

Relative Merits of Minimum Cost Spanning Trees and Steiner Trees

G. Anandhi¹, S. K. Srivatsa²

¹Research Scholar, Vels University, Chennai – 117, India

²Senior Professor, St. Joseph's College of Engg , Chennai – 117, India

Abstract: This paper discusses about the concepts of Minimum cost spanning tree and what happens if it is used in place of Steiner tree. We show that the MSTP is equivalent to a Steiner Tree Problem (STP) in an adequate layered graph. We also adapt the proposed approach for the Diameter-Constrained Minimum Spanning Tree Problem (DMSTP). Although the HMSTP and the DMSTP have symmetric edge costs, we will focus on so-called directed formulations.

Keywords: HMSTP (Minimum Spanning Tree Problem), Subtrees, MBST (Minimum Bottleneck Spanning Tree), DMSTP (Diameter-Constrained Minimum Spanning Tree).

1. Introduction

The Hop-constrained Minimum Spanning Tree Problem (HMSTP) is defined as follows:

Given a graph $G = (V, E)$ with node set $V = \{0, 1, \dots, n\}$ and edge set E as well as a positive cost c_e associated with each edge e of E and a natural number H , we wish to find a spanning tree T of the graph with minimum total cost and such that the unique path from a specified root node, node 0, to any other node has no more than H hops (edges).[1] Diameter-constrained Minimum Spanning

Tree Problem (DMSTP) is defined as follows: given a graph $G = (V, E)$ with node set $V = \{1, \dots, n\}$ and edge set E as well as a positive cost c_e associated with each edge e of E and a natural number D , we wish to find a spanning tree T of the graph with minimum total cost and such that the unique path from any node i to any node j has no more than D hops (edges).[2]

A single graph can have many different spanning trees. A minimum spanning tree (MST) or minimum weight spanning tree is then a spanning tree with weight less than or equal to the weight of every other spanning tree. Any undirected graph has a minimum spanning forest, which is a union of minimum spanning trees for its connected components. A spanning tree for that graph would be a subset of those paths that has no cycles but still connects to every house. There might be several spanning trees possible. A minimum spanning tree would be one with the lowest total cost.

The DCMST problem in directed networks constructs the spanning tree $T(s)$ rooted at s that has minimum total cost among all possible spanning trees rooted at s which have a maximum end-to-end delay less than or equal to a given delay constraint D . The same problem can be expressed as a decision problem as follows.

Delay-Constrained Minimum Spanning Tree (DCMST) Problem: Given a directed network $G = (V, E)$, a positive cost $c(e)$ for each $e \in E$, a positive delay $d(e)$ for each $e \in E$,

a source node $s \in V$, a positive delay constraint D , and a positive value B , is there a spanning tree $T(s)$ that satisfies:
 $\text{Cost}(T(s)) \leq B$,
 $\text{Max_Delay}(T(s)) \leq \Delta$?

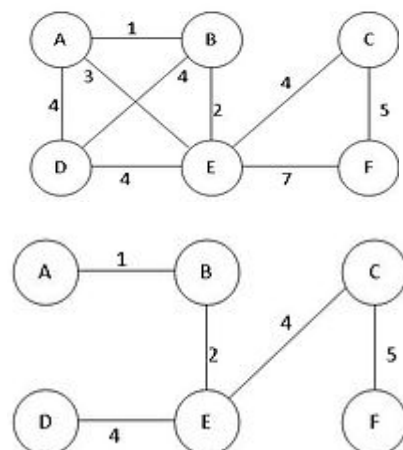
Theorem 1 DCMST is NP-complete unless all link costs are equal.

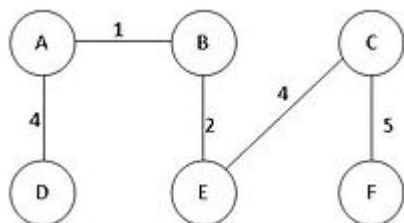
Proof: The DCMST problem in undirected networks (DCMST-undirected) is a restricted version of DCMST. DCMST-undirected is NP-complete and therefore DCMST is also NP-complete. [1] We propose a simple and efficient heuristic for the DCMST problem to avoid the exponentially growing execution times of the optimal solutions. We call it the bounded delay broadcasting (BDB) heuristic.

1.1 Steinerized Minimum Spanning Tree Algorithm

Minimum Spanning Tree T may not be a feasible solution for Steiner tree problem for minimal steiner points. Since some edges in T may have longer length. To make feasible solution we add steiner points in each edge, break them into smaller pieces each having certain length. Resulting tree is steinerized Minimum Spanning Tree.

