IX Congreso de Ingeniería de Organización
Gijón, 8 y 9 de septiembre de 2005

EAGH (Empirically Adjusted Greedy Algorithms)

Albert Corominas

Instituto de Organización y Control de Sistemas Industriales, Departamento de Organización de Empresas y ETS de Ingeniería Industrial de Barcelona. Universidad Politécnica de Cataluña. albert.corominas@upc.edu

Resumen

Con el fin de determinar la mejor heurística greedy en un conjunto infinito de heurísticas de este tipo, se propone un procedimiento que consiste en considerar una función heurística con parámetros y optimizar una función de los mismos. La aplicación del procedimiento a un problema de secuenciación ha producido muy buenos resultados.

Palabras clave: algoritmos *greedy*, optimización no lineal.

1. Introducción.

Se propone un procedimiento para el diseño de algoritmos heurísticos (AH) *greedy* adaptados a ejemplares procedentes de una población determinada. Más concretamente, se trata de determinar la mejor heurística *greedy* en un conjunto infinito de heurísticas de este tipo definido por una función en que las variables son magnitudes asociadas a los ejemplares y que incluye parámetros. Dado que el comportamiento de la heurística depende del valor atribuido a dichos parámetros, se suscita un problema de optimización en que las variables son los parámetros de la función que define la familia de heurísticas.

Como es sabido, los algoritmos heurísticos (a los que se han aplicado también, entre otras, las denominaciones de procedimientos heurísticos, métodos heurísticos –como en Silver, 2004– o simplemente heurísticas) han gozado de un gran predicamento para la resolución de problemas de optimización combinatoria, a causa del fracaso de los procedimientos exactos y también de una cierta interpretación de la teoría de la complejidad de problemas, según la cual el hecho de que un problema sea NP-duro permite renunciar sin más al intento de resolverlo de forma exacta. Como quiera que los algoritmos pueden clasificarse en exactos y heurísticos y lo que caracteriza a estos últimos es que no ofrecen la garantía de que la solución que obtienen sea óptima, las heurísticas presentan una gran variedad, lo que ha dado lugar a la publicación de diversos trabajos de síntesis, de los cuales (Silver, 2004) es, hasta donde alcanza nuestro conocimiento, el más reciente.

Los progresos en la teoría y en el *hardware* y el *software* (Bixby, 2002) han hecho ganar terreno a los procedimientos exactos, lo que es fácil de comprobar mediante el análisis de los trabajos publicados, entre los que cada vez son más frecuentes aquéllos que aplican procedimientos exactos incluso a problemas que se habían abordado, hasta hace poco, casi exclusivamente por medio de heurísticas. No obstante, los algoritmos heurísticos tienen todavía un gran interés, y siguen siendo objeto de investigación, porque son el único recurso para resolver problemas en que los procedimientos exactos siguen fracasando, porque permiten encontrar buenas soluciones en tiempos breves y también porque los procedimientos exactos suelen ser más eficientes si disponen de una buena solución inicial heurística. De todos modos, los algoritmos heurísticos presentan importantes limitaciones, derivadas en general de la falta de propiedades teóricas fuertes. La heurística no suele proporcionar información sobre la calidad de la solución obtenida, que sólo puede evaluarse si se dispone de una cota. Por otra parte, es frecuente que para un mismo problema se haya propuesto un número elevado de heurísticas y no es fácil determinar la más adecuada; la práctica más corriente consiste en aplicar las heurísticas que se desea comparar a un número más o menos grande de ejemplares, generados mediante algún procedimiento específico (Vg.: ley uniforme para los tiempos de proceso en el problema *flow-shop* o en problemas de equilibrado de líneas) y, por consiguiente, resulta imposible generalizar la conclusión a otros tipos de ejemplares. De hecho, se puede decir que no existe un método riguroso y que tenga aceptación general para la comparación de heurísticas (Barr et al., 1995; Hooker, 1995; Rardin y Uzsoy, 2001).

