Approximating k-hop Minimum Spanning Trees in Euclidean Metrics

Sören Laue* Domagoj Matijević*

Abstract

In the minimum-cost k-hop spanning tree (k-hop MST) problem, we are given a set S of n points in a metric space, a positive small integer k and a root point $r \in S$. We are interested in computing a rooted spanning tree of minimum cost such that the longest root-leaf path in the tree has at most k edges. We present a polynomial-time approximation scheme for the plane. Our algorithm is based on Arora's et al. [5] technique for the Euclidean k-median problem.

1 Introduction

We are given set S of n points in d-dimensional Euclidean space, a fixed positive integer k and a root node $r \in S$. The k-hop spanning tree of S is a tree T rooted at r and spanning all points of S, such that number of edges on any root-leaf path is not greater than k. The cost of T is the sum of its edge lengths. In this paper we consider the k-hop spanning tree problem of minimum cost (k-hop MST).

Based on the methods of Arora et al. [5] for the Euclidean k-median problem, we present a polynomial-time approximation scheme for the k-hop MST problem in the plane, when k is a constant.

As a byproduct of our algorithm, we also provide a polynomial-time approximation scheme for the geometric versions of the following more general problems:

The multi-level concentrator location problem. Here, we are given a set S of nodes, a set $C \subset S$ of clients and a k sets of facilities $F = F_1 \cup \ldots \cup F_k \subseteq S$ with the opening facility costs f_j for each facility $j \in F$. The task is to open subsets of facilities $F'_i \subseteq F_i$, $1 \leq i \leq k$ and assign each client to the closest level one facility in F'_1, and assign each of the level $(i-1)$ facilities to the closest level i facilities F'_i, such that the opening facilities costs plus the sum of the distances is minimized.

The bounded depth minimum Steiner tree problem. Given a set S of nodes, a set D of Steiner points and a root node $r \in S$, the task is to construct a minimum cost tree of depth k, rooted at r that spans the set S and possibly using some Steiner points from the set D.

It is not difficult to see that the k-hop MST is just a special case of the above two problems, and any solutions for them would immediately imply a solution for the k-hop MST problem.

*Max-Planck-Institut für Informatik, Saarbrücken, Germany, {soeren, dmatijev}@mpi-inf.mpg.de

Motivation

Minimum-cost spanning trees are pervasive and their efficient construction appears important in many practical applications. For example, in *multicast-routing* problem in the area of computer networks (see, e.g. [8, 7]) a number of clients and a server are connected by a common communication network. The server wishes to transmit identical information to all client nodes. Most solutions to the multicast problem involve computing a tree rooted at the server and spanning the client nodes. The server then transmits the data to its immediate children in the tree and intermediate nodes forward incoming data to their respective descendants in the tree. Tree-routing schemes allow for fast data delivery while keeping the total network load low. Kompella et al. [14] consider the problem of computing multicast-trees that minimize the overall network cost as well as the maximum transmission latency on any path in the tree connecting the server to a client node. It is not hard to see that a multi-hop transmission with too many hops will increase the latency of the communication. Thus, assuming that all links in the network have roughly the same transmission delay (which is a reasonable assumption in local area networks), limiting the number of hops in the transmission to some small integer k helps in achieving *fast* and *reliable* communication protocols.

Related Work

In the classic *metric facility location problem*, we are given a set of clients C and a set of facilities F with metric edge costs c_{ij}, for all $i \in F, j \in C$ and opening cost f_i for all facilities $i \in F$. The goal is to open a subset of facilities $F' \subseteq F$ such that the sum of opening facility costs, plus the sum of the costs of assigning each client to its closest facility in F' is minimized. The best known approximation algorithm is by Mahdian et al. [15] that achieves 1.52 approximation ratio. Note that the 2-hop MST is a special case of the facility location problem, e.g. replace each facility cost f_i by the distance from i to the root r. Thus, all the approximation results for facility location problem apply immediately to the 2-hop MST. Guha and Khuller [10] proved that the existence of a polynomial time 1.463-approximation algorithm for the metric facility location problem would imply that P = NP. This hardness result also applies for the 2-hop MST problem.

