ARTICLE IN PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Operations Research Letters ▮▮▮ (▮▮▮▮) ▮▮▮–▮▮▮

Operations Research Letters

www.elsevier.com/locate/dsw

Approximating k-hop minimum-spanning trees☆

Ernst Althaus[a,1], Stefan Funke[b,2], Sariel Har-Peled[c], Jochen Könemann[d,3], Edgar A. Ramos[c], Martin Skutella[b,*]

[a] *Université Henri Poincaré, LORIA, B.P. 239, F-54506 Vandoeuvre-lès-Nancy, France*

[b] *Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany*

[c] *University of Illinois at Urbana-Champaign, USA*

[d] *Department of Combinatorics and Optimization, University of Waterloo, 200 University Avenue West, Waterloo, Canada ON N2L 3G1*

Received 14 April 2004; accepted 12 May 2004

Abstract

Given a complete graph on n nodes with metric edge costs, the *minimum-cost k-hop spanning tree* (kHMST) problem asks for a spanning tree of minimum total cost such that the longest root-leaf-path in the tree has at most k edges. We present an algorithm that computes such a tree of total expected cost $O(\log n)$ times that of a minimum-cost k-hop spanning-tree.

Keywords: Approximation algorithms; Minimum spanning trees; Depth restriction; Metric space approximation

1. Introduction

We are given an undirected complete graph G on a set of n vertices V, a non-negative metric cost function c on the edges, an integer $k > 0$, and a root node $r \in V$. We say that a rooted tree T is a *k-hop spanning tree of G* if the number of edges on any root-leaf-path is at most k. In this paper we consider the kHMST problem where the goal is to compute a minimum-cost k-hop spanning tree that is rooted at node r.

Minimum-cost spanning trees are pervasive in many practical applications. To name just one example, consider the *multicast-routing* problem arising in computer networks (see, e.g., [6,7]) where 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 spanning the server and 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

☆ We would like to thank an anonymous referee for many helpful comments and pointers to the literature. This work was supported in part by the IST Program of the EU under contract numbers IST-1999-14186 (ALCOM-FT) and IST-2001-30012 (APPOL II).

* Corresponding author.

E-mail addresses: althaus@loria.fr (E. Althaus), funke@mpi-sb.mpg.de (S. Funke), sariel@cs.uiuc.edu (S. Har-Peled), jochen@uwaterloo.ca (J. Könemann), eramosn@cs.uiuc.edu (E.A. Ramos), skutella@mpi-sb.mpg.de (M. Skutella).

[1] This work was done when the author was at Max-Planck-Institut für Informatik, Saarbrücken, Germany.

[2] This work was done while the second author was visiting the University of Illinois at Urbana-Champaign.

[3] This work was done while visiting Max-Planck-Institut für Informatik, Saarbrücken, Germany.

0167-6377/$ - see front matter
doi:10.1016/j.orl.2004.05.005

tree. Tree-routing schemes allow for fast data delivery while keeping the total network load low.

Kompella et al. [12] consider the problem of computing multicast-trees that minimize the overall network cost as well as the maximum transmission delay on any path in the tree connecting the server to a client node. Assuming that all links in the network have roughly the same transmission delay (which is a reasonable assumption in local area networks), it is not hard to see that this problem can be cast in the kHMST framework.

1.1. Related work

A related problem is the so-called *bounded-diameter minimum spanning tree problem* where the goal is to compute a spanning tree of minimum cost such that the number of edges on any path in the tree is bounded by some given number. We note that an approximation algorithm for the kHMST problem implies a solution to the $2k$- and $(2k+1)$-bounded-diameter spanning tree problem of same approximation ratio by applying the kHMST-algorithm a polynomial number of times.

Marathe et al. [15] consider the following generalization of the bounded-diameter minimum spanning tree problem. In the *bounded-diameter minimum-cost Steiner tree problem* (BDST) we are given an undirected graph $G=(V,E)$, a subset of vertices $\mathscr{V} \subseteq V$, non-negative costs c_e (not necessarily metric) and non-negative lengths ℓ_e for all edges $e \in E$, and a parameter $D>0$. A feasible solution consists of a tree spanning $\mathscr{V}$ whose longest path has length at most D. The authors give an algorithm that computes a tree T of total cost $O(\log n)$ times that of a minimum-cost feasible solution to BDST. They also prove that the length of any path in T is at most $O(D \log n)$. For the case that all $\ell_e = 1$, Kortsarz and Peleg [13] show how to obtain a solution of cost $O(|\mathscr{V}|^{\varepsilon})$ times the optimum solution for an arbitrary fixed parameter $\varepsilon \in (0,1)$. The resulting Steiner tree has diameter at most D. For constant D, Kortsarz and Peleg obtain an $O(\log n)$-approximation.