El objetivo de este trabajo consiste en determinar la mejor heurística *greedy*, para ejemplares (*instances*) procedentes de una población especificada, entre las pertenecientes a un conjunto infinito. Cada heurística se caracteriza por unos valores concretos de unos parámetros; la suma de los valores de la función objetivo para un conjunto de ejemplares "de entrenamiento" es, por consiguiente, una función de los parámetros y se buscan los valores de los mismos que optimicen dicha función.

La organización del resto de este trabajo es como sigue: La Sección 2 incluye una breve discusión sobre las heurísticas *greedy*, la Sección 3 describe la propuesta de procedimiento de selección de heurísticas que se ha denominado EAGH y la Sección 4 contiene unas sucintas conclusiones.

2. Heurísticas *greedy*.

Muchos algoritmos heurísticos propuestos en la literatura son de tipo *greedy* y, además, los algoritmos *greedy* son un componente esencial de los algoritmos GRASP (*Greedy Randomized Adaptative Search Procedure*; Feo y Resende, 1995).

No cabe aquí profundizar en la definición de algoritmo *greedy* y basta decir que estos algoritmos determinan una solución mediante unas iteraciones en cada una de las cuales se toma una decisión irreversible. Dicha decisión se basa en un indicador, más o menos complejo, que es función de los datos y, a veces, de las decisiones previas (en cuyo caso se dice que el algoritmo es adaptativo). Algunos autores consideran que la "miopía", es decir, el no tener en cuenta más que las consecuencias inmediatas de una decisión, es una característica inherente a los algoritmos *greedy;* no obstante, el cálculo del indicador puede tener parcialmente en cuenta las consecuencias de la decisión que se toma en una iteración sobre las iteraciones posteriores (se ha utilizado la expresión *look-ahead* para calificar a los algoritmos con esta propiedad).

En algunos casos hay razones claras para elegir el indicador en que se basa la decisión que se adopta en cada iteración; en otros, son varios los indicadores que pueden considerarse o que parecen razonables (por ejemplo, en problemas de equilibrado de líneas: duración de la tarea, número de tareas siguientes, etc.). No obstante, existe el riesgo de que la heurística *greedy* proporcione, no ya malas soluciones, sino incluso las peores (Bang-Jensen et al., 2004). Cuando hay varios indicadores posibles se pueden generar infinitos mediante, por ejemplo, una ponderación de los indicadores elementales, pero la elección de los pesos no es trivial. De hecho, estos se asignan a veces implícitamente y sin justificación plena (por ejemplo, en un problema de *flow-shop*, al calcular el tiempo de proceso en una máquina ficticia a partir de los tiempos de proceso en las máquinas reales). En general, un indicador para una heurística *greedy* puede ser una función de los datos y de unos parámetros. Así pues, para muchos problemas de optimización combinatoria se dispone potencialmente de infinitas heurísticas *greedy* y el resultado de aplicar estas heurísticas a un conjunto determinado de ejemplares es una función de los mencionados parámetros.

Sea un problema de optimización combinatoria P (como, por ejemplo, el problema unidimensional de la mochila o KP):

$$\begin{gathered} max\, f(X) \\ X \in E \end{gathered}$$

En un algoritmo *greedy* se toma una decisión, en cada iteración k, entre las pertenecientes a un conjunto de decisiones factibles F_k (en el caso del KP, se decide qué objeto se introduce en la mochila, entre los que no han sido ya incluidos o rechazados), de modo que la decisión, i_k^*, es una entre las que cumplen $i_k^* = \underset{i \in F_k}{\operatorname{argmin}} h\left(d_i^k\right)$, donde h se puede denominar la función heurística y d_i^k son los datos correspondientes a la decisión i en la iteración k (en el KP, los datos correspondientes al objeto i son su peso, w_i, y su valor, v_i, y la función heurística, por ejemplo, $h(w_i, v_i) = \dfrac{v_i}{p_i}$).

La función heurística puede hacerse depender de un conjunto de parámetros, $\wp$, con $\left|\wp\right| = n$, y entonces la decisión correspondiente a la iteración k es una de las que cumple:

$$i_k^* = \underset{i \in F_k}{\operatorname{argmin}} h\left(d_i^k, \wp\right)$$

(en el KP, por ejemplo, $h(w_i, v_i, \alpha, \alpha', \beta, \beta', \gamma, \gamma') = w_i^{\alpha} \cdot v_i^{\alpha'} + \beta \cdot w_i^{\gamma} + \beta' \cdot w_i^{\gamma'}$).

