

Espacios de Búsqueda en un Árbol Binario para Resolver Problemas de Optimización Discreta

María Elena Gómez-Torres¹, J. Crispín Zavala-Díaz², Marco Antonio Cruz-Chávez³

¹Instituto Tecnológico de Zacatepec
Calzada Instituto Tecnológico 27, Centro, 62780, Zacatepec Morelos, México
marie112212002@yahoo.com.mx
²FCAeI, ³CIICAp, UAEM
Av. Universidad 1001, Chamilpa, 62209, Cuernavaca Morelos, México
{crispin_zavala, mcruz}@uaem.mx

Resumen. En el presente artículo mostramos la forma de cómo obtener un espacio de búsqueda en un árbol binario para solucionar un problema de optimización discreta. Para ejemplificar el comportamiento de cómo se va reduciendo el espacio de búsqueda para obtener la solución óptima global utilizamos el planteamiento del problema de la mochila 0-1. La generación del espacio de búsqueda en el árbol binario esta dado por los subconjuntos de un conjunto potencia de los índices de la función objetivo. Dicho espacio de búsqueda se va reduciendo por medio de una prueba de optimalidad hasta llegar a encontrar la mejor solución. Por los resultados obtenidos se puede comprobar que se logró reducir el espacio de búsqueda eficientemente, propiciando con esto las bases para el desarrollo de un algoritmo que sea capaz de solucionar cualquier problema de búsqueda de tipo discreto.

1. Introducción

Para solucionar los problemas de optimización discreta un factor muy importante es el acotamiento del espacio de búsqueda, donde sea factible determinar la solución óptima. Entre más rápido se determine o se reduzca el espacio de búsqueda el algoritmo será más eficiente.

La determinación del espacio de búsqueda donde se busca la solución óptima depende directamente del algoritmo que se utilice para definir ésta [1]. En este trabajo presentamos la forma de acotar el espacio de búsqueda a través de la definición de reglas de acotamiento para solucionar un problema de optimización discreta.

2. Descripción de la generación del espacio de búsqueda

Los índices de los elementos que forman la función objetivo los podemos agrupar en el conjunto $I = \{1, 2, \dots, m\}$ donde m es el número de elementos de la función objetivo. El espacio de búsqueda de este conjunto estará dado por todos los posibles sub-

conjuntos que pueden ser parte de la solución, esto es el conjunto potencia del conjunto I denotado por $P(I)$. A este conjunto potencia lo podemos representar como un cubo $C(m)$ de dimensión m , el cual está definido por $C(m)=[\emptyset, I]$, donde \emptyset representa el vértice mínimo e I representa el vértice máximo.

Para ejemplificar lo anterior consideremos el conjunto $I = \{1,2,3,4\}$ donde el conjunto potencia está dado por:

$$P(I) = \{\{\emptyset\}, \{1\}, \{2\}, \{3\}, \{4\}, \{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}, \{2,4\}, \{3,4\}, \{1,2,3\}, \{1,2,4\}, \{1,3,4\}, \{2,3,4\}, \{1,2,3,4\}\}$$

y lo representamos por medio del cubo de la figura 1.

