

Método y algoritmo de árbol de cubos y sus aplicaciones para resolver diferentes problemas de optimización discreta

Vladimir Khachaturov¹, José Crispín Zavala Díaz²

¹ Profesor-Investigador, Facultad de Ciencias-UAEM,
Cuernavaca, Morelos, México
vr_khach@yahoo.com

² Profesor-Investigador, Facultad de Contabilidad, Administración e
Informática-UAEM, Cuernavaca, Morelos, México
jc_zavala2002@yahoo.com

Resumen. Presentamos un método para solucionar problemas de optimización discreta, que llamamos "el árbol de cubos". El método se fundamenta en la teoría de conjuntos y en la teoría de latices. El método se puede utilizar para analizar problemas de optimización discreta de grandes dimensiones porque está estructurado de tal forma que fácilmente se puede paralelizar. Para ejemplificar su aplicación se muestra la solución del problema de la selección optimal del subconjunto de objetos. Los resultados de los experimentos en un procesador de cientos de tareas nos enseñan que el número de cálculos no fue más que m^4 . Nuestras valoraciones nos muestran que para 32 procesadores la dimensión de la tarea se podría incrementar hasta 4 veces mas.

1 Planteamiento del problema

Sea el conjunto finito $I = \{1, 2, \dots, m\}$ y sea el conjunto $B(I)$ el conjunto de todos los subconjuntos, esto es: $B(I) = \{\omega \mid \omega \subseteq I\}$ y $|B(I)| = 2^m$

Al conjunto $B(I)$ es posible imaginarlo como un cubo de dimensión m , donde los vértices del cubo son subconjuntos $\omega \subseteq I$, estos subconjuntos tienen un orden parcial entre ellos dado por la teoría de conjuntos con las operaciones: $\subseteq$, $\cup$ y $\cap$.

El cubo de dimensión m lo denotamos como $C(m)$, es claro que los elementos (vértices) del cubo son subconjuntos $\omega \subseteq I$, es posible escribirlo como el intervalo:

$$C(m) = [\emptyset, I] = \{\omega \subseteq I \mid \emptyset \subset \omega \subseteq I\}$$

Igualmente para el intervalo $[\omega_1, \omega_2]$, donde $\omega_1 \subset \omega_2 \subseteq I$, corresponde a un cubo $C(k)$

$$C(k) = [\omega_1, \omega_2] = \{\omega \subseteq I \mid \omega_1 \subset \omega \subset \omega_2\}, \text{ con la dimensión } k = |\omega_2 \setminus \omega_1|.$$

Todos los elementos (vértices) de $C(k)$ son vértices de $C(m)$. Si tenemos dos cubos $C(k_1) = [\omega_1^1, \omega_2^1]$, $C(k_2) = [\omega_1^2, \omega_2^2]$ y $\omega_1^2 \subset \omega_1^1$, $\omega_2^1 \subset \omega_2^2$ entonces se puede escribir $C(k_1) \subset C(k_2)$. Para $k \leq m$: $C(k) \subset C(m)$.

En este trabajo estudiaremos la tarea de partir el cubo $C(m)$ en los conjuntos Π de cubos de dimensión menor, donde para cada dos cubos la intersección es el conjunto vacío y la unión de todos los cubos es igual a $C(m)$.

Supongamos $\Pi = \{C_1, C_2, \dots, C_r\}$, donde C_p es un cubo donde $p = 1, 2, \dots, r$. Vamos a decir que Π es la partición si:

- $C_i \cap C_j = \emptyset$ para cada $i \neq j, 1 \leq i \leq r, 1 \leq j \leq r$
- $\bigcup_{p=1}^r C_p = C(m)$

2 Procedimiento para la construcción del árbol de cubos

Sea el cubo inicial $C(m) = [\emptyset, I]$, $I = \{1, 2, \dots, m\}$, vamos a construir un medio especial con el conjunto parcialmente ordenado de cubos $C(l)$, $l=m-k$ donde $k=0, 1, \dots, m$, de diferentes dimensiones. Al nivel superior $l=m$ ($k=0$) le corresponde el cubo único $C(m)$ de dimensión m . En el nivel siguiente $l=m-1$ ($k=1$) se distribuyen dos cubos disjuntos: $C_1(m-1)$ y $C_2(m-1)$ de dimensión $(m-1)$ cada uno, estos cubos se pueden hacer de m distintas formas, nosotros hacemos lo siguiente: para el cubo $C_1(m-1)$ consideramos un intervalo $[1, I]$ y al cubo $C_2(m-1)$ le corresponde un intervalo $[\emptyset, I \setminus \{1\}]$. Estos dos cubos son subconjuntos disjuntos, $C_1(m-1) \cap C_2(m-1) = \emptyset$, porque todos los elementos del cubo $C_1(m-1)$ tienen al subconjunto $\{1\}$ y todos los elementos del cubo $C_2(m-1)$ no tienen al mismo elemento $\{1\}$.

El número de elementos de cada uno de los cubos C_1 y C_2 es igual a $2^{(m-1)}$, entonces el número de elementos de ambos cubos es 2^m , por eso: $C_1(m-1) \cup C_2(m-1) = C(m)$. Por tanto, el conjunto de los dos cubos $C_1(m-1)$ y $C_2(m-1)$ es la partición del cubo $C(m)$. Notamos que cada cubo del nivel $l=m-k$ tiene la dimensión $l=m-k$, entonces tiene 2^l vértices.

Posteriormente se forman los niveles siguientes $(m-2)$, en cada cubo $C_1(m-1)$ y $C_2(m-1)$ se hacen las mismas operaciones. Cada cubo se parte en dos cubos de una dimensión menor, entonces en el nivel $(m-2)$ recibimos a cuatro que son una partición del cubo $C(m)$.

