Multi-Level Steiner Trees

Reyan Ahmed
University of Arizona, Tucson, United States

Patrizio Angelini
Universität Tübingen, Tübingen, Germany
https://orcid.org/0000-0002-7602-1524

Faryad Darabi Sahneh
University of Arizona, Tucson, United
https://orcid.org/0000-0003-2521-4331

Alon Efrat
University of Arizona, Tucson, United States

David Glickenstein
University of Arizona, Tucson, United States

Martin Gronemann
Universität zu Köln, Cologne, Germany

Niklas Heinsohn
Universität Tübingen, Tübingen, Germany

Stephen G. Kobourov
University of Arizona, Tucson, United States
https://orcid.org/0000-0002-0477-2724

Richard Spence
University of Arizona, Tucson, United States
rcspence@email.arizona.edu

Joseph Watkins
University of Arizona, Tucson, United States

Alexander Wolff
Universität Würzburg, Würzburg, Germany
https://orcid.org/0000-0001-5872-718X

Abstract

In the classical Steiner tree problem, one is given an undirected, connected graph $G = (V, E)$ with non-negative edge costs and a set of *terminals* $T \subseteq V$. The objective is to find a minimum-cost edge set $E' \subseteq E$ that spans the terminals. The problem is APX-hard; the best known approximation algorithm has a ratio of $\rho = \ln(4) + \varepsilon < 1.39$. In this paper, we study a natural generalization, the *multi-level Steiner tree* (MLST) problem: given a nested sequence of terminals $T_1 \subset \cdots \subset T_k \subseteq V$, compute nested edge sets $E_1 \subseteq \cdots \subseteq E_k \subseteq E$ that span the corresponding terminal sets with minimum total cost.

The MLST problem and variants thereof have been studied under names such as Quality-of-Service Multicast tree, Grade-of-Service Steiner tree, and Multi-Tier tree. Several approximation results are known. We first present two natural heuristics with approximation factor $O(k)$. Based on these, we introduce a composite algorithm that requires 2^k Steiner tree computations. We determine its approximation ratio by solving a linear program. We then present a method that guarantees the same approximation ratio and needs at most $2k$ Steiner tree computations. We compare five algorithms experimentally on several classes of graphs using four types of graph generators. We also implemented an integer linear program for MLST to provide ground truth.

17th International Symposium on Experimental Algorithms (SEA 2018).
Editor: Gianlorenzo D'Angelo; Article No. 15; pp. 15:1–15:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Our combined algorithm outperforms the others both in theory and in practice when the number of levels is small ($k \leq 22$), which works well for applications such as designing multi-level infrastructure or network visualization.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms, Theory of computation → Approximation algorithms analysis, Theory of computation → Routing and network design problems

Keywords and phrases Approximation algorithm, Steiner tree, multi-level graph representation

Digital Object Identifier 10.4230/LIPIcs.SEA.2018.15

Related Version An extended version of this manuscript is available at arXiv:1804.02627 [1].

Funding This work is partially supported by NSF grants CCF-1423411 and CCF-1712119.

Acknowledgements The authors wish to thank the organizers and participants of the First Heiligkreuztal Workshop on Graph and Network Visualization for the stimulating atmosphere.

1 Introduction

Let $G = (V, E)$ be an undirected, connected graph with non-negative edge costs $c\colon E \to \mathbb{R}^+$, and let $T \subseteq V$ be a set of vertices called *terminals*. A *Steiner tree* is a tree in G that spans T. The *network (graph) Steiner tree problem* (ST) is to find a minimum-cost Steiner tree $E' \subseteq E$, where the cost of E' is $c(E') = \sum_{e \in E'} c(e)$. ST is one of Karp's initial NP-hard problems [13]; see also a survey [23], an online compendium [12], and a textbook [20].

Due to its practical importance in many domains, there is a long history of exact and approximation algorithms for the problem. The classical 2-approximation algorithm for ST [11] uses the *metric closure* of G, i.e., the complete edge-weighted graph G^* with vertex set T in which, for every edge uv, the cost of uv equals the length of a shortest u–v path in G. A minimum spanning tree of G^* corresponds to a 2-approximate Steiner tree in G.

Currently, the last in a long list of improvements is the LP-based approximation algorithm of Byrka et al. [6], which has a ratio of $\ln(4) + \varepsilon < 1.39$. Their algorithm uses a new iterative randomized rounding technique. Note that ST is APX-hard [5]; more concretely, it is NP-hard to approximate the problem within a factor of 96/95 [8]. This is in contrast to the geometric variant of the problem, where terminals correspond to points in the Euclidean or rectilinear plane. Both variants admit polynomial-time approximation schemes (PTAS) [2, 16], while this is not true for the general metric case [5].

In this paper, we consider a natural generalization of ST where the terminals appear on "levels" and must be connected by edges of appropriate levels. We propose new approximation algorithms and compare them to existing ones both theoretically and experimentally.

▶ **Definition 1** (Multi-Level Steiner Tree (MLST) Problem)**.** Given a connected, undirected graph $G = (V, E)$ with edge weights $c\colon E \to \mathbb{R}^+$ and k nested terminal sets $T_1 \subset \cdots \subset T_k \subseteq V$, a *multi-level Steiner tree* consists of k nested edge sets $E_1 \subseteq \cdots \subseteq E_k \subseteq E$ such that E_1 spans T_1, …, E_k spans T_k. The cost of an MLST is defined by $c(E_1) + c(E_2) + \cdots + c(E_k)$. The MLST problem is to find an MLST $E_{\mathrm{OPT},1} \subseteq \cdots \subseteq E_{\mathrm{OPT},k} \subseteq E$ with minimum cost.

Since the edge sets are nested, we can also express the cost of an MLST as follows:

$$kc(E_1) + (k-1)c(E_2 \backslash E_1) + \cdots + c(E_k \backslash E_{k-1}).$$

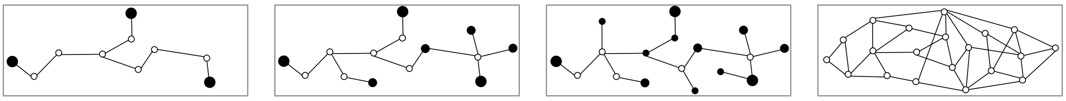


Figure 1 An illustration of a 3-level MLST for the graph at the right. Solid and open circles represent terminal and non-terminal nodes, respectively. Note that the level 1 tree (left) is contained in the level 2 tree (mid), which is in turn contained in the level 3 tree (right).

This emphasizes that the total cost $c(e)$ of an edge that appears at level ℓ is $(k-\ell+1)c(e)$.

