

Multi-Level Steiner Trees ^{*†}

Reyan Ahmed¹, Patrizio Angelini², Faryad Darabi Sahneh¹, Alon Efrat¹,
 David Glickenstein¹, Martin Gronemann³, Niklas Heinsohn², Stephen G. Kobourov¹,
 Richard Spence¹, Joseph Watkins¹, and Alexander Wolff⁴

¹University of Arizona, Tucson, United States

²Universität Tübingen, Tübingen, Germany

³Universität zu Köln, Köln, Germany

⁴Universität Würzburg, Würzburg, Germany

April 10, 2018

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 $T \subseteq V$, the *terminals*. The objective is to find a minimum-cost edge set $E' \subseteq E$ that spans the terminals. The problem is APX-hard [Bern & Plassman, IPL 1989]; the best known approximation algorithm has a ratio of $\rho = \ln(4) + \varepsilon < 1.39$ [Byrka et al., J. ACM 2013]. In this paper, we study a natural generalization, the *multi-level Steiner tree* (MLST) problem: given a nested sequence of terminals $T_1 \subset \dots \subset T_k \subseteq V$, compute nested edge sets $E_1 \subseteq \dots \subseteq E_k \subseteq E$ that span the corresponding terminal sets with minimum total cost. (Note that, for $\ell = 1, \dots, k$, edges in E_ℓ contribute $(k - \ell + 1)$ -fold to this cost).

The MLST problem and variants thereof have been studied under names such as Quality-of-Service Multicast tree, Grade-of-Service Steiner tree, Multi-Tier tree, etc. 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 Erdős–Rényi, random geometric, Watts–Strogatz, and Barabási–Albert network generation models for varying $|V|$, k , and terminal selection methods. We also implemented an integer linear program for MLST to provide ground truth. Our combined algorithm outperforms the others both in theory and in practice when the number of levels is up to $k \leq 22$.

1 Introduction

Let $G = (V, E)$ be an undirected, connected graph with non-negative edge costs $c: E \rightarrow \mathbb{R}^+$, and let $T \subseteq V$ be a set of vertices called *terminals*. A *Steiner tree* is a subgraph of G that spans T . The (*Network or Graph*) *Steiner tree problem* (STP) is to find a minimum-cost Steiner tree $E' \subseteq E$, where the cost of E' is $c(E') = \sum_{e \in E'} c(e)$. STP is one of Karp’s initial NP-hard problems [12]. There is a survey on Steiner tree problems [22], an online compendium [11], and a textbook [19].

^{*}This material is based on work supported by the National Science Foundation through the TRIPODS project under Grant No. CCF-1423411.

[†]This paper has been accepted in 17th International Symposium on Experimental Algorithms (SEA 2018)

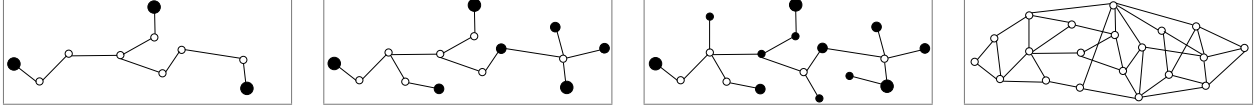


Figure 1: An illustration of a 3-level MLST for the graph at the right. Solid and open circles represent terminal and non-terminal nodes, respectively. Note that the level 1 tree (left) is contained in the level 2 tree (mid), which is in turn contained in the level 3 tree (right).

Due to its practical importance in many domains, there is a long history of exact and approximation algorithms for the problem. Let us quickly recall the classical 2-approximation algorithm for STP [10]. Consider the *metric closure* of G , that is, 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. [5], which has a ratio of $\ln(4) + \varepsilon < 1.39$. Their algorithm uses a new iterative randomized rounding technique. Note that STP is APX-hard [4]; more concretely, it is NP-hard to approximate the problem within a factor of 96/95 [7]. 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) [1, 15], while this is not true for the general metric case [4].

In this paper, we consider a natural generalization of STP 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 1.1 (Multi-Level Steiner Tree (MLST) Problem). Given a connected, undirected graph $G = (V, E)$ with edge weights $c: E \rightarrow \mathbb{R}^+$ and k nested terminal sets $T_1 \subset \dots \subset T_k \subseteq V$, a *multi-level Steiner tree* consists of k nested edge sets $E_1 \subseteq \dots \subseteq E_k \subseteq E$ such that E_1 spans T_1, \dots, E_k spans T_k . The cost of an MLST is defined by $c(E_1) + c(E_2) + \dots + c(E_k)$. The MLST problem is to find an MLST $E_{\text{OPT},1} \subseteq \dots \subseteq E_{\text{OPT},k} \subseteq E$ with minimum cost.

