Optimización estocástica
Zaira Carolina Martinez Vargas

Tarea 3:
Ramificación y acotación

2023-03-06

1. Introducción

Ramificación y acotación (Branch and bound) es una técnica que consiste en la enumeración en un árbol en el cual el espacio de búsqueda se divide de forma sucesiva, dando lugar a problemas lineales que se resuelven por nodo. El procedimiento establece una cota del valor óptimo de la función objetivo, la idea es eliminar los nodos o al respectivo subárbol, si la cota del nodo (o la raíz del subárbol) supera la cota óptima. De esta forma, el espacio de búsqueda disminuye. Para el problema de Bin Packing, se suele utilizar la técnica de Branch and Price, que consiste en la combinación de la técnica de generación de columnas y de ramificación y acotación. La ventaja de esta combinación es que se disminuye la simetría en el espacio. Así, se dice que el determinar una solución para un contenedor se puede modelar como el problema de encontrar un camino en un grafo acíclico directo.

En este trabajo, se utiliza la técnica planteada en el capítulo 8 del libro [1], que utiliza sólo ramificación y acotación para resolver el problema de Bin Packing.

2. Metodología

Como se mencionó, se necesita calcular una cota para aplicar la técnica de ramificación y acotación. Se utiliza el límite inferior más fuerte (stronger lower bound) L_2, dado por:

Teorema 1. *Dada cualquier instancia I del Bin Packing Problem (BPP), la capacidad c los contenedor, y cualquier entero α, $0 \leq \alpha \leq c/2$.*

$$J_1 = \{j \in N : w_j > c - \alpha\}, J_2 = \{j \in N : c - \alpha \geq w_j \leq c/2\}, J_3 = \{j \in N : c/2 \geq w_j \geq \alpha\};$$

entonces

$$L(\alpha) = |J_1| + |J_2| + \left(0, \left\lceil \frac{\sum_{j \in J_3} w_j - (|J_2|c - \sum_{i \in J_2 w_j})}{c} \right\rceil \right)$$

es un límite inferior.

Esta cota es una forma de evitar explorar un subárbol. También se utiliza otra cota L_3 que no se utilizó en este trabajo y otra de las herramientas que se utiliza es la dominación de nodos. Si un nodo domina a otro y el que domina tiene una cota L_3 o L_2 mayor que la cota de todo el arreglo, el dominado ya no tendrá la mejor solución y por lo tanto, no se explora ninguno de los nodos.

Para la técnica de ramificación y acotación, en la raíz se añadió el valor más grande, y después se armó el árbol como se muestra en la referencia mencionada anteriormente, utilizando el arreglo:

$$arr = \{49, 41, 34, 33, 29, 26, 26, 22, 20, 19\},$$
$$n = 10, c = 100, L_2 = 3.$$

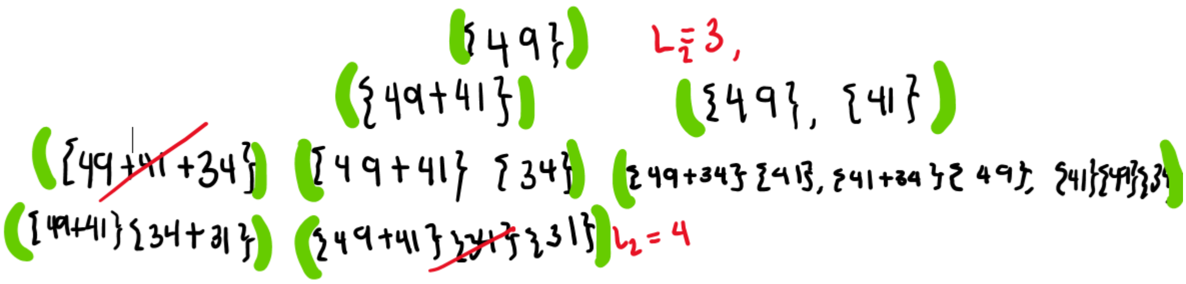
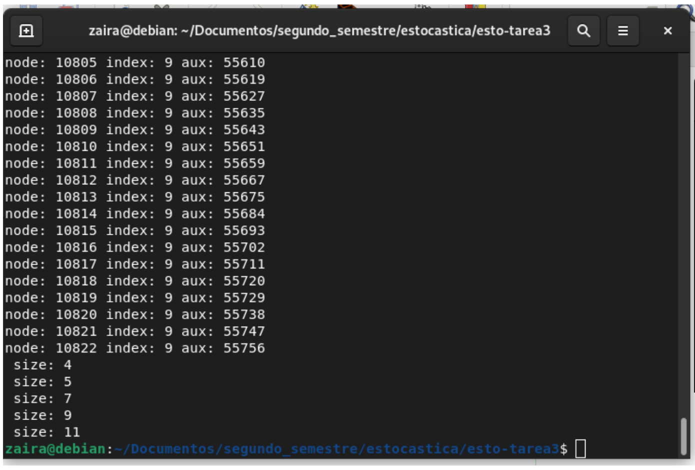


