Construct a minimum spanning tree covering a specific subset of the vertices

Asked 12 years, 2 months ago Modified 6 years, 4 months ago Viewed 13k times



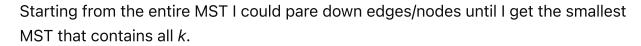
I have an undirected, positive-edge-weight graph (V,E) for which I want a minimum spanning tree covering a subset k of vertices V (the Steiner tree problem).





I'm not limiting the size of the spanning tree to k vertices; rather I know exactly which k vertices must be included in the MST.







I can use Prim's algorithm to get the entire MST, and start deleting edges/nodes while the MST of subset k is not destroyed; alternatively I can use Floyd-Warshall to get allpairs shortest paths and somehow union the paths. Are there better ways to approach this?

algorithm graph-theory graph-algorithm tree

Share Improve this question

Follow

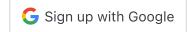
edited Aug 7, 2017 at 10:18

asked Oct 7, 2011 at 9:21



Join Stack Overflow to find the best answer to your technical question, help others answer theirs.

Sign up with email



Sign up with GitHub

Sign up with Facebook



- 4 If I remove the unwanted vertices I might also lose intermediate edges that connect k vertices that are far apart. For example if I have: k--o--o--k where o represents an unnecessary vertex and k represents one I need, if I deleted the middle o there would be no way to construct the MST between my k vertices. − rxmnnxfpvg Oct 7, 2011 at 9:32 ✓
- So you interested in the minimum spanning tree, which doesn't necessarily span all vertices, only the vertices in k? aioobe Oct 7, 2011 at 9:35
- Exactly. The MST that includes all of k at least, and then as little else as possible.
 rxmnnxfpvg Oct 7, 2011 at 9:36
- Hi could you solve your problem? If possible can you help with the pseudo code/code? I have similar problem but the graph is unweighted. phoenix Mar 14, 2015 at 12:13
- The question is unclear about whether k is a number or a set. Will you please clarify? Palec Dec 31, 2015 at 10:34

3 Answers

Sorted by:

Highest score (default)





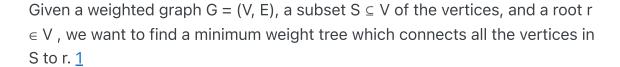
There's a lot of confusion going on here. Based on what the OP says:

25

I'm not limiting the size of the spanning tree to k vertices; rather I know exactly which *k* vertices must be included in the MST.



This is the Steiner tree problem on graphs. This is not the k-MST problem. The Steiner tree problem is defined as such:



As others have mentionned, this problem is NP-hard. Therefore, you can use an approximation algorithm.

Early/Simple Approximation Algorithms

Two famous methods are **Takahashi's method** and **Kruskal's method** (both of which

Join Stack Overflow to find the best answer to your technical question, help others answer theirs.





- Kruskal JB: On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. In Proceedings of the American Mathematical Society, Volume 7.; 1956:48–50.
- Rayward-Smith VJ, Clare A: On finding Steiner vertices. Networks 1986, 16:283–294.

Shortest path approximation by Takahashi (with modification by Rayward-Smith)

```
INPUT: a graph G = (V, E), set of terminals S = \{s_1, ..., s_k\} \subseteq V, number of repeats r
OUTPUT: a Steiner tree constructed from G
for r times do
     choose a random terminal s_1 \in S
     construct a sub-graph G' = (s_1, \{\})
     t := 1
     while t \ll |S| do
                   determine terminal s_{t+1} \notin G', which is closest to any node in G'
                   add s_{t+1} and shortest path P joining s_{t+1} with G' to G'
                   t := t + 1
      end while
     construct a minimum spanning tree T_r induced from the nodes and edges in G'
      remove non-terminals of degree 1 from T_{\mu}
end for
\hat{r} := \arg\min_{r} |T_r|
return T.
```

Kruskal's approximation algorithm (with modification by Rayward-Smith)

```
INPUT: a graph G = (V, E) with a terminal set S = \{s_1, ..., s_k\} \subseteq V

OUTPUT: a Steiner tree constructed from G

construct a forest F of k sub-graphs f_1, ..., f_k consisting of one terminal each.

while does not exist a f_i \in F such that all terminals s_1, ..., s_k \in f_i do

For all i \neq j: determine the shortest path between all nodes in f_i to all those in f_j find the minimum length path P among all computed paths from the last step construct f_n = f_i \cup f_i \cup P and add it to forest F
```

Join Stack Overflow to find the best answer to your technical question, help others answer theirs.