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

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

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

$$\text{OPT} = k\text{OPT}_1 + (k-1)\text{OPT}_2 + \dots + \text{OPT}_k$$

where $\text{OPT}_1 = c(E_{\text{OPT},1})$ and $\text{OPT}_\ell = c(E_{\text{OPT},\ell} \setminus E_{\text{OPT},\ell-1})$ for $2 \leq \ell \leq k$. 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 obvious applications in designing multi-level infrastructures of low cost that connect users who need connections among themselves of high-enough quality for their purpose. Apart from this application in network design, multi-scale representations of graphs are useful in applications such as geography or network visualization, where users want to examine complex (street, river, or abstract) networks at different levels of detail. Here, the cost models the “stability” of the visualization.

Previous Work. Variants of the MLST problem have been studied previously under various names, such as *Multi-Level Network Design (MLND)* [2], *Multi-Tier Tree (MTT)* [14], *Quality-of-Service (QoS) Multicast Tree* [6], and *Priority-Steiner Tree* [8].

In MLND, the vertices of the given graph are partitioned into k levels, and the task is to construct a k -level network. For $1 \leq \ell \leq k$, let $c^\ell(e)$ be the cost of edge e if it is in level ℓ (“primary” and “secondary” costs for 2 levels). 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, $0 \leq c^k(e) \leq \dots \leq c^1(e)$ 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. 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 $c^2(e) = 0$. Let ρ be the ratio of the best approximation algorithm for (single-level) STP, that is, currently $\rho = \ln(4) + \varepsilon < 1.39$. Balakrishnan et al. [2] gave a $4/3\rho$ -approximation algorithm for 2-level MLND with proportional edge costs, that is, $c^\ell(e) = c^k(e)(k - \ell + 1)$.

For MTT, which is equivalent to MLND, Mirchandani [14] 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 $c^1, \dots, c^k: E \rightarrow \mathbb{R}^+$. 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 [6] 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. [6] introduced what they call the *rate model*, where edge costs are proportional. They show that the rate model 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 the MLST problem can be formulated as an instance of the rate model by Charikar et al.

First, Charikar et al. 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. Next, Charikar et al. improved the ratio to $e\rho$, where e is the base of the natural logarithm. The $e\rho$ -approximation is similar to the 4ρ -approximation, but uses randomized doubling. It can be derandomized.

Instead of taking the union of the Steiner trees on each rounded level, Karpinski et al. [13] contract them into the source in each step. This yields a 2.454ρ -approximation. Karpinski et al. also gave a $(1.265 + \varepsilon)\rho$ -approximation for the 2-level case. (Since the results of Karpinski et al. are not given with respect to ρ , but depend on several Steiner tree approximation algorithms – among them the best approximation algorithm with ratio 1.549 [20] available at the time – we obtained the numbers given here by dividing their results by 1.549 and stating the factor ρ . Strictly speaking, their algorithm could perform worse when plugging in the algorithm of Byrka et al. [5], but we did not redo their analysis.)

For the more general Priority-Steiner Tree problem, where edge costs are not necessarily proportional, Charikar et al. [6] gave a $\min\{2 \ln |T|, k\rho\}$ -approximation algorithm. Chuzhoy et al. [8] showed that Priority-Steiner Tree does not admit an $O(\log \log n)$ -approximation algorithm unless $\text{NP} \subseteq \text{DTIME}(n^{O(\log \log \log n)})$.

For Euclidean MLST, Xue et al. [24] gave a recursive algorithm that uses any algorithm for

Euclidean Steiner Tree (EST) as a subroutine. With a PTAS [1, 15] 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 STP computations.

Therefore, we propose a procedure that achieves the same approximation ratio as the composite heuristic but needs only $2k$ STP computations. In particular, it achieves a ratio of 1.5ρ for $k = 3$ levels, which settles a question posed by Karpinski et al. [13] who were asking whether the $1.5224 + \varepsilon$ -approximation of Xue et 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 STP [18] 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 STP 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. [2]; however, we generalize the analysis to an arbitrary number of levels.

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

In the bottom-up approach, we compute a Steiner tree $E_{\text{BOT},k}$ spanning the terminals T_k in level k . Then, for each level ℓ , we obtain $E_{\text{BOT},\ell}$ as the smallest subtree of $E_{\text{BOT},k}$ 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 STP (that is, $\rho = 1$ corresponds to using an exact STP subroutine).