3. EAGH (Empirically Adjusted Greedy Algorithms)

Por consiguiente, la elección de la mejor heurística, entre las pertenecientes al conjunto determinado por una función de los datos y de unos parámetros, se puede abordar como la optimización de una función de dichos parámetros. Ésta debe ser, normalmente, la suma de los valores de la función objetivo del problema de optimización combinatoria, aunque cabe también, tal vez entre otras, la posibilidad de ponderar dichos valores.

Consideremos un conjunto de N ejemplares del problema P. Sea $X_{e,h(\bar{\wp})}$ es la solución del problema de optimización combinatoria que se obtiene al aplicar el algoritmo *greedy* al ejemplar problema de optimización combinatoria, con la función heurística $h\left(d_i^k, \bar{\wp}\right)$. Se puede definir entonces la función:

$$\varphi(\wp) = \sum_{e=1}^{N} f\left(X_{e,h(\wp)}\right)$$

No puede esperarse, en general, que esta función goce de ninguna propiedad especial que pueda ser conocida. Supondremos únicamente que podemos calcular, mediante la aplicación de la heurística al conjunto de ejemplares, el valor de la función para cada juego de valores de los parámetros. En consecuencia, para buscar buenos valores de los parámetros sólo podremos utilizar un algoritmo directo (es decir, que haga uso únicamente de los valores de la función), tal como el de Nelder y Mead (N&M) o de los poliedros flexibles.

Esta idea es parecida a una de las que están en la base del programa CALIBRA (Adenso-Díaz y Laguna, 2005), con el que se determinan los valores de los parámetros de una metaheurística. Pero en este caso, el de las metaheurísticas, los parámetros son explícitos y es bien conocida la dificultad para encontrar valores adecuados para ellos. En el caso de las heurísticas *greedy* los parámetros son implícitos; de hecho esta última afirmación implica una forma de "ver" cada heurística *greedy* como un elemento de un conjunto de infinitos elementos (por supuesto, son múltiples los conjuntos de que puede formar parte una heurística dada).

Esquemáticamente, el procedimiento propuesto consiste en:

(i) Generar un conjunto de ejemplares "de entrenamiento".

(ii) Determinar el mejor valor de los parámetros mediante un algoritmo directo (optimización de la función $\varphi(\wp)$).

(iii) Validar el resultado mediante la aplicación de la heurística a un nuevo conjunto de ejemplares procedente de la misma población.

El paso (ii) comprende las iteraciones del algoritmo directo de optimización a la función que se considere adecuada. El algoritmo que se ha elegido es el de Nelder y Mead (Nelder y Mead, 1965; Box et al,. 1969; Corominas et al., 1997). Hay otros algoritmos que podrían utilizarse en el procedimiento, pero el de N&M ha sido ampliamente aplicado, con buenos resultados, desde que se publicó y referencias recientes (Anjos et al., 2004; Chelouah y Siarry, 2005) indican que todavía es uno de los instrumentos utilizados para resolver problemas de optimización con funciones objetivo no diferenciables. El algoritmo N&M parte de $n+1$ puntos que han de formar un hipertetraedro en el espacio n-dimensional de los parámetros; en cada iteración se genera uno o más puntos en dicho espacio y se calcula el valor de la función en cada uno de estos puntos. El cálculo de la función en un punto supone aplicar a todos los ejemplares del conjunto de entrenamiento la heurística, con el valor de los parámetros correspondiente a las coordenadas de dicho punto. El algoritmo conduce a un punto que probablemente es próximo a un óptimo local; dado que las propiedades de la función no son conocidas, puede suponerse que, en general, es multimodal, por lo cual no puede asegurarse la aproximación al óptimo global. Esta dificultad es común, por otra parte, a todos los problemas de optimización local.