We denote the cost of an optimal MLST by OPT. We can write

$$\mathrm{OPT} = k\mathrm{OPT}_1 + (k-1)\mathrm{OPT}_2 + \cdots + \mathrm{OPT}_k$$

where $\mathrm{OPT}_1 = c(E_{\mathrm{OPT},1})$ and $\mathrm{OPT}_\ell = c(E_{\mathrm{OPT},\ell} \setminus E_{\mathrm{OPT},\ell-1})$ for $2 \le \ell \le k$. Thus OPT_ℓ represents the cost of edges on level ℓ but not on level $\ell - 1$ in the minimum cost MLST. Figure 1 shows an example of an MLST for $k = 3$.

Applications. This problem has natural applications in designing multi-level infrastructure of low cost. Apart from this application in network design, multi-scale representations of graphs are useful in applications such as network visualization, where the goal is to represent a given graph at different levels of detail.

Previous Work. Variants of the MLST problem have been studied previously under various names, such as *Multi-Level Network Design (MLND)* [3], *Multi-Tier Tree (MTT)* [15], *Quality-of-Service (QoS) Multicast Tree* [7], and *Priority-Steiner Tree* [9].

In MLND, the vertices of the given graph are partitioned into k levels, and the task is to construct a k-level network. For $1 \le \ell \le k$, let $c^\ell(e)$ be the cost of edge e if it is in level ℓ. The vertices on each level must be connected by edges of the corresponding level or higher, and edges of higher level are more costly, that is, $0 \le c^k(e) \le \cdots \le c^1(e)$ for any edge e. The cost of an edge partition is the sum of all edge costs, and the task is to find a partition of minimum cost. Let ρ be the ratio of the best approximation algorithm for (single-level) ST, that is, currently $\rho = \ln(4) + \varepsilon < 1.39$. Balakrishnan et al. [3] gave a $4/3\rho$-approximation algorithm for 2-level MLND with proportional edge costs, that is, $c^\ell(e) = c^k(e)(k-\ell+1)$. Note that the definitions of MLND and MLST treat the bottom level differently. While MLND requires that *all* vertices are connected eventually, this is not the case for MLST. In this respect, MLST is more general than MLND, which makes it harder to approximate. On the other hand, MLND is more flexible in terms of edge costs. Whereas the Steiner tree problem is a special case of the MLST problem for $k = 1$, the same problem is a special case of MLND for $k = 2$, by setting $c^2(e) = 0$.

For MTT, which is equivalent to MLND, Mirchandani [15] presented a recursive algorithm that involves 2^k Steiner tree computations. For $k = 3$, the algorithm achieves an approximation ratio of 1.522ρ independently of the edge costs $c^1, \ldots, c^k \colon E \to \mathbb{R}^+$. For proportional edge costs, Mirchandani's analysis yields even an approximation ratio of 1.5ρ for $k = 3$. Recall, however, that this assumes $T_k = V$, and setting the edge costs on the bottom level to zero means that edge costs are *not* proportional.

In the QoS Multicast Tree problem [7] one is given a graph, a source vertex s, and a level between 1 and k for each terminal (1 meaning important). The task is to find a minimum-cost Steiner tree that connects all terminals to s. The level of an edge e in this tree is the minimum over the levels of the terminals that are connected to s via e. The cost of the edges and of the tree are as above. As a special case, Charikar et al. [7] study the *rate model*, where edge costs are proportional, and show that the problem remains NP-hard if all vertices (except the source) are terminals (at some level). Note that if we choose as source any vertex at the top level T_1, then MLST can be seen as an instance of the rate model.

Charikar et al. [7] gave a simple 4ρ-approximation algorithm for the rate model. Given an instance φ, their algorithm constructs an instance φ' where the levels of all vertices are rounded up to the nearest power of 2. Then the algorithm simply computes a Steiner tree at each level of φ' and prunes the union of these Steiner trees into a single tree. The ratio can be improved to $e\rho$, where e is the base of the natural logarithm, using randomized doubling.

Instead of taking the union of the Steiner trees on each rounded level, Karpinski et al. [14] contract them into the source in each step, which yields a 2.454ρ-approximation. They also gave a $(1.265+\varepsilon)\rho$-approximation for the 2-level case. (Since these results are not stated with respect to ρ, but depend on several Steiner tree approximation algorithms – among them the best approximation algorithm with ratio 1.549 [21] available at the time – we obtained the numbers given here by dividing their results by 1.549 and stating the factor ρ.)

For the more general Priority-Steiner Tree problem, where edge costs are not necessarily proportional, Charikar et al. [7] gave a $\min\{2\ln|T|, k\rho\}$-approximation algorithm. Chuzhoy et al. [9] showed that Priority-Steiner Tree does not admit an $O(\log\log n)$-approximation algorithm unless $\mathrm{NP} \subseteq \mathrm{DTIME}(n^{O(\log\log\log n)})$. For Euclidean MLST, Xue at al. [24] gave a recursive algorithm that uses any algorithm for Euclidean Steiner Tree (EST) as a subroutine. With a PTAS [2, 16] for EST, the approximation ratio of their algorithm is $4/3+\varepsilon$ for $k=2$ and $(5+4\sqrt{2})/7+\varepsilon \approx 1.5224+\varepsilon$ for $k=3$.

Our Contribution. We introduce and analyze two intuitive approximation algorithms for MLST – bottom-up and top-down; see Section 2.1. The bottom-up heuristic uses a Steiner tree at the bottom level for the higher levels after pruning unnecessary edges at each level. The top-down heuristic first computes a Steiner tree on the top level. Then it passes edges down from level to level until the bottom level terminals are spanned.

We then propose a composite heuristic that generalizes these and examines all possible 2^{k-1} (partial) top-down and bottom-up combinations and returns the one with the lowest cost; see Section 2.2. We propose a linear program that finds the approximation ratio of the composite heuristic for any fixed value of k. We compute the explicit approximation ratios for up to 22 levels, which turn out to be better than those of previously known algorithms. The composite heuristic requires, however, 2^k ST computations.

Therefore, we propose a procedure that achieves the same approximation ratio as the composite heuristic but needs only $2k$ ST computations. In particular, it achieves a ratio of 1.5ρ for $k=3$ levels, which settles a question posed by Karpinski et al. [14] who were asking whether the $1.5224+\varepsilon$-approximation of Xue at al. [24] can be improved for $k=3$. Note that Xue et al. treated the Euclidean case, so their ratio does not include the factor ρ. We generalize an integer linear programming (ILP) formulation for ST [19] to obtain an exact algorithm for MLST; see Section 3. We experimentally evaluate several approximation and exact algorithms on a wide range of problem instances; see Section 4. The results show that the new algorithms are also surprisingly good in practice. We conclude in Section 5.