For the Euclidean facility location problem a randomized PTAS based on Arora's [3, 4] technique for the Euclidean TSP is presented in [5], for the points in the plane. Unfortunately, for d-dimensional geometric instances and $d > 2$, the algorithms runs only in quasi-polynomial time. However, Kolliopoulos and Rao [12] were able to construct a nearly linear time randomized PTAS for facility location problem for any d-dimensional Euclidean space. The small errors in this paper were fixed by the authors in [13]. Another planar subdivision, namely the guillotine subdivision, was introduced by Mitchell [16] for geometric optimization problems in the plane.

Zhang [17] gives a 1.77-approximation algorithm for the metric *two-level concentrator location problem* which is a generalization of the 3-hops MST.

The first constant factor approximation for the bounded depth steiner tree problem and likewise for the k-hop MST in general metric spaces is presented by Kantor and Peleg in [11]. They achieve an approximation ratio of $1.52 \cdot 9^{k-2}$.

Althaus et al. [2] present an approximation algorithm that computes a k-hop spanning tree in general metric spaces of total expected cost $O(\log n)$ times the cost of the optimal k-hop MST. They approximate the metric space into a tree metric using the result by Fakcharoenphol et al. [9] who showed that any metric space can be probabilistically approximated by a family of tree metrics such that the expected stretch in the cost is at most $O(\log n)$.

Clementi et al. [6] present an algorithm that computes with high probability a constant approximation for constant k for random instances in the plane.

Our Contribution

In this work we present the first PTAS for the k-hop MST problem in the plane. We extend the technique of Arora et al. [5] for the Euclidean k-median problem and show that the $(1+\varepsilon)$ solution for the k-hops MST problem can be computed in polynomial time.

In Section 2 we review the quadtree dissection from [5] and show that there exist a $(1+\varepsilon)$ solution to the k-hop MST problem with respect to the given dissection. Furthermore, in Section 3 we show how to compute such an approximate solution with a dynamic programming algorithm in time $O\left(\left(\frac{n}{\varepsilon}\right)^{O(1/\varepsilon)}\right)$.

We also extend our algorithm to the related *multi-level concentrator location problem* and *bounded depth minimum Steiner tree problem* in Section 4.

2 Preliminaries

In this part we describe the quadtree dissection from [5] and show the existence of approximately optimal solutions with a simple structure based on a given dissection. Let S denote a set of n points in the plane. The *bounding box* is the smallest axis-aligned square that contains all points of S. In the following, we assume that the bounding box of the points has side-length $L = n/\varepsilon$ and all points of S lie on gridpoints of the unit grid defined on the bounding box. Note that the cost increase of the optimum is negligible since moving each point to the closest grid point will increase the minimum cost k-hop MST by at most $\varepsilon \cdot OPT$.

A *dissection* of a square is a recursive partition of the square into lower level squares. More precisely, we view the dissection as a hierarchical decomposition of the plane into squares/boxes. A box in a dissection is any square that can be obtained by a recursive splitting process that starts with the bounding box and generally splits an existing dissection box by 2 axis-orthogonal lines passing through its center into 4 identical subboxes. Such a decomposition naturally defines a 4-ary tree. Each line is assigned a level. There are 2^i level i lines that partition level i boxes into level $i+1$ boxes. The *size* of a box is its side length. A nice property of the dissection boxes is that any two boxes either have disjoint interiors or one is contained inside the other. Note that there are $O(L^2)$ nodes in the tree and its depth is $\log L = O(\log(n/\varepsilon))$.

We randomize the levels in the dissection of the bounding box the same way as in [3, 4, 5]. Namely, randomly pick two integers $0 \le a, b < L$. The (a,b)-shift of the dissection is defined by shifting x and y coordinates of all lines by a and b respectively, and then reducing modulo L.

In other words, the middle vertical line of the dissection is moved from the x-coordinate $L/2$ to the x-coordinate $a + (L/2)\text{mod}L$, and the middle horizontal line from the y-coordinate $L/2$ to the y-coordinate $b + (L/2)\text{mod}L$. The rest of the dissection is then wrapped-around, so that the left edge of the dissection comes to rest at the x-coordinate a, and the lower edge of the dissection comes to rest at the y-coordinate b. Note that we treat a wrapped-around square in the shifted dissection as a single region.