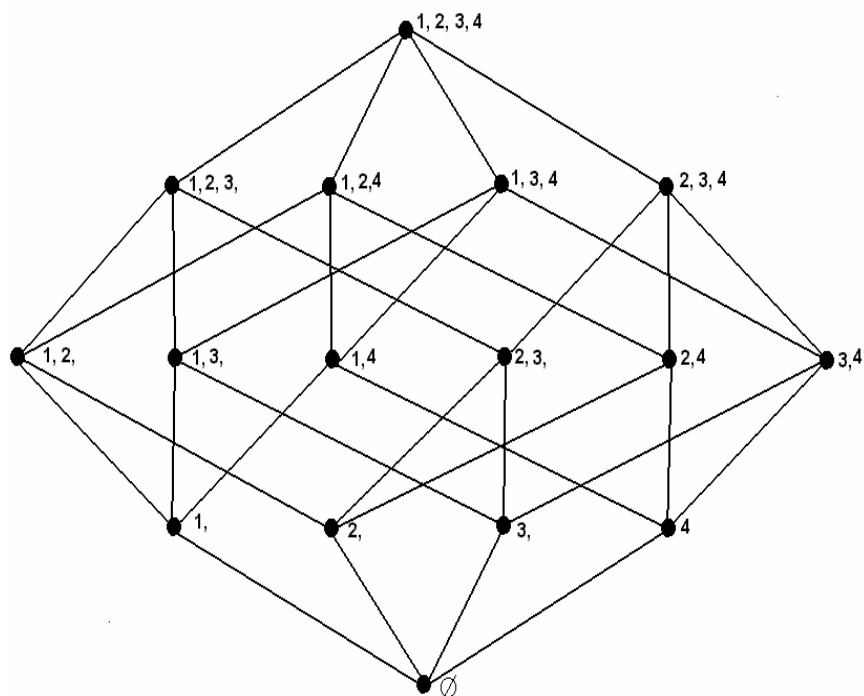


Fig. 1. Cubo $C(m)$ de dimensión 4

Para reducir el espacio de búsqueda el cubo $C(m)$ se parte en dos cubos de dimensión menor, donde la intersección es el conjunto vacío y la unión de todos los cubos es igual a $C(m)$ (fig. 2). Los dos cubos disjuntos: $C_1(m-1)$ y $C_2(m-1)$ de dimensión $(m-1)$ cada uno se pueden hacer de m distintas formas:

- Para el cubo $C_1(m-1)$ se considera un intervalo $[\{1\}, I]$. (fig. 2), cuyos elementos son: $\{\{1\}, \{1,2\}, \{1,3\}, \{1,4\}, \{1,2,3\}, \{1,2,4\}, \{1,3,4\}, \{1,2,3,4\}\}$
- Al cubo $C_2(m-1)$ le corresponde un intervalo $[\emptyset, I \setminus \{1\}]$. (fig. 2), cuyos elementos son: $\{\{\emptyset\}, \{2\}, \{3\}, \{4\}, \{2,3\}, \{2,4\}, \{3,4\}, \{2,3,4\}\}$

Como se observa todos los subconjuntos del conjunto $C_1(m-1)$ tienen al elemento 1. Mientras que en el conjunto $C_2(m-1)$ ningún subconjunto tiene a ese elemento 1. Por tanto $C_1(m-1) \cap C_2(m-1) = \emptyset$

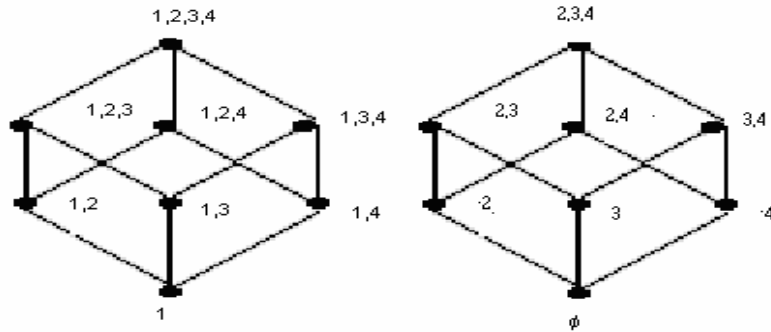


Fig. 2. $C_1(m-1)$ es el de la izquierda y el cubo $C_2(m-1)$ es el de la derecha

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 ($m-2$) se forman cuatro cubos que son una partición del cubo $C(m)$. (fig. 3)

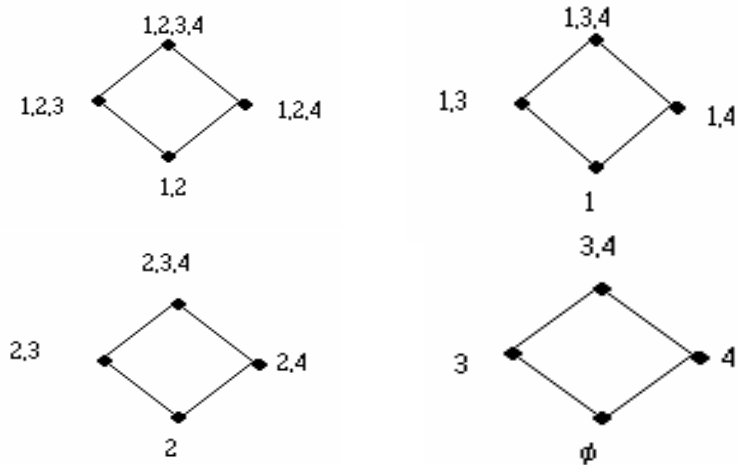


Fig. 3. Representación del nivel ($m-2$) de los cubos $C_1(m-1)$ y $C_2(m-1)$

Así sucesivamente hasta llegar al último nivel, donde el número de cubos de dimensión cero es igual a 2^m . Entonces es igual al número de vértices del cubo inicial $C(m)$. (fig. 4)

