TALLER N°3

Evaluación 3 - Taller de Programación

Simplex con números enteros

usach

Taller de programación 1-2023

Fecha: 11/07/2023

Autor: Lucas Mesías

TALLER N°3

Simplex con números enteros

Explicación breve del algoritmo

El algoritmo explora las posibilidades de soluciones a un problema de optimización de variables, donde se debe maximizar una función y ciertas variables deben ser enteras, según restricciones entregadas en un archivo, similarmente a la primera evaluación, se estructura como un árbol y se crean nodos para navegar hacia las soluciones con mayor posibilidad de dar un mejor resultado.

Heurísticas o técnicas utilizadas

El algoritmo simplex resuelve el problema para números flotantes, y nosotros deseamos que algunas de las variables sean enteras, por lo que los procedimientos solo se aplican a estas variables.

Se usó el número más fraccionario (cercano a 0.5) para escoger la variable a redondear, usando la siguiente fórmula: $|X - trunc(X)| - 0.5$

Luego para encontrar el lower bound, se truncaron todas las variables del conjunto de variables enteras y se aplicó simplex.

El upper bound es simplemente el resultado del simplex sobre las variables sin redondear.

Si lower bound y upper bound son iguales, la ejecución del programa se acaba y se encontró la solución.

El programa usa branch and bound para almacenar un conjunto de nodos posibles, de los cuales se escoge el de mayor límite superior, lo que aumenta las posibilidades de encontrar antes una solución correcta, al escoger un nodo se crean dos ramas, donde la variable fraccionaria es redondeada y truncada, generando 2 caminos por donde puede haber una solución.

Se usa un heap de máximos para almacenar los nodos, el heap se ordena según el upper bound, de esta forma se revisan los nodos con mayor probabilidad de dar un resultado más rápido y cercano del árbol, de manera eficiente.

Funcionamiento del programa

El programa muestra un menú donde el usuario puede elegir distintas opciones:

Cargar un archivo mostrando todos los nodos recorridos.

Cargar un archivo y ejecutar el algoritmo para dicho archivo 100 veces, sin escrituras a la terminal, para observar el rendimiento del programa.

Salir del programa.

Aspectos de implementación y eficiencia

El programa no se optimizó, se podría haber implementado una tabla de hash, alguna forma de reconocer si una rama no entregará mejores resultados, o de reconocer caminos ya recorridos. Pero no se hizo.

Ejecución del código

En el proyecto se utilizaron las librerías *iostream*, *iomanip*, *fstream, sstream*, *string, ctime, vector* y *cmath* para obtener datos por consola, el nombre del archivo a leer, realizar la lectura del archivo y crear el heap inicial. En este proyecto se usó el archivo Simplex.cpp y Heap.cpp entregados por el profesor a lo largo del semestre. No se incluyeron tests para los objetos Heap, puesto que fué entregado por el profesor, y Node, porque es muy simple y no tiene métodos.

Para ejecutar el código, se descomprime el archivo zip, ya sea usando click derecho y la opción de extraer, o con el comando 'unzip LucasMesias212666599', luego se abre una terminal dentro de la carpeta recién extraída y se usa el comando 'make'. Se compilarán todos los archivos, y se podrá ejecutar cualquiera de los siguientes programas:

- ./main
- ./test_Simplex