Si seguimos este proceso llegamos al nivel $m-3$ con ocho cubos, dos cubos del nivel $m-3$ para cada uno de los cubos del nivel $m-2$. En general para el nivel $(m-k)$, $0 \leq k \leq m$, se pueden formar 2^k cubos, donde cada dos cubos son conjuntos disjuntos y la unión de todos ellos forman el cubo inicial $C(m)$. Entonces todos los cubos del nivel $(m-k)$ es la partición del cubo $C(m)$.

En el último nivel ($l=0$) el número de cubos de dimensión cero es igual a 2^m . Entonces es igual al número de vértices del cubo inicial $C(m)$. El conjunto de ellos es la partición, porque es claro que cada intersección es igual a cero y la unión de ellos es el conjunto inicial $C(m)$.

Construimos un árbol donde sus vértices son los cubos definidos con el procedimiento descrito. En cada etapa donde recibimos cubos, en el nivel siguiente se construye un lado entre el cubo del cual nacieron con los dos cubos de una dimensión menor en uno del cubo del nivel superior. Como resultado recibimos un árbol con

vértices, donde cada vértice es un cubo y ellos son de un orden parcial definido por el lado que los conectan. Supongamos este árbol como $A(m)$, vamos a estudiar algunas propiedades del árbol $A(m)$.

3 Algunas propiedades del árbol de cubos

3.1 Estructura jerárquica

De la formalización del algoritmo de la construcción del árbol de cubos concluimos que el árbol tiene una estructura jerárquica. El árbol tiene $m+1$ niveles $l = m - k$ ($k = 0, 1, \dots, m-1, m$), en el nivel superior ($l=m$) está un sólo vértice, en los niveles siguientes el número de vértices (número de cubos) es igual a 2^k ($k = 0, 1, \dots, m-1, m$).

3.2 Número de vértices del árbol $A(m)$

El número de todos los vértices del árbol $A(m)$ es igual a: $1 + 2 + 4 + \dots + 2^m$. Entonces $|A(m)| = 2^{m+1} - 1$.

3.3 Generación de diferentes variantes de particiones del árbol $A(m)$

El conjunto de vértices de cada nivel del árbol $A(m)$ es la partición del cubo inicial $C(m)$. Entonces en cada árbol se tiene al mismo tiempo m diferentes variantes de las particiones del cubo inicial $C(m)$. Además se puede formar muchas otras variantes de la partición. Por ejemplo se puede tomar como base una variante de la dimensión que le corresponde a cualquier nivel ($l > 0$) del árbol y sustituir, dentro de esta variante base, uno de los elementos (o varios elementos) con dos cubos de dimensión menor a uno, que nosotros recibimos con el procedimiento de construcción del árbol. Esta propiedad del árbol de cubos permite organizar eficazmente los cálculos en las máquinas paralelas.

3.3 Dependencia del número de vértices del árbol conforme a su nivel

La dependencia del número de vértices del árbol $A(m)$ conforme al nivel l ($N(l)$, $l=m-k$, donde $k = 0, 1, 2, \dots, m$) está dada por:

$$N(k) = 2|x(l(k))| = 2^{m-l(k)} = 2^k$$

Donde

$l(k)=m-k$, ($k = m, m-1, \dots, 1, 0$) es el número del nivel
 $|x(l(k))| = 2^{m-l(k)} = 2^k$

Si reflejamos esta función (dependencia) $N(l)$ en los sistemas coordenados (x, l) , generamos una gráfica, como se muestra en la figura 1 para $m=5$. Esta figura recuerda el contorno de un pino que tiene un tronco que corresponde a una franja $-\frac{1}{2} \leq x \leq \frac{1}{2}$.

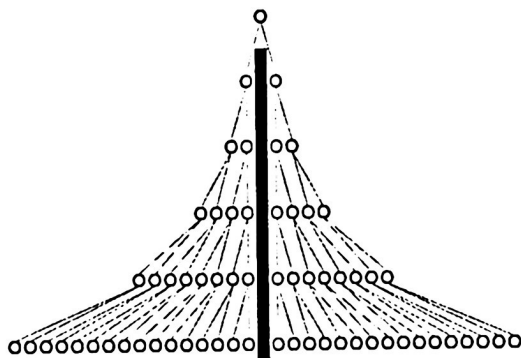


Figura 1. El árbol de cubos para $m=5$. En el centro el tronco va de $-\frac{1}{2} \leq x \leq \frac{1}{2}$.

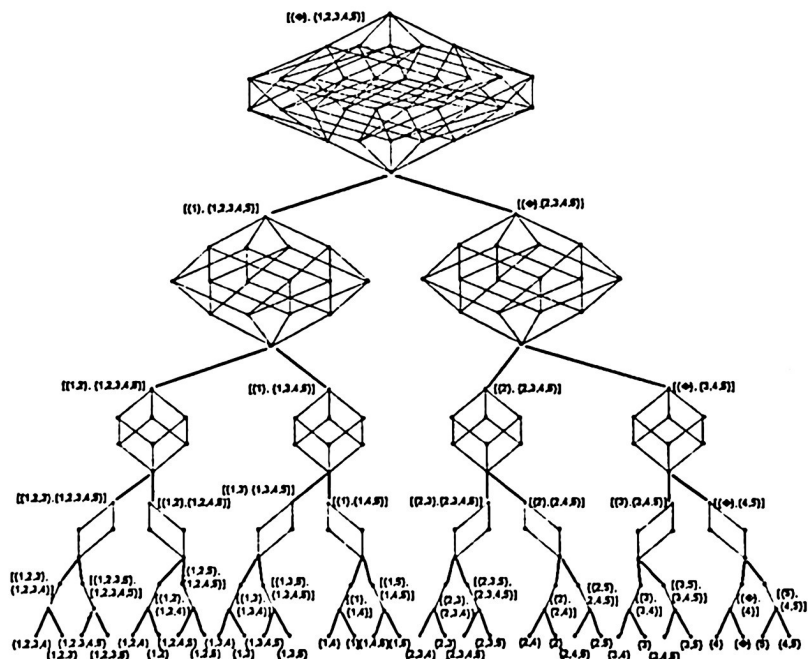


Figura 2. El árbol de cubos.