Theorem 1. 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. Let TOP be the total cost produced by the top-down approach, and let $\text{TOP}_\ell = c(E_{\text{TOP},\ell} \setminus E_{\text{TOP},\ell-1})$ denote the cost of edges on level ℓ but not level $\ell - 1$, produced by the top-down approach, so that $\text{TOP} = 2\text{TOP}_1 + \text{TOP}_2$. 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.

Lemma 1. The following inequalities relate TOP with OPT:

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

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

Proof. (1) follows from the fact that $E_{\text{TOP},1}$ is a ρ -approximation for STP over T_1 , that is, $\text{TOP}_1 \leq \rho \text{MIN}_1 \leq \rho \text{OPT}_1$. 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 $e \in E_{\text{TOP},1}$. Thus, $\text{TOP}_2 \leq \rho \text{MIN}'_2 \leq \rho \text{MIN}_2$. Additionally, since $E_{\text{OPT},2}$ spans T_2 by definition, we have $\text{MIN}_2 \leq \text{OPT}_1 + \text{OPT}_2$, so $\text{TOP}_2 \leq \rho(\text{OPT}_1 + \text{OPT}_2)$ as desired. \square

Combining (1) and (2), we have $\text{TOP} = 2\text{TOP}_1 + \text{TOP}_2 \leq 3\rho \text{OPT}_1 + \rho \text{OPT}_2 \leq 3\rho \text{OPT}_1 + \frac{3}{2}\rho \text{OPT}_2 = \frac{3}{2}\rho \text{OPT}$, and hence the top-down approach provides a $\frac{3}{2}\rho$ -approximation when $k = 2$.

Analogously to Lemma 1 in the two-level case, we have that $\text{TOP}_\ell \leq \rho(\text{OPT}_1 + \text{OPT}_2 + \dots + \text{OPT}_\ell)$ for all $1 \leq \ell \leq k$. This holds as $\text{TOP}_\ell \leq \rho \text{MIN}'_\ell$, where MIN'_ℓ denotes the cost of the minimum Steiner tree over T_ℓ , with edges from $E_{\text{TOP},\ell-1}$ having weight set to zero. Then $\text{TOP}_\ell \leq \rho \text{MIN}'_\ell \leq \rho \text{MIN}_\ell \leq \rho(\text{OPT}_1 + \text{OPT}_2 + \dots + \text{OPT}_\ell)$.

Combining yields

$$\begin{aligned} \text{TOP} &= k\text{TOP}_1 + (k-1)\text{TOP}_2 + \dots + 1\text{TOP}_k \\ &\leq k\rho \text{OPT}_1 + (k-1)\rho(\text{OPT}_1 + \text{OPT}_2) + \dots + 1\rho(\text{OPT}_1 + \text{OPT}_2 + \dots + \text{OPT}_k) \\ &= \rho \left(\binom{k+1}{2} \text{OPT}_1 + \binom{k}{2} \text{OPT}_2 + \dots + \binom{2}{2} \text{OPT}_k \right) \\ &\leq \frac{k+1}{2} \rho \text{OPT}. \end{aligned}$$

In Fig. 2 we provide an example showing that our analysis is tight for $\rho = 1$.

Lemma 2. The following inequality relates BOT with OPT for the two-level case:

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

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

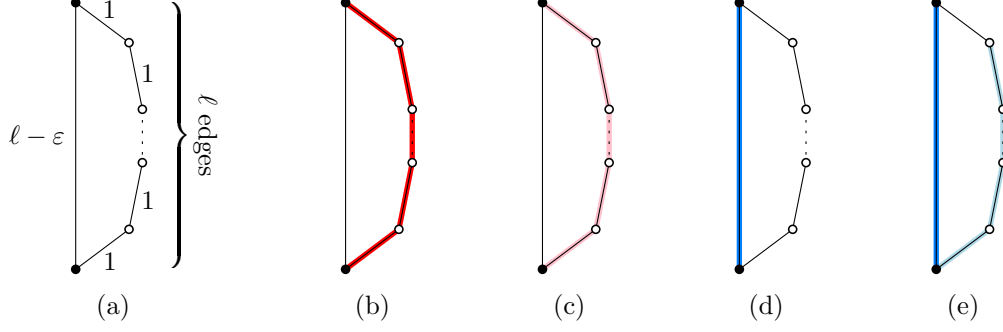


Figure 2: Analysis of the top-down approach (light and dark blue) in the 2-layer case is asymptotically tight (optimal solution light and dark red). The dark vertices and edges are on the top level, the white vertices and light edges are on the bottom level. In this example, $\text{OPT} = 2\ell$, while $\text{TOP} = 2(\ell - \varepsilon) + \ell - 1 = 3\ell - 2\varepsilon - 1$.