2 Approximation Algorithms

In this section we propose several approximation algorithms for MLST. In Section 2.1, we show that the natural approach of computing edge sets either from top to bottom or vice versa, already give $O(k)$-approximations; we call these two approaches *top-down* and *bottom-up*, and denote their cost by TOP and BOT, respectively. Then, we show that running the two approaches and selecting the solution with minimum cost produces a better approximation ratio than either top-down or bottom-up.

In Section 2.2, we propose a composite approach that mixes the top-down and bottom-up approaches by solving ST on a certain subset of levels, then propagating the chosen edges to higher and lower levels in a way similar to the previous approaches. We then run the algorithm for each of the 2^{k-1} possible subsets, and select the solution with minimum cost. For relatively small values of k ($k \leq 22$), our results improve over the state of the art.

2.1 Top-Down and Bottom-Up Approaches

We present top-down and bottom-up approaches for computing approximate multi-level Steiner trees. The approaches are similar to the MST and Forward Steiner Tree (FST) heuristics by Balakrishnan et al. [3]; however, we generalize the analysis to an arbitrary number of levels.

In the top-down approach, we compute an exact or approximate Steiner tree $E_{\mathrm{TOP},1}$ spanning T_1. Then we modify the edge weights by setting $c(e) := 0$ for every edge $e \in E_{\mathrm{TOP},1}$. In the resulting graph, we compute a Steiner tree $E_{\mathrm{TOP},2}$ spanning T_2. This extends $E_{\mathrm{TOP},1}$ in a greedy way to span the terminals in T_2 not already spanned by $E_{\mathrm{TOP},1}$. Iterating this procedure for all levels yields a solution $E_{\mathrm{TOP},1} \subseteq \cdots \subseteq E_{\mathrm{TOP},k} \subseteq E$ with cost TOP.

In the bottom-up approach, we compute a Steiner tree $E_{\mathrm{BOT},k}$ spanning the terminals T_k in level k. Then, for each level ℓ, we obtain $E_{\mathrm{BOT},\ell}$ as the smallest subtree of $E_{\mathrm{BOT},k}$ that spans all the terminals in T_ℓ, giving a solution with cost BOT.

A natural approach is to run both top-down and bottom-up approaches and select the solution with minimum cost. This yields an approximation ratio better than those from top-down or bottom-up. Let $\rho \geq 1$ denote the approximation ratio for ST (that is, $\rho = 1$ corresponds to using an exact ST subroutine).

▶ **Theorem 2.** *For $k \geq 2$ levels, the top-down approach is a $\frac{k+1}{2}\rho$-approximation to MLST, the bottom-up approach is a $k\rho$-approximation, and taking the minimum of* TOP *and* BOT *is a $\frac{k+2}{3}\rho$-approximation.*

Proof. We give the proof for an arbitrary number of levels in the full version [1]; here we treat only the case $k = 2$. We have $\mathrm{OPT} = 2\mathrm{OPT}_1 + \mathrm{OPT}_2$. Let TOP be the total cost produced by the top-down approach, and let $\mathrm{TOP}_\ell = c(E_{\mathrm{TOP},\ell} \backslash E_{\mathrm{TOP},\ell-1})$ denote the cost of edges on level ℓ but not level $\ell - 1$, produced by the top-down approach, so that $\mathrm{TOP} = 2\mathrm{TOP}_1 + \mathrm{TOP}_2$. Define BOT and BOT_ℓ analogously. Let MIN_ℓ denote the cost of a minimum Steiner tree over terminals T_ℓ with original edge weights, independently of other levels, so that $\mathrm{MIN}_1 \leq \mathrm{MIN}_2 \leq \ldots \leq \mathrm{MIN}_k$.

▶ **Lemma 3.** *The following inequalities relate* TOP *with* OPT*:*

$$\mathrm{TOP}_1 \leq \rho \mathrm{OPT}_1 \tag{1}$$

$$\mathrm{TOP}_2 \leq \rho(\mathrm{OPT}_1 + \mathrm{OPT}_2) \tag{2}$$

Proof. (1) follows from the fact that $E_{\mathrm{TOP},1}$ is a ρ-approximation for ST over T_1, that is, $\mathrm{TOP}_1 \leq \rho\mathrm{MIN}_1 \leq \rho\mathrm{OPT}_1$. To show (2), note that TOP_2 is at most ρ times the cost (denote MIN_2') of a minimum Steiner tree over T_2 in the instance obtained by setting $c(e) = 0$ for each $e \in E_{\mathrm{TOP},1}$. Thus, $\mathrm{TOP}_2 \leq \rho\mathrm{MIN}_2' \leq \rho\mathrm{MIN}_2$. Additionally, since $E_{\mathrm{OPT},2}$ spans T_2 by definition, we have $\mathrm{MIN}_2 \leq \mathrm{OPT}_1 + \mathrm{OPT}_2$, so $\mathrm{TOP}_2 \leq \rho(\mathrm{OPT}_1 + \mathrm{OPT}_2)$ as desired. ◀

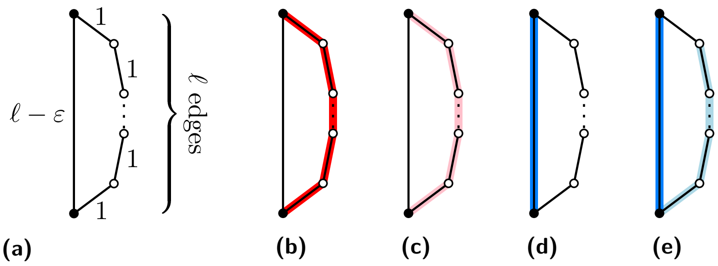


Figure 2 The analysis of the top-down approach (light and dark blue) is asymptotically tight for two layers (optimal solution in light and dark red). The dark vertices and edges are on the top level, the white vertices and light edges are on the bottom level. Here, $\mathrm{OPT} = 2\ell$, while $\mathrm{TOP} = 2(\ell - \varepsilon) + \ell - 1 = 3\ell - 2\varepsilon - 1$.

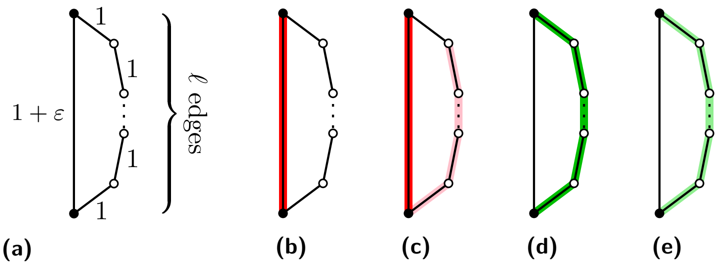


Figure 3 The analysis of the bottom-up approach (light and dark green) is asymptotically tight for two layers (optimal solution in light and dark red). Here, $\mathrm{OPT} = \ell + 1 + 2\varepsilon$, while $\mathrm{BOT} = 2\ell$.

