Universidad Simón Bolívar
Departamento de Computación y Tecnología de la Información
Diseño de Algoritmos I (CI5651)
Prof.: Guillermo Palma
Autores: - Guillermo Betancourt, carnet 11-10103
- Gabriel Giménez, carnet 12-11006

Informe - Proyecto II – PRPP

1. Detalles de implementación:

Para la resolución del problema propuesto, se implementó en C++ el solucionador del problema de PRPP con el algoritmo propuesto en el enunciado, basado en la técnica de Branch and Bound.

Para la implementación del grafo, se creó primeramente una clase “lado”, que contenía el índice de los vértices a los que conectaba, el costo y el beneficio del lado. Luego se creó una clase “vertice” que contenía el índice del vértice, que tenía un vector de lados para almacenas sus lados incidentes. Luego, para la representación del grafo, se utilizó un vector de vértices y un entero que contenía el máximo beneficio sobre el grafo. Para el almacenamiento de las posibles soluciones, se creó una clase “solución” que contenía un vector de enteros que contenía con los índices de los vértices en el orden del recorrido, un vector de lados que contenía los lados en el orden del recorrido.

Se implementaron los algoritmos propuestos en el enunciado para realizar Branch and Bound sobre el espacio de soluciones factibles para hallar el ciclo el ciclo de mayor beneficio sobre un grafo dado. Se realizó una búsqueda en profundidad por vértices, partiendo desde el vértice depósito del grafo, y se iba explorando una rama hasta conseguir llegar de nuevo al depósito. Si la solución era mejor que la solución actual en términos de mayor beneficio, se sustituía. La solución inicial tomada fue la que arrojó el algoritmo propuesto para el proyecto I de la asignatura.

Se implementaron igualmente las funciones propuestas para el algoritmo de Branch and Bound, como la búsqueda por DFS de las soluciones, las función para calcular un ciclo negativo en una solución parcial, para verificar el cumplimiento del acotamiento, para ver si un lado está en una solución parcial. Para el cálculo de ciclos repetidos, se implementó una función de acuerdo con la explicación proveída en la sección 1 del enunciado.

Es importante destacar que para compilación y ejecución del programa, se deben realizar los siguientes comandos:

Para compilar el código, desde la carpeta Proyecto2-1110103-1211006 ejecute:

$ make

Seleccione una instancia y cópiela en la carpeta 4_BranchAndBound

$ cp /5_Instancias/[carpeta_de_la_instancia]/[instancia] /4_BranchAndBound

Para correr el programa, ejecute:

$ cd 4_BranchAndBound
$ ./bb [instancia]

Al finalizar la ejecución, se escribirá en un archivo llamado **[instancia]-salida.txt** la mejor solución encontrada por la heurística del proyecto 1, y se escribirá en un archivo llamado **[instancia]-bb.txt** la solución óptima. Si el tiempo de corrida llega a superar los 7200 segundos (2 horas), el programa escribirá en **[instancia]-bb.txt** la mejor solución encontrada hasta ese momento.

2. Resultados experimentales y análisis:

En el archivo **Tablas.pdf** podrá encontrar las tablas de los resultados para cada una de las instancias solicitadas en el enunciado del proyecto, incluyendo el valor óptimo de la instancia, el valor de la heurística utilizada en el primer proyecto, la desviación de dicha heurística, el mejor resultado conseguido por el Branch and Bound, y el tiempo empleado por dicho algoritmo. Nótese que en las tablas, hay tiempos de ejecución demarcados como **- T -**, esto significa que el algoritmo fue ejecutado por 2 horas para dicha instancia y no había conseguido recorrer todo el espacio de soluciones restantes para decidir si la solución que poseía al momento de detener el algoritmo era la mejor.

Respecto a hardware del computador utilizado para hacer las corridas del algoritmo, se utilizó un computador con procesador Intel Core i5-4460 a 3.20 GHz, con 8 GB de memoria RAM, bajo el sistema operativo Windows 10.

Respecto al análisis de las tablas, se calcularon algunas estadísticas sencillas a partir de las tablas obtenidas para ver el desempeño del algoritmo en términos del valor de la mejor solución y del tiempo empleado para hallarla. En promedio, se alcanzó un 95.64% del valor óptimo para las soluciones de las instancias calculadas. Correr todas las instancias tomó un tiempo de cómputo total de 2 días, 9 horas, 12 minutos y 20 segundos, aproximadamente.