El contorno de este pino se compone de dos aproximaciones $x(l(k))=2^{k-1}$ para $x(l(k)) \geq 0$ y $x(l(k))=-2^{k-1}$ para $x(l(k)) \leq 0$. Por eso $N(x(l(k)))=2^{k-1}-(-2^{k-1})=2^k$. Notamos que si aumentamos la magnitud de m , el "pino" se desliza a lo largo del eje "l". El incremento de la capacidad de vértices del cubo sucede en los últimos niveles del "pino" nuevo.

En la figura 2 se muestra como disminuye en una dimensión los cubos de niveles consecutivos. En la parte superior el grado del cubo es de cinco, el cual se encuentra conectado a dos cubos de grado cuatro, cada uno de éstos a su vez están conectados con dos cubos de grado tres, así sucesivamente hasta el último nivel, donde los cubos son de grado cero.

3.4 Número de vértices como una función de los intervalos de los niveles del árbol

No es difícil determinar, para dos pinos con $m=m_1$ y $m=m_2$ el número de vértices del árbol de cubos en los intervalos $[m_1-k, m_1]$ y $[m_2-k, m_2]$ es el mismo, para cualquier $0 \leq k \leq \min(m_1, m_2)$ y es igual a $2^{k+1}-1$, entonces no depende del valor de m

3.5 Subárboles del árbol $A(m)$ y sus propiedades

Cada vértice del árbol forma un subárbol del árbol $A(m)$. El vértice del árbol $A(m)$ - cubo $C(m)$ - se denomina también la raíz del árbol $A(m)$, porque de él nacen todos los vértices del árbol.

Un subárbol del árbol $A(m)$ es un árbol que nació a través de cualquier vértice del árbol $A(m)$. El número de todos los subárboles es igual al número de todos los vértices del árbol, esto es igual a $2^{m+1}-1$, cada subárbol que nace de la raíz $C(m-k)$ tiene $2^{m-k+1}-1$ vértices ($k=0, 1, \dots, m$).

Entonces para $k=0$ tenemos la raíz del árbol principal $A(m)$ que tiene $2^{m+1}-1$ vértices (ver punto 3.2). Para $k=m$ cada uno de los 2^m subárboles tienen un solo vértice.

3.5.1 Sea $C^1(l_1)$ y $C^2(l_2)$, $l_2 \leq l_1$, dos raíces que corresponden a dos subárboles $A^1(l_1)$ y $A^2(l_2)$.

- Si $C^1(l_1) = C^2(l_2)$, entonces $A^1(l_1) = A^2(l_2)$.
- Si $C^1(l_1) \cap C^2(l_2) = \emptyset$, entonces $A^1(l_1) \cap A^2(l_2) = \emptyset$, que significa que los subárboles $A^1(l_1)$ y $A^2(l_2)$ no tienen vértices y lados comunes.
- Si $C^2(l_2) \subset C^1(l_1)$, entonces todos los vértices y lados del subárbol $A^2(l_2)$ también son elementos del subárbol $A^1(l_1)$.

3.5.2 Si excluimos del árbol $A(m)$ el subárbol $A(l_0)$, $l_0 < m$, excluirémos a todos los subárboles $A(l)$ con $l < l_0$, para los cuales $C(l) \subset C(l_0)$.

3.5.3 Si alguna propiedad no cumple para ningún elemento $\omega \in C(l_0)$, entonces esta propiedad no cumple para ningún elemento $\omega \in C(l)$ para todo $C(l) \subset C(l_0)$, porque $C(l_0)$ tiene todos los elementos para cualquier $C(l) \subset C(l_0)$.

3.5.4 Si alguna propiedad cumple para algún elemento $\omega_b \in C(l_0)$, entonces existe una cadena de cubos: $C^0(0) \subset C^0(1) \subset C^0(2) \dots C^0(l_0-1) \subset C^0(l_0)$, en la cual para cada $C^0(l)$, $0 \leq l \leq l_0$, existe un elemento donde esta propiedad también se cumple. Por ejemplo, la cadena de cubos para los cuales $C^0(0) = \omega_b$

3.6 El árbol de cubos como latices

Introducimos en el árbol de cubos un vértice, el conjunto vacío, y lo conectamos por medio de lados con todos los cubos del nivel cero ($l(k)=0$, $k=m$). Por el teorema siguiente determinamos que la estructura que recibimos es un lattice [1,2].

Teorema Después de introducir en el árbol de cubos el vértice que corresponde al conjunto vacío, la estructura que recibimos es una lattice.

Demostración. En correspondencia con la formación del árbol de cubos m , cada dos cubos $C_1 = [\omega_1^1, \omega_2^1]$, $C_2 = [\omega_1^2, \omega_2^2]$ y $C_1 \neq C_2$ pueden tener una de las dos propiedades:

1. $C_1 \subset C_2$, $\omega_1^2 \subset \omega_1^1$ y $\omega_2^2 \supset \omega_2^1$
2. $C_1 \cap C_2 = \emptyset$

Por el primer punto se tiene: $\supremo(C_1, C_2) = C_2$, $\infimo(C_1, C_2) = C_1$.

