

# 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*

---

## Resumen

---

### 1. Introducción al problema

El k-MST o k-minimum spanning tree problem, árbol de expansión de peso mínimo k en español es un problema computacional que pide un árbol de mínimo costo con exactamente  $k$  vértices que forme un subgrafo del grafo original. [? ]

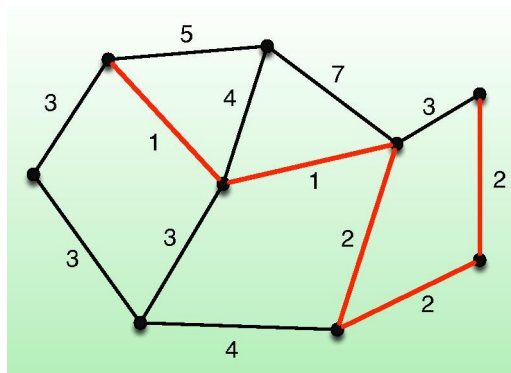


Figura 1: 6-MST del grafo G. Fuente: Wikipedia Commons

### 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 probarlo este debe pertenecer a NP, además que un problema NP-completo pueda reducirse al mismo.

2.1. *Demostrar que  $k\text{-MST} \in NP$*

2.2. *Transformación NP-completo  $\alpha$   $k\text{-MST}$*

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 también 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 árbol dado un *Sub-set* de vértices los cuales formarán este árbol. Además, pueden añadirse nuevos vértices para lograr las conexiones entre estos, llamados *Steiner-vertices*.

El objetivo del problema será crear un árbol de menor peso posible, tomando en cuenta los pesos, y los vértices dados. Los vértices deberán ser exactamente los dados en el Sub-set, pero podrán ser añadidos los Steiner-vertices fueran necesarios. La principal diferencia con el  $k\text{-MST}$  es que aquí recibimos un conjunto específico de vectores para conformar nuestro árbol, pudiendo usar vértices fuera de la relación para conectarlos. El  $k\text{-MST}$  no recibe esta relación, sólo el número de vértices exactos que necesita.

2.3. *Transformación*

*Steiner-problem*

*Entrada:*

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

*Salida:*

- Árbol de menor peso con los vértices de  $S$  y los Steiner-vertices si fueran necesarios.

*k-MST*

*Entrada:*

- Grafo no-dirigido  $G$  con aristas de peso.
- Número  $k$

*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:

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

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  $T_n$ . Por ello:

$$S \subseteq T_n$$

Sin embargo, calcular todas las permutaciones de una cantidad  $n$  de elementos es un proceso con una complejidad  $O(n!)$ , que no entra dentro de complejidad polinomial. Es necesario otro tratamiento para que el k-MST opere con los vértices que el algoritmo Steiner pide. Otra idea es añadir un árbol con aristas de peso 0 en cada vértice que pertenezca a la relación de vértices de Steiner. De esta forma, el k-MST utilizará estos vértices sí o sí como parte de su solución. Calculamos  $t$  (número de vértices en cada uno de estos árboles de arista 0) como  $t = n - r - 1$ , siendo  $n$  el número de vértices totales y  $r$  el tamaño del sub-set de Steiner.

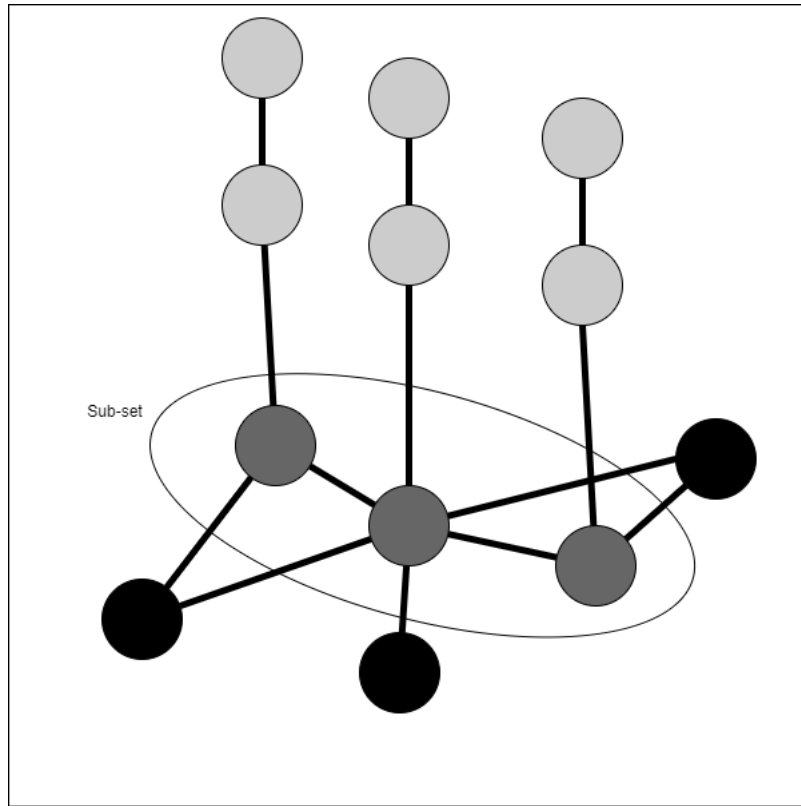


Figura 2: Transformación de la entrada de Steiner a entrada de k-MST. Elaboración propia.

3. Algoritmo de fuerza bruta

4. Algoritmo aproximado

## Bibliography

## Referencias

[1] [https://en.wikipedia.org/wiki/K-minimum\\_spanning\\_tree](https://en.wikipedia.org/wiki/K-minimum_spanning_tree)