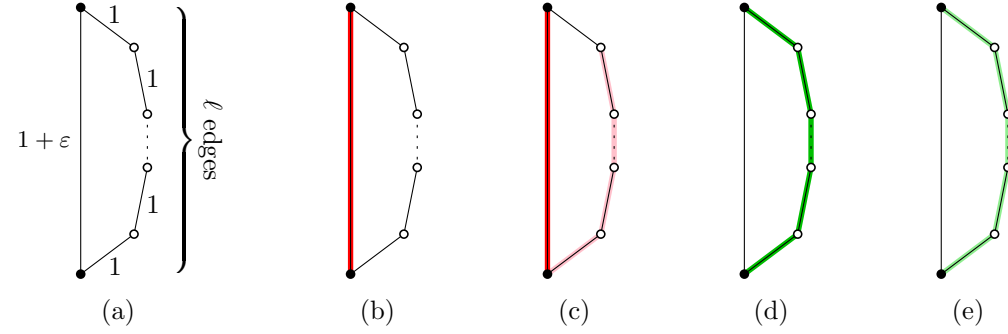


Figure 3: Analysis of the bottom-up approach (light and dark green) in the 2-layer case is asymptotically tight (optimal solution in light and dark red). In this example, $\text{OPT} = \ell + 1 + 2\varepsilon$, while $\text{BOT} = 2\ell$.

Hence, $\text{BOT} = 2\text{BOT}_1 + \text{BOT}_2 \leq 2(\text{BOT}_1 + \text{BOT}_2) \leq 2\rho(\text{OPT}_1 + \text{OPT}_2) \leq 2\rho(2\text{OPT}_1 + \text{OPT}_2) = 2\rho\text{OPT}$.

Analogously to Lemma 2, we have $\text{BOT}_1 + \text{BOT}_2 + \dots + \text{BOT}_k \leq \rho \text{MIN}_k \leq \rho(\text{OPT}_1 + \text{OPT}_2 + \dots + \text{OPT}_k)$. Hence, we have

$$\begin{aligned} \text{BOT} &\leq k(\text{BOT}_1 + \text{BOT}_2 + \dots + \text{BOT}_k) \\ &\leq k\rho(\text{OPT}_1 + \text{OPT}_2 + \dots + \text{OPT}_k) \\ &\leq k\rho\text{OPT}. \end{aligned}$$

Again, the approximation ratio (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,

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

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

To prove Theorem 1 for arbitrary k , we again use the fact that $\min\{\text{TOP}, \text{BOT}\} \leq \alpha \text{TOP} + (1 - \alpha) \text{BOT}$ for any $0 \leq \alpha \leq 1$. Applying the inequalities for BOT and TOP, we get:

$$\min\{\text{TOP}, \text{BOT}\} \leq \sum_{i=1}^k \left[\left(\binom{i+1}{2} - k \right) \alpha + k \right] \rho \text{OPT}_{k-i+1}$$

Since we are comparing $\min\{\text{TOP}, \text{BOT}\}$ to $t \cdot \text{OPT}$ for some approximation ratio $t > 1$, we can compare coefficients and find the smallest $t \geq 1$ such that the system of inequalities

$$\begin{aligned} \left(\binom{k+1}{2} - k \right) \rho \alpha + k \rho &\leq kt \\ \left(\binom{k}{2} - k \right) \rho \alpha + k \rho &\leq (k-1)t \\ &\vdots \\ \left(\binom{2}{2} - k \right) \rho \alpha + k \rho &\leq t \end{aligned}$$

has a solution $\alpha \in [0, 1]$. Adding the first inequality to $k/2$ times the last inequality gives $\frac{k^2+2k}{2} \rho \leq \frac{3kt}{2} \Rightarrow t \geq \frac{k+2}{3} \rho$, and it can be shown algebraically that $(t, \alpha) = (\frac{k+2}{3} \rho, \frac{2}{3})$ simultaneously satisfies the above inequalities. □

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 $E_{\text{TOP},1}$ that spans T_1 , and then propagate this choice to levels $2, \dots, k$ by setting the cost of these edges to 0. On the other hand, in the bottom-up approach, we choose a set of edges $E_{\text{BOT},k}$ that spans T_k , which is propagated to levels $k-1, \dots, 1$. 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 $\mathcal{Q} = \{\ell_1, \ell_2, \dots, \ell_m\}$ with $1 \leq \ell_1 < \ell_2 < \dots < \ell_m = k$ 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_{ℓ_1-1}, \dots, E_1 similar 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_{ℓ_2-1} to E_{ℓ_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 setting, with $\mathcal{Q} = \{1, 2, \dots, k\}$ 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.