Combining (1) and (2), we have $\mathrm{TOP} = 2\mathrm{TOP}_1 + \mathrm{TOP}_2 \leq 3\rho\mathrm{OPT}_1 + \rho\mathrm{OPT}_2 \leq 3\rho\mathrm{OPT}_1 + \frac{3}{2}\rho\mathrm{OPT}_2 = \frac{3}{2}\rho\mathrm{OPT}$, and hence the top-down approach provides a $\frac{3}{2}\rho$-approximation when $k = 2$. In Fig. 2 we provide an example showing that our analysis is tight for $\rho = 1$.

▶ **Lemma 4.** *The following inequality relates* BOT *with* OPT*:*

$$\mathrm{BOT}_1 + \mathrm{BOT}_2 \leq \rho(\mathrm{OPT}_1 + \mathrm{OPT}_2)$$

Proof. This follows from the fact that $\mathrm{BOT}_1 + \mathrm{BOT}_2 \leq \rho\mathrm{MIN}_2$, and that the tree with cost $\mathrm{OPT}_1 + \mathrm{OPT}_2$ spans T_2 with cost at least MIN_2. ◀

Hence, $\mathrm{BOT} = 2\mathrm{BOT}_1 + \mathrm{BOT}_2 \leq 2(\mathrm{BOT}_1 + \mathrm{BOT}_2) \leq 2\rho(\mathrm{OPT}_1 + \mathrm{OPT}_2) \leq 2\rho(2\mathrm{OPT}_1 + \mathrm{OPT}_2) = 2\rho\mathrm{OPT}$. Again, the approximation ratio of 2 (for $\rho = 1$) is asymptotically tight; see Figure 3.

We show that taking the better of the two solutions returned by the top-down and the bottom-up approach provides a $\frac{4}{3}\rho$-approximation to MLST for $k = 2$. To prove this, we use the fact that $\min\{x, y\} \leq \alpha x + (1 - \alpha)y$ for any real numbers x, y, and $\alpha \in [0, 1]$. Thus,

$$\begin{aligned}\min\{\mathrm{TOP}, \mathrm{BOT}\} &\leq \alpha(3\rho\,\mathrm{OPT}_1 + \rho\,\mathrm{OPT}_2) + (1 - \alpha)(2\rho\,\mathrm{OPT}_1 + 2\rho\,\mathrm{OPT}_2)\\ &= (2 + \alpha)\rho\,\mathrm{OPT}_1 + (2 - \alpha)\rho\,\mathrm{OPT}_2\end{aligned}$$

Setting $\alpha = \frac{2}{3}$ gives $\min\{\mathrm{TOP}, \mathrm{BOT}\} \leq \frac{8}{3}\rho\,\mathrm{OPT}_1 + \frac{4}{3}\rho\,\mathrm{OPT}_2 = \frac{4}{3}\rho\,\mathrm{OPT}$. Combining the graphs in Figures 2 and 3, we can show that, asymptotically, the ratio $\frac{4}{3}$ is tight.

For $k > 2$ levels, the inequalities in Lemmas 3 and 4 generalize; we provide the proof in the full version [1]. ◀

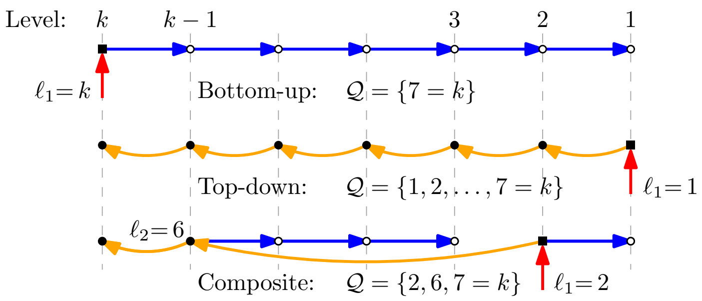


Figure 4 Illustration of a composite heuristic for an arbitrary choice of $\mathcal{Q} = \{\ell_1, \ell_2, \cdots, \ell_m\}$. Blue arrows pointing right indicate bottom-up propagations (prune E_{ℓ_i} to get $E_{\ell_{i-1}}$). Orange curved arrows pointing left indicate top-down propagations (set to 0 the cost of edges in E_{ℓ_i} when computing $E_{\ell_{i+1}}$). Red arrows indicate where the algorithms starts. Bottom-up and top-down heuristics are special cases with $\mathcal{Q} = \{k\}$, and $\mathcal{Q} = \{1, 2, \ldots, k\}$, respectively.

2.2 Composite Algorithm

We describe an approach that generalizes the above approaches in order to obtain a better approximation ratio for $k > 2$ levels. The main idea behind this composite approach is the following: In the top-down approach, we choose a set of edges $E_{\mathrm{TOP},1}$ that spans T_1, and then propagate this choice to levels $2, \ldots, k$ by setting the cost of these edges to 0. On the other hand, in the bottom-up approach, we choose a set of edges $E_{\mathrm{BOT},k}$ that spans T_k, which is propagated to levels $k-1, \ldots, 1$. The idea is that for $k > 2$, we can choose a set of intermediate levels and propagate our choices between these levels in a top-down manner, and to the levels lying in between them in a bottom-up manner.

Formally, let $\mathcal{Q} = \{\ell_1, \ell_2, \ldots, \ell_m\}$ with $1 \le \ell_1 < \ell_2 < \cdots < \ell_m = k$ be a subset of levels sorted in increasing order. We first compute a Steiner tree $E_{\ell_1} = ST(G, T_{\ell_1})$ for level ℓ_1, and then use it to construct trees $E_{\ell_1 - 1}, \ldots, E_1$ similarly to the bottom-up approach. Then, we set the weights of E_{ℓ_1} to zero (as in the top-down approach) and compute a Steiner tree $E_{\ell_2} = ST(G', T_{\ell_2})$ for level ℓ_2 in the reweighed graph. Again, we can use E_{ℓ_2} to construct the trees $E_{\ell_2 - 1}$ to $E_{\ell_1 + 1}$. Repeating this procedure until spanning $E_{\ell_m} = E_k$ results in a solution to MLST. Note that the top-down and bottom-up heuristics are special cases of this approach, with $\mathcal{Q} = \{1, 2, \ldots, k\}$ and $\mathcal{Q} = \{k\}$, respectively. Figure 4 provides an illustration of the propagations in the top-down, in the bottom-up, and in a general heuristic. Let $\mathrm{CMP}(\mathcal{Q})$ be the cost of the MLST returned by the composite approach over some set $\mathcal{Q}$.

