Optimización con Ramificación y Poda con Backtracking

Introducción

Esta técnica de diseño, cuyo nombre en castellano proviene del término inglés Branch and Bound, se aplica normalmente para resolver problemas de optimización. Ramificación y Poda, al igual que el diseño Vuelta Atrás (backtracking), realiza una enumeración parcial del espacio de soluciones basándose en la generación de un árbol de expansión.
Una característica que le hace diferente al diseño de Vuelta Atrás es la posibilidad de generar nodos siguiendo distintas estrategias. Recordemos que el diseño Vuelta Atrás realiza la generación de descendientes de una manera sistemática y de la misma forma para todos los problemas, haciendo un recorrido en profundidad del árbol que representa el espacio de soluciones. El diseño Ramificación y Poda en su versión más sencilla puede seguir un recorrido en anchura (estrategia LIFO) o en profundidad (estrategia FIFO), o utilizando el cálculo de funciones de coste para seleccionar el nodo que en principio parece más prometedor (estrategia de mínimo coste o LC).

Como veremos a continuación, además de estas estrategias, la técnica de Ramificación y Poda utiliza cotas para podar aquellas ramas del árbol que no conducen a la solución óptima. Para ello calcula en cada nodo una cota del posible valor de aquellas soluciones alcanzables desde ése. Si la cota muestra que cualquiera de estas soluciones tiene que ser necesariamente peor que la mejor solución hallada hasta el momento no necesitamos seguir explorando por esa rama del árbol, lo que permite realizar el proceso de poda.

En consecuencia, y a la vista de todo esto, podemos afirmar que lo que le da valor a esta técnica es la posibilidad de disponer de distintas estrategias de exploración del árbol y de acotar la búsqueda de la solución, que en definitiva se traduce en eficiencia. La dificultad está en encontrar una buena función de coste para el problema, buena en el sentido de que garantice la poda y que su cálculo no sea muy costoso. Si es demasiado simple probablemente pocas ramas puedan ser excluidas. Dependiendo de cómo ajustemos la función de coste mejor algoritmo se deriva.

Conceptos básicos

El algoritmo de Ramificación y Poda (branch and bound) es una técnica basada en el recorrido de un árbol de soluciones que: Realiza un recorrido sistemático en un árbol de soluciones; El recorrido no tiene por qué ser necesariamente en profundidad sino que seguirá una estrategia de ramificación, guiada por estimaciones del beneficio, que se realizarán para cada nodo; Se usan técnicas de poda para eliminar nodos que no lleven a la solución óptima; La poda se realiza estimando en cada nodo las cotas de beneficio que se pueden obtener a partir del mismo; Un concepto fundamental para entender el algoritmo de ramificación y poda es el de nodo vivo. Nodo vivo del árbol de expansión es un nodo con posibilidades de ser ramificado, es decir, un nodo que no ha sido podado. Para determinar en cada momento que nodo va a ser expandido y dependiendo de la estrategia de búsqueda seleccionada, necesitaremos almacenar todos los nodos vivos en alguna estructura que podamos recorrer.

Metodología de resolución

El método general a aplicar para la resolución de un problema mediante este algoritmo es el siguiente:

1. Para cada nodo i, tendremos: - Cota superior (CS(i)) y Cota inferior (CI(i)) del beneficio (o coste) óptimo que se podrían alcanzar a partir de ese nodo. Estas cotas determinarán el momento en el que se podrá realizar una poda. Estimación del beneficio (o coste) óptimo que se puede obtener a partir de ese nodo. Esta estimación se puede calcular como una media de las cotas anteriores y puede ayudar a decidir qué parte del árbol se explorará primero.