For the bounded-diameter minimum spanning tree (also under non-metric cost function) Bar-Ilan et al. [3] achieve an $O(\log n)$-approximation algorithm for the special case of this problem where $D \in \{4,5\}$. The authors also show that there is no $o(\log n)$-approximation algorithm for this problem, unless $\mathtt{P} = \mathtt{NP}$. Alfandari and Paschos [2] present a $\frac{5}{4}$-approximation for the 2-hop case where all edge costs are within $\{1,2\}$.

Hassin and Levin [11] introduce the following more general hop-constrained spanning tree problem: Given an undirected graph $G=(V,E)$, costs c_e for all edges $e \in E$, and a symmetric requirement matrix $(u_{ij}) \in \mathbb{N}^{n \times n}$; the goal is to find a spanning tree T in G such that for all $i,j \in V$, the unique i-j-path in T has at most u_{ij} edges. The authors consider the special case of this problem where $u_{ij} \in \{1,2,\infty\}$, for all $i,j \in V$, and present a constant factor approximation algorithm for this case.

The problem of computing diameter-constrained trees has also been studied empirically. We point the reader to a recent paper by Gouveia and Magnanti [9] and the references therein.

Closely related is the *metric facility location problem* (MFL): We are given a bipartite graph $G=(F \cup C, E)$ with metric edge costs c_{ij}, for all $i \in F$, $j \in C$, and opening costs f_i for all facilities $i \in F$. The goal is to open certain facilities $F' \subset F$ such that the sum of the costs of the facilities in F' plus the sum of the costs of assigning each client to its closest facility in F' is minimized. Guha and Khuller [10] prove that no approximation algorithm with performance guarantee better than 1.463 can exist, unless $\mathtt{NP} \subseteq \mathtt{DTIME}(n^{O(\log\log n)})$. On the other hand, Mahdian et al. [14] give a 1.52-approximation for this problem. It can be seen that MFL is a generalization of 2HMST, nevertheless the hardness results from [10] also apply to the 2HMST problem.

In the *k-level metric facility location problem* we have a set D of demand points where clients are located. Moreover, we have pairwise disjoint sets $F_1, \ldots, F_k$ of potential locations for facilities at levels $1, \ldots, k$. Each demand point needs to be assigned to one path visiting open facilities at levels $1, \ldots, k$ in this order. The service cost for this demand point is the length of the chosen path in a given metric. There is a cost for opening a facility depending on the location. The objective is to minimize the total opening cost plus the sum of service costs. Aardal et al. [1] present a 3-approximation algorithm for that problem. In a very recent paper Zhang [16] presented a 1.77-approximation for the case $k=2$.

There is an optimum solution to the k-level metric facility location problem which is a forest of k-hop trees rooted at level-k facilities. Notice, however, that the contribution of an edge to the cost of a solution depends on the number of demand points using that edge.

Zhang [16] also gives a 1.77-approximation algorithm for the 2-*level concentrator location problem* which is a generalization of the 3HMST problem. Here, in contrast to the 2-level facility location problem, the cost of an edge has to be paid only once even if more than one customer wants to use that edge. The result of Zhang thus implies a 1.77-approximation for the 3HMST problem.

1.2. Our contributions

Our main theorem can be stated as follows:

Theorem 1. *Given an undirected complete graph G on vertex set V, a metric cost function c, and an integer parameter $k > 0$; there is a randomized algorithm that computes a feasible k-hop spanning tree T of G whose expected cost is* $\mathrm{O}(\log n)\mathtt{opt}_k$ *where* $\mathtt{opt}_k$ *denotes the minimum-cost of any k-hop spanning tree of G. The running time of this algorithm is* $\mathrm{O}(n^5 k)$.

Our algorithm first uses a recent result by Fakcharoenphol et al. [8] who show that any metric space can be probabilistically approximated by a family of tree metrics. Their result is based on the notion of *hierarchically well-separated trees* which is due to Bartal [4,5].