For any choice of $\mathcal{Q}$, we have $\mathrm{CMP}(\mathcal{Q}) \le \rho \sum_{i=1}^{m} (k - \ell_{i-1}) \mathrm{MIN}_{\ell_i}$, with the convention $\ell_0 = 0$. The proof of this claim is similar to that of Lemma 3: when we compute E_{ℓ_1} and propagate its edges to all levels, we incur a cost of at most $\rho k \mathrm{MIN}_{\ell_1}$. When we compute E_{ℓ_2}, we also construct the trees $E_{\ell_2 - 1}, \ldots, E_{\ell_1 + 1}$. Using the lower bound $\mathrm{OPT} \ge \sum_{\ell=1}^{k} \mathrm{MIN}_\ell$, we can find an upper bound for the approximation ratio t. Without loss of generality, assume $\sum_{\ell=1}^{k} \mathrm{MIN}_\ell = 1$, so that $\mathrm{OPT} \ge 1$. Also, since all the equations and inequalities scale by ρ, we let $\rho = 1$. Hence, we have

$$t = \frac{\mathrm{CMP}(\mathcal{Q})}{\mathrm{OPT}} \le \frac{\rho \sum_{i=1}^{m} (k - \ell_{i-1}) \mathrm{MIN}_{\ell_i}}{\sum_{\ell=1}^{k} \mathrm{MIN}_\ell} = \sum_{i=1}^{m} (k - \ell_{i-1}) \mathrm{MIN}_{\ell_i}.$$

As observed above, both the top-down and the bottom-up approaches (which, due to Theorem 2, are $\frac{k+1}{2}$- and k-approximations, respectively) are two of the 2^{k-1} heuristics possible in the composite approach. For the top-down heuristic, $\mathrm{TOP} = \mathrm{CMP}(\{1, 2, \ldots, k\}) \le$

$k\text{MIN}_1 + (k-1)\text{MIN}_2 + \ldots + \text{MIN}_k \leq \frac{k+1}{2}$, with equality when $\text{MIN}_1 = \text{MIN}_2 = \ldots = \text{MIN}_k = \frac{1}{k}$. For the bottom-up heuristic, $\text{BOT} = \text{CMP}(\{k\}) \leq k\text{MIN}_k \leq k$.

An important choice of $\mathcal{Q}$ is $\mathcal{Q} = \{k-2^q+1 : 0 \leq q \leq q_{\max} = \lfloor \log_2 k \rfloor\}$. For $k = 2^{q_{\max}+1}-1$, the weakest upper bound occurs when $\text{MIN}_1 = \cdots = \text{MIN}_{k-2^{q_{\max}}} = 0$ and $\text{MIN}_{k-2^q_{\max}+1} = \cdots = \text{MIN}_k = 1/2^{q_{\max}}$ resulting in $t \leq \sum_{q=0}^{q_{\max}} 2^{q+1} - 1/2^{q_{\max}} \leq 2^{q_{\max}+2}/2^{q_{\max}} = 4$. Indeed, this choice of $\mathcal{Q}$ produces the 4ρ-approximation (QoS) given by Charikar et al. [7].

When $k = 2$, the only $2^{2-1} = 2$ composite heuristics are top-down and bottom-up (see Section 2.1). For $k \geq 2$, the set $\{1, \ldots, k\}$ has 2^{k-1} subsets that contain k, so there are 2^{k-1} different choices of $\mathcal{Q}$. The composite algorithm executes all of them and picks the solution with minimum cost (denoted CMP):

$$\text{CMP} = \min_{\substack{\mathcal{Q} \subseteq \{1,\ldots,k\} \\ k \in \mathcal{Q}}} \text{CMP}(\mathcal{Q}).$$

More generally, for $k \geq 2$, the composite heuristic produces a t-approximation, where t is the largest real number that simultaneously satisfies the 2^{k-1} inequalities

$$t \leq \sum_{i=1}^{m} (k - \ell_{i-1})\text{MIN}_{\ell_i},$$

for all subsets $\{\ell_1, \ldots, \ell_m\} \subseteq \{1, 2, \ldots, k\}$ that contain k and for all choices of $\text{MIN}_1, \ldots, \text{MIN}_k$ such that $\text{MIN}_1 \leq \text{MIN}_2 \leq \cdots \leq \text{MIN}_k$ and $\sum_{\ell=1}^{k} \text{MIN}_\ell = 1$. The system of 2^{k-1} inequalities can be expressed in matrix form as

$$M_k \boldsymbol{s} \geq t \cdot \mathbf{1}_{2^{k-1} \times 1},$$

where $\boldsymbol{s} = [\text{MIN}_1, \text{MIN}_2, \cdots, \text{MIN}_k]^T$ and M_k is a $(2^{k-1} \times k)$-matrix that can be constructed recursively as

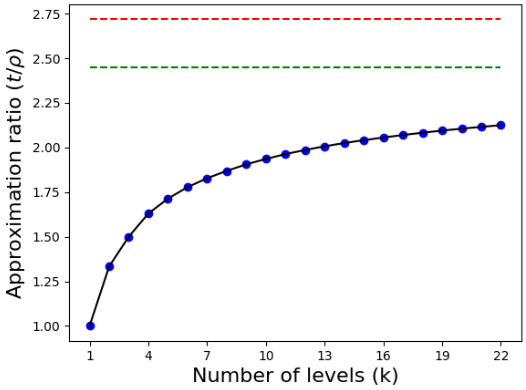
$$M_k = \begin{bmatrix} k \cdot \mathbf{1}_{2^{k-2} \times 1} & M_{k-1} \\ \mathbf{0}_{2^{k-2} \times 1} & P_{k-1} + M_{k-1} \end{bmatrix} \text{ with } P_k = \begin{bmatrix} \mathbf{1}_{2^{k-2} \times 1} & \mathbf{0}_{2^{k-2} \times (k-1)} \\ \mathbf{0}_{2^{k-2} \times 1} & P_{k-1} \end{bmatrix},$$

starting with the 1×1 matrices $M_1 = [1]$ and $P_1 = [1]$. Therefore, for each value of k, we can find the approximation ratio of the composite algorithm by solving a linear program (LP). We summarize our discussion as follows.

▶ **Theorem 5.** *For any $k = 2, \ldots, 22$, the composite algorithm yields a t-approximation to MLST, where the values of t are listed in Figure 5.*

Neglecting the factor ρ for now, the approximation ratio $t = 3/2$ for $k = 3$ is better than the ratio of $(5 + 4\sqrt{2})/7 + \varepsilon \approx 1.5224 + \varepsilon$ guaranteed by Xue et al. [24] for the Euclidean case. (The additive constant ε in their ratio stems from using Arora's PTAS as a subroutine for Euclidean ST, which corresponds to the multiplicative constant ρ for using an ST algorithm as a subroutine for MLST.) Recall that an improvement for $k = 3$ was posed as an open problem by Karpinski et al. [14]. Also, for each of the cases $4 \leq k \leq 22$ our results in Theorem 5 improve the approximation ratios of $e\rho \approx 2.718\rho$ and 2.454ρ guaranteed by Charikar et al. [7] and by Karpinski et al. [14], respectively. On the other hand, our ratios increase with k, while their results hold for every k. The graph of the approximation ratio of the composite algorithm (see Figure 5) for $k = 1, \ldots, 22$ suggests that it will stay below 2.454ρ for values of k much larger than 22.