For any choice of \mathcal{Q} , we have $\text{CMP}(\mathcal{Q}) \leq \rho \sum_{i=1}^m (k - \ell_{i-1}) \text{MIN}_{\ell_i}$, with the convention $\ell_0 = 0$. Using the lower bound $\text{OPT} \geq \sum_{\ell=1}^k \text{MIN}_{\ell}$, we can find an upper bound for the approximation ratio t . Without loss of generality, assume $\sum_{\ell=1}^k \text{MIN}_{\ell} = 1$. Also, since all the equations and inequalities scale by ρ , we let $\rho = 1$. Hence, we have:

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

As observed above, both the top-down and the bottom-up algorithms are special cases of the 2^{k-1} heuristics generated in the composite approach. For the top-down heuristic, $\text{TOP} = \text{CMP}(\{1, 2, \dots, k\}) \leq k\text{MIN}_1 + (k-1)\text{MIN}_2 + \dots + \text{MIN}_k$. The worst case happens when $\text{MIN}_1 = \text{MIN}_2 = \dots = \text{MIN}_k = 1/k$, for which we find $t \leq (k + (k-1) + \dots + 1)/k = (k+1)/2$. For the bottom-up heuristic, $\text{BOT} = \text{CMP}(\{k\}) \leq k\text{MIN}_k$. The worst case happens when $\text{MIN}_1 = \dots = \text{MIN}_{k-1} = 0$ and $\text{MIN}_k = 1$, for which we find $t \leq k$.

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 worst case happens when $\text{MIN}_1 = \dots = \text{MIN}_{k-2^{q_{\max}}} = 0$ and $\text{MIN}_{k-2^{q_{\max}+1}} = \dots = \text{MIN}_k = 1/2^{q_{\max}}$. This results 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} can be deduced from the 4ρ -approximation algorithm (QoS) of Charikar et al. [6].

Note that, when $k = 2$, the only $2^{2-1} = 2$ generated heuristics are the top-down and the bottom-up, and hence $\text{CMP} = \min\{\text{TOP}, \text{BOT}\}$. This does not hold for larger values of k , since the composite approach evaluates more heuristics and obtains a better ratio. Overall, $\{1, \dots, k\}$ has 2^{k-1} possible subsets, so there are 2^{k-1} different heuristics. The composite algorithm executes all of them and picks the solution with minimum cost. Thus:

$$\text{CMP} = \min_{\mathcal{Q} \subseteq \{1, \dots, k\}} \text{CMP}(\mathcal{Q}),$$

where $\text{CMP}(\mathcal{Q})$ denotes the cost for a specific subset \mathcal{Q} of $\{1, \dots, k\}$.

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

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

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

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

starting with $M_1 = [1], P_1 = [1]$. Therefore, for each value of k , we can find the approximation ratio of the composite algorithm by solving a linear program (LP).

Corollary 1. For any $k = 2, \dots, 22$, 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 STP, which corresponds to the multiplicative constant ρ for using an STP algorithm as a subroutine for MLST.) Recall that an improvement for $k = 3$ was posed as an open problem by Karpinski et al. [13]. Also, for each of the cases $4 \leq k \leq 22$ our results in Corollary 1 improve the approximation ratios of $e\rho \approx 2.718\rho$ and 2.454ρ guaranteed by Charikar et al. [6] and by Karpinski et al. [13], respectively. On the other hand, our ratios increase with k , while their results hold for every k . The function in Figure 5 suggests that the approximation ratio of the composite algorithm will stay