Definition 1. Let H be a tree that is rooted at r and let c^H be a cost function on the edges of H. Then H is said to be an ℓ-hierarchically well-separated tree (ℓ-HST) for a parameter $\ell > 1$ if there exists $C > 0$ such that the cost of all edges e of level h is equal to $c_e^H = C/\ell^h$, for all $h \geqslant 0$.

For any two leaves u, v of H, we denote by P_{uv}^H the unique u-v-path in H and by $c_{uv}^H = c(P_{uv}^H)$ the sum of the edge costs of the path. Fakcharoenphol et al. prove the following statement:

Theorem 2. *Given a metric (V, c), there is a distribution $\mathscr{H}$ over metrics induced by* 2-HST*s with the following properties: For any two nodes $u, v \in V$,*

(i) $c_{uv} \leqslant c_{uv}^H$ *for any metric $(H, c^H) \in \mathscr{H}$, and*
(ii) $E_{\mathscr{H}}[c_{uv}^H] = \mathrm{O}(\log n) c_{uv}$.

The elements $(H, c^H) \in \mathscr{H}$ have the additional property that the nodes of the original metric space (V, c) are the leaves of H and every leaf of H has the same level. Finally there is an efficient algorithm to sample from $\mathscr{H}$.

In the bulk of this paper we develop an exact algorithm for the kHMST problem in the special case of cost-functions c that are induced by a 2-HST. An application of Theorem 2 then yields Theorem 1.

2. Computing minimum-cost k-hop trees in HSTs

We assume throughout the rest of the paper that we are given a complete graph G with vertex set $V = \{1, 2, \ldots, n\}$ and a cost function $c = c^H$ that is induced by a metric space (H, c^H) where H is a 2-HST, i.e., $c_{uv} = c_{uv}^H = c(P_{uv}^H)$, for all $u, v \in V$. Moreover, we assume a fixed embedding of H such that the leaves are labeled $1, 2, \ldots, n$ from left to right. For $1 \leqslant i \leqslant j \leqslant n$, let $[i, j] := \{i, \ldots, j\}$.

Observation 1. *For $1 \leqslant h < i < j \leqslant n$, it holds that* $c_{hj} \geqslant \max\{c_{hi}, c_{ij}\}$.

Proof. By definition of the cost function c and Definition 1, the cost coefficient c_{hj} only depends on the level of the lowest common ancestor of the leaves h, j in the 2-HST H. Since the lowest common ancestor of h, j is also an ancestor of i (it is either the lowest common ancestor of h, i or the lowest common ancestor of i, j, or both), the result follows. □

Our goal is to compute a minimum-cost k-hop spanning tree in G rooted at r. In fact, the procedure described below works for an arbitrary cost function c for which Observation 1 is true.

The following lemma states that it is possible to limit the search for an optimum solution to those k-hop trees where each subtree consists of consecutive vertices in the ordering $1, \ldots, n$. This insight is essential for the dynamic programming formulation presented below.

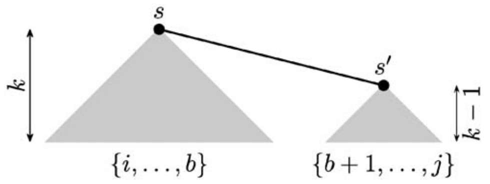

Fig. 1. The figure explains the structural result described in part (i) of Lemma 1. An optimum k-hop tree splits the set of vertices $\{i,\dots,j\}$ into two subsets $\{i,\dots,b\}$ and $\{b+1,\dots,j\}$.

Lemma 1. *For $1 \leqslant i \leqslant s \leqslant j \leqslant n$ and $k > 0$, there exists a minimum-cost k-hop spanning tree T with root s covering all vertices in $[i,j]$ which satisfies the following conditions*:

(i) *If $s < j$ and $s' > s$ is the largest (rightmost) child of s in T, then there exists a vertex $b \in [s, s'-1]$ such that the subtree of T rooted at s' spans the interval $[b+1, j]$.*

(ii) *If $s > i$ and $s' < s$ is the smallest (leftmost) child of s in T, then there exists a vertex $b \in [s', s-1]$ such that the subtree of T rooted at s' spans the interval $[i, b]$.*

Fig. 1 depicts the situation discussed in part (i) of the lemma.