Modern/More Advanced Approximation Algorithms

In biology, more recent approaches have treated the problem using the cavity method, which has led to a "modified belief propagation" method that has shown good accuracy on large data sets:

- Bayati, M., Borgs, C., Braunstein, A., Chayes, J., Ramezanpour, A., Zecchina, R.: Statistical mechanics of steiner trees. Phys. Rev. Lett. 101(3), 037208 (2008) 15.
- For an application: Steiner tree methods for optimal sub-network identification: an empirical study. BMC Bioinformatics. BMC Bioinformatics 2013 30;14:144. Epub 2013 Apr 30.

In the context of search engine problems, approaches have focused on efficiency for very large data sets that can be pre-processed to some degree.

- G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S. Sudarshan. Keyword Searching and Browsing in Databases using BANKS. In ICDE, pages 431–440.
- G. Kasneci, M. Ramanath, M. Sozio, F. M. Suchanek, and G. Weikum. STAR: Steiner-tree approximation in relationship graphs. In ICDE'09, pages 868–879, 2009

Share Improve this answer Follow

user2398029 6 749 8 49 81

Thank you so much for this. This post led me to a nice R implementation in the SteinerNet package – Jeff Bezos Apr 18, 2020 at 3:58 ▶



The problem you stated is a famous NP-hard problem, called <u>Steiner tree in graphs</u>. There are no known solutions in polynomial time and many believe no such solutions exist.



Share Improve this answer Follow

edited Dec 31, 2015 at 10:11
Palec
12.9k 8 69 138

answered Jan 24, 2012 at 2:56 meh 135 1 2

Join Stack Overflow to find the best answer to your technical question, help others answer theirs.





@Palec Actually, that is wrong. "I'm not limiting the size of the spanning tree to k vertices; rather I know exactly which k vertices must be included in the MST." This problem is the Steiner tree problem. – user2398029 Jan 28, 2016 at 1:26

3 Also, -1 to @meh because the fact that the problem is NP-hard doesn't mean we can't get useful solutions with approximation algorithms. This answer does not help the OP in solving his problem. – user2398029 Jan 28, 2016 at 1:54



Run Prim's algorithm on the restricted graph (k, E') where $E' = \{(x, y) \in V : x \in k \text{ and } y \in k\}$). Constructing that graph takes O(|E|).



Share Improve this answer Follow







This might work alright some of the time, but it's not even guaranteed that the E' is connected -- and even if it is, it might be possible to save arbitrarily much distance by introducing a Steiner point (i.e., a vertex not in k). (Less than "arbitrarily much" if the distances obey the Triangle Inequality, but nothing says they have to.) – j_random_hacker Dec 21, 2015 at 14:17

@j_random_hacker interested in posting an alternative solution? – user2398029 Dec 25, 2015 at 5:46

@user2398029: I upvoted meh's answer (and I don't know why "Bill the Lizard" deleted adi's much earlier answer saying mostly the same thing). Basically this is an NP-hard problem to solve optimally; if you google "Steiner tree approximation" you can probably get some OK algorithms. – j_random_hacker Dec 25, 2015 at 14:30

@user2398029: It might be helpful to look at chapter 3 of this link from adi's answer: cc.gatech.edu/fac/Vijay.Vazirani/book.pdf. (I (re)post this here since I can see deleted posts, but I'm not sure what the rep cutoff is for that.) – j_random_hacker Dec 25, 2015 at 14:33

Join Stack Overflow to find the best answer to your technical question, help others answer theirs.