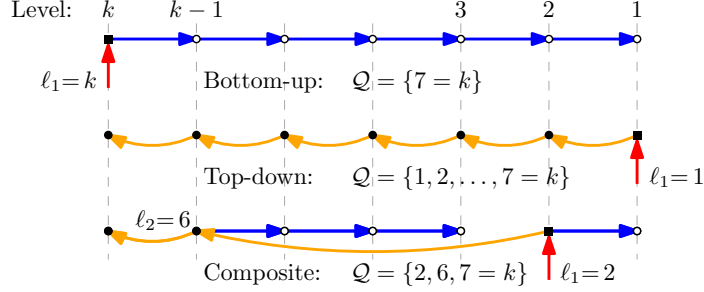


Figure 4: Illustration of a composite heuristic for an arbitrary choice of $\mathcal{Q} = \{\ell_1, \ell_2, \dots, \ell_m\}$. Blue arrows pointing right indicate bottom-up propagations (prune E_{ℓ_i} to get $E_{\ell_{i-1}}$). Orange curved arrows pointing left indicate top-down propagations (set to 0 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 $\mathcal{Q} = \{1, 2, \dots, k\}$, respectively.

below 2.454ρ for values of k much larger than 22. We found that the tangent line to the function at $k = 19$ intersects $t/\rho = 2.454$ only at $k = 50$.

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

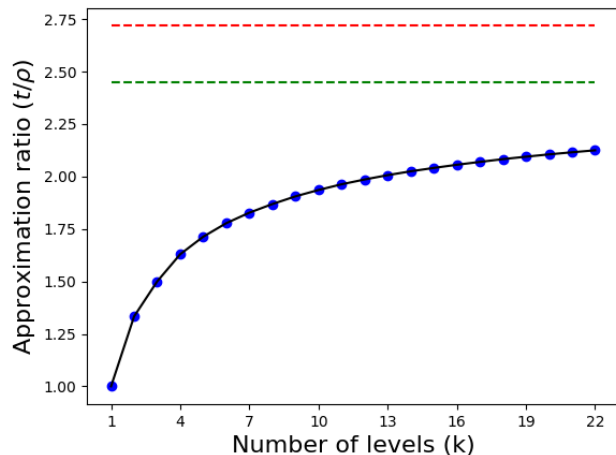
Theorem 2. For a given instance of the MLST problem, a specific choice of \mathcal{Q}^* can be found through k STP computations for which $\text{CMP}(\mathcal{Q}^*)$ is guaranteed the theoretical approximation ratio of the composite heuristic.

Proof. Given a graph $G = (V, E)$ with cost function c , and terminal nodes $T_1 \subset T_2 \subset \dots \subset T_k \subseteq V$, compute the Steiner trees on every level and find costs $\text{MIN}_\ell = c(\text{ST}(G, T_\ell))$. Since this $\mathbf{s} = [\text{MIN}_1, \dots, \text{MIN}_k]^T$ is not necessarily the optimal solution (assuming a worst-case scenario) to the LP for computing the approximation ratio t , there exists at least one constraint for which $\sum_{i=1}^m (k - \ell_{i-1}) \text{MIN}_{\ell_i} \leq t \sum_{\ell=1}^k \text{MIN}_\ell$. An instance of such constraint corresponds to the index $1 \leq q \leq 2^{k-1}$ with the minimum entry in the vector $M_k \mathbf{s}$. Let $\mathcal{Q}^* \subseteq \{1, \dots, k\}$ be the index set corresponding to non-zero entries in the q^{th} row of M_k . We have $\text{CMP}(\mathcal{Q}^*) \leq t \cdot \text{OPT}$. \square

2.3 Hybrid Algorithm

We implemented the additional hybrid heuristic to overcome a potential limitation of all our other heuristics. Namely, in each of these algorithms we consider the levels in a certain order, and propagate the choices done at each level to those following in this order, but we never consider any level twice. In this way, if we make a bad choice at some point, we will never be able to recover it. The hybrid approach, instead, has the property that levels may be considered more than once. Also, the choices done at a certain level are still propagated to the following levels, but in a weaker way than in the other heuristics. In particular, instead of setting their weights to 0, we just set them to a fraction of their original cost.

More specifically, in order to compute $\text{HYBRIDMLST}(G, T_1, \dots, T_k)$, we first compute $(\mathcal{T}_2, \dots, \mathcal{T}_k) = \text{HYBRIDMLST}(G, T_2, \dots, T_k)$ recursively. Then, we modify weights of G so that, for each level ℓ , the weight of the edges corresponding to T_ℓ are scaled by $\frac{\ell-1}{k}$. We then compute the Steiner tree $\mathcal{T}_1^{\text{hyb}}$ on this reweighed graph, which will be part of our final solution. In order to compute the other trees of the solution, we restore the original weights for the edges in G , except for those of



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. [6] and $t/\rho = 2.454$ (green dashed line) guaranteed by the algorithm of Karpinski et al. [13]. The table to the right lists the exact values for the ratio t/ρ .