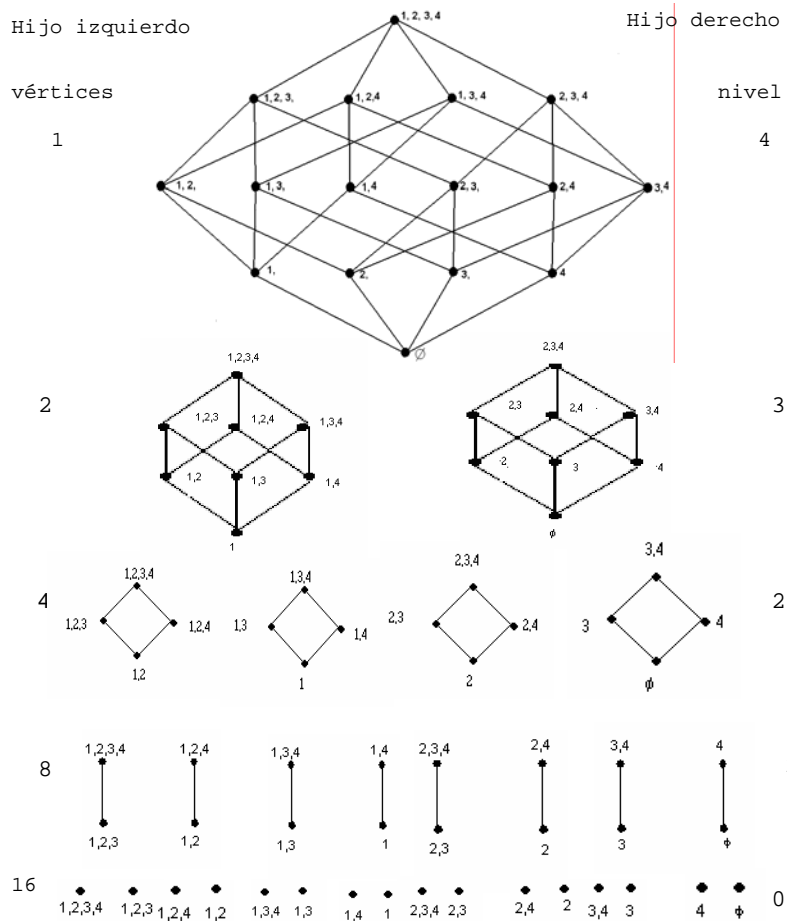


Fig. 4. Representación de los espacios de búsqueda de dimensión 4

3. Utilización del espacio de búsqueda para resolver problemas de optimización discreta

Con el objetivo de explicar la utilización del espacio de búsqueda, se hace uso del planteamiento del problema de la mochila 0-1 (Knapsack Problem = KP). Dado un conjunto de n elementos y una mochila con una utilidad del elemento p_i , un volumen del elemento w_i , y una capacidad de la mochila c . Seleccionar un subconjunto de elementos tal que:

$$\max z = \sum_{i=1}^n p_i x_i \tag{1}$$

Sujeto a:

$$\sum_{i=1}^n w_i x_i \leq c \tag{2}$$

Donde $x_i \in \{0,1\}$, $i \in \{1, \dots, n\}$

Para mostrar como se utiliza la reducción del espacio de búsqueda resolvemos este problema para un conjunto de 5 elementos con sus respectivas utilidades p_i , (3, 8, 7, 6 y 2), y volúmenes w_i , (6, 2, 1, 12, 2) y una mochila con una capacidad de $c=11$.

Se determina la solución entera de (4) como una solución inicial con la finalidad de encontrar la mejor solución óptima, ya que al comparar la función objetivo con la solución entera, si ésta última resulta ser mayor que la función objetivo esta pasa a ser la solución óptima. Por otra parte se calcula la función lineal con (5), la cual se utiliza para acotar el árbol en caso de que no se encuentre una solución lineal mayor que la función objetivo, de no ser así aún es posible encontrar una mejor solución óptima y se continúa ramificando el árbol. Ambas soluciones se determinan por medio de un método algebraico que consiste en obtener los coeficientes y ordenarlos de mayor a menor [2], con lo cual la expresión queda como en la relación (3).

