Problema K-minimum spanning tree

Roberto Juan Cayro Cuadros, Gabriel Alexander Valdivia Medina, Giulia Alexa Naval Fernández, Rodrigo Alonso Torres Sotomayor

Universidad Católica San Pablo

Abstract

Keywords:

---

1. Introducción al problema

2. Demostración NP-completo

No es posible suponer la naturaleza del problema, y establecer que es NP-hard o NP-completo, sin la evidencia correspondiente, para probrarlo este debe pertenecer a NP, ademas que un problema NP-completo pueda reducirse al mismo.

2.1. Demostrar que k-MST $\in$ NP

2.2. Transformación NP-completo α k-MST

2.2.1. Steiner problem

Es un problema NP-completo de los 21 problemas de Karp, usado en problemas de optimización y mayormente enfocado en estructuras de grafos aunque tambien visto en aplicaciones de modelación de redes con más de 2 terminales. El problema consiste en que, dado un grafo no-dirigido de aristas con peso, generar un arbol dado un *Sub-set* de vertices los cuales formarán este arbol. Además pueden añadirse nuevos vertices para lograr las conexiones entre estos, llamados *Steiner-vertices*. El objetivo del problema será crear un arbol de menor peso posible, tomando en cuenta los pesos, y los vertices dados. Los vertices deberán ser exactamente los dados en el Sub-set, pero podrán ser añadidos los Steiner-vertices fueran necesarios

2.3. Transformación

Steiner-problem

ENTRADA:

- Grafo no-dirigido G con aristas de peso.
- Sub-set de vertices S.

SALIDA:

- Arbol de menor peso con los vertices de S y los Steiner-vertices si fueran necesarios.

k-MST

ENTRADA:

- Grafo no-dirigido G con aristas de peso.
- numero k

SALIDA:

- Arbol de menor peso con k-vertices y k-1 aristas.

2.3.1. Reducción

Dada la entrada G para Steiner, se puede tomar el mismo grafo para k-MST, puesto que tiene las aristas pesadas y un número determinado de vertices. Aún debe aplicarse una transformación polinomial, de modo que usaremos su complemento cuyas aristas serán de peso igual a size of sub-set, de esta forma aseguramos la transformación y no afectara la salida porque siempre busaremos el arbol de menor peso , se usará el tamaño del sub-set de vertices siendo este igual a k.

Por consiguiente logramos reducir el problema de Steiner tree a k-mst, al tranformar su entrada en los parámetros de k-MST. Pero resta confirmar que la solución de k-MST también podría solucionar a Steiner-tree. La salida de k-mst será un arbol con el menor peso posible, sin embargo, debemos tener en cuenta que el número k denota la cantidad de vertices necesarios más no que vertices, por lo tanto el numero total de permutaciones de los vertices de G serán combinaciones de el número total de vertices(n) en k vertices:

$$Total\ de\ permutaciones(Tn) = \binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

Esto es importante porque sabiendo que k es el size of sub-set (S) , y hablamos de todas las permutaciones de ese tamaño, podemos concluir que el S debe estar incluido en Tn. Por ello:

$$S \subseteq Tn$$

3. Algoritmo de fuerza bruta

4. Algoritmo aproximado

Bibliography