1.2 Possible Multiplicity





This figure1 shows there may be more than one minimum spanning tree in a graph. In the figure, the two trees below the graph are two possibilities of minimum spanning tree of the given graph. *There may be several minimum spanning trees of the same weight having a minimum number of edges*; in particular, if all the edge weights of a given graph are the same, then every spanning tree of that graph is minimum. If there are n vertices in the graph, then each tree has $n-1$ edges.

1.3 Uniqueness

If each edge has a distinct weight then there will be only one, unique minimum spanning tree. This can be proved by induction or contradiction. This generalizes to spanning forests as well. If the edge weights are not unique, only the (multi-)set of weights in minimum spanning trees is unique, that is the same for all minimum spanning trees. A proof of uniqueness by contradiction is as follows.

1. Assume MST A is not unique.
2. There is another spanning tree with equal weight, say MST B.
3. Let e_1 be an edge that is in A but not in B.
4. As B is a MST, $\{e_1\} \cup B$ must contain a cycle C.
5. Then B should include at least one edge e_2 that is not in A and lies on C.
6. Assume the weight of e_1 is less than that of e_2 .
7. Replace e_2 with e_1 in B yields the spanning tree $\{e_1\} \cup B - \{e_2\}$ which has a smaller weight compared to B.
8. Contradiction. As we assumed B is a MST but it is not.

If the weight of e_1 is larger than that of e_2 , a similar argument involving tree $\{e_2\} \cup A - \{e_1\}$ also leads to a contradiction. Thus, we conclude that the assumption that there can be a second MST was false.

1.4 Minimum-cost subgraph

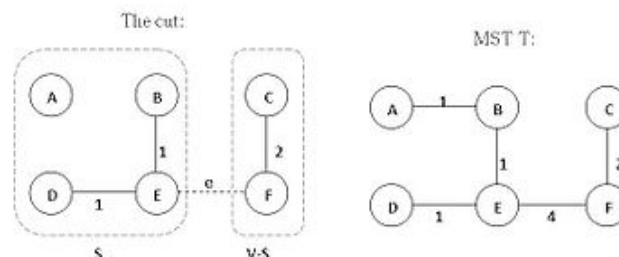
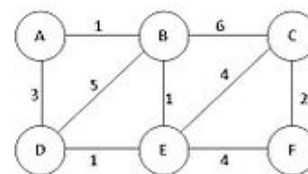
If the weights are positive, then a minimum spanning tree is in fact a minimum cost subgraph connecting all vertices, since subgraphs containing cycles necessarily have more total weight.

1.5 Cycle Property

For any cycle C in the graph, if the weight of an edge e of C is larger than the weights of all other edges of C, then this edge cannot belong to an MST. Assuming the contrary, i.e. that e belongs to an MST T1, then deleting e will break T1 into two subtrees with the two ends of e in different subtrees. The remainder of C reconnects the subtrees, hence there is an edge f of C with ends in different subtrees, i.e., it reconnects the subtrees into a tree T2 with weight less than

that of T1, because the weight of f is less than the weight of e .

1.6 Cut Property



This figure2 shows the cut property of MSP. T is the only MST of the given graph. If $S = \{A, B, D, E\}$, thus $V-S = \{C, F\}$, then there are 3 possibilities of the edge across the cut $(S, V-S)$, they are edges BC, EC, EF of the original graph. Then, e is one of the minimum-weight-edge for the cut, therefore $S \cup \{e\}$ is part of the MST T. For any cut C in the graph, if the weight of an edge e of C is strictly smaller than the weights of all other edges of C, then this edge belongs to all MSTs of the graph. To prove this, assume the contrary: in the figure at right, make edge BC (weight 6) part of the MST T instead of edge e (weight 4). Adding e to T will produce a cycle, while replacing BC with e would produce MST of smaller weight. Thus, a tree containing BC is not a MST, a contradiction that violates our assumption. By a similar argument, if more than one edge is of minimum weight across a cut, then each such edge is contained in a minimum spanning tree.

1.7 Minimum-Cost Edge

If the edge of a graph with the minimum cost e is unique, then this edge is included in any MST. Indeed, if e was not included in the MST, removing any of the (larger cost) edges in the cycle formed after adding e to the MST would yield a spanning tree of smaller weight.

2. Algorithms

There are now two algorithms commonly used Prim's algorithm and Kruskal's algorithm. All three are greedy algorithms that run in polynomial time, so the problem of finding such trees is in FP, and related decision problems such as determining whether a particular edge is in the MST or determining if the minimum total weight exceeds a certain value are in P. Another greedy algorithm not as commonly used is the reverse-delete algorithm, which is the reverse of Kruskal's algorithm.

