

# TALLER N°3

Knapsack Problem  
(Problema de la mochila)



Taller de programación 1-2024

Fecha: 15-08-24

Autor: Benjamín Zúñiga Jofré



## TALLER N°3

---

### Problema de la mochila

#### Explicación breve del algoritmo

El programa consiste el conjunto de objetos que más ganancias producen sin exceder la capacidad máxima del problema

Para lograr encontrar este conjunto de objetos se utiliza la clase Simplex con el método Branch and Bound que sirve para resolver problemas de programación lineal, este problema cuenta con una restricción adicional a un problema de programación lineal pues el vector solución solo debe contener unos y ceros, pues estos valores indican si el objeto pertenece o no a la solución respectivamente.

#### Heurísticas o técnicas utilizadas

Dentro de las técnicas utilizadas para mejorar los tiempos del programa, se encuentra el uso del anteriormente mencionado Branch and Bound, el cual consiste en ir podando problemas que no tienen el potencial de dar la mejor solución, esto es medido mediante una cota superior e inferior, las cuales se calculan resolviendo el problema de manera normal y relajando la solución, es decir, truncando los valores del vector solución al entero más próximo (aplicando la función piso), mediante comparaciones de cotas inferiores y superior se pueden ir descartando problemas que eventualmente no generaran la mejor solución. También se ordenan los problemas a resolver de manera descendiente según sus cotas superiores, para así resolver el problema con potencial de mejor solución en cada momento

#### Funcionamiento del programa

El programa funciona primeramente creando un objeto de la clase Simplex mediante la lectura de un archivo el cual tiene un formato específico, este archivo posee primeramente el número de cada tipo de desigualdades que representan el problema a resolver, para así poder leer todas las líneas del archivo, luego posee dos líneas, donde la primera posee la solución y es seguida del valor que tiene cada variable dentro del problema y la segunda posee la capacidad máxima de la mochila seguida de los pesos en negativo de cada variable del problema, este formato se debe a la clase y como genera las desigualdades (restricciones) y finalmente el archivo es seguido de desigualdades que indican que el valor máximo para cada variable es uno. Posteriormente se resuelve el problema, representado por la clase Simplex, mediante el método de Branch and Bound, el cual está implementado de la siguiente manera, primero se copia el objeto Simplex que representa el problema inicial y se resuelve mediante el método Simplex, luego se crea un vector para almacenar los problemas por resolver junto a un vector para guardar la mejor solución generada y otros dos vectores, uno para guardar las cotas superiores de los problemas a resolver y otro para guardar las soluciones generadas, posteriormente se inicia un ciclo, el cual dura hasta que el vector de problemas a resolver sea



vacío y comienza buscando el índice del problema con mayor cota superior, luego lo establece como problema actual y lo elimina del conjunto de problemas a resolver, al igual que su cota del conjunto de cotas superiores de problemas a resolver, después se comprueba si el problema tiene solución para así seguir con el problema actual, en caso de que si existe solución, se guarda la solución del problema escogido y si es mejor que la guardada anteriormente, se actualiza esta última, luego se comprueba si la solución no ha sido generada en caso de que si lo haya sido se continua con otro problema, sino se calcula la cota inferior del problema actual y así se comprueba si la cota inferior es igual a la superior, lo cual indicaría que la solución sin restricción de enteros ya es entera y al haber revisado en orden de los problemas con mayor cota superior, esta solución sería la con mayor valor posible, en caso de ser así se retorna esta solución, sino se continua y se busca la variable mas fraccionaria dentro de la solución para así restringirla a uno o cero, es decir, que pertenezca o no la solución, luego se copia el problema actual dos veces y a una copia se le agrega la restricción de la variable más fraccionaria a cero y a la otra copia la restricción con uno, luego se resuelven las copias y si el cota superior de la solución es mayor a la cota inferior del problema actual se guarda la solución en su respectivo problema, se guarda el problema en el conjunto de problemas a resolver y se guarda la cota superior en el conjunto de cotas de problemas a resolver, finalmente en caso de que se acaba el ciclo y no se encontró una solución óptima entera se relaja la solución más grande encontrada y se buscan objetos que quepan todavía en la mochila para así aumentar un poco la solución.

## Aspectos de implementación y eficiencia

El programa se podría considerar eficiente hablando de tiempo pues no demora tiempo excesivo en resolver cada problema, pero es ineficiente en cuanto a encontrar la solución optima puesto que nunca se llega a encontrar, se cree que es debido al funcionamiento del método Simplex ya que a pesar de resolverse problemas con más restricciones siempre se escoge como base de la solución los mismo objetos pudiendo así llegar a entrar en bucle infinito el programa ya lo único que varia entre solución y solución es la variable más fraccionaria que va rotando entre distintas cantidades dependiendo el problema original.

## Ejecución del código

Para ejecutar el código basta con abrir una terminal del sistema con la dirección de la carpeta que contiene el código del programa y ejecutar el comando "make main" para compilar el programa principal, ejecutar el comando "make testClique" para compilar una prueba de la clase o ejecutar "make" para compilar los dos archivos a la vez, luego ejecutar "./main" o "./testSimplex" según corresponda y por último ingresar el nombre del archivo que contiene el problema a resolver, todo esto considerando que se usa un computador con alguna distribución del sistema operativo Linux.