Por el segundo punto se tiene: el $\supremo(C_1, C_2)$ es un nuevo cubo $\tilde{C} = [\omega_1^1 \cap \omega_1^2, \omega_2^1 \cup \omega_2^2]$ si el cubo $\tilde{C} \in A(m)$, si $\tilde{C} \notin A(m)$ $\supremo(C_1, C_2) = C(m) = [\emptyset, 1]$ e $\infimo(C_1, C_2) = \emptyset$. Entonces nosotros determinamos que después de introducir el conjunto vacío en el árbol de cubos m , nosotros recibimos la estructura en la cual cualquier par de elementos siempre tienen supremo e infimo, por eso la estructura que recibimos es una lattice [1,2].

4 Método del árbol de cubos para los problemas de optimización

Con el objetivo de explicar cómo el método de árbol de cubos se aplica para determinar la solución de problemas de optimización discreta, utilizamos el planteamiento del problema de la selección optimal del subconjunto de objetos con un parámetro, que es una generalización de la tarea clásica conocida como "la tarea del maletín".

Sujeto a:
$$\sum_{j=1}^m y_j \leq b, \quad y_k \in \{0, b_{i_k}\}$$

De esta última transformación se obtienen las soluciones integral $C(I)$ y lineal $C(L)$. Dado que b_{i_k} es un entero, se tendrá un límite q tal que:

$$\sum_{k=1}^q b_{i_k} \leq b < \sum_{k=1}^{q+1} b_{i_k}$$

Porque $\sum_{k=1}^m b_{i_k} = B > \lambda_0 B = b$ entonces $q < m$ siempre. Con lo cual se obtienen los elementos que son la solución.

$$y_k = b_{i_k}, \text{ cuando } x_{i_k} = 1 \text{ para } k = 1, 2, \dots, q$$

$$y_k = 0, \text{ cuando } x_{i_k} = 0 \text{ para } k = q+1, \dots, m$$

La magnitud de la función objetivo para la solución integral estará dada por los términos donde $x_{i_k} = 1$ ó $x_{i_k} = 0$

$$C(I) = \sum_{k=1}^q c_{i_k}$$

Para la solución lineal (no integral) tenemos:

$$C(L) = C(I) + \frac{c_{i(q+1)}}{b_{i(q+1)}} \left(b - \sum_{k=1}^q b_{i_k} \right)$$

Claro que $C(L) \geq C(I)$. Puede ser $C(L) = C(I)$ sólo cuando $b = \sum_{k=1}^q b_{i_k}$. A cada solución integral, $x_{i_k} = 1$, le corresponde el subconjunto $\omega^p \subset I$, donde $i_k \in \omega^p$ para $x_{i_k} = 1$ y $i_k \notin \omega^p$ para $x_{i_k} = 0$. A cada ω^p le corresponde su $C(\omega^p)$, donde $C(I, \omega^0) = \sum_{i \in \omega^0} c_i$ y $C(L, \omega^p) \geq C(I, \omega^p)$.

Este algoritmo se puede utilizar para cualquier subárbol $A(I)$ con su raíz $C(I)$ de la manera siguiente.

Cada una de las raíces se puede representar de la forma:

$$C(I) = (\omega_1, \omega_2) = \{\omega_1 \mid \omega_1 \subset \omega \subset \omega_2\}, \text{ donde } [\omega_2 \setminus \omega_1] = I, \omega_1 \subset I \text{ y } \omega_2 \subset I.$$

Planteamiento del problema, para el cubo $C(I)$ es necesario buscar:

$$\sum_{i \in \omega_1} c_i + \max_{i \in \omega_2 \setminus \omega_1} \sum c_i x_i$$

Sujeto a:
$$\sum_{i \in \omega_2 \setminus \omega_1} b_i x_i \leq b - \sum_{i \in \omega_1} b_i x_i, \quad x_i \in \{0, 1\}, \quad i \in \omega_2 \setminus \omega_1$$

Sujeto a:
$$\sum_{j=1}^m y_j \leq b, \quad y_k \in \{0, b_{i_k}\}$$

De esta última transformación se obtienen las soluciones integral $C(I)$ y lineal $C(L)$. Dado que b_{i_k} es un entero, se tendrá un límite q tal que:

$$\sum_{k=1}^q b_{i_k} \leq b < \sum_{k=1}^{q+1} b_{i_k}$$

Porque $\sum_{k=1}^m b_{i_k} = B > \lambda_0 B = b$ entonces $q < m$ siempre. Con lo cual se obtienen los elementos que son la solución.

$$y_k = b_{i_k}, \text{ cuando } x_{i_k} = 1 \text{ para } k = 1, 2, \dots, q$$

$$y_k = 0, \text{ cuando } x_{i_k} = 0 \text{ para } k = q+1, \dots, m$$

La magnitud de la función objetivo para la solución integral estará dada por los términos donde $x_{i_k} = 1$ ó $x_{i_k} = 0$

$$C(I) = \sum_{k=1}^q c_{i_k}$$

Para la solución lineal (no integral) tenemos:

$$C(L) = C(I) + \frac{c_{i(q+1)}}{b_{i(q+1)}} \left(b - \sum_{k=1}^q b_{i_k} \right)$$

Claro que $C(L) \geq C(I)$. Puede ser $C(L) = C(I)$ sólo cuando $b = \sum_{k=1}^q b_{i_k}$. A cada solución integral, $x_{i_k} = 1$, le corresponde el subconjunto $\omega^p \subset I$, donde $i_k \in \omega^p$ para $x_{i_k} = 1$ y $i_k \notin \omega^p$ para $x_{i_k} = 0$. A cada ω^p le corresponde su $C(\omega^p)$, donde $C(I, \omega^0) = \sum_{i \in \omega^0} c_i$ y $C(L, \omega^p) \geq C(I, \omega^p)$.

Este algoritmo se puede utilizar para cualquier subárbol $A(I)$ con su raíz $C(I)$ de la manera siguiente.