Since the number of heuristics in the composite algorithm grows exponentially with k, it is computationally efficient only for small k. Indeed, for k levels, the composite heuristic requires 2^k ST computations. In the following, we show that we can achieve the same approximation guarantee with at most $2k$ ST computations.



k	t/ρ	k	t/ρ
1	1.000	12	1.986
2	1.333	13	2.007
3	1.500	14	2.025
4	1.630	15	2.041
5	1.713	16	2.056
6	1.778	17	2.070
7	1.828	18	2.083
8	1.869	19	2.094
9	1.905	20	2.106
10	1.936	21	2.116
11	1.963	22	2.125

Figure 5 Approximation ratios for the composite algorithm for $k = 1, \dots, 22$ (blue curve), compared to the ratio $t/\rho = e$ (red dashed line) guaranteed by the algorithm of Charikar et al. [7] and $t/\rho = 2.454$ (green dashed line) guaranteed by the algorithm of Karpinski et al. [14]. The table to the right lists the exact values for the ratio t/ρ.

▶ **Theorem 6.** *For a given instance of the MLST problem, a specific choice of Q^* can be found through k ST computations for which $\mathrm{CMP}(Q^*)$ is guaranteed the theoretical approximation ratio of the composite heuristic.*

Proof. Given a graph $G = (V, E)$ with cost function c, and terminal sets $T_1 \subset T_2 \subset \cdots \subset T_k \subseteq V$, compute a Steiner tree on each level and set $\mathrm{MIN}_\ell = c(ST(G, T_\ell))$. Since $\boldsymbol{s} = [\mathrm{MIN}_1, \dots, \mathrm{MIN}_k]^T$ is not necessarily the optimal solution to the LP for computing the approximation ratio t, there must be at least one constraint for which $\sum_{i=1}^m (k - \ell_{i-1})\mathrm{MIN}_{\ell_i} \leq t \sum_{\ell=1}^k \mathrm{MIN}_\ell$. The minimum entry in the vector $M_k\boldsymbol{s}$ corresponds to such a constraint. Let $q \in \{1, \dots, 2^{k-1}\}$ be the index of this entry, and let $\mathcal{Q}^* \subseteq \{1, \dots, k\}$ be the index set corresponding to non-zero entries in the q^{th} row of M_k. Then we have $\mathrm{CMP}(\mathcal{Q}^*)/\mathrm{OPT} \leq (\sum_{i=1}^m (k - \ell_{i-1})\mathrm{MIN}_{\ell_i})/(\sum_{\ell=1}^k \mathrm{MIN}_\ell)$, which yields $\mathrm{CMP}(\mathcal{Q}^*) \leq t \cdot \mathrm{OPT}$. ◀

3 Exact Algorithm

Recall the well-known flow formulation for ST [3, 19]. It assumes that the input graph is directed, which we can achieve by simply replacing each undirected edge by two directed edges in opposite directions of the same cost. Recall that T is the set of terminals. Let s be a fixed terminal node, the *source*. Then the ILP formulation for ST is as follows.

$$\text{Minimize} \sum_{(u,v)\in E} c(u,v) \cdot y_{uv}$$

$$\text{subject to} \sum_{vw\in E} x_{vw} - \sum_{uv\in E} x_{uv} = \begin{cases} |T| - 1 & \text{if } v = s \\ -1 & \text{if } v \in T \setminus \{s\} \\ 0 & \text{else} \end{cases} \quad \text{for } v \in V$$

$$0 \leq x_{uv} \leq (|T| - 1) \cdot y_{uv}, \text{ and } y_{uv} \in \{0, 1\}$$

In MLST, if an edge is selected on level ℓ, it must be selected on all levels below, that is, on levels $\ell+1,\ldots,k$. The flow variables x_{uv}^{ℓ} and the binary variables y_{uv}^{ℓ} are now additionally indexed by the level ℓ. The intended meaning of $y_{uv}^{\ell}=1$ is that edge uv is selected on level ℓ. We constrain the graph on level ℓ to be a subgraph of the graph on level $\ell+1$ as follows:

$$y_{uv}^{\ell+1} \geq y_{uv}^{\ell} \qquad \text{for } \ell \in \{1,2,\ldots,k-1\} \text{ and } (u,v) \in E$$

We also modify the objective function in the natural way:

$$\text{Minimize } \sum_{\ell=1}^{k} \sum_{uv \in E} c(u,v) \cdot y_{uv}^{\ell}$$

In the full version of our paper [1], we provide two further ILP formulations of MLST. Among the three, the above formulation uses the smallest number of constraints.

4 Experimental Results

Graph Data Synthesis. The graph data we used in our experiment are synthesized from graph generative models. In particular, we used four random network generation models: Erdős–Renyi [10], random geometric [18], Watts–Strogatz [22], and Barabási–Albert [4]. These networks are very well studied in the literature [17].

In each graph instance, we assign integer edge weights $c(e)$ randomly and uniformly between 1 and 10 inclusive. Even though the generated graphs are almost surely connected, it is possible to get a disconnected graph. Therefore, in our experiment, we only use connected graphs and discard the rest. Computational challenges of solving an ILP limit the size of the graphs to a few hundred in practice.

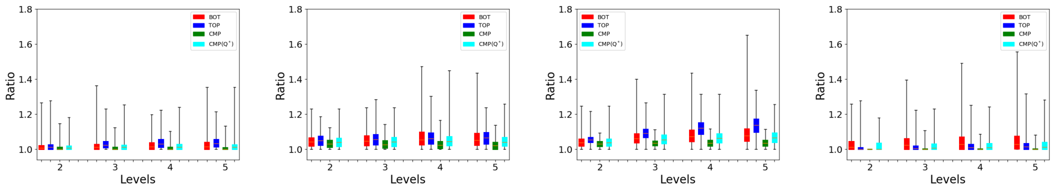
Selection of Levels and Terminal Nodes. For each generated graph, we generated MLST instances with $k=2,3,4,5$ levels. We adopted two strategies for selecting the terminals on the k levels: *linear* vs. *exponential*. In the linear scenario, we select the terminals on each level by randomly sampling $\lfloor |V|(\ell+1)/(k+1) \rfloor$ nodes on level ℓ so that $|T_{\ell+1}|-|T_{\ell}| \approx |T_{\ell}|-|T_{\ell-1}|$. In the exponential case, we select the terminals at each layer by sampling uniformly randomly $\lfloor |V|/2^{k-\ell} \rfloor$ nodes so that $|T_{l+1}|/|T_l| \approx |T_l|/|T_{l-1}|$.