If the edge weights are integers, then deterministic algorithms are known that solve the problem in $O(m + n)$ integer operations. In a comparison model, in which the only allowed operations on edge weights are pairwise comparisons, found a linear time randomized algorithm based on a combination of Borůvka's algorithm and the

reverse-delete algorithm. Whether the problem can be solved deterministically in linear time by a comparison-based algorithm remains an open question, however. The fastest non-randomized comparison-based algorithm with known complexity, by Bernard Chazelle, is based on the soft heap, an approximate priority queue. Its running time is $O(m \alpha(m,n))$, where m is the number of edges, n is the number of vertices and α is the classical functional inverse of the Ackermann function. The function α grows extremely slowly, so that for all practical purposes it may be considered a constant no greater than 4; thus Chazelle's algorithm takes very close to linear time. Research has also considered parallel algorithms for the minimum spanning tree problem. With a linear number of processors it is possible to solve the problem in $O(\log n)$ time. Minimum Spanning Forest (MSF) algorithm for undirected weighted graphs. This algorithm leverages Prim's algorithm in a parallel fashion, concurrently expanding several subsets of the computed MSF. PMA minimizes the communication among different processors by not constraining the local growth of a processor's computed subtree. In effect, PMA achieves a scalability that previous approaches lacked. PMA, in practice, outperforms the previous state-of-the-art GPU-based MSF algorithm, while being several order of magnitude faster than sequential CPU-based algorithms.

Other specialized algorithms have been designed for computing minimum spanning trees of a graph so large that most of it must be stored on disk at all times. They rely on efficient external storage sorting algorithms and on graph contraction techniques for reducing the graph's size efficiently.

The problem can also be approached in a distributed manner. If each node is considered a computer and no node knows anything except its own connected links, one can still calculate the distributed minimum spanning tree [2].

3. Applications

Minimum spanning trees have direct applications in the design of networks, including computer networks, telecommunications networks, transportation networks, water supply networks, and electrical grids (which they were first invented for, as mentioned above). They are invoked as subroutines in algorithms for other problems, including the Christofides algorithm for approximating the travelling salesman problem, approximating the multi-terminal minimum cut problem (which is equivalent in the single-terminal case to the maximum flow problem), and approximating the minimum-cost weighted perfect matching.

Other practical applications based on minimal spanning trees include:

- Taxonomy, one of the earliest motivating applications.
- Cluster analysis: clustering points in the plane, single-linkage clustering (a method of hierarchical clustering), graph-theoretic clustering, and clustering gene expression data.
- Constructing trees for broadcasting in computer networks.
- Image registration and segmentation [see minimum spanning tree-based segmentation.

- Curvilinear feature extraction in computer vision.
- Handwriting recognition of mathematical expressions.
- Circuit design: implementing efficient multiple constant multiplications, as used in finite impulse response filters.
- Regionalisation of socio-geographic areas, the grouping of areas into homogeneous, contiguous regions.
- Comparing ecotoxicology data.
- Topological observability in power systems.
- Measuring homogeneity of two-dimensional materials.
- Minimax process control.

In pedagogical contexts, minimum spanning tree algorithms serve as a common introductory example of both graph algorithms and greedy algorithms due to their simplicity.

4. Related Problems

A related problem is the k -minimum spanning tree (k -MST), which is the tree that spans some subset of k vertices in the graph with minimum weight. A set of k -smallest spanning trees is a subset of k spanning trees (out of all possible spanning trees) such that no spanning tree outside the subset has smaller weight. (Note that this problem is unrelated to the k -minimum spanning tree.) The Euclidean minimum spanning tree is a spanning tree of a graph with edge weights corresponding to the Euclidean distance between vertices which are points in the plane (or space).

The rectilinear minimum spanning tree is a spanning tree of a graph with edge weights corresponding to the rectilinear distance between vertices which are points in the plane (or space). In the distributed model, where each node is considered a computer and no node knows anything except its own connected links, one can consider distributed minimum spanning tree. Mathematical definition of the problem is the same but has different approaches for solution. The capacitated minimum spanning tree is a tree that has a marked node (origin, or root) and each of the subtrees attached to the node contains no more than a c nodes. c is called a tree capacity. Solving CMST optimally requires exponential time, but good heuristics such as Esau-Williams and Sharma produce solutions close to optimal in polynomial time. The degree constrained minimum spanning tree is a minimum spanning tree in which each vertex is connected to no more than d other vertices, for some given number d . The case $d = 2$ is a special case of the travelling salesman problem, so the degree constrained minimum spanning tree is NP-hard in general. For directed graphs, the minimum spanning tree problem is called the Arborescence problem and can be solved in quadratic time using the Chu-Liu/Edmonds algorithm.

