejemplar, son necesarias 36,229 estaciones; EAGH proporciona mejores resultados que la mejor heurística para cada ejemplar, ya que se necesitan 146 estaciones de trabajo menos. Además, si se utiliza, para cada ejemplar, la mejor de las heurísticas iniciales más EAGH, son necesarias 36,016 estaciones; en este caso, calibrar EAGH y añadirla al conjunto de heurísticas disponible para resolver el problema disminuye en 213 el número estaciones de trabajo necesarias.

El tiempo de cálculo necesario para calibrar la función $h\left(a_i^k,\Pi\right)$, que depende del valor de δ, así como el número de iteraciones del algoritmo N&M, son, respectivamente, de entre 13,762 y 20,702 segundos, y entre 94 y 181 iteraciones, en un Pentium IV a 3.4 GHz con 512 Mb RAM. Concretamente, para la función heurística $h_3\_0.1$ se han efectuado 100 iteraciones en 13,850 segundos. Si se estudia la evolución de los valores $\varphi_T(\Pi)$ y $\varphi_{I\setminus T}(\Pi)$, respecto al número de iteraciones, se comprueba que las condiciones de finalización impuestas en el algoritmo N&M programado (tamaño del hipertetaedro y dispersión de los valores de la función en sus vértices pequeños) han sido muy estrictas; con un número muy reducido de iteraciones, y, de esta forma, en un tiempo de cálculo menor, se podría calibrar una función heurística $h\left(a_i^k,\Pi\right)$ que proporcione mejoras destacables: con 3 iteraciones se alcanza el 88.20% de la mejora total que se obtiene del valor de $\varphi_T(\Pi)$ y con 71, el 100%.

La función heurística $h\left(a_i^k,\Pi\right)$ obtenida es la siguiente:

$$h_3\_0.1 = \quad 0.00698\cdot RPW_i^{0.960}+0.00281\cdot NS_i^{1.032}+1.04657\cdot t_i^{1.041}+0.00004\cdot NIS_i^{1.026}$$

$$-0.01219 \cdot \left(RPW_i \Big/ \left(NS_i + 1\right) \right)^{1.021} + 0.02237 \cdot \left(NS_i \Big/ \left(UB_i - LB_i\right) \right)^{1.006}$$

3. Aplicación de EAGH al problema de *flow-shop* permutativo

A continuación se considera el problema de minimizar el makespan, $C_{\max}$, en un *flow-shop* permutativo: procesar n piezas en m máquinas, de forma que la secuencia en que las piezas "ven" las máquinas es la misma para todas ellas y la secuencia de las piezas es la misma para todas las máquinas. Así pues, una solución del problema es una permutación de las n piezas.

Se han programado tres heurísticas greedy de la literatura que consisten en calcular el tiempo de proceso de cada pieza i en dos máquinas ficticias (S_{1i} y S_{2i}, respectivamente), basándose en los tiempos de proceso de i en las máquinas reales j (p_{ij}). De esta forma se reduce el problema con $m > 2$ máquinas a un problema de 2 máquinas, el cual se resuelve aplicando el algoritmo de Johnson (Johnson, 1954). El cálculo de S_{1i} y S_{2i} es, para las heurísticas utilizadas, el siguiente:

i) Companys (1966): $S_{1i} = \sum_{j=1}^{m-1} (m-j) \cdot p_{ij}$ y $S_{2i} = \sum_{j=1}^{m-1} j \cdot p_{i,j+1}$

ii) Dannenbring (1977): $S_{1i} = \sum_{j=1}^{m} (m-j+1) \cdot p_{ij}$ y $S_{2i} = \sum_{j=1}^{m} j \cdot p_{ij}$

iii) Campbell et al. (1970): $S_{1i}(K) = \sum_{j=1}^{K} p_{ij}$ y $S_{2i}(K) = \sum_{j=m-K+1}^{m} p_{ij}$ con $K = m-1$

Como se puede comprobar, las tres heurísticas elementales (denominadas h_{Co}, h_{Da} y h_{Ca}, respectivamente) son función de los mismos atributos: el número de máquinas, m, y los tiempos de proceso de las piezas en las máquinas, p_{ij}. De hecho, en la heurística original de Campbell et al. se resuelve el problema ficticio de dos máquinas $\forall K \in [1, m-1]$ y se selecciona la mejor solución encontrada; en este trabajo, y para poder comparar y asignar parámetros, se fija $K = m-1$.