Note that the solution to the k-hop MST problem consists of collections of line segments. We will only allow the segments to bend and pass through a set of prespecified points called *portals*. More precisely, place $2^i m$ equally spaced portals on each level i line. Moreover, at the corner of each dissection box place a portal. Note that each level $i+1$ box in the dissection has m portals on its two level $i+1$ edges and strictly less than m portals on its two level i edges. In general, any box in the dissection has at most $4m$ portals.

A solution to the k-hop MST problem is called *portal respecting* if it crosses a dissection box only at portals.

Suppose we are given the optimal set of line segments that describe an optimal k-hop MST solution. To make the solution portal respecting, we need to deflect each edge that crosses a side of a box in the dissection to the nearest portal. Note that if the size of the box is l, we need to deflect each edge by at most l/m to make it pass through a portal.

Since the shifts a and b are chosen randomly, we have that the probability that each vertical/horizontal line l in the grid is from the level i: $Pr[l \text{ is at level } i] = 2^i/L$. Using this fact, Arora et al. [5] show the following Structure theorem:

Lemma 1 *For any collection of line segments, random shifts a and b and $m \ge 1$, the bending process will, with probability at least 1/2, deflect the segments by at most $O(\log L/m)$ times the sum of the length of the line segments.*

Since the above Lemma 1 holds for any set of line segments, it also implies the following:

Corollary 2 *Let $r \in S$ denote the root node and let shifts a and b be chosen uniformly at random. Let $m = O\left(\frac{\log(n/\varepsilon)}{\varepsilon}\right)$ for any $\varepsilon > 0$. Then, with probability of at least $1/2$ the cost of the optimal portal respecting solution for the k-hop MST problem is at most $(1+\varepsilon)\cdot OPT$, where OPT denotes the optimal cost of the k-hop MST.*

3 The Algorithm

In this section we will describe the algorithm to compute an optimal portal respecting k-hop MST which is with probability of at least $1/2$ a $(1+\varepsilon)$-approximation to the optimal k-hop MST.

Consider any optimal k-hop MST. We assign each node a level depending on the number of hops to the root r, where r is assigned level 0, its immediate neighbors are assigned 1 and so on. We also assign levels to the edges. An edge from a level $i-1$ node to a level i node is assigned the level i. Hence, we have nodes from level 0 to k and edges from level 1 to k.

Consider now a box in the dissection as described in the previous section. Remember that edges are only allowed to cross the boundary of the box at portals. The optimal solution inside the box is fully determined if we know for each portal and each level i the distance from the portal to the nearest node of level i outside the box. Conversely, the optimal solution outside this box is fully determined if we know for each portal and each level i the distance from the portal to the nearest node of level i inside this box.

Hence, if we fix all distances at the portals of a box to all nearest nodes of levels 0 to $k-1$, only the solution inside this box with minimal cost can be part of an optimal solution. This enables us to design the following dynamic program.

We store in the table

$$\text{Table}(B, \text{inside}_0, \ldots, \text{inside}_{k-1}, \text{outside}_0, \ldots, \text{outside}_{k-1})$$

the cheapest solution for box B that respects the given inside and outside function, where inside_i denotes the distance function on the portals to the closest node of level i inside box B. outside_i is defined analogously. In other words inside_i describes what box B can provide to the outside and outside_i describes what can be provided to box B from outside. For the distance function inside_i we can still allow an additional additive error of l/m as the distance between two neighboring portals is already l/m. Remember, that the size of box B is l and we have placed m portals on its boundary. Thus, we have $\text{inside}_i(p) \in \{0, l/m, 2l/m, \ldots, 2l, \infty\}$ for a portal p and two neighboring portals differ by at most l/m. We assign ∞ as a value for $\text{inside}_i(p)$, if no node of level i is inside the corresponding box. Hence, we have at most $2m \cdot 3^{4m}$ possible assignments per box for each inside_i function.