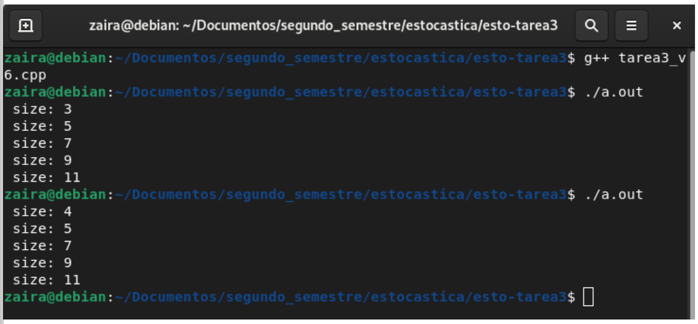
Figura 1: Ejemplo de una parte del árbol que se formará con el arreglo mencionado. Los paréntesis verdes se utilizan para denotar nodos, el valor 49 está en un nodo, y los valores (49, 41) están en un nodo. Los nodos que tienen una línea roja, es que fueron eliminados y por lo tanto, no se sigue llenando el subárbol, uno de ellos se eliminó porque la suma de los elementos es mayor que la capacidad del contenedor y otro por el tamaño de la cota.

En este caso, cuando las cotas eran iguales a la cota inicial, se utilizó el algoritmo de First-fit-decreasing (FFD) para encontrar una solución, dejando la combinación de elementos que tenía esa cota. Por ejemplo, la cota del elemento que se eliminó, se ingresó el arreglo $= \{90, 34, 31, 29, 26, 26, 22, 20, 19\}$, y se encontró que $L_2 = 4$, además, nos da una pista de la combinación, los valores 41 y 49 no dan el óptimo si están en el mismo contenedor. En cambio, se encuentra que la combinación 41+33, y elementos restantes, sigue teniendo la misma cota. Por lo que, la combinación de estos si debe de estar en el mismo contendor.

3. Resultados

Una desventaja de este método, es que la dominación de los nodos es una estructura complicada de implementar y que si se utilizan vectores, y no se va eliminando espacio, este crece mucho. Como en el caso del arreglo utilizado, se formaron aproximadamente 55 mil nodos, de los cuales, sólo 4 cumplían con la cota y no superaban el tamaño del contenedor entre la combinación de elementos. Ya que se encontró una combinación con la misma cota, se aplicó el método de FFD con distinto inicio, para buscar el tamaño óptimo.





En la captura de pantalla de la izquierda, se muestra que hay 10822 nodos que no superan el tamaño del contendor, el índice es la profundidad a la que se encuentra el nodo y la variable auxiliar son los valores que tomaban los nodos no eliminados, por ejemplo, se observó en el árbol que el nodo 1 se eliminó, y el conteo de nodos avanzó a 2, y la variable auxiliar es la que marca esta pauta. En la captura de pantalla de la derecha, se muestran combinaciones con uno de los arreglos, iniciando en un nodo distinto, y se encontró el valor óptimo 3, en la segunda iteración. Así, es posible saber cuál será la solución óptima para el arreglo.
Por lo que no se probó el método con las instancias utilizadas en las tareas anteriores por el tamaño del vector aux que se formaría.

4. Conclusiones

Fue un procedimiento tardado y complicado, en algún punto pensé que la dominación de nodos era más sencilla y por desgracia se llevó tiempo y no fue posible. También, no supe cómo evitar que el conteo de la variable aux aumentara tanto sin cambiar mucho el procedimiento, pero ya no hubo tiempo. Se espera arreglar este método antes de la última tarea para poder compararlo al utilizar la técnica BNP. También, se espera mejorar la implementación y si es posible, añadir la cota extra de L3.

Referencias

[1] Silvano Martello and Paolo Toth. Algorithms for knapsack problems. *North-Holland Mathematics Studies*, 132:213–257, 1987.