Proof. To simplify notation, we assume in the following that all edges in a rooted tree are directed away from the root. Among all minimum-cost k-hop spanning trees T with root s covering the vertices in $[i,j]$, choose one minimizing the second objective function $\sum_{(u,v)\in T} |u-v|$. We show that this optimum k-hop tree T fulfills requirement (i). Using symmetric arguments, one can analogously show that it fulfills requirement (ii). For a vertex v in T, let $d(v)$ be the number of edges on the unique s-v-path in T. We assume that $s < j$; otherwise, we are done. Let $s' > s$ be the rightmost child of s and denote the subtree of T rooted at s' by T'.

We first argue that the subtree T' contains all vertices in $[s', j]$. By contradiction, assume that there is a vertex in $[s', j]$ which is not contained in T'. Among all such vertices v, choose one of minimum depth $d(v)$. Then, there exists an edge (u,v) in T with $u < s' < v$. Since s' is the rightmost child of s, we get $u \neq s$ and thus $d(u) \geqslant d(s')$. We can therefore replace edge (u,v) with edge (s',v) without increasing cost (by Observation 1) and without increasing the depth of any vertex in T. Notice, however, that this replacement decreases the second objective function and therefore leads to a contradiction.

Similarly, we argue that the subtree T' does *not* contain a vertex from $[i,s]$. By contradiction, choose a vertex v violating this condition with minimum $d(v)$. Then there exists an edge (u,v) in T' with $v < s < u$ which can be replaced with edge (s,v). By Observation 1, this leads to a contradiction.

It remains to consider the vertices between s and s'. Assume by contradiction that there exist two vertices v, v' with $s < v < v' < s'$ such that v but not v' is contained in the subtree T'. Among all such pairs v, v', choose one minimizing $d(v) + d(v')$. Then there exist two edges (u,v) and (u',v') with $u' < v < v' < u$. We distinguish three cases (see also Fig. 2): (i) If $d(u) = d(u')$, we replace edges (u,v) and (u',v') with (u,v') and (u',v). (ii) If $d(u) < d(u')$, then $d(v) \leqslant d(u')$ and we can replace (u',v') with (v,v'). (iii) If $d(u) > d(u')$, then $d(u) \geqslant d(v')$ and we can replace (u,v) with (v',v). By Observation 1, all three cases yield a contradiction. □

For $1 \leqslant i \leqslant s \leqslant j \leqslant n$, let $C[s,i,j,k]$ denote the cost of a minimum k-hop spanning tree rooted at s covering all vertices in $[i,j]$. By definition,

$$C[s,i,j,k] = \begin{cases} 0 & \text{if } s = i = j, \\ \infty & \text{if } k = 0 \text{ and } i < j. \end{cases}$$

By Lemma 1, $C[s,1,n,k]$ can be computed recursively using the following dynamic programming formulation. For $i < j$ and $k > 0$,

$$C[s,i,j,k] = \min_{s \neq s' \in [i,j]} \begin{cases} \min_{b \in [s,s'-1]} & C[s,i,b,k] + c_{ss'} & \text{if } s < s', \\ & +C[s', b+1, j, k-1] & \\ \min_{b \in [s',s-1]} & C[s',i,b,k-1] + c_{ss'} & \text{if } s' < s. \\ & +C[s,b+1,j,k] & \end{cases}$$

In order to gain some intuition for this expression, assume that the minimum on the right-hand side is assumed for $s' > s$. In this case, the inner minimum can be thought of as deciding on the rightmost child s' of s and the *split position* b such that the elements in $[i,b]$ end up in the remaining subtree of s and the elements

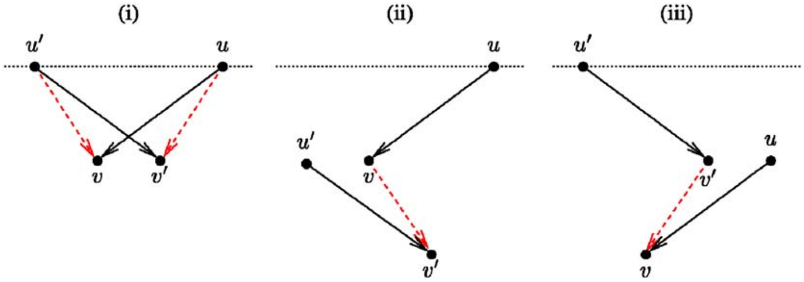

Fig. 2. An illustration of the three cases occurring in the proof of Lemma 1.

in $[b+1, j]$ in the subtree of s'; the latter must have depth $k-1$, though; see also Fig. 1.