$\mathcal{T}_1^{\text{hyb}}$, which are set to 0. Let G' be this reweighed graph. We then apply the same procedure to the instance $\text{HYBRIDMLST}(G', T_2, \dots, T_k)$ to eventually obtain $(\mathcal{T}_2^{\text{hyb}}, \dots, \mathcal{T}_k^{\text{hyb}})$. Note that $\text{HYBRIDMLST}(G, T) = \text{STEINERTREE}(G, T)$.

We observe that the hybrid approach for $k = 2$ produces an approximation with the same ratio as $\min(\text{TOP}, \text{BOT})$ and the composite, namely $\frac{4}{3}$, as we show in Theorem 3, while for larger values of k we have no guarantee. On the other hand, we will see in the following that this approach was quite stable in our tests, performing well on all the test suites, often better than all the other approaches.

We analyze the hybrid algorithm and prove that, for $k = 2$, it guarantees the same approximation ratio as $\min(\text{TOP}, \text{BOT})$ and the composite algorithm.

Namely, we start by computing a Steiner tree $E_{\text{BOT},2}$ that spans the set of terminals T_2 . Then, we construct a graph $G' = (V, E, c')$, where c' is defined as

$$c'(e) \mapsto \begin{cases} c(e)/2 & \text{if } e \in E_{\text{BOT},2} \\ c(e) & \text{otherwise.} \end{cases}$$

Now we compute the Steiner tree $E'_{\text{TOP},1}$ that spans T_1 on G' . As a last step, we construct a graph $G'' = (V, E, c'')$, where

$$c'' \mapsto \begin{cases} 0 & \text{if } e \in E'_{\text{TOP},1} \\ c(e) & \text{otherwise.} \end{cases}$$

and compute the Steiner-tree $E''_{\text{BOT},2}$ spanning T_2 on G'' .

Theorem 3. The hybrid algorithm yields a $\frac{4}{3}$ -approximation to the 2-level MLST.

Proof. The optimal solution for the 2-level MLST is lower-bounded as $(\text{BOT}_2 + \text{BOT}_1) + \text{TOP}_1 \leq \text{OPT}$, where BOT_ℓ is the cost of the optimal Steiner-tree on level ℓ and TOP_ℓ respectively. Let HYB_1 denote the cost of the subtree of $E''_{\text{BOT},2}$ spanning all terminals of T_1 . We can split $\text{HYB}_1 = \text{HYB}_{1,a} + \text{HYB}_{1,b}$, where $\text{HYB}_{1,a}$ denotes the cost of edges in $E''_{\text{BOT},2} \cap E_{\text{BOT},2}$ and $\text{HYB}_{1,b}$ denotes

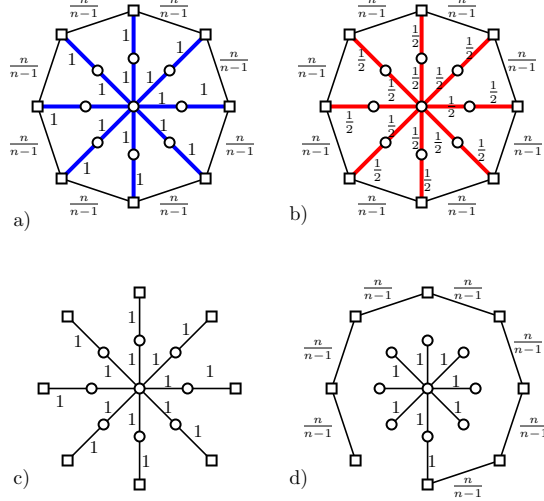


Figure 6: The approximation ratio $4/3$ of HYBRID for $k = 2$ levels is tight. a) A hub-and-spoke graph. Terminal nodes in T_1 are square shaped and terminal nodes in $T_2 \setminus T_1$ are circle shaped. The Steiner tree on the bottom level is the spike graph shown in blue. b) Reweighed graph G' . The Steiner tree on the top level is shown in red. c) Solution of the HYBRID heuristic algorithm which has cost $4n$. d) The optimal MLST solution of cost $3n + 1$. The ratio of the two approaches $4/3$.