A maximum spanning tree is a spanning tree with weight greater than or equal to the weight of every other spanning tree. Such a tree can be found with algorithms such as Prim's or Kruskal's after multiplying the edge weights by -1 and solving the MST problem on the new graph. A path in the maximum spanning tree is the widest path in the graph between its two endpoints: among all possible paths, it maximizes the weight of the minimum-weight edge. Maximum spanning trees find applications in parsing algorithms for natural languages and in training algorithms for conditional random fields. The dynamic MST problem

concerns the update of a previously computed MST after an edge weight change in the original graph or the insertion/deletion of a vertex. [3]

5. The Diameter Constrained Spanning Tree Problem

In this section we adapt the previous methods to a variation of the HMSTP, the Diameter constrained Spanning Tree problem (DMSTP). Given a prescribed graph $G = (V, E)$ with node set V and edge set E as well as a positive cost c_e associated with each edge e of E , we wish to find a minimal spanning tree with a bound D on the diameter of the tree, which is the maximum number of edges in any of its paths.[2] When $D = 2$ or 3 , the problem is easy to solve. However, it is NP-Hard when $D \geq 4$. As noted before, the DMSTP differs from the HMSTP in the sense that here we constrain the path between each pair of nodes while in the HMSTP; only the paths from the special node are constrained. This observation suggests that the DMSTP appears to be much more complex than the HMSTP.

However, several approaches for the DMSTP (see, for instance, [2, 3, 4, 5]) have used the properties of tree centers in order to transform the DMSTP into special versions of the HMSTP. For instance, with respect to situations with parameter D even then, following center property proves to be very useful.

6. Minimum bottleneck spanning tree

A bottleneck edge is the highest weighted edge in a spanning tree. A spanning tree is a minimum bottleneck spanning tree (or MBST) if the graph does not contain a spanning tree with a smaller bottleneck edge weight. A MST is necessarily a MBST (provable by the cut property), but a MBST is not necessarily a MST.

7. Conclusions

We formulated the problem as a DCMST problem in directed networks, and then we proved that this problem is NP-complete. We also proved that the graph acquired with the help of spanning tree is always better than the steiner trees.

References

- [1] Steiner Tree Problems in Computer Networks, Dingzhu Du, Xiadong Hu
- [2] Modelling Hop-Constrained and Diameter-Constrained Minimum Spanning Tree Problems as Steiner Tree Problems over Layered Graphs, Luis Gouveia, Luidi Simonetti, Eduardo Uchoa – March 2008
- [3] N.R.. Achuthan, L. Caccetta, P. A. Caccetta, and J. F. Geelen. Computational methods for the diameter restricted minimum weight spanning tree problem. *Australasian Journal of Combinatorics*, 10:51–71, 1994.
- [4] L. Gouveia and T. L. Magnanti. Network flow models for designing diameter-constrained minimum-spanning and Steiner trees. *Networks*, 41(3):159–173, 2003.
- [5] C.. Santos, A. Lucena, and C. C. Ribeiro. Solving diameter constrained minimum spanning tree problems in dense graphs. In *Experimental and Efficient Algorithms*, volume 3059/2004 of *Lecture Notes in Computer Science*, pages 458–467. Springer, Berlin, 2004.
- [6] M. Gruber and G. R. Raidl. A new 0-1 ILP approach for the bounded diameter minimum spanning tree problem. In *Proceedings of the 2nd International Network Optimization Conference*, pages 178–185, Lisbon, 2005.
- [7] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.
- [8] N. R. Achuthan, L. Caccetta, P. A. Caccetta, and J. F. Geelen. Computational methods for the diameter restricted minimum weight spanning tree problem. *Australasian Journal of Combinatorics*, 10:51–71, 1994.
- [9] G. Dahl. The 2-hop spanning tree problem. *Operations Research Letters*, 23:21–26, 1998.
- [10] G. Dahl, L. Gouveia, and C. Requejo. On formulations and methods for the hop-constrained minimum spanning tree problem. In P. Pardalos and M. Resende, editors, *Handbook of Optimization in Telecommunications*, pages 493–515. Springer, 2006.
- [11] L. Gouveia. Multicommodity flow models for spanning trees with hop constraints. *European Journal of Operational Research*, 95(1):178–190, 1996.
- [12] L. Gouveia. Using variable redefinition for computing lower bounds for minimum spanning and Steiner trees with hop constraints. *INFORMS Journal on Computing*, 10(2):180–188, 1998.
- [13] T. F. Noronha, A. C.Santos, and C. C. Ribeiro. Constraint programming for the diameter constrained minimum spanning tree problem. In *Proceedings of IV Latin-American Algorithms, Graphs and Optimization Symposium*, *Electronic Notes in Discrete Mathematics*, Puerto Varas, Chile, 2008.