A slightly different reasoning holds for the outside_i functions. Here, the maximal distance from a portal to an outside node can be at most $2L$. Again, we can allow an additional additive error of l/m. Hence, we have $\text{outside}_i(p) \in \{0, l/m, 2l/m, \ldots, 2L, \infty\}$. This sums up to at most $2Lm/l$ different values and at most $2Lm/l \cdot 3^{4m}$ possible assignments per box for each outside_i function. This could be reduced by making the gap between two consecutive values larger as the distance becomes larger, since for larger distances we anyway have a larger additional error due to a larger interportal distance, but we omit this here. In total we have $T = 4Lm^2 \cdot 3^{8mk}$ entries in table Table per box B.

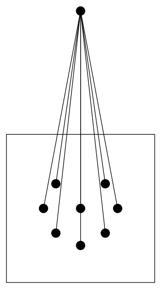
Figure 1: All nodes actually lie on top of each other and the edges pass through one portal.

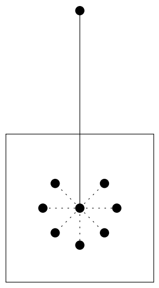
Figure 2: All nodes actually lie on top of each other and the dotted lines have length 0.

Computing the table

We compute the table bottom up. There are two different base cases:

1. The root r is inside the box B. We set

$$\text{Table}(B, \text{inside}_0, \ldots, \text{inside}_{k-1}, \text{outside}_0, \ldots, \text{outside}_{k-1})$$

to cost 0 if

1. $\text{inside}_0(p)$ is at least the distance from each portal p to the root r, and
2. inside_i is ∞ for $i \geq 1$.

2. The box B contains at least one node but no root.

Note, that all nodes lie in the center of box B and thus on top of each other due to the initial perturbation. If $\text{inside}_i(p)$ is at least the distance between the nodes in the box and each portal p for all p then we connect a node q to the portal p' such that $\text{dist}(q, p') + \text{outside}_{i-1}(p')$ is minimal among all portals of this box. We store this cost in the corresponding Table entry. If however, all $\text{inside}(p)$ is ∞, i.e. this box does not provide any reachable node to the outside, we have to distinguish two cases. In the first case it is cheaper to connect all nodes to a level $k-1$ node as depicted in figure 1. In the second case, it is cheaper to connect one node q to a node of level at most $k-2$ and then connecting all other nodes inside the box to this node q as in figure 2. Which case we have can be determined by looking at the corresponding outside_i functions. We store the cost in the corresponding Table entry.

If we are not in the base case, the entry of

$$\text{Table}(B, \text{inside}_0, \ldots, \text{inside}_{k-1}, \text{outside}_0, \ldots, \text{outside}_{k-1})$$

can be computed from the corresponding table entries

$$\text{Table}(B_j, \text{inside}_0^{(j)}, \ldots, \text{inside}_{k-1}^{(j)}\ \text{outside}_0^{(j)}, \ldots, \text{outside}_{k-1}^{(j)}),$$

for $1 \leq j \leq 4$, where B_1, B_2, B_3 and B_4 are the four sub-boxes of B. Once all inside$_i$ and outside$_i$ functions are fixed we go through all possible inside$_i^{(j)}$ and outside$_i^{(j)}$ functions that comply with distance functions of box B. As we only have approximate distances stored we again introduce an additive error of at most l/m per line segment. However, this error is at most the error that occurs while making an edge portal respecting for this box and hence, can be neglected here. We sum up the corresponding costs for B_1, B_2, B_3 and B_4 and store the minimal in the corresponding

$$\text{Table}(B, \text{inside}_0, \ldots, \text{inside}_{k-1}, \text{outside}_0, \ldots, \text{outside}_{k-1})$$

entry. The time spend per box then amounts to $O(T^5)$

As there are L^2 boxes in the dissection the total running time amounts to $O(L^2 \cdot T^5) = O\left(\left(\frac{n}{\varepsilon}\right)^{O(1/\varepsilon)}\right)$.

We conclude with the main theorem

Theorem 3 *The k-hop minimum spanning tree problem in the Euclidean plane admits a polynomial time approximation scheme.*

4 Generalizations

The bounded depth minimum Steiner tree problem

Our approach easily generalizes to the shallow Steiner tree problem. Here, one is also allowed to use Steiner points in the bounded-hop MST. We just have to change the base case in our algorithm. If we only have Steiner points inside a box we have two options. Either use the Steiner point or do not use it. This can be easily decided upon the distance functions on the portals.