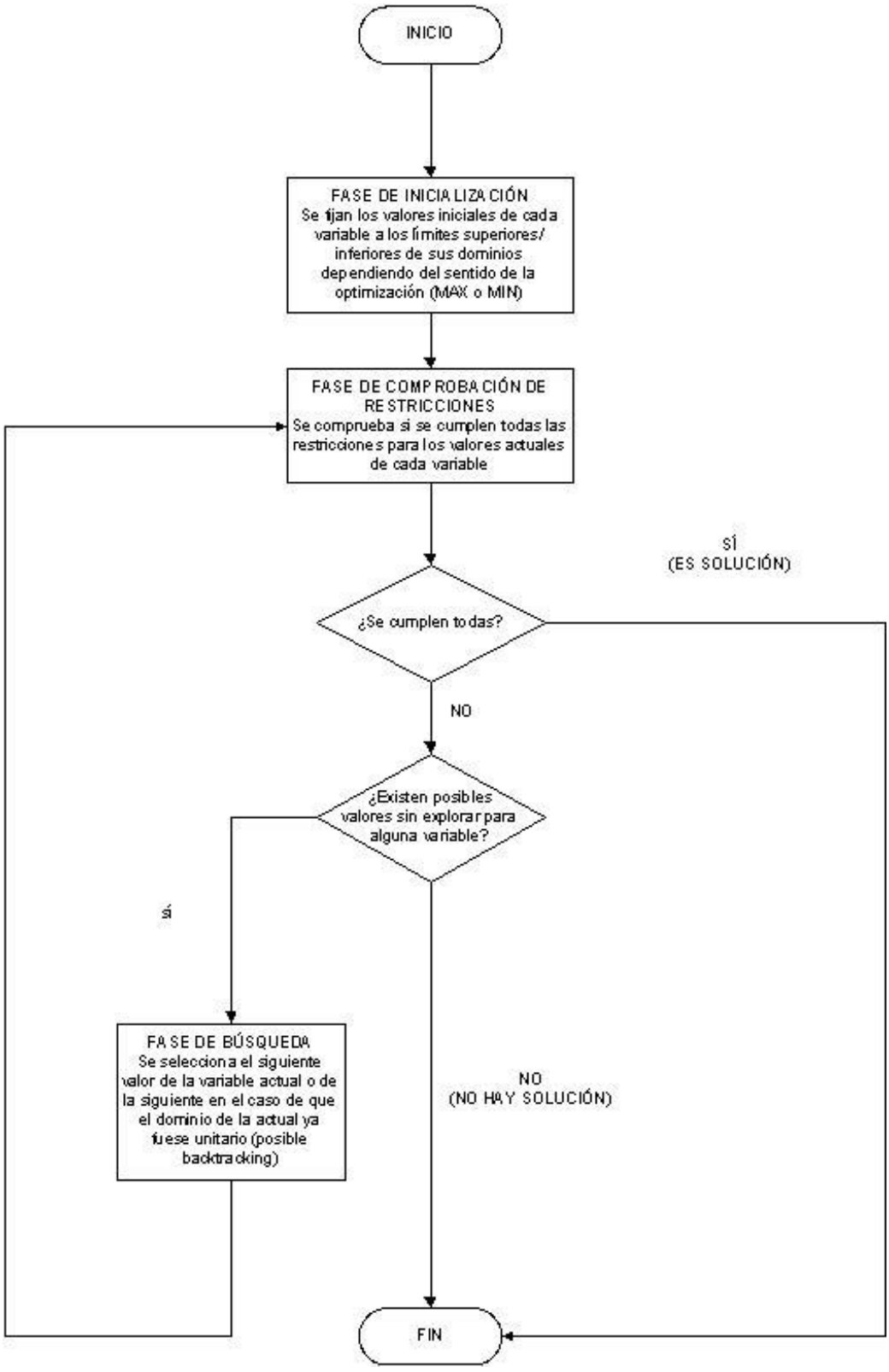
2. Estrategia de poda: Supongamos un problema de maximización de la función objetivo en un punto intermedio del recorrido del árbol de soluciones en el que, además, nos podemos encontrar algún nodo a partir del cual no se pueda llegar a ninguna solución válida. Se han recorrido varios nodos 1..n, estimando para cada uno la cota superior y la cota inferior para todo nodo j entre 1 y n. Se podrá podar un nodo i, si: CS (i) <= Beneficio (j), para algún j, solución final (factible).