$$\frac{p_1}{w_1}, \frac{p_2}{w_2}, \dots, \frac{p_m}{w_m} \tag{3}$$

Cálculo del nodo raíz, nivel 5

3 2 5 4 1	Índices de la posición original
$q_i = \frac{7}{1}, \frac{8}{2}, \frac{2}{2}, \frac{6}{12}, \frac{3}{6}$	Se ordenan (3)
$\sum p_i = 7 + 8 + 2 = 17$	Se suman las utilidades
$\sum w_i = 1 + 2 + 2 = 5$	Se suman los volúmenes de cada elemento que cumplen con (2)
$x_1=0, x_2=1, x_3=1, x_4=0, x_5=1$	Son los elementos que entran como una solución temporal

La solución entera esta dada por:

$$C(I) = \max \sum_{j=1}^q p_{ij} \tag{4}$$

La solución lineal por:

$$C(L) = C(I) + \frac{C - \sum_{i=1}^q w_{ij}}{w_{i(q+1)}} (p_{i(q+1)}) \quad (5)$$

Donde: (q+1) indica la posición del siguiente elemento que no entró como parte de la solución temporal.

Como el primer cálculo es en la raíz, la función entera se considera la solución óptima temporal. Por lo tanto la función entera pasa a ser la función objetivo [4]

$$C(I) (4): 7 + 8 + 2 = 17$$

$$C(L) (5): 17 + \frac{11-5}{12} (6) = 20$$

Se continúa ramificando el siguiente nivel del árbol binario.

Cálculo del hijo izquierdo $C_1(m-1)$, nivel 4. Se considera al primer elemento fijo, formando parte de la solución.

Elemento fijo 1: $p_1=3$ $w_1=6$

$$\text{Elementos ordenados: } q_i = \frac{7}{1}, \frac{8}{2}, \frac{2}{2}, \frac{6}{12}$$

Se procede a calcular una nueva restricción (c_{nueva}), así como la función entera de la siguiente manera:

$$c_{nueva} = c - w_1 = 11 - 6 = 5 \quad (6)$$

Se procede a buscar los elementos que cumplan con la restricción sin tomar en cuenta el elemento 1

$$\begin{aligned} \sum p_i &= 7 + 8 + 2 = 17 && \text{Se suman las utilidades} \\ \sum w_i &= 1 + 2 + 2 = 5 && \text{Se suman los volúmenes de cada elemento que cumplen con (2)} \end{aligned}$$

Se determina la solución entera:

$$C(I) = \sum_{i=1}^1 p_i x_i + \max \sum_{i=2}^5 p_i x_i = (3 + 7 + 8+2)=20 \quad (7)$$

$$C(L) (5): 20 + \frac{11 - (6 + 5)}{12} (6) = 20$$

Como en estos casos ya se tiene un nuevo resultado con que compararse con el anterior (nodo raíz) se procede a realizar la prueba de optimalidad, que consiste en validar las siguientes condiciones [4]:

$$\begin{aligned} \text{Si } F.O_{\cdot \text{NodoRaíz}} < F.L_{\cdot \text{Hijolzquierdo}} & \text{ entonces} && \text{(condición 1)} \\ \text{Si } F.E_{\cdot \text{Hijolzquierdo}} > F.O_{\cdot \text{NodoRaíz}} & \text{ entonces} && \text{(condición 2)} \\ F.E. & = F.O. \end{aligned}$$

Donde:

F.O. = Función Objetivo

F.L. = Función Lineal

F.E. = Función Entera

Como se observa en los resultados se obtiene una mejor solución, la función objetivo será igual a 20, y los elementos que forman parte de la solución $x_1=1$, $x_2=1$, $x_3=1$, $x_4=0$, $x_5=1$.

Cálculo del hijo derecho $C_2(m-1)$, nivel 4. Se descarta el primer elemento y se toman los elementos restantes y procede a ordenar los elementos:

$$\text{Se ordenan los elementos } q_i = \frac{7}{1}, \frac{8}{2}, \frac{2}{2}, \frac{6}{12}$$

$$c = 11$$

Se encuentran los elementos que cumplan con la restricción (2)

$$\sum p_i = 7 + 8 + 2 = 17$$

$$\sum w_i = 1 + 2 + 2 = 5$$

$$C(I) (7): 7 + 8 + 2 = 17$$

$$C(L) (5): 17 + \frac{11 - 5}{12} (6) = 20$$

Los elementos de la función son: $x_1=0$, $x_2=1$, $x_3=1$, $x_4=0$, $x_5=1$

Se compara con la solución óptima temporal y se evalúa la prueba de optimalidad:

$$\begin{aligned} \text{Si } F.O_{\cdot \text{Hijolzquierdo}} < F.L_{\cdot \text{HijoDerecho}} & \text{ (condición 3)} \\ \text{Si } F.E_{\cdot \text{HijoDerecho}} > F.O_{\cdot \text{Hijolzquierdo}} & \text{ (condición 4)} \\ F.E. & = F.O. \end{aligned}$$

No cumple con la condición por lo que se corta esta rama.