The correctness of our dynamic programming formulation follows from Lemma 1. Furthermore, observe that we can partially order the entries of the dynamic programming table according to the tuple $(k, |i-j|)$ and determining the content of a cell of this table only requires the knowledge of *smaller* cells with respect to that order. We conclude with a proof of Theorem 1.

Proof of Theorem 1. Consider a minimum-cost k-hop spanning tree T^*. Due to linearity of expectation, its expected cost under the tree metric (H, c^H) is $\mathrm{O}(\log n)\mathtt{opt}_k$ and we actually compute an optimal spanning tree under (H, c^H). For the running time, observe that for the dynamic program we have to fill a table of size n^3k and determining the content of a cell takes time $\mathrm{O}(n^2)$. Constructing the tree metric (H, c^H) only takes $\mathrm{O}(n^2)$ time. □

3. Conclusion and open problems

A number of unresolved issues remain: It follows from [10] that the kHMST problem is likely not approximable within arbitrarily small constants. Is there a constant factor approximation for the kHMST problem with metric cost functions? Moreover, the negative result in [13] shows that no approximation algorithm for the kHMST problem with arbitrary cost-functions is likely to have a performance ratio of $o(\log n)$. However, all hardness proofs rely on strict bounds on the number of hops of the output tree. In the spirit of [15] we can ask : Is there an algorithm for the kHMST problem with arbitrary non-negative cost-functions that computes $\mathrm{O}(k)$-hop trees with cost $\mathrm{O}(\mathtt{opt}_k)$?

References

[1] K.I. Aardal, F.A. Chudak, D.B. Shmoys, Improved approximation algorithms for the k-level facility location problem, Inform. Process. Lett. 72 (1999) 161–167.
[2] L. Alfandari, V.T. Paschos, Approximating minimum spanning tree of depth two, Int. Trans. Oper. Res. 6 (1999) 607–622.
[3] J. Bar-Ilan, G. Kortsarz, D. Peleg, Generalized submodular cover problems and applications, Theoret. Comput. Sci. 250 (1–2) (2001) 179–200.
[4] Y. Bartal, Probabilistic approximation of metric spaces and its algorithmic applications, in: Proceedings, IEEE Symposium on Foundations of Computer Science, 1996, pp. 184–193.
[5] Y. Bartal, On approximating arbitrary metrics by tree metrics, in: Proceedings, ACM Symposium on Theory of Computing, 1998, pp. 161–168.
[6] S.E. Deering, D.R. Cheriton, Multicast routing in datagram internetworks and extended LANs, ACM Trans. Comput. Syst. 8 (2) (1990) 85.
[7] S. Deering, D. Estrin, D. Farinacci, An architecture for wide-area multicast routing, in: Proceedings of SIGCOMM, 1994.
[8] J. Fakcharoenphol, S. Rao, K. Talwar, A tight bound on approximating arbitrary metrics by tree metrics, in: Proceedings, ACM Symposium on Theory of Computing, 2003, pp. 448–455.
[9] L. Gouveia, T.L. Magnanti, Network flow models for designing diameter-constrained minimum-spanning and steiner trees, Networks 41 (2003) 159–173.
[10] S. Guha, S. Khuller, Greedy strikes back: Improved facility location algorithms, J. Algorithms 31 (1999) 228–248.
[11] R. Hassin, A. Levin, Minimum spanning tree with hop restrictions, J. Algorithms 48 (2003) 220–238.
[12] V.P. Kompella, J.C. Pasquale, G.C. Polyzos, Multicast routing for multimedia communication, IEEE/ACM Trans. Networks 1 (3) (1993) 286–292.

[13] G. Kortsarz, D. Peleg, Approximating the weight of shallow steiner trees, DAMATH: Discrete Applied Mathematics and Combinatorial Operations Research and Computer Science, Vol. 93, 1999.

[14] M. Mahdian, Y. Ye, J. Zhang, A 1.52-approximation algorithm for the uncapacitated facility location problem, in: Proceedings, International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, 2002, pp. 127–137.

[15] M.V. Marathe, R. Ravi, R. Sundaram, S.S. Ravi, D.J. Rosenkrantz, H.B. Hunt III, Bicriteria network design problems, J. Algorithms 28 (1) (1998) 142–171.

[16] J. Zhang, Approximating the two-level facility location problem via a quasi-greedy approach, Proceedings of the 15th ACM-SIAM Symposium on Discrete Algorithms (SODA), 2004.