El algoritmo diseñado, que define el conjunto H de infinitas heurísticas, consiste en resolver, con el algoritmo de Johnson, el problema ficticio de 2 máquinas resultante de calcular el valor de S_{1i} y S_{2i} de forma parametrizada. Así, EAGH es aplicado en la fase de calibración de la función heurística que proporciona los valores de S_{1i} y S_{2i}, como sigue:

$$S_{1i} = \sum_{j=1}^{m} \alpha_j \cdot p_{ij} \qquad S_{2i} = \sum_{j=1}^{m} \beta_j \cdot p_{ij}$$

Por ejemplo, el valor de los parámetros α_j y β_j para cada una de las heurísticas iniciales, con $m = 4$, es: $\alpha_{Co} = (3,2,1,0)$, $\beta_{Co} = (0,1,2,3)$; $\alpha_{Da} = (4,3,2,1)$, $\beta_{Da} = (1,2,3,4)$; $\alpha_{Ca} = (1,1,1,0)$, $\beta_{Ca} = (0,1,1,1)$.

Para realizar el experimento computacional se ha generado de forma aleatoria un conjunto I de 2,000 ejemplares, del cual se extrae un subconjunto $T \subset I$ de 1,000 ejemplares; el conjunto T y el $I \setminus T$ son utilizados, respectivamente, como conjunto de entrenamiento y de validación. Los ejemplares de I, para cada valor de $m \in [3,8]$, presentan las siguientes características: un valor de n entre 5 y 100 piezas, según una ley uniforme discreta; y tiempos de proceso p_{ij} como valores enteros distribuidos uniformemente entre 1 y 100.

Para cada heurística inicial, y cada valor de m, se han resuelto los 1,000 ejemplares del conjunto de entrenamiento T. La heurística h_{Co} es la que, para todo número de máquinas m probado, proporciona los mejores resultados; a continuación se sitúa la heurística h_{Da} que, para todos los valores de m, es la segunda mejor.

Posteriormente se ha calibrado la función heurística que calcula el valor de S_{1i} y S_{2i} con los ejemplares del conjunto T y se ha aplicado a los ejemplares del conjunto $I \setminus T$. Como vértice inicial del hipertetraedo en el algoritmo N&M se ha tomado los valores de los parámetros que corresponden a la heurística h_{Co}: $\alpha_{Co} = (m-1, m-2, ..., 1, 0)$ y $\beta_{Co} = (0, 1, ..., m-2, m-1)$; por otro lado se han utilizado tres valores de la longitud de la arista del hipertetraedro regular que se construye en N&M (δ). Se observa que con cualquier valor de δ los resultados obtenidos con EAGH son de mejor calidad que los que proporciona la mejor de las heurísticas iniciales, h_{Co}; y que para todos los valores de m las mejores soluciones son obtenidas con valores de δ iguales a 1 ó 2.

En este caso, el tiempo de ejecución de las heurísticas utilizadas también puede ser considerado despreciable. Si se compara, para cada ejemplar del conjunto $I \setminus T$, el resultado proporcionado por EAGH con el resultado de la mejor de las heurísticas iniciales para dicho ejemplar, se observa que EAGH proporciona mejores resultados cuando $m \geq 5$ y ligeramente perores cuando $m = 3$ y $m = 4$.

El tiempo de cálculo (y el número de iteraciones del algoritmo N&M) necesario para calibrar la función que calcula el valor de S_{1i} y S_{2i} depende del valor de m y del valor de δ, y ha estado entre 259 y 329 segundos (y entre 84 y 112 iteraciones) para $m = 3$ y entre 1,674 y 2,163 segundos (y entre 266 y 388 iteraciones) para $m = 8$.