Cada una de las raíces se puede representar de la forma:

$$C(I) = (\omega_1, \omega_2) = \{\omega_1 \mid \omega_1 \subset \omega \subset \omega_2\}, \text{ donde } [\omega_2 \setminus \omega_1] = I, \omega_1 \subset I \text{ y } \omega_2 \subset I.$$

Planteamiento del problema, para el cubo $C(I)$ es necesario buscar:

$$\sum_{i \in \omega_1} c_i + \max_{i \in \omega_2 \setminus \omega_1} \sum c_i x_i$$

Sujeto a:
$$\sum_{i \in \omega_2 \setminus \omega_1} b_i x_i \leq b - \sum_{i \in \omega_1} b_i x_i, \quad x_i \in \{0, 1\}, \quad i \in \omega_2 \setminus \omega_1$$

donde: $x_i = 1$ para $i \in \omega_1$, y $x_i = 0$ para $i \in \omega_2$

Buscamos $\max_{i \in \omega_1 \setminus \omega_2} \sum c_i x_i$ igual que para el cubo $C(m)$. Como resultado

recibimos, para el cubo $C(l) = C(\omega_1, \omega_2)$, las soluciones óptimas lineal $C(L, \omega^0, l)$ e integral aproximado con magnitud $C(l, \omega^0, l)$. Claro, siempre $C(L, \omega^0, l) \geq C(l, \omega^0, l)$ para $\omega_1 \subset \omega^0 \subset \omega_2$.

4.2.2 Reglas para el rechazo de subárboles no óptimos

En correspondencia con la propiedad 3.6.3, si en el cubo $C(l_0)$ no existen soluciones óptimas, entonces no hay soluciones óptimas en ningún cubo $C(l) \subset C(l_0)$. Por tanto si de alguna manera afirmamos que el cubo $C(l_0)$ no tiene soluciones óptimas, entonces no hace falta revisar todo el subárbol $A(l_0)$, en consecuencia no hace falta revisar los $2^{l_0+1} - 1$ cubos, vértices de este subárbol.

Para el problema que investigamos se pueden formular las siguientes reglas de rechazo de subárboles no óptimos.

Supongamos conocidas algunas soluciones aproximadas de nuestra tarea, las cuales están en el subconjunto $\tilde{\omega} \subset I$ con sus correspondientes magnitudes de las funciones de la solución integral $C(l, \tilde{\omega})$ y de la solución no integral (lineal) $C(L, \tilde{\omega})$, donde $C(l, \tilde{\omega}) \leq C(L, \tilde{\omega})$

Sea el subárbol $A(l)$ con su raíz $C(l) = C(\omega_1, \omega_2)$, para $C(l)$ conocemos las magnitudes de sus soluciones $C(L, \omega^0, l)$ y $C(l, \omega^0, l)$, donde $C(L, \omega^0, l) \geq C(l, \omega^0, l)$.

Regla de rechazo Si $C(l, \tilde{\omega}) \geq C(L, \omega^0, l)$, entonces se puede excluir la revisión del subárbol $A(l)$, porque entre todos los cubos que son vértices del subárbol $A(l)$ no existe ninguna solución "mejor" que $\tilde{\omega}$, es decir, no existe una solución con una significación del funcional entero mayor que $C(l, \tilde{\omega})$. La demostración de esta regla sigue la desigualdad siguiente:

$$C(l, \tilde{\omega}) \geq C(L, \omega^0, l) \geq C(l, \omega^0, l)$$

y de la propiedad de los subárboles 3.6.3.

Esta regla se utiliza permanentemente en el proceso de búsqueda de la solución óptima de la tarea inicial.

4.2.3 Algoritmo de búsqueda de la solución óptima

El algoritmo de búsqueda de las soluciones óptimas se constituye de cuatro bloques principales (etapas).

Primer Bloque. Diferentes búsquedas de las soluciones integrales aproximadas. La realización de este bloque es una etapa muy importante en la búsqueda de la solución óptima, porque la presencia de una buena solución integral permite aumentar la eficacia del algoritmo de búsqueda de la solución óptima, ya que aumenta la eficacia de la regla de rechazo.

Estos algoritmos de búsqueda deben ser bastante rápidos, por ejemplo, óptimo local, algoritmos avaros (greedy), algoritmos de búsqueda causal y otros más. Uno de estos algoritmos está descrito en el punto 4.2.1 y en el punto 4.2.4. se muestra otro método de búsqueda para las soluciones aproximadas.

Como resultados de la realización de este bloque determinamos y guardamos en la memoria la solución integral $\tilde{\omega}$ y su correspondiente significación $C(\tilde{\omega})$, la que determinamos como la solución óptima temporal.

Segundo Bloque. Esquema de selección de los vértices del árbol $A(m)$. Existen diferentes algoritmos para seleccionar los vértices de un árbol, por ejemplo las que se presentan en las referencias [4, 8, 11]. Nosotros utilizamos los algoritmos, que se han aplicado durante muchos años, para solucionar diferentes tareas de optimización combinatoria [5,6,7,9,10]. El esquema común está escrito en el punto 2 del presente trabajo.

Tercer Bloque. Realización de la regla de rechazo. Para cada vértice $C(l)=C(\omega_l, \omega_l)$ determinamos las significaciones $C(L, \omega^l, l)$, $C(l, \omega^l, l)$ y la solución integral aproximada ω^l , $\omega_l \subset \omega^l \subset \omega_l$. Después comparamos $C(L, \omega^l, l)$ con la significación del óptimo temporal $C(l, \tilde{\omega})$.

