Cátedra Nissan

-PROTHIUS-

Introducción al BDP

Joaquín Bautista Valhondo, Ramón Companys Pascual y Albert Corominas Subias

WP-07/2010
(Rec. DIT 92/04- BCC - 1992)

Departamento de Organización de Empresas

Universidad Politécnica de Cataluña

Publica:

Universitat Politècnica de Catalunya
www.upc.edu

Edita:

Cátedra Nissan
www.nissanchair.com
director@nissanchair.com

TITOL:

INTRODUCCION AL BDP

Autors: J. Bautista
R. Companys
A. Corominas

Document Intern de Treball
(D.I.T.) 92/04

Barcelona, 1992

Departament d'Organització d'Empreses
Universitat Politècnica de Catalunya

Trabajo realizado dentro del Proyecto de Investigación
PB-0504 (DGICYT)

NOTA DE TRABAJO
INTRODUCCION AL BDP

por J. Bautista, R. Companys & A. Corominas

DEPARTAMENT D'ORGANITZACIO D'EMPRESES
ESCOLA TECNICA SUPERIOR D'ENGINYERS INDUSTRIALS DE BARCELONA
UNIVERSITAT POLITECNICA DE CATALUNYA

(versión inicial 25-05-1991; revisión 13-04-1992)

1. CONSIDERACIONES PREVIAS

La resolución de muchos problemas combinatorios puede reducirse a la búsqueda de un camino óptimo (mínimo o máximo) entre dos vértices específicos en un grafo valorado en los arcos o/y en los vértices. Entre dichos problemas cabe citar: el problema MONDEN, el problema del viajante de comercio (TSP), el equilibrado de líneas de montaje, etc. Por otra parte dichos problemas (y el grafo asociado) admiten dos procedimientos (que pueden estar estrechamente relacionados):

<u>procedimiento</u> Q: consistente en la obtención rápida de una solución heurística (generalmente "greedy") razonablemente buena, es decir próxima a la solución óptima,

<u>procedimiento</u> K: consistente en la obtención de una cota (inferior en el caso de búsqueda de un mínimo) de los caminos entre un vértice cualquiera del grafo y uno de los vértices específicos,

En estas condiciones es posible la aplicación del BDP ("bounded dynamic programming") que reduce considerablemente el volumen y tiempo de cálculo de la solución óptima respecto a los métodos tradicionales de determinación de caminos extremos en los grafos.

1.1. Comentario.

Existe una aplicación biyectiva entre el proceso de construcción de un camino del grafo y el de construcción de una solución posible del problema combinatorio, el vértice E está asociado al inicio de la construcción y el S a la disponibilidad de una solución completa. Dicha solución tiene un valor igual al del camino elegido para ir de E a S. En un TSP con 20 ciudades los caminos entre E y S pueden representar las 19! permutaciones circulares de las 20 ciudades.

Por tanto BDP será posible en la medida en que la solución del problema combinatorio pueda construirse progresivamente y puedan aplicarse procedimientos del tipo Q y K.

Además de existir varios caminos, tal vez muchos, entre E y S (tantos como soluciones posibles del problema combinatorio), dado un vértice i del grafo, también existirán varios caminos posibles de E a i y varios de i a S. En principio cunado en el proceso de exploración del grafo alcancemos el vértice i, los caminos de E a i habrán sido detenidamente considerados y dispondremos del mejor de ellos asociado al mejor valor. Los caminos de i a S no habrán sido explorados todavía, pero el procedimiento K nos proporcionará una cota de su valor.

En el ejemplo TSP con 20 ciudades un vértice i corresponderá a los caminos hamiltonianos con k+1 de las 20 ciudades (la tomada como origen arbitrario, A, y k explícitamente definidas más, de las cuales una, X, se considerará en forma especial). El camino óptimo de E a i es la imagen del camino hamiltoniano de menor valor que pasa por dichas ciudades con el origen arbitrario en el extremo inicial y X en el extremo final. Los siguientes de i corresponden a añadir una ciudad más de las 19-k que quedan (si k<19; si k=19 el único siguiente es S, que corresponde a añadir la ciudad A para cerrar el circuito). Un arco (i,j) tendrá asociado un valor correspondiente a la distancia desde X hasta la nueva ciudad final del camino hamiltoniano. Los caminos de i a S son la imagen de los caminos hamiltonianos que pasan por 21 - k ciudades (las 19-k no tenidas en cuenta anteriormente más X y A) y que empiezan en X y terminan en A. Dichos caminos no habrán sido evaluados detenidamente y a través del procedimiento K obtendremos una cota inferior de su valor.

Un mismo problema combinatorio puede estar asociado a diferentes grafos, que dependerán de la forma de concebir la construcción de soluciones.