3. Estrategia de ramificación: - De forma implícita, se tendrá en cuenta a la hora de llevar a cabo la ramificación del árbol de soluciones, la lista de nodos vivos en cada momento. - La lista de nodos vivos contiene la lista de nodos

que ya han sido generados pero que todavía no han sido explorados, para cada variable del problema a resolver. Son los nodos pendientes de tratar por el algoritmo.
Las posibles estrategias de ramificación a emplear son las siguientes:
- Estrategia FIFO (first in first out): la lista de nodos vivos es una cola y, por lo tanto, el recorrido del árbol se realiza en anchura.
- Estrategia LIFO (last in first out): la lista de nodos vivos es una pila y, por lo tanto, el recorrido del árbol se realiza en profundidad.
- Estrategia LC (least cost): se selecciona de toda la lista de nodos vivos aquel con el que se obtenga un mayor beneficio (o menor coste) para explorar a continuación. - Dadas las características del problema a resolver en nuestro caso, la estrategia de ramificación empleada ha sido la estrategia LC.

Algoritmo General

Básicamente, en un algoritmo de Ramificación y Poda básico, se realizan tres etapas: La primera de ellas, denominada de Selección, se encarga de extraer un nodo de entre el conjunto de los nodos vivos. La forma de escogerlo va a depender directamente de la estrategia de búsqueda que decidamos para el algoritmo. En la segunda etapa, la Ramificación, se construyen los posibles nodos hijos del nodo seleccionado en el paso anterior. Por último se realiza la tercera etapa, la Poda, en la que se eliminan algunos de los nodos creados en la etapa anterior. Esto contribuye a disminuir en lo posible el espacio de búsqueda y así atenuar la complejidad de estos algoritmos basados en la exploración de un árbol de posibilidades. Aquellos nodos no podados pasan a formar parte del conjunto de nodos vivos, y se comienza de nuevo por el proceso de selección. El algoritmo finaliza cuando encuentra la solución, o bien cuando se agota el conjunto de nodos vivos.


INICIO
FASE DE INICIALIZACIÓN
Se fijan los valores iniciales de cada variable a los límites superiores/ inferiores de sus dominios dependiendo del sentido de la optimización (MAX o MIN)
FASE DE COMPROBACIÓN DE RESTRICCIONES
Se comprueba si se cumplen todas las restricciones para los valores actuales de cada variable
¿Se cumplen todas?
SÍ
(ES SOLUCIÓN)
NO
¿Existen posibles valores sin explorar para alguna variable?
sí
FASE DE BÚSQUEDA
Se selecciona el siguiente valor de la variable actual o de la siguiente en el caso de que el dominio de la actual ya fuese unitario (posible backtracking)
NO
(NO HAY SOLUCIÓN)
FIN

Estrategias de Poda

Nuestro objetivo principal será eliminar aquellos nodos que no lleven a soluciones buenas. Podemos utilizar dos estrategias básicas. Supongamos un problema de maximización donde se han recorrido varios nodos i=1,…,n. estimando para cada uno la cota superior CS(xi) e inferior CI(x_i).

Estrategia 1

Si a partir de un nodo x_i se puede obtener una solución válida, entonces se podrá podar dicho nodo si la cota superior CS(x_i) es menor o igual que la cota inferior CI(x_j) para algún nodo *j* generado en el árbol.

Por ejemplo: Supongamos el problema de la mochila, el cual se va a desarrollar en la sección de ejemplos, donde utilizamos un árbol binario. Entonces:

Si a partir de x_i se puede encontrar un beneficio máximo de CS(x_i) = 4 y a partir de x_j, se tiene asegurado un beneficio mínimo de CI(x_j) = 5, esto nos llevará a la conclusión de que se puede podar el nodo x_i sin que perdamos ninguna posible solución óptima.

Estrategia 2

Si se obtiene una posible solución válida para el problema con un beneficio B_j, entonces se podrán podar aquellos nodos x_i cuya cota superior CS(x_i) sea menor o igual que el beneficio que se puede obtener B_j (este proceso sería similar para la cota inferior).