Si $C(l, \tilde{\omega}) \geq C(L, \omega^l, l)$ excluimos del cálculo al subárbol $A(l)$, entonces no continuamos formando dos nuevos cubos del vértice $C(l)$, y pasamos a formar nuevos cubos de otro vértice en correspondencia al esquema común de formalización del árbol $A(m)$ (ver punto dos).

Si no existen otros cubos terminamos la búsqueda de la solución óptima y pasamos al bloque 4.

Cuarto bloque. Retención de las soluciones óptimas temporales y la salida del cálculo. Cada vez, después de la determinación de la solución integral, para cada vértice $C(l)$ del árbol $A(m)$ la significación de $C(l, \omega^l, l)$ se compara con la significación $C(l, \tilde{\omega})$. Si $C(l, \omega^l, l) > C(l, \tilde{\omega})$, entonces ω_l con su $C(l, \omega^l, l)$ la retenemos como la solución siguiente óptima temporal en lugar de $\tilde{\omega}$.

Después de concluir el cálculo, la última solución óptima temporal $\tilde{\omega}_u$, $C(l, \tilde{\omega}_u)$ es la solución de la tarea inicial $\omega_{opt} = \tilde{\omega}_u$, $C(l, \omega_{opt}) = C(l, \tilde{\omega}_u)$ son las salidas del algoritmo.

4.2.4 Los experimentos computacionales, versión secuencial

Los experimentos computacionales consistieron en obtener la solución integral, lineal y los subconjuntos de la solución óptima para cada λ_0 seleccionada. Se consideraron dos tipos de problemas, el primero de ellos fue determinar los coeficientes b_i y c_i por medio de una progresión aritmética, de tal forma que nosotros conocemos la suma de ellos para cualquier número de elementos, es decir la solución óptima. Una vez que se determinaron los coeficientes, éstos se distribuyeron en diferentes posiciones para dirigir la búsqueda a cubos diferentes. En todas las pruebas, el AAC determinó la solución óptima correcta, con lo cual comprobamos que el algoritmo del árbol de cubos se implementó correctamente.

El segundo experimento consistió en determinar los coeficientes (b_i, c_i) en forma aleatoria, los resultados que se presentan son para este segundo experimento. El primer experimento que se presenta para $m=500$, consiste en determinar la solución entera óptima para 20 intervalos de λ . Para este problema y el siguiente se utilizó la regla de rechazo general $C(L, \alpha^p, l) - C(I, \tilde{\omega}) \leq \varepsilon C(I, \tilde{\omega})$. Cuando $\varepsilon=0$ la regla de rechazo se utiliza para determinar la solución exacta, y cuando $\varepsilon>0$ busca la solución aproximada con un valor no mayor a $\varepsilon C(I, \tilde{\omega})$. En la figura 3 se muestra la gráfica de la solución exacta, donde en las abscisas está λ y en el eje de las ordenadas está el número de iteraciones con el cual convergió el algoritmo.

Como es de suponerse, los cálculos experimentales confirman nuestros supuestos de que los problemas más difíciles requieren de muchos cálculos, esto es cuando $\lambda=1/2$.

El número de soluciones (N) que calculamos nunca fue mayor a m' , $N < m'$. En forma indirecta nos muestra que el algoritmo que utilizamos es efectivo. Tomando como base estos valores experimentales ($N < m'$, para $\lambda=1/2$), nosotros recibimos en forma analítica los valores para otras significaciones de λ siguientes:

$$N \approx m^{x_k}, \text{ donde } x_k = 4(1 - \log_m 2^k), \quad k = 0, 1, 2, \dots$$

$$\text{para } 0 < \lambda_k \leq \frac{1}{2}, \text{ donde } \lambda_k = 2^{-(k+1)} \text{ y para } \frac{1}{2} \leq \lambda_k < 1, \text{ donde } \lambda_k = 1 - 2^{-(k+1)}.$$

Es claro que para $\lambda_k = 0$ la solución óptima es $x_i = 0$ para todos $i = 1, 2, \dots, m$, y para $\lambda_k = 1$ la solución óptima es $x_i = 1$ para todos $i = 1, 2, \dots, m$.

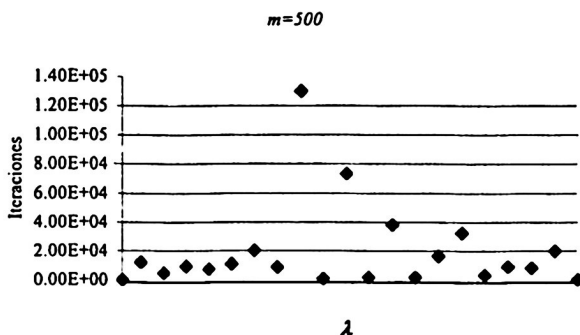


Figura 3. Iteraciones en función de λ para $m = 500$.

La magnitud de la solución óptima entera se muestra en la figura 4, como se observa la magnitud de la solución entera se incrementa monótonamente hasta alcanzar el máximo, cuando $\lambda=1$.

El segundo experimento que se presenta es para $m=3,200$ con diferentes valores de ε . Se determinan las soluciones aproximadas para $\varepsilon>0$ (6%, 4%, 2% y 1%) y la solución exacta $\varepsilon=0$. El cálculo se realizó para $\lambda = 0.0035$ a causa de que es posible obtener la solución exacta en un procesador, ya que como se mostró en el primer ejemplo, para una λ mayor y cercana a $1/2$ se requiere de un número de iteraciones

considerablemente alto. Los resultados de este segundo ejemplo se muestran en la tabla 1.

En la segunda columna de la tabla 1 están los números de iteraciones en que se calculo el valor mayor y en la tercera columna está el número de iteraciones en que se determino que este valor máximo era la solución optima. En la cuarta columna está el valor de la solución entera y en la quinta columna está el intervalo en el cual la solución optima se encuentra.