El formato del problema y el del grafo deben admitir el Principio de BELLMAN: si el camino óptimo de E a S pasa por i, los tramos Ei e iS son óptimos considerando los caminos de E a i y de i a S respectivamente.

Las posibles dificultades que acechan son:

- la no disponibilidad explícita del grafo, definido implicitamente a través de unas reglas,

- el número desmesurado de vértices y caminos del mismo, que impiden su enumeración y la utilización de algoritmos de caminos mínimos basados en iteraciones exhaustivas.

2. FORMATO DEL GRAFO MONDEN.

El grafo G asociado al problema MONDEN, al que hemos aplicado con éxito BDP, posee las siguientes propiedades:

g1) el grafo es orientado, valorado en los arcos y sin bucles ni circuitos,

g2) existe un vértice único E sin precedentes y un vértice único S sin siguientes,

g3) todos los caminos de E a S tienen exactamente el mismo número U de arcos.

En consecuencia podemos asociar niveles a los vértices del grafo. Si asignamos el nivel 0 al vértice E, S tendrá el nivel U y todos los arcos (i,j) unirán un vértice de nivel k con otro de nivel k+1.

En lo que sigue consideraremos que el grafo asociado al problema combinatorio que pretendemos resolver satisface las condiciones anteriores y que buscamos un camino mínimo. Llamaremos a(i,j) al valor asociado al arco (i,j).

3. PODADO DEL GRAFO

Sea z0 la mejor solución obtenida en el problema combinatorio mediante un procedimiento heurístico. Sea i un vértice del grafo de nivel k > 0 y sea v(i) el valor acumulado a lo largo de un camino mínimo desde E hasta i.

<u>Podado 1</u>: Dado un vértice i de nivel k si

$$v(i) \geq z0$$

los caminos que pasan por i no pueden conducir a un valor mejor que z0; por tanto no es necesario considerar dichos caminos, el vértice i puede eliminarse de G así como los arcos que inciden en él.

Sea c(i) una cota inferior del valor de los caminos del grafo que unen i con S,

<u>Podado 2</u>: Dado un vertice i de nivel k si

$$v(i) + c(i) \geq z0$$

los caminos que pasan por i no pueden conducir a un valor mejor que z0; por tanto no es necesario considerar dichos caminos, el vértice i puede eliminarse de G así como los arcos que inciden en él.

Aplicando los podados anteriores algún vértice puede quedar "colgado" (no conectado a E) lo que permite eliminarlo también.

Podado 3: Todo vértice j de G distinto de E que en virtud de los podados 1, 2 y/o 3 quede sin precedentes o siguientes puede eliminarse de G así como todos los arcos que inciden en él.

4. PROCEDIMIENTO BDP

El procedimiento BDP construye progresivamente el grafo procurando generar explícitamente sólo aquellos vértices del mismo por los que potencialmente puede pasar el camino óptimo, lo que reduce considerablemente la porción descrita y explorada del grafo.

FASE 1 - Hacer k=0, generar E y situarlo en la lista ABIERTOS.
Poner el puntero de E hacia E; v(E) = 0
Crear la lista CERRADOS vacía
Calcular c(E)
Si v(E) + c(E) = z0 FIN (el valor óptimo es z0)

FASE 2 - Sea k el nivel en curso, tomar un vértice de ABIERTOS de nivel k; si hay alguno ir a FASE 4, si no hay ninguno hacer k = k+1

FASE 3 - Si no hay ningún vértice en ABIERTOS de nivel k FIN (el valor óptimo es z0), en caso contrario:
Si k = U FIN, el único vértice a este nivel es el vértice S y el valor v(S) es el óptimo buscado; mediante los punteros construímos regresivamente el camino óptimo
Si k<U tomar un vértice cualquiera en ABIERTOS de nivel k e ir a FASE 4

FASE 4 - Sea i el vértice tomado de nivel k; generar sus siguientes G(i) y pasar i a CERRADOS con su puntero

FASE 5 - Si G(i) es vacío volver a FASE 2, en caso contrario tomar un vértice j, j ∈ G(i)

FASE 5.1 - Calcular

$$w(j) = v(i) + a(i,j)$$

Si w(j) ≥ z0 eliminar j de G(i), ir a FASE 5

FASE 5.2 - Calcular c(j)

Si w(j) + c(j) ≥ z0 eliminar j de G(i), ir a FASE 5

FASE 5.3 - Buscar si j había sido generado ya y está en <u>ABIERTOS</u>

FASE 5.3.1 - Si está comparar w(j) con v(j)

si w(j) ≥ v(j) eliminar j de G(i), ir a FASE 5

si w(j) < v(j) substituir en <u>ABIERTOS</u> v(j) por w(j) y poner el puntero de j orientado hacia i, eliminar j de G(i) e ir a FASE 5

FASE 5.3.2 - Si no está, poner j en <u>ABIERTOS</u> asociado al valor v(j)=w(j) y poner el puntero de j dirigido hacia i; eliminar j de G(i) e ir a FASE 5.

