DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

TP 2: Recorridos y árbol generador mínimo

28 de Mayo, 2023 — Algoritmos y Estructuras de Datos III

Integrante	LU	Correo electrónico
Zaid, Pablo	869/21	pablozaid2002@gmail.com
Arienti, Federico	316/21	fa.arienti@gmail.com

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón I/Planta Baja)
Intendente Güiraldes 2610 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (++54 +11) 4576-3300
http://www.exactas.uba.ar

RESUMEN

En la teoría de grafos, el problema del *árbol generador mínimo*[1] —o *minimum spanning tree*, en inglés— se refiere al problema de encontrar, para un grafo conexo $G = (V,\ E)$ con función de peso $w : E \to \mathbb{R}$ asociada, un subgrafo conexo y acíclico de G —es decir, un árbol generador— que minimice la suma total del peso de sus aristas.

Existen diversos métodos para la resolución de este problema. Entre ellos, los algoritmos *golosos* de *Prim* y de *Kruskal*, que se basan en la selección de aristas de peso mínimo *seguras*[2] para la construcción de una solución.

El siguiente informe evalúa el problema de los *módems*, explicado en el apartado siguiente, y lo reformula como una variante del problema de *árbol generador mínimo* que aprovecha el invariante del algoritmo de *Kruskal*. Además, evalúa la eficiencia de la solución propuesta de manera empírica en función de la aplicación de diferentes estructuras de datos: *priority queue* y *disjoint set*; y heurísticas: *union by rank* y *path compression*[3].

Palabras clave: *árbol generador mínimo, algoritmo de Kruskal.*

Índice

[1]Ver Thomas H. Cormen; Charles E. Leiserson; Ronald L. Rivest y Clifford Stein. Introduction to algorithms. 2009. Sección 23: *Minimum Spanning Trees.*

[2]Una arista es *segura* si y sólo si al agregarla a un subgrafo de un árbol generador, el resultado sigue siendo un subgrafo de algún árbol generador.

[3]Ver nota al pie (1). Sección 21: *Data Structures for Disjoint Sets.*

1. El problema de los módems

El problema de los *módems* que consideraremos tiene la siguiente premisa. Dado un conjunto de N oficinas

$$O := \{o_1 \ldots o_N\}$$

donde, para cada $1 \leq i \leq N$, la oficina o_i se representa por su posición $(x_i,\ y_i)$ en el plano cartesiano —con unidad en centímetros—, queremos encontrar el costo mínimo a emplear en cables UTP y en cables de fibra óptica para conectar todas las oficinas a internet[4].

Para ello, vamos a contar con $W < N$ módems y vamos a disponer de cables UTP y de fibra óptica, cuyo costo por centímetro será U y V pesos, respectivamente. Dado que los cables UTP tienen ciertas restricciones, estos se podrán utilizar si y sólo si la distancia entre dos oficinas es menor a R centímetros.

Por ejemplo, si

$$O = \{(0,\ 0),\ (0,\ 1),\ (1,\ 0)\},\ R = 2,\ W = 1,\ U = 1 \text{ y } V = 1$$

luego una solución puede situar un módem en la oficina $(0,\ 0)$ y conectarla a cada una de las restantes con un centímetro de cable UTP, por un costo total de dos pesos.

1.1. Modelado como un problema de árbol generador mínimo.

Dada la descripción anterior, vamos a demostrar que el problema se puede modelar como una variante del problema del árbol generador mínimo. Lo detallamos a continuación.

Sea G un grafo pesado completo cuyos vértices son el conjunto de oficinas O. Por cada par de vértices $o_i,\ o_j \in O$, definimos el costo de la arista $(o_i,\ o_j)$ como

$$w(o_i,\ o_j) = \begin{cases} d_{ij} \cdot U & \text{si } U \leq V \text{ y } d_{ij} < R \\ d_{ij} \cdot V & \text{si no} \end{cases}$$

donde d_{ij} es la distancia euclideana entre ambas oficinas.

Luego, de haber, al menos, una oficina con un módem, G modela una solución no mínima en la que todas las oficinas están conectadas entre si con la opción más barata disponible entre un cable UTP y uno de fibra óptica. Si $W = 1$, sigue que, para mantener a todas las oficinas conectadas y minimizar el costo empleado en los cables, basta encontrar un árbol generador mínimo de G.

Sin embargo, si $W > 1$, podemos reducir aún más el costo si, en vez de encontrar un árbol, encontramos un bosque de W componentes conexas cuyo peso sea mínimo.

Notar que este bosque se puede generar removiendo las $W - 1$ aristas de peso máximo de un árbol generador mínimo T de G. Esto se debe a que esta operación genera un grafo de $N - W$ aristas, por lo que se compone de W componentes conexas que son árboles y tiene peso mínimo ya que T es mínimo y estamos removiendo sus aristas de peso máximo.

Vamos a demostrar que lo podemos encontrar modificando el algoritmo de *Kruskal* para que termine en la iteración $N - W$.

1.2. Demostración de correctitud.

1.3. El algoritmo.

Para ello, proponemos el siguiente algoritmo.

[4]Notar que una oficina tendrá acceso a internet si y sólo si tiene un módem o si está conectada a una oficina con acceso a internet.

1.4. Complejidad temporal y espacial.

2. Evaluación empírica