The multi-level concentrator location problem

If we assign levels to the Steiner points and also opening costs for using a Steiner point we are left with the multi-level concentrator location problem. This problem can also be solved using our approach. We just have to add the opening cost to the corresponding Table entry. For the initial perturbation it suffices to have a lower bound on the optimal cost which is polynomial in the number of nodes n. The corresponding k-level facility location problem obviously is an n-approximation. Aardal et al. [1] showed how to compute a 3-approximation for this problem. Hence, the initial bounding square has size $L = 3n^2/\varepsilon$ and the running time adapts accordingly.

5 Conclusions and Open Problems

We provided the first polynomial time approximation scheme for the k-hop minimum spanning tree and related problems in the plane. The algorithm follows along the lines of Arora et al. [5]. Thus, the algorithm can be generalized to higher dimensions but with only quasi-polynomial running time. It would be interesting to find a PTAS also for higher dimensions.

References

[1] Karen Aardal, Fabian A. Chudak, and David B. Shmoys. A 3-approximation algorithm for the k-level uncapacitated facility location problem. *Inf. Process. Lett.*, 72(5-6):161–167, 1999.

[2] Ernst Althaus, Stefan Funke, Sariel Har-Peled, Jochen Könemann, Edgar A. Ramos, and Martin Skutella. Approximating k-hop minimum-spanning trees. *Operations Research Letters*, 33(2):115–120, March 2005.

[3] Sanjeev Arora. Polynomial time approximation schemes for euclidean tsp and other geometric problems. In *FOCS '96*, page 2, 1996.

[4] Sanjeev Arora. Nearly linear time approximation schemes for euclidean tsp and other geometric problems. In *FOCS '97*, page 554, 1997.

[5] Sanjeev Arora, Prabhakar Raghavan, and Satish Rao. Approximation schemes for euclidean k-medians and related problems. In *STOC '98*, pages 106–113, 1998.

[6] Andrea E. F. Clementi, Miriam Di Ianni, Angelo Monti, Massimo Lauria, Gianluca Rossi, and Riccardo Silvestri. Divide and conquer is almost optimal for the bounded-hop mst problem on random euclidean instances. In *SIROCCO*, pages 89–98, 2005.

[7] Stephen Deering, Deborah Estrin, Dino Farinacci, Van Jacobson, Ching-Gung Liu, and Liming Wei. An architecture for wide-area multicast routing. In *SIGCOMM '94*, pages 126–135, 1994.

[8] Stephen E. Deering and David R. Cheriton. Multicast routing in datagram internetworks and extended lans. *ACM Trans. Comput. Syst.*, 8(2):85–110, 1990.

[9] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *STOC '03*, pages 448–455, 2003.

[10] Sudipto Guha and Samir Khuller. Greedy strikes back: improved facility location algorithms. *J. Algorithms*, 31(1):228–248, 1999.

[11] Erez Kantor and David Peleg. Approximate hierarchical facility location and applications to the shallow steiner tree and range assignment problems. In *CIAC*, pages 211–222, 2006.

[12] Stavros G. Kolliopoulos and Satish Rao. A nearly linear-time approximation scheme for the euclidean kappa-median problem. In *ESA '99*, pages 378–389, 1999.

[13] Stavros G. Kolliopoulos and Satish Rao. A nearly linear-time approximation scheme for the euclidean k-median problem. submitted for journal (`http://cgi.di.uoa.gr/ sgk/papers/kmedian-3rd.pdf`), 2006.

[14] Vachaspathi P. Kompella, Joseph C. Pasquale, and George C. Polyzos. Multicast routing for multimedia communication. *IEEE/ACM Trans. Netw.*, 1(3):286–292, 1993.

[15] Mohammad Mahdian, Yinyu Ye, and Jiawei Zhang. Improved approximation algorithms for metric facility location problems. In *APPROX '02*, pages 229–242, 2002.

[16] Joseph S. B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric TSP, k-MST, and related problems. *SIAM Journal on Computing*, 28(4):1298–1309, 1999.

[17] Jiawei Zhang. Approximating the two-level facility location problem via a quasi-greedy approach. In *SODA '04*, pages 808–817, 2004.