A modo de ejemplo, a continuación se muestra la mejor función calibrada para calcular el valor de S_{1i} y S_{2i}, para $m = 3$ $(h_{Co}\_2(3))$ y $m = 5$ $(h_{Co}\_1(5))$:

$h_{Co}\_2(3)$:

$$S_{1i} = 2.6649 \cdot p_{i1} + 1.1375 \cdot p_{i2} - 0.4428 \cdot p_{i3}$$
$$S_{2i} = -0.2283 \cdot p_{i1} + 1.2641 \cdot p_{i2} + 2.7202 \cdot p_{i3}$$

$h_{Co}\_1(5)$:

$$S_{1i} = 5.0196 \cdot p_{i1} + 3.3642 \cdot p_{i2} + 1.6831 \cdot p_{i3} + 0.0833 \cdot p_{i4} - 1.9141 \cdot p_{i5}$$

$$S_{2i} = -1.5317 \cdot p_{i1} + 0.1497 \cdot p_{i2} + 1.7482 \cdot p_{i3} + 3.2318 \cdot p_{i4} + 4.8048 \cdot p_{i5}$$

4. Conclusiones

Una heurística greedy, utilizada para resolver un problema dado de optimización combinatoria, puede ser vista como un elemento de un conjunto de infinitas heurísticas, H, el cual es definido mediante una función que depende de varios parámetros. En Corominas (2005) se propone EAGH (Empirically Adjusted Greedy Algorithms), un procedimiento para determinar el mejor elemento de H para un conjunto de ejemplares de calibración del problema de optimización combinatoria a resolver.

En este trabajo, EAGH es aplicado a dos conocidos problemas de optimización combinatoria: el problema de equilibrado de líneas de montaje y el de flow-shop permutativo. Ambos son problemas muy estudiados, para los que se han publicado heurísticas muy efectivas y difíciles de mejorar; de esta forma, la aplicación de EAGH se presenta con el objetivo principal de mostrar ejemplos que faciliten la comprensión del funcionamiento y aplicabilidad de EAGH, y no para obtener una heurística que supere a la mejor de las publicadas. De todas formas EAGH proporciona mejoras notables respecto a las heurísticas greedy elementales utilizadas para diseñar la función heurística $h\left(a_i^k, \Pi\right)$ a calibrar.

Referencias

Bixby, R.E. (2002). Solving real-world linear programs: a decade and more of progress. *Operations Research* 50 (1), 3-15.

Campbell, H.G.; Dudek, R.A.; Smith, M.L. (1970). A heuristic algorithm for the n-job. m-machine sequencing problem. *Management Science*, 16, B630-637.

Companys, R. (1966). Métodos heurísticos en la resolución del problema del taller mecánico. *Estudios empresariales* 5 (66/2), 3-14.

Corominas, A.; Companys, R.; Coves, A.M.; Ferrer, J.; Roselló, X. (1997). *Mètodes quantitatius d'organització industrial: problemes no lineals.* Edicions UPC.

Corominas, A. (2005). Empirically Adjusted Greedy Algorithms (EAGH): A new approach to solving combinatorial optimisation problems. *Working paper IOC-DT-P-2005-22*. Universidad Politécnica de Cataluña. Barcelona.

Corominas, A.; Pastor, R.; Sánchez, A. (2005). Heurística, resultado de calibración de heurísticas, para la determinación de secuencias en una máquina multiproducto sujeta a fallos y con costes cuadráticos. *Actas del IX Congreso de Ingeniería de Organización* (CIO 2005), Gijón, 8 y 9 de septiembre de 2005.

Dannenbring, D.G. (1977). An evaluation of flow-shop sequencing heuristics. *Management Science*, 23, 1174-1182.

Johnson, S.M. (1954), Optimal two ─and three─ stages production schedules with setup times included. *Naval Research Logistics Quarterly*, 1, 61-68.

Nelder, J.A.; Mead, R. (1965). A simplex method for function minimization. *The Computer Journal* 7, 308-313.

Talbot, F.B.; Patterson, J.H.; Gehrleiv, W.V. (1986). A comparative evaluation of heuristic line balancing techniques. *Management Science*, 32, 431-453.