Estrategias de Ramificación

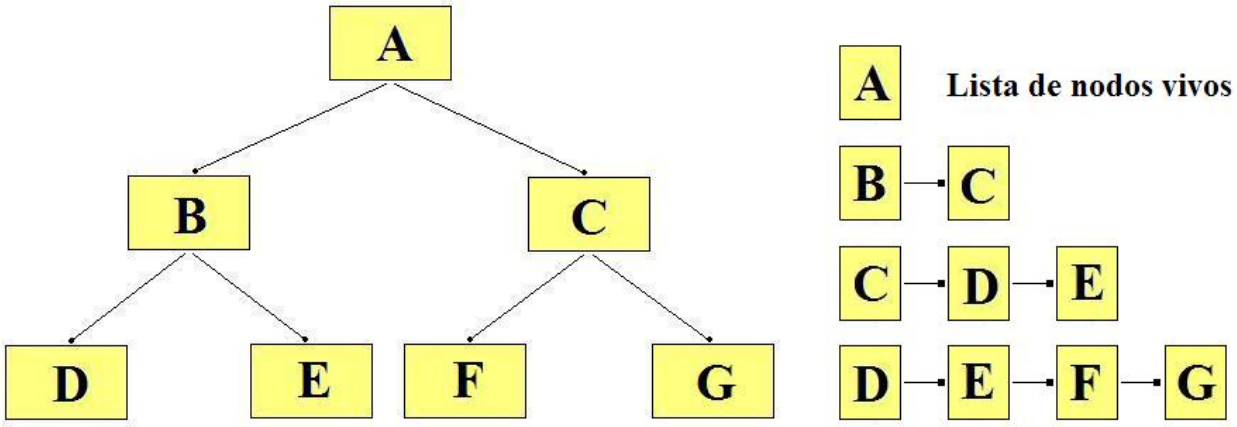
Como se comenta en la introducción de éste apartado, la expansión del árbol con las distintas estrategias está condicionada por la búsqueda de la solución óptima. Debido a esto todos los nodos de un nivel deben ser expandidos antes de alcanzar un nuevo nivel, cosa que es lógica ya que para poder elegir la rama del árbol que va a ser explorada, se deben conocer todas las ramas posibles.

Todos estos nodos que se van generando y que no han sido explorados se almacenan en lo que se denomina Lista de Nodos Vivos (a partir de ahora LNV), nodos pendientes de expandir por el algoritmo.

La LNV contiene todos los nodos que han sido generados pero que no han sido explorados todavía. Según como estén almacenados los nodos en la lista, el recorrido del árbol será de uno u otro tipo, dando lugar a las tres estrategias que se detallan a continuación.

Estrategia FIFO

En la estrategia FIFO (First In First Out), la LNV será una cola, dando lugar a un recorrido en anchura del árbol.