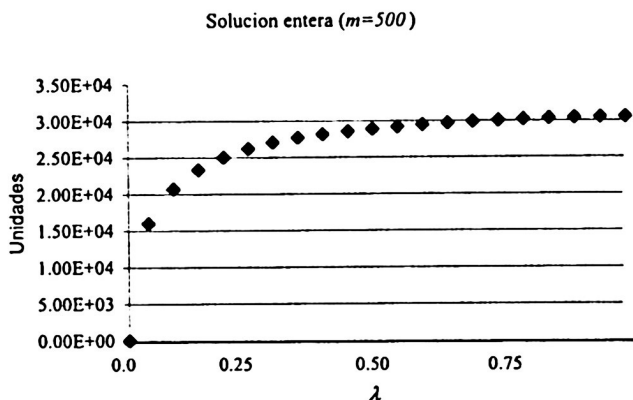


Figura 4. Magnitud de la solución entera exacta para $m=500$

El intervalo de la solución estará dado por el porcentaje considerado para obtener la solución aproximada ($\varepsilon C(I, \tilde{\omega})$). Por ejemplo para $\varepsilon=6\%$ el límite superior del intervalo es $2,202 + 0.06(2,202) = 2,334$, el cual está en la quinta columna de la tabla 1. Para el porcentaje siguiente el límite superior del intervalo es $2,401$, pero dado que la solución no será mayor al límite superior anterior, entonces consideramos éste como el límite superior del intervalo. Los resultados para los demás porcentajes se muestran en la quinta columna.

Tabla 1. Solución aproximada y exacta para $m=3,200$ para una $\lambda = 0.0035\%$

$\varepsilon(\%)$	Iteración máximo	Iteración optimo	$C(I)_{aprox}$	$C(I)$
6	4	5	2,202	$2,202 \leq C(I)_{opt} \leq 2,334$
4	62	1339	2,309	$2,309 \leq C(I)_{opt} \leq 2,334$
2	338	1341	2,309	$2,309 \leq C(I)_{opt} \leq 2,334$
1	1338	1343	2,309	$2,309 \leq C(I)_{opt} \leq 2,332$
0	1378	2655	2,329	$C(I)_{opt} = C(I)_{aprox} =$
(exacta)				2,329

En esa tabla 1 se observa que en todos los casos la solución exacta está dentro de los intervalos calculados de las soluciones aproximadas. El costo de obtener la

solución exacta se ve reflejada en el número de iteraciones requeridas para determinar el valor máximo y para determinar que este máximo es la solución óptima.

A causa de la diferencia tan grande de iteraciones, entre la solución exacta y la aproximada, para este problema ($m=3,200$) es posible calcular las soluciones enteras aproximadas para cualquier λ siempre y cuando $\varepsilon \geq 1\%$. Para calcular la solución óptima exacta de cualquier λ se necesita de cómputo paralelo por la gran cantidad de cálculos y memoria requeridos.

A causa de que el método del árbol de cubos es eficaz es posible aplicarlo en determinar la solución óptima de problemas de optimización mas complejos como la selección del subconjunto de objetos con un parámetro fijo, multirestricciones y multicriterial.

4.2.5 Algoritmo de búsqueda en el intervalo de estabilidad de la solución óptima con un parámetro fijo

En los párrafos siguientes enseñamos varias modificaciones del método de árbol de cubos que pueden ayudar a resolver problemas de optimización mas complejos, multirestricciones y multicriterial. Estas modificaciones se fundamentan en el método "aproximative-combinatorial" que se detalla en los artículos [9, 10].

Este algoritmo puede ser modificado para determinar, no sólo la solución óptima, también puede determinar la región de estabilidad de la solución óptima. A este subconjunto de soluciones la denotamos por $\Omega(R)$, donde:

$$\Omega(R) = \{\omega \in I \mid C(I, \omega_{opt}) - R \leq C(\omega) \leq C(I, \omega_{opt})\} \text{ donde } R \geq 0.$$

Entonces $\Omega(R)$ es el conjunto de todas las soluciones integrales no diferentes para la significación de la función a la solución óptima en mas que $R \geq 0$.

Para buscar $\Omega(R)$ es necesario modificar las Reglas para el rechazo de subárboles no óptimales del Método de árbol de Cubos. La modificación que se tiene que incluir en los bloques 3 y 4 es la siguiente:

Para todos los subárboles $A(l)$ para los cuales:

$$C(I, \tilde{\omega}) > C(L, \omega^0, l) + R$$

los excluimos del cálculo (bloque 3).

En el proceso del algoritmo, se almacena en la memoria todas las soluciones para las cuales (bloque 4):

$$C(I, \omega^0, l) \geq C(I, \omega_{opt}) - R$$

Después de realizar los cálculos se tendrá almacenada en la memoria el conjunto $\Omega(R)$.

Este conjunto se puede utilizar para buscar el óptimo de la tarea cuando se tiene mas de una restricción. Por ejemplo, se tienen k restricciones lineales y conocemos la solución $\omega_k \subset I$ con el funcional $C(I, \omega_k)$ que satisface todas las restricciones, pero no puede ser el óptimo.