To summarize, a single instance of an input to MLST is characterized by four parameters: network generation model NGM $\in$ {ER,RG,WS,BA}, number of nodes $|V|$, number of levels k, and the terminal selection method TSM $\in$ {Linear,Exponential}.

Algorithms and Outputs. We implemented the bottom-up, top-down, and composite heuristics described in Section 2 and the simple 4ρ-approximation algorithm by Charikar et al. [7] for the QoS Multicast Tree problem, all in Python.

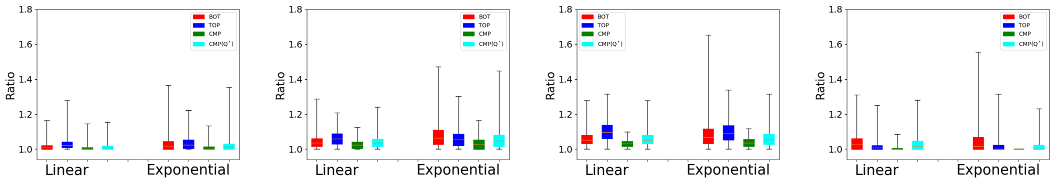
For evaluating the heuristcs, we also implemented the ILP described in Section 3 using CPLEX 12.6.2 as ILP solver. We distributed the experiment on a high performance computer (HPC) into multiple tasks. A single task performs the computation of 5 to 50 graphs. The number of graphs varies because for smaller graphs we can combine more graphs in a single task. For larger graphs, however, the time limit for a single task is not enough if the number of graphs is too large.

For each instance of MLST, we compute the costs of the MLST from the ILP solution (OPT), the bottom-up solution (BOT), the top-down solution (TOP), the composite heuristic



(a) Barabási–Albert **(b)** Erdős–Rényi **(c)** Geometric **(d)** Watts–Strogatz

Figure 6 Performance of BOT, TOP, CMP, and CMP($\mathcal{Q}^*$) w.r.t. the number k of levels.



(a) Barabási–Albert **(b)** Erdős–Rényi **(c)** Geometric **(d)** Watts–Strogatz

Figure 7 Performance of BOT, TOP, CMP, and CMP($\mathcal{Q}^*$) w.r.t. the terminal selection method.

(CMP), the guaranteed performance heuristic (CMP($\mathcal{Q}^*$)) heuristic, and the simple 4ρ-approximate Quality-of-Service heuristic (QoS) of Charikar et al. [7]. For the ST computation we used the 2-approximation algorithm of Gilbert and Pollak [11].

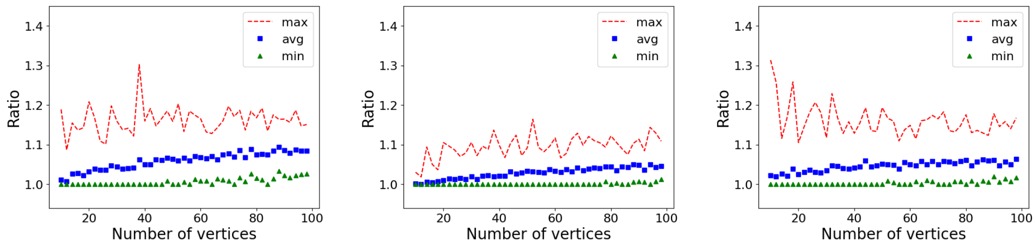
After completing the experiment, we compared the results of the heuristics with exact solutions. We show the performance ratio APP/OPT for each heuristic, and how they depend on parameters of the experiment setup. For example, we investigate how the performance ratio changes as $|V|$ increases. Since each instance of the experiment setup involves randomness at different steps, we generated 5 instances for any fixed setup (e.g., Geometric graph, $|V| = 100$, 5 levels, linear terminal selection).

We did not compare the running times of our implementations in detail since our Python code is not optimized in this respect. As a rough measure, however, we list the number of Steiner tree computations performed by each algorithm in the worst case – BOT: 1, TOP: k, CMP: 2^k, CMP($\mathcal{Q}^*$): $2k$, and QoS: k.

Results. First, we examined how the performance of the heuristics compared with the exact solution as the number of the levels k changed. In our experiments, k varies between 2 and 5. We show the results using box plots in Figure 6. As expected, the performance of the heuristics gets slightly worse as k increases. The bottom-up approach had the worst performance, while the composite heuristic performed very well in practice.

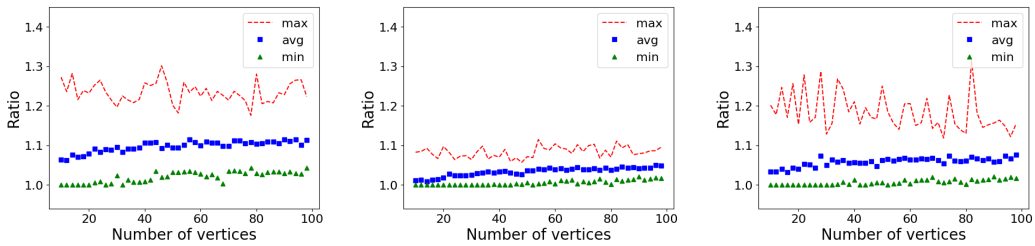
Second, we examined how the performance of the heuristics compared with the exact solution for different terminal selection methods, either LINEAR or EXPONENTIAL. We show the results using box plots in Figure 7. Overall, the heuristics performed worse when the sizes of the terminal sets decrease exponentially.

Third, we investigated how the heuristics perform with respect to the graph size $|V|$, for each of the network models ER, RG, WS, and BA; see Figures 8–11. Note that the y-axes of the graphs in these figures have a different scale than the graphs in Figures 6 and 7. Since several instances share the same network size, we show minimum, maximum, and mean values. Overall, the performance of the heuristics slightly deteriorated as $|V|$ increased. Due to lack of space, we omit the bottom-up heuristic here, which tends to be comparable to or slightly worse than the top-down heuristic. Again, the composite heuristic yielded the



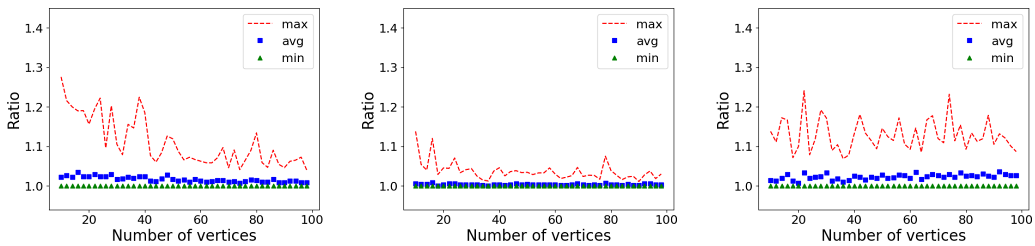
(a) Top-down **(b)** Composite **(c)** CMP($\mathcal{Q}^*$)

Figure 8 Performance of TOP, CMP, and CMP($\mathcal{Q}^*$) on Erdős–Rényi graphs.