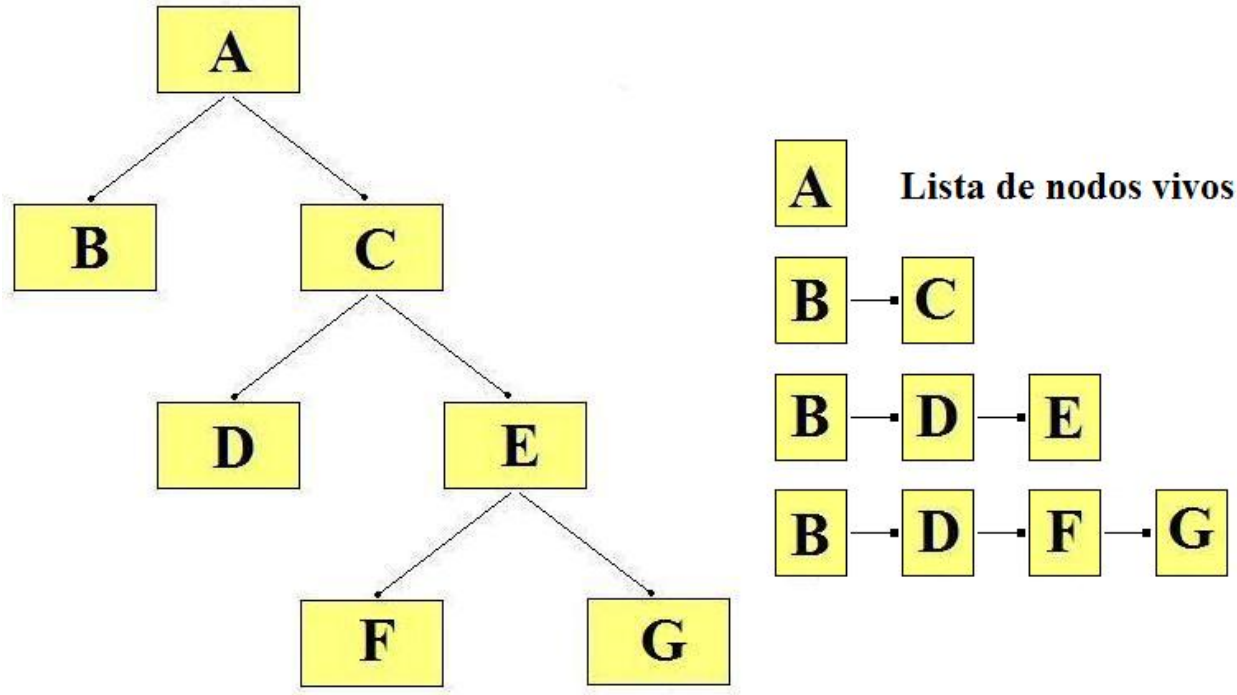


En la figura 1 se puede observar que se comienza introduciendo en la LNV el nodo A. Sacamos el nodo de la cola y se expande generando los nodos B y C que son introducidos en la LNV. Seguidamente se saca el primer nodo que es

el B y se vuelve a expandir generando los nodos D y E que se introducen en la LNV. Este proceso se repite mientras que quede algún elemento en la cola.

Estrategia LIFO

En la estrategia LIFO (Last In First Out), la LNV será una pila, produciendo un recorrido en profundidad del árbol.



En la figura 2 se muestra el orden de generación de los nodos con una estrategia LIFO. El proceso que se sigue en la LNV es similar al de la estrategia FIFO, pero en lugar de utilizar una cola, se utiliza una pila.

Estrategia de Menor Coste o LC

Al utilizar las estrategias FIFO y LIFO se realiza lo que se denomina una búsqueda “a ciegas”, ya que expanden sin tener en cuenta los beneficios que se pueden alcanzar desde cada nodo. Si la expansión se realizase en función de los beneficios que cada nodo reporta (con una “visión de futuro”), se podría conseguir en la mayoría de los casos una mejora sustancial.

Es así como nace la estrategia de Menor Coste o LC (Least cost), selecciona para expandir entre todos los nodos de la LNV aquel que tenga mayor beneficio (o menor coste). Por tanto, ya no estamos hablando de un avance “a ciegas”.

Esto nos puede llevar a la situación de que varios nodos puedan ser expandidos al mismo tiempo. De darse el caso, es necesario disponer de un mecanismo que solucione este conflicto:

-Estrategia LC-FIFO: Elige de la LNV el nodo que tenga mayor beneficio y en caso de empate se escoge el primero que se introdujo.

-Estrategia LC-LIFO: Elige de la LNV el nodo que tenga mayor beneficio y en caso de empate se escoge el último que se introdujo.

Fuentes y contribuyentes del artículo

Ramificación y poda *Fuente*: http://es.wikipedia.org/w/index.php?oldid=73296034 *Contribuyentes*: Alexman321, Balles2601, Bpk, CommonsDelinker, David0811, Er Komandante, FranDaniKikeDavid, Gothmog, HanPritcher, Laura Fiorucci, Manz, Pablete im, Tessier, Yrithinnd, 27 ediciones anónimas

Fuentes de imagen, Licencias y contribuyentes

Imagen:branch&bound.jpg *Fuente*: http://es.wikipedia.org/w/index.php?title=Archivo:Branch&bound.jpg *Licencia*: Public Domain *Contribuyentes*: Mocte13, 2 ediciones anónimas
Imagen:ArbolFIFO.JPG *Fuente*: http://es.wikipedia.org/w/index.php?title=Archivo:ArbolFIFO.JPG *Licencia*: Creative Commons Attribution 3.0 *Contribuyentes*: FranDaniKikeDavid
Imagen:EstratLIFO.JPG *Fuente*: http://es.wikipedia.org/w/index.php?title=Archivo:EstratLIFO.JPG *Licencia*: Creative Commons Attribution 3.0 *Contribuyentes*: FranDaniKikeDavid

Licencia