4.1. Comentario

Sería posible alterar FASE 3 y FASE 4 para explorar a lo largo y no a lo ancho usando al pasar del nivel k al k+1 el vértice de nivel k en ABIERTOS con mejor evaluación (menor) v(i) + c(i). En consecuencia habría que buscar si un vértice se había generado ya tanto en ABIERTOS como en <u>CERRADOS</u> retocando (propagando), si procede, los valores a lo largo de los caminos existentes mediante los punteros (que deberían ser de doble dirección).

El número de vértices que aparecen explícitamente en las listas es función de dos factores:

- la bondad de la solución inicial z0, es decir de lo cerca o lejos que se encuentre del valor óptimo (procedimiento Q),

- la calidad de la cota c(i) (procedimiento K),

En algunas circunstancias el número de vértices que aparecen en las listas puede superar los límites razonables, pudiendose recurrir a su reducción mediante una ventana.

4.2. Utilización de una ventana

Si el número de vértices de <u>ABIERTOS</u> de un nivel cualquiera k lo limitamos a un cierto valor H (ancho de la ventana) por razones de tiempo o tamaño de memoria el procedimiento puede transformarse en heurístico al no garantizar el óptimo. La regla adicional consistirá en contar los vértices de nivel k situados en <u>ABIERTOS</u>. En el momento en que haya H y proceda escribir uno nuevo (FASE 5.3.2) se eliminará aquél de los H+1 cuyo valor v(j) + c(j) sea el peor (mayor). Esta eliminación, dado que trabajamos con cotas, introduce incertidumbre en el éxito del procedimiento ya que puede impedir que hallemos el camino óptimo al privarnos de un vértice por donde éste pasa, vértice que comparado con otros de su nivel no parecía de los más favorecidos. Los fracasos por dicha causa del procedimiento pueden adoptar dos formas: alcanzar un valor de v(S) definiendo un camino desde E que no sean ni el valor ni el camino óptimos, o bien llegar a un nivel k vacío de vértices (debido a las sucesivas eliminaciones) lo que conduce a la conclusión de que el mejor valor conocido continua siendo z0, no siendo éste el óptimo. En el primer caso el fracaso es relativo ya que $v(S) < z0$. Una forma más frustante de la incertidumbre que puede introducir una ventana, común con otros procedimientos, es la de permitirnos obtener el valor óptimo sin que dispongamos de argumentos para garantizar que lo es.

Denominaremos z1(k) el mejor valor (menor) de los vértices eliminados a causa de la ventana al nivel k, y z1 al mínimo de todos ellos.

<u>Teorema</u> - Si aplicamos el procedimiento BDP con una ventana de anchura H y alcanzamos el valor v(S) sin que haya tanido que eliminarse ningún vértice a causa de la ventana o tal que:

$$v(S) \leq z1$$

v(S) es el valor óptimo y el camino obtenido con los punteros es un camino óptimo.

<u>Corolario</u> - Si en las condiciones anteriores $v(S) > z1$ la separación máxima de v(S) respecto al valor óptimo es $v(S) - z1$. (En cualquier caso si alcanzamos dicho valor $v(S) < z0$ y constituye una solución heurística mejor que la disponible anteriormente con la que podemos reiniciar el proceso).

5. GENERALIZACIONES

Las limitaciones más importantes impuestas han sido las relativas a la forma del grafo (g1, g2, g3). Es posible relajar g1 y g3. Los comentarios que siguen no pretenden alcanzar una formalización estricta sino señalar algunas líneas para futuros desarrollos.

R1 - Si las valoraciones son sobre los vértices o a la vez sobre los arcos y los vértices únicamente deben alterarse los valores a(i,j) en consecuencia,