Es necesario buscar el óptimo $\omega_{opt} \in I$ con su $C(I, \omega_{opt})$ para la tarea con todas sus restricciones. Resolvemos n tareas presentadas en el punto 4.1, para cada una de las restricciones y buscamos para cada una de ellas las soluciones óptimas: $\omega_{opt}^1 \subset I$ con el funcional $C(I, \omega_{opt}^1)$, $\omega_{opt}^2 \subset I$ con el funcional $C(I, \omega_{opt}^2)$, ..., $\omega_{opt}^n \subset I$ con el funcional $C(I, \omega_{opt}^n)$. Posteriormente buscamos para cada tarea sus $R_j = C(I, \omega_{opt}^j) - C(I, \omega_0)$, claro que $R_j \geq 0$ para todos los $j = 1, 2, \dots, n$. Para cada tarea buscamos los conjuntos:

$$\Omega_j(R_j) = \{\omega \subset I \mid C(I, \omega_{opt}^j) - R_j \leq C(\omega) \leq C(I, \omega_{opt}^j)\}$$

Posteriormente buscamos la intersección

$$\Omega_0 = \bigcap_{j=1}^n \Omega_j(R_j)$$

$\Omega_0 \neq \emptyset$ porque siempre $\omega_b \in \Omega_j$, $j = 0, 1, 2, \dots, n$.

En correspondencia con el método combinatorio aproximacional [9], en este caso siempre $\omega_{opt} \in \Omega_0$, por eso

$$C(I, \omega_{opt}) = \max_{\omega \in \Omega_0} C(\omega)$$

Si tenemos n criterios lineales $C_1(\omega)$, $C_2(\omega)$, ..., $C_n(\omega)$, para cada uno de ellos buscamos las soluciones óptimas

$$C_1(I, \omega_{opt}^1), C_2(I, \omega_{opt}^2), \dots, C_n(I, \omega_{opt}^n)$$

Si tenemos una solución $\omega_b \subset I$ para el cual conocemos $C_1(\omega_b)$, $C_2(\omega_b)$, ..., $C_n(\omega_b)$ calculamos $R_j = C_j(\omega_{opt}^j) - C_j(\omega_b)$, buscamos:

$$\Omega_j(R_j) = \{\omega \mid C_j(I, \omega_{opt}^j) - R_j \leq C_j(\omega) \leq C_j(I, \omega_{opt}^j)\}$$

Posteriormente determinamos:

$$\Omega_0 = \bigcap_{j=1}^n \Omega_j(R_j)$$

$\Omega_0 \neq \emptyset$ porque $\omega_b \in \Omega_0$. Todos $\omega \in \Omega_0$ son soluciones que tienen $C_j(\omega)$ mejor ($\geq$) que solución $\omega_b \in \Omega_0$ para todos los criterios $C_j(\omega)$ al mismo tiempo, y peor ($\leq$) con la solución $\omega_{opt}^j \in \Omega$ no mas que $R_j \geq 0$.

Todos los $\omega \notin \Omega_0$ son peores que la solución ω_b en al menos un criterio.

5 Conclusiones

El método y el algoritmo del árbol de cubos son eficaces para resolver problemas de optimización discreta. En la gran cantidad de problemas resueltos se determinó la solución exacta sin requerir un número elevado de iteraciones, en el peor de los casos fue $O(m^4)$.

Las reglas de rechazo en las cuales se fundamenta este método, permiten que se puedan modificar para resolver otro tipo de problemas, como se mostró en la determinación del intervalo de estabilidad de la solución óptima. Con lo cual se muestra que el método del árbol de cubos es flexible para resolver un gran número de problemas de optimización discreta.

El método del árbol de cubos, fundamentado matemáticamente, es una buena opción para resolver problemas que se venían resolviendo con otros métodos exactos, pero con la diferencia de que nuestro método se puede aplicar a solucionar problemas de optimización discreta de dimensiones mayores. Además, en el caso de que el problema sea lo suficientemente grande y requiera de una gran cantidad de memoria y procesamiento, que no sea posible satisfacerla con una computadora de un solo procesador, es posible utilizar el método con una variante para determinar la solución aproximada, donde la solución óptima estará en el intervalo dado por la precisión requerida.

Como se mostró en el procedimiento del método árbol de cubos, las reglas de rechazo nos indican que en varios subárboles se podría encontrar la solución óptima, donde cada subárbol es un conjunto disjunto. Por tal motivo es posible calcular cada uno en forma independiente, lo cual se puede aprovechar para utilizar las computadoras paralelas, donde cada procesador calcule un subárbol. Nuestros primeros análisis y resultados, nos indican que para una máquina paralela con 32 procesadores es posible incrementar el número de variables de 3 a 4 veces más que la versión secuencial.

Referencias

1. Birkhoff G., "Lattice Theory", second edition Amer., Math Soc., Providence, 1948
2. Grätzer G. "General Lattice Theory" Akademie-Verlag Berlin, 1978.
3. Cook W. J., Cunningham W. H., Pulleyblank W. R., Schrijver A. "Combinatorial Optimization" A Wiley-Interscience Publication, John Wiley and Sons Inc, New York 1998.
4. Nemhauser G. L., Wolsey L. A. "Integer and Combinatorial Optimization" A Wiley-Interscience Publication, John Wiley and Sons Inc., New York 1999
5. Khachaturov V. R. "The modeling and methods of solutions of multiextremal allocations problems and the use of supermodular function properties in the Boolean lattices", Algorithms and Languages of algorithms, Moscu: "Nauka", 1986
6. Khachaturov V. R. "Multiextremal allocation problems (models and solution methods)". Studia Scientiarum Mathematicarum Hungarica 27, 1992.

7. Khachaturov et others. "Combinatorial methods and algorithms for solving problems of discrete optimization with large dimensionality" Moscu: "Nauka", 2000.
8. Pisinger D., "An exact algorithm for large multiple knapsack problems", *European Journal of Operational Research*, 114 528-541 (1999)
9. Khachaturov V. R. "The Approximative-Combinatorial Meted and Some of its Applications", Moscow, Computational Mathematics and Mathematical Phisics, Vol 14, 1974.
10. Khachaturov V. R. "The Approximative-Combinatorial Meted for Decomposition and Composition of Systems and Finite Topology Spaces, Lattices and Optimisation", Moscow, Computational Mathematics and Mathematical Phisics, 1985.