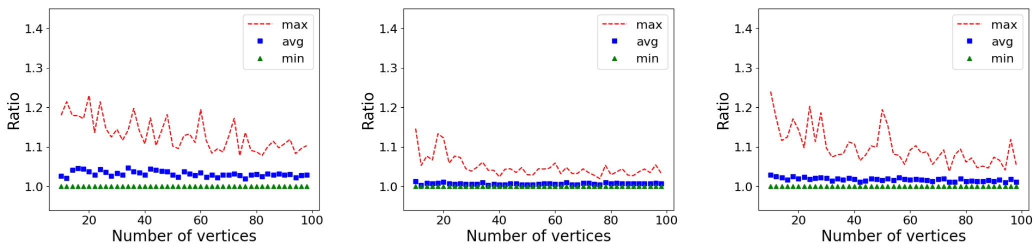
(a) Top-down **(b)** Composite **(c)** CMP($\mathcal{Q}^*$)

Figure 9 Performance of TOP, CMP, and CMP($\mathcal{Q}^*$) on Geometric graphs.



(a) Top-down **(b)** Composite **(c)** CMP($\mathcal{Q}^*$)

Figure 10 Performance of TOP, CMP, and CMP($\mathcal{Q}^*$) on Watts–Strogatz graphs.



(a) Top-down **(b)** Composite **(c)** CMP($\mathcal{Q}^*$)

Figure 11 Performance of TOP, CMP, and CMP($\mathcal{Q}^*$) on Barabási–Albert graphs.

best performance; top-down and CMP($\mathcal{Q}^*$) were comparable. Data for the other heuristics is available in the full version [1].

5 Conclusions

We presented several heuristics for the MLST problem and analyzed them both theoretically and experimentally. Natural open problems include determining inapproximability results for MLST, determining a closed-form expression for the approximation ratio of the composite heuristic (Section 2.2), and generalizing the notion of multi-level graphs to related problems (such as the node-weighted Steiner tree problem).

References

1 Reyan Ahmed, Patrizio Angelini, Faryad Darabi Sahneh, Alon Efrat, David Glickenstein, Martin Gronemann, Niklas Heinsohn, Stephen G. Kobourov, Richard Spence, Joseph Watkins, and Alexander Wolff. Multi-level Steiner trees, 2018. `arXiv:1804.02627`.

2 Sanjeev Arora. Polynomial time approximation schemes for Euclidean Traveling Salesman and other geometric problems. *J. ACM*, 45(5):753–782, 1998. `doi:10.1145/290179.290180`.

3 Anantaram Balakrishnan, Thomas L. Magnanti, and Prakash Mirchandani. Modeling and heuristic worst-case performance analysis of the two-level network design problem. *Management Sci.*, 40(7):846–867, 1994. `doi:10.1287/mnsc.40.7.846`.

4 Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

5 Marshall Bern and Paul Plassmann. The Steiner problem with edge lengths 1 and 2. *Inform. Process. Lett.*, 32(4):171–176, 1989. `doi:10.1016/0020-0190(89)90039-2`.

6 Jaroslaw Byrka, Fabrizio Grandoni, Thomas Rothvoß, and Laura Sanità. Steiner tree approximation via iterative randomized rounding. *J. ACM*, 60(1):6:1–6:33, 2013. `doi:10.1145/2432622.2432628`.

7 Moses Charikar, Joseph (Seffi) Naor, and Baruch Schieber. Resource optimization in QoS multicast routing of real-time multimedia. *IEEE/ACM Trans. Networking*, 12(2):340–348, 2004. `doi:10.1109/TNET.2004.826288`.

8 Miroslav Chlebík and Janka Chlebíková. The Steiner tree problem on graphs: Inapproximability results. *Theoret. Comput. Sci.*, 406(3):207–214, 2008. `doi:10.1016/j.tcs.2008.06.046`.

9 Julia Chuzhoy, Anupam Gupta, Joseph (Seffi) Naor, and Amitabh Sinha. On the approximability of some network design problems. *ACM Trans. Algorithms*, 4(2):23:1–23:17, 2008. `doi:10.1145/1361192.1361200`.

10 Paul Erdös and Alfréd Rényi. On random graphs I. *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959.

11 Edgar N. Gilbert and Henry O. Pollak. Steiner minimal trees. *SIAM J. Appl. Math.*, 16(1):1–29, 1968. `doi:10.1137/0116001`.

12 Mathias Hauptmann and Marek Karpinski (eds.). A compendium on Steiner tree problems, 2015. URL: `http://theory.cs.uni-bonn.de/info5/steinerkompendium/`.

13 Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller, James W. Thatcher, and Jean D. Bohlinger, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972. `doi:10.1007/978-1-4684-2001-2_9`.

14 Marek Karpinski, Ion I. Mandoiu, Alexander Olshevsky, and Alexander Zelikovsky. Improved approximation algorithms for the quality of service multicast tree problem. *Algorithmica*, 42(2):109–120, 2005. `doi:10.1007/s00453-004-1133-y`.

15 Prakash Mirchandani. The multi-tier tree problem. *INFORMS J. Comput.*, 8(3):202–218, 1996.

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 J. Comput.*, 28(4):1298–1309, 1999. doi:10.1137/S0097539796309764.

17 Mark E.J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003. doi:10.1137/S003614450342480.

18 Mathew Penrose. *Random geometric graphs*, volume 5 of *Oxford Studies in Probability*. Oxford University Press, 2003.

19 Tobias Polzin and Siavash Vahdati Daneshmand. A comparison of Steiner tree relaxations. *Discrete Appl. Math.*, 112(1):241–261, 2001. doi:10.1016/S0166-218X(00)00318-8.

20 Hans Jürgen Prömel and Angelika Steger. *The Steiner Tree Problem.* Vieweg and Teubner Verlag, 2002.

21 Gabriel Robins and Alexander Zelikovsky. Tighter bounds for graph Steiner tree approximation. *SIAM J. Discrete Math.*, 19(1):122–134, 2005. doi:10.1137/S0895480101393155.

22 Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998. doi:10.1038/30918.

23 Pawel Winter. Steiner problem in networks: A survey. *Networks*, 17(2):129–167, 1987. doi:10.1002/net.3230170203.

24 Guoliang Xue, Guo-Hui Lin, and Ding-Zhu Du. Grade of service Steiner minimum trees in the Euclidean plane. *Algorithmica*, 31(4):479–500, 2001. doi:10.1007/s00453-001-0050-6.