R2 - Si todos los caminos entre E y S no tienen el mismo número de arcos es posible establecer niveles pero todos los siguientes inmediatos de un vértice de nivel k no deberán encontrarse forzosamente en el nivel k+1; esto posiblemente complicará la utilización de la ventana, cuya anchura deberá relacionarse con el total de vértices en <u>ABIERTOS</u> y no con el total de vértices de un nivel dado en <u>ABIERTOS</u>, pero el procedimiento a utilizar puede ser esencialmente el mismo, R3 - Si el grafo puede tener circuitos (de valor no negativo en caso de buscar un mínimo) no podemos establecer una clasificación por niveles de los vértices y la estrategia de exploración es un elemento crítico. Un vértice puede sufrir alteraciones (en su valor y en su puntero) después de haber generado sus siguientes inmediatos por lo que tanto <u>ABIERTOS</u> como <u>CERRADOS</u> (y eliminados) son provisionales, lo que exige punteros con doble dirección; si un vértice es reevaluado en el proceso de propagación sus siguientes deben ser regenerados, pues es posible que alguno haya sido eliminado. El algoritmo no termina hasta que S ha sido evaluado y <u>todos los vértices</u> han pasado a <u>CERRADOS</u> (o han sido eliminados). La utilización de ventanas es más delicada, pudiendose emplear una para el conjunto <u>ABIERTOS</u> más <u>CERRADOS</u>. (Si es posible conocer cuando un vértice está definitivamente cerrado pues sus precedentes no pueden modificarse más, será útil descomponer <u>CERRADOS</u> en dos sublistas).

6.- UTILIZACION DE LA BDP ("BOUNDED DYNAMIC PROGRAMMING").

La Programación Dinámica (DP) y el branch-and-bound (B&B) son dos procedimientos muy generales de optimización para problemas combinatorios. En **IBARAKI** se pone de manifiesto las relaciones entre ambos: ciertos esquemas de aplicación del branch-and-bound generan grafos que coinciden con los que podrían obtenerse mediante la DP y, por otra parte, cabe la posibilidad de utilizar cotas en un esquema de Programación Dinámica; en cierto modo, pues, la DP y el B&B pueden considerarse como casos particulares de un tipo de procedimiento más general. Esta observación es interesante conceptualmente pero no sugiere inmediatamente consecuencias de orden práctico.

La utilización de cotas en un esquema de PD no es una idea nueva. En **MORIN-MARSTEN** se describe este enfoque y su aplicación al TSP y a un problema de la mochila no lineal ("nonlinear knapsack problem") y los mismos autores (en **MARSTEN-MORIN**) generalizan el procedimiento, al que denominan *híbrido* ("hybrid approach"). Más recientemente (en **CARRAWAY-SCHMIDT**), siguiendo las ideas expuestas en **MORIN-MARSTEN** se introducen cotas en un esquema de PD propuesto en **NEMHAUSER-ULLMANN** para resolver un problema de asignación de recursos a proyectos interdependientes. En todos los casos, evidentemente, el uso de cotas permite eliminar vértices por lo cual, dada una cierta disponibilidad de memoria, es factible resolver problemas de dimensiones algo mayores; no obstante, en problemas en que el número de vértices crece exponencialmente con la dimensión de los ejemplares, el aumento de las dimensiones tratables es reducido y, en definitiva, la programación dinámica, con cotas o sin ellas, sólo permite resolver problemas de pequeñas dimensiones (por ejemplo, en **CARRAWAY-SCHMIDT** se da cuenta de la resolución de ejemplares del problema con no más de 12 proyectos).

La consideración explícita de la limitación en la disponibilidad de memoria permite resolver problemas de grandes dimensiones, sin renunciar a priori a la posibilidad de obtener el óptimo y de tener la certeza de que ha sido alcanzado. Al procedimiento basado en la utilización de cotas en un esquema de programación dinámica y con un tratamiento que tiene en cuenta la limitación en el número de vértices del grafo que se puede almacenar lo hemos denominado Programación Dinámica Acotada (BDP); el hecho de dar un nombre propio al procedimiento parte de la conocida observación según la cual algo que no tiene nombre es innombrable y por tanto no existe a fines prácticos: es decir, el hecho de que un procedimiento tenga un nombre facilita que sea tenido en cuenta como un elemento en la caja de herramientas del "problem solver".

Robert L. CARRAWAY, Robert L. SCHMIDT.- "An Improved Discrete Dynamic Programming Algorithm for Allocating Resources among Interdependent Projects". Management Science, Vol. 37, No. 9, September 1991, pp. 1195-1200.

Toshihide IBARAKI.- Enumerative Approaches to Combinatorial Optimization. Part I: Annals of Operations Research, Vol. 10, No. 1-4, 1987; Part II: Annals of Operations Research, Vol. 11, No. 1-4, 1987.

Roy E. MARSTEN, Thomas L. MORIN.- "A Hybrid Approach to Discrete Mathematical Programming". Mathematical Programming 14, 1978, pp. 21-40.

Thomas L. MORIN, Roy E. MARSTEN.- "Branch-and-Bound Strategies for Dynamic Programming". Operations Research, Vol. 24, No. 4, July-August 1976, pp. 611-627.

G. NEMHAUSER, Z. ULLMANN.- "Discrete Dynamic Programming and Capital Allocation". Management Science, Vol. 15, 1969, pp. 494-505.