Se sigue ramificando a partir del hijo izquierdo con el objetivo de encontrar una solución óptima global.

Cálculo del hijo izquierdo, nivel 3. Se toman como fijos los elementos 1 y 2, éstos forman parte de la solución, $p_1 = 3$, $w_1 = 6$, $p_2 = 8$, $w_2 = 2$

Los elementos restantes se ordenan: $q_i = \frac{7}{1}, \frac{2}{2}, \frac{6}{12}$

Se calcula c_{nueva} : $c_{nueva} = c - (w_1 + w_2) = 11 - (6+2) = 3$,

Se procede a buscar los elementos que cumplan con la restricción sin tomar en cuenta el elemento 1 y 2.

$$\sum p_i = 7 + 2 = 9$$

$$\sum w_i = 1 + 2 = 3$$

$$C(I) = (3 + 8) + 7 + 2 = 20$$

$$C(L) = 20 + \frac{11 - (8 + 3)}{12} (6) = 20$$

Los elementos que forman parte de la solución $x_1=1, x_2=1, x_3=1, x_4=0, x_5=1$
Se compara con la solución óptima temporal, de acuerdo a la prueba de optimalidad.
No cumple, por lo que se corta esta rama

Cálculo del hijo derecho, nivel 3. Se toma como fijo el elemento 1 ya que forma parte de la solución y se quita el elemento 2, los demás se vuelven a ordenar.

$$q_i = \frac{7}{1}, \frac{2}{2}, \frac{6}{12}$$

Se procede a buscar los elementos que cumplan con la restricción sin tomar en cuenta el elemento 1 ni el 2.

$$\sum p_i = 7 + 2 = 9$$

$$\sum w_i = 1 + 2 = 3$$

$$C(I) = (3 + 9) = 12$$

$$C(L) = 12 + \frac{11 - (6 + 3)}{12} (6) = 13$$

Los elementos que son parte de la solución: $x_1=1, x_2=0, x_3=1, x_4=0, x_5=1$

Se compara con la solución óptima temporal. En vista de que ya no es posible encontrar una mejor solución óptima global, el proceso concluye hasta aquí, por lo que la solución óptima global se encuentra en el nivel 4.

Para una mejor comprensión del ejemplo se muestra una representación gráfica (Fig. 5) y una tabla (tabla 1) de cómo se llegó al óptimo global.

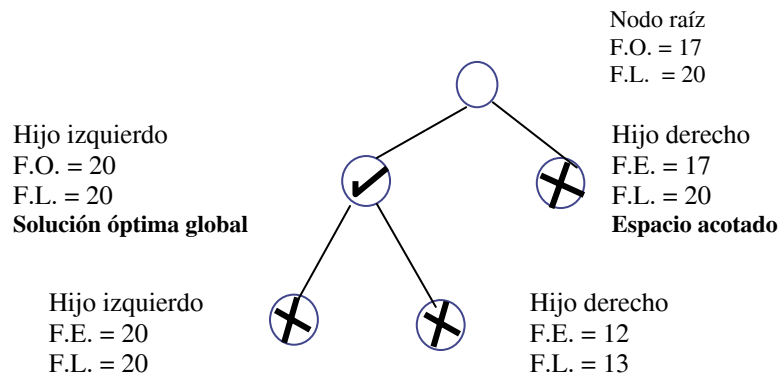
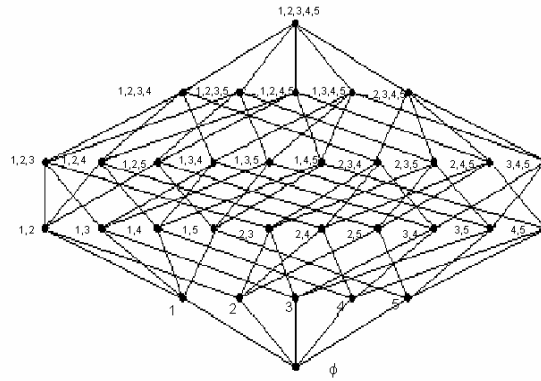
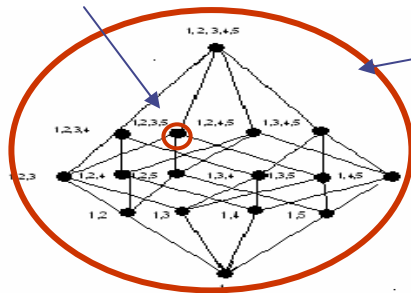