Uno o dos puntos iniciales del algoritmo de N&M pueden corresponder a las heurísticas *greedy* ya conocidas, entre las pertenecientes al conjunto definido por la función h, que resulten mejores para el conjunto de ejemplares de entrenamiento (hacer corresponder tres o más puntos a heurísticas conocidas puede dar una forma poco deseable al hipertetraedro inicial, el cual conviene que sea regular). Así se asegura que la calidad del resultado proporcionado por EAGH no es inferior a la correspondiente a la mejor heurística del conjunto, puesto que en el algoritmo N&M la última solución obtenida no puede ser peor que la mejor de las iniciales.

Estas ideas se han aplicado a un problema de secuenciación, con resultados muy satisfactorios, muy superiores a los obtenidos con las heurísticas *greedy* basadas en indicadores elementales (Corominas et al., 2005).

Por supuesto, el procedimiento EAGH puede aplicarse también a un único ejemplar (es el caso particular $N=1$) por lo que entonces ha de considerarse como un algoritmo heurístico específico (que podemos denominar EAGH-1, en contraposición con EAGH-N) más que como un procedimiento para seleccionar el mejor elemento de un conjunto de heurísticas; deja de tener sentido, en este caso, la distinción entre las funciones f y φ. Este algoritmo no puede ser peor que los correspondientes a los vértices del hipertetraedro inicial del algoritmo N&M, aunque la probabilidad de encontrar unos valores de los parámetros que supongan una mejora con relación al mejor de dichos vértices es menor que cuando $N \gg 1$ (por ejemplo, puede suceder que todos los vértices del hipertetraedro tengan asociado el mismo valor de la función f, en cuyo caso es poco probable que N&M pueda progresar).

Así pues, dados N ejemplares de un problema se puede aplicar EAGH-N para resolverlos y determinar un juego de valores de los parámetros con el que se define una heurística para aplicar a nuevos ejemplares. O bien, aplicar EAGH-1 a cada ejemplar, lo cual puede proporcionar, en conjunto, mejores soluciones, pero sin posibilidad de extrapolación a nuevos ejemplares.

4. Perspectivas

Aplicar EAGH a un conjunto de problemas de optimización combinatoria y determinar los mejores valores de los parámetros para ejemplares procedentes de poblaciones especificadas.

Referencias

Adenso-Díaz, B.; Laguna, M. (2005) Fine-tuning of algorithms using fraccional experimental designs and local search. Próxima publicación en *Operations Research*.

Anjos, M.F.; Cheng, R.C.H.; Currie, C.S.M. (2004) Maximizing revenue in the airline industry under one-way pricing. *Journal of the Operational Research Society* 55, 535-541.

Bang-Jensen, J.; Gutin, G.; Yeo, A. (2004) When the greedy algorithm fails. *Discrete optimization* 1, 121-127.

Barr, R.; Golden, B.; Kelly, J.; Resende, M.; Stewart Jr, W. (1995) Designing and reporting on computational experiments with heuristic methods. *Journal of Heuristics* 1, 9-32.

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

Box, M.J; Davies, D.; Swann, W.H. (1969) Non-linear optimization techniques. Oliver & Boyd.

Chelouah, R.; Siarry, P. (2005) A hybrid method combining continous tabu search and Nelder-Mead simplex algorithms for the global minimization of multiminima functions. *European Journal of Operational Research* 161, 636-654.

Corominas, A.; Companys, R.; Coves, A.-M.; Ferrer, J.; Roselló, X. (1997) *Mètodes quantitatius d’organitzación industrial : pRoblemes no lineals*. Edicions UPC.
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. CIO2005, IX Congreso de Ingeniería de Organización, Gijón.
Feo, T.A.; Resende, M.G.C. (1995) Greedy randomized adaptative search procedures. *Journal of Global Optimization* 6, 109-133.
Hooker, J.N. (1995) Testing heuristics: we have it all wrong. *Journal of Heuristics* 1, 33-42.
Nelder, J.A.; Mead, R. (1965) a simplex method for function minimization. *The Computer Journal* 7, 308-313.
Rardin, R.L.; Uzsoy, R. (2001) Experimental evaluation of heuristic optimization algorithms: a tutorial. *Journal of Heuristics* 7, 261-304.
Silver, E.A. (2004). An overview of heuristic solution methods. *Journal of the Operational Research Society* 55 (9) 936-956.