

Tarea computacional

*Profesor: Fernando Ordóñez**Auxiliar: Renaud Chicoisne*

1 Enunciado

1.1 Introduccion

El objetivo de esta tarea es implementar métodos de resolución para el problema de camino a costo mínimo, de flujo máximo y de flujo a costo mínimo . El grafo en lo cual deberán ejecutar sus algoritmos es un grafo de Santiago usado por el Cuerpo de Bomberos de Santiago para despachar sus carros hacia las emergencias.

Sea $G = (V, E)$ el grafo ocupado. Sea $C \subset V$ el conjunto de los nodos de oferta, y $D \subset V$ el conjunto de nodos de demanda. Cada nodo de oferta $i \in C$ tiene una cierta cantidad $c_i \in \mathbb{N}^*$ de flujo para dar. Asimismo, cada nodo de demanda $i \in D$ necesita exactamente $d_i \in \mathbb{N}^*$ unidades de flujo.

Para cada arco $(i, j) \in E$, pueden pasar a lo maximo $u_{ij} \in \mathbb{N}^*$ unidades de flujo, y cada unidad de flujo que pasa por este arco se demora $t_{ij} \in \mathbb{R}_+^*$ en atravesarlo.

1.2 Shortest path

Implementar uno de los algoritmos de camino más corto que existen y aplicarlo para encontrar el árbol de caminos mas cortos entre el nodo de oferta 53481 y todos los nodos de demanda.

1.3 Maximum flow

Implementar uno de los algoritmos de flujo máximo que existen y aplicarlo para encontrar el flujo máximo entre el conjunto de todos los nodos de oferta hasta el conjunto de todos los nodos de demanda. Su respuesta debe indicar el valor del flujo máximo, el flujo en cada arco e identificar el corte de capacidad mínima asociado.

1.4 Minimum cost flow

Implementar uno de los algoritmos de flujo a costo mínimo que existen y aplicarlo para encontrar el flujo de costo mínimo en el grafo G .

2 Formato de input

Junto con este documento vienen dos archivos que contienen toda la información necesaria para resolver los problemas precedentes.

2.1 Archivo de nodos

El archivo `nodos.csv` contiene la información de cada nodo con el formato siguiente:

`<#nodo>,<demanda/oferta>`

Donde la columna de demanda/oferta contiene un número negativo si es un nodo de demanda, positivo si es de oferta, y cero si es de transición.

2.2 Archivo de arcos

El archivo `arcos.csv` contiene la información de cada arco del grafo con el formato siguiente:

`<#arco>,<#nodo inicial>,<#nodo final>,< u_{ij} >,< t_{ij} >`

3 Formato de output

3.1 Maximum flow

Guardar las informaciones del flujo máximo encontrado en un archivo `max_flow.csv` con el formato siguiente:

`<#arco>,< x_{ij} >`

Con x_{ij} la cantidad de flujo pasando por el arco (i, j) . Si no hay flujo pasando por un arco (i, j) , se tiene que guardar $x_{ij} = 0$.

Asimismo, el corte mínimo encontrado tiene que ser guardado en un archivo `min_cut.csv` con el formato siguiente:

`<#arco>,< y_{ij} >`

Con $y_{ij} = 1$ si el arco (i, j) pertenece al corte mínimo, 0 sino.

3.2 Shortest path

Guardar las informaciones del árbol de rutas mínimas encontrado en un archivo `min_path.csv` con el formato siguiente:

`<#nodo>,<father(#nodo)>`

Con `<father(#nodo)>` el índice del padre del nodo `<#nodo>` en el árbol de caminos mínimos. Por convención, la raíz del árbol es su propio padre y si un nodo no pertenece al árbol se guarda su padre con índice `-1`.

3.3 Minimum cost flow

Guardar las informaciones del flujo de costo mínimo encontrado en un archivo `min_cost_flow.csv` con el formato siguiente:

`<#arco>,<xij>`

Con x_{ij} la cantidad de flujo pasando por el arco (i, j) . Si no hay flujo pasando por un arco (i, j) , se tiene que guardar $x_{ij} = 0$.

4 Entrega

Entregue un archivo ejecutable que lea los archivos de texto de entrada y resuelva estos problemas. Entregue también una hoja `readme.txt` describiendo como ejecutar estos programas.

Finalmente, entregue un informe `informe.pdf` que comporte a lo más 3 páginas (una por algoritmo) que describa para cada uno de los problemas el algoritmo implementado y un resumen de los resultados obtenidos, incluyendo valores óptimos de la función objetivo y tiempos de ejecución.

La entrega se hará por U-Cursos hasta el Viernes 25 de Noviembre a las 23:59.