Fig. 5. Resultado de la función entera, lineal y objetivo, para cada nodo del árbol

La siguiente figura muestra todos los nodos del árbol binario de dimensión 5 para efectos del ejemplo anterior y se indican los elementos que maximizan la función objetivo (Fig.6).



Estos son los elementos que forman la solución óptima.



Este es el espacio de búsqueda reducido donde se encuentra la solución óptima.

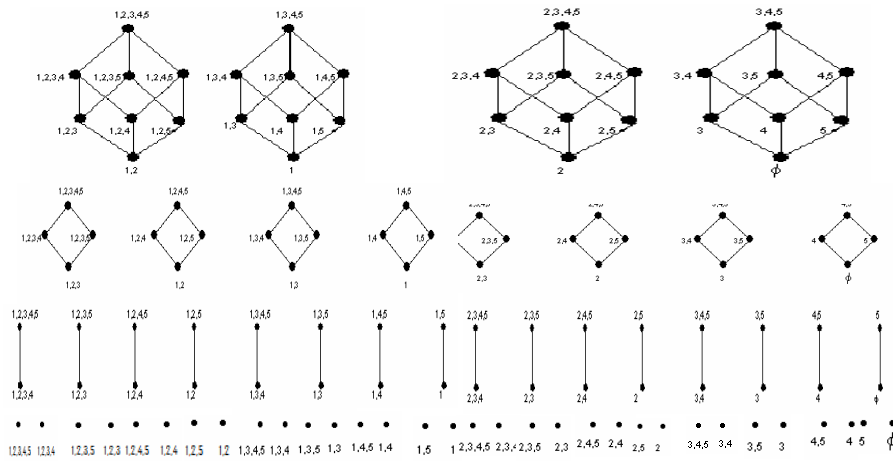
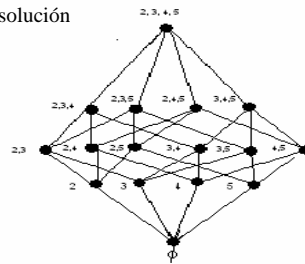


Fig. 6. Representación del árbol binario de dimensión 5 y el espacio de búsqueda donde se encuentran los elementos que forman parte de la solución óptima global

4. Conclusiones

Se concluye que al ir reduciendo el espacio de búsqueda en un árbol binario es posible encontrar la solución exacta en un espacio de búsqueda reducido, y al no ser necesario recorrer todo el árbol binario, es posible encontrar la mejor solución de manera más rápida. Cabe resaltar que para efectos de explicar como utilizar el espacio de búsqueda que generamos con un árbol binario, se utilizó el planteamiento del problema de la mochila. No obstante los espacios de búsqueda en árboles binarios pueden ser utilizados para encontrar la solución de cualquier otro problema de tipo discreto.

Por lo que se pretende con investigaciones futuras demostrar que es posible desarrollar un algoritmo que sea capaz de resolver problemas discretos aplicando las reglas de acotamiento presentadas en este artículo, explorando el espacio de búsqueda eficientemente y reduciendo el tamaño del espacio de búsqueda.

Referencias

- [1] Adenso D. Fernández, et al, "Optimización Heurística y Redes Neuronales", ed. Paraninfo S.A, Madrid, España 1996, ISBN 84-283-2269-4
- [2] Vladimir Khachaturov, Jose Crispín Zavala-Díaz, "Método y algoritmo del árbol de cubos y sus aplicaciones para resolver diferentes problemas de optimización discreta" Research on computing science, Advances in computer science in Mexico, IPN, ISSN 1665-9899, pp 111-126
- [3] Silvano Martello, Paolo Toth, "Knapsack Problems, Algorithms and Computer Implementations", ed. John Wiley & Sons, DEIS University of Bologna 1990, ISBN 0-471-92420-2
- [4] Crispín Zavala-Díaz, Marco Cruz-Chávez, María Elena Gómez-Torres, "Algoritmo híbrido Branch and Bound para determinar la solución, exacta o aproximada, del 0/1 knapsack problem con un parámetro" En etapa de desarrollo.