the cost of $E''_{BOT,2} \setminus E_{BOT,2}$ in comparison to the initial solution on G . Since $E'_{BOT,1}$ corresponds to a Steiner-tree spanning T_1 on the reweighed graph G' , we have $\frac{1}{2}HYB_{1,a} + HYB_{1,b} \leq \frac{1}{2}BOT_1$, since $E_{BOT,1}$ spans T_1 and each edge supports half of its cost in the reweighed graph. Another tree that spans T_1 is $E_{TOP,1}$ and thereby $\frac{1}{2}HYB_{1,a} + HYB_{1,b} \leq TOP_1$ holds. The combination of the two inequalities suggest:

$$HYB_1 = HYB_{1,a} + HYB_{1,b} \leq \min\left\{\frac{1}{2}BOT_1, TOP_1\right\} + \frac{1}{2}HYB_{1,a}$$

Using edges of $E_{BOT,2}$, we can construct a tree that expands $E''_{BOT,1}$ to a tree which covers T_2 . Therefore, $HYB_2 \leq BOT_1 + BOT_2 - HYB_{1,a}$. Here $HYB_{1,a}$ is subtracted, since $HYB_{1,a}$ is already part of $E''_{BOT,1}$.

Since $\min\left\{\frac{1}{2}BOT_1, TOP_1\right\} \leq \frac{2}{3}\left(\frac{1}{2}BOT_1\right) + \frac{1}{3}TOP_1 = \frac{1}{3}(BOT_1 + TOP_1) \leq \frac{1}{3}(BOT_2 + BOT_1 + TOP_1) \leq \frac{1}{3}OPT$, we can deduce from the previous inequalities, that

$$\begin{aligned} 2HYB_1 + HYB_2 &\leq BOT_1 + BOT_2 + 2\min\left\{\frac{1}{2}BOT_1, TOP_1\right\} \\ &\leq OPT + \min\left\{\frac{1}{2}BOT_1, TOP_1\right\} \\ &\leq \frac{4}{3}OPT \end{aligned}$$

□

As shown in Figure 6, we can construct a graph for which a $\frac{4}{3}$ approximation ratio is realized through the hybrid heuristic. Therefore, the $\frac{4}{3}$ ratio is tight.

3 Exact Algorithm

We first provide an integer linear programming (ILP) based on cuts which is very easy to understand.

3.1 ILP based on Cuts

This is the standard ILP formulation of the single-level Steiner tree problem. With $\delta(S) := \{(u, v) \in E \mid u \in S, v \notin S\}$, we refer to the set of edges that have exactly one endpoint in S .

$$\begin{aligned} & \text{Minimize } \sum_{e \in E} c(e) \cdot x_e \\ & \text{subject to } \sum_{e \in \delta(S)} x_e \geq 1 \quad \forall S \subset V \text{ with } \emptyset \neq S \cap T \neq T \\ & \quad \quad \quad x_e \in \{0, 1\} \quad \forall e \in E \end{aligned}$$

This formulation generalizes to the k -level multilevel Steiner problem by introducing for every edge k variables that indicate if the corresponding edge is in E_i for $1 \leq i \leq k$.

$$\begin{aligned} & \text{Minimize } \sum_{l=1}^k \sum_{e \in E} c(e) \cdot x_{e,l} \\ & \text{subject to } \sum_{e \in \delta(S)} x_{e,l} \geq 1 \quad \forall S \subset V \text{ with } \emptyset \neq S \cap T_l \neq T_l \text{ and } 1 \leq l \leq k \\ & \quad \quad \quad x_{e,l+1} \geq x_{e,l} \quad \forall e \in E \text{ and } 1 \leq l < k \\ & \quad \quad \quad x_{e,l} \in \{0, 1\} \quad \forall e \in E \text{ and } 1 \leq l \leq k \end{aligned}$$

3.2 ILP based on Multi-commodity Flow

Recall the well-known flow formulation for STP [2, 18]. 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. Let s be a fixed terminal node, the *source*. An ILP formulation of STP using multi-commodity flow [23] is as follows:

$$\text{Minimize } \sum_{(i,j) \in E} c_{ij} y_{ij} \tag{3}$$

$$\text{subject to } \sum_{v_h \in N(v_i)} x_{ih}^p - \sum_{v_j \in N(v_i)} x_{ji}^p = \begin{cases} 1 & i = s \\ -1 & i \in T \setminus \{s\} \\ 0 & i \in V \setminus T \end{cases} \quad \text{for } i \in V \tag{4}$$

$$x_{ij}^p \leq y_{ij} \quad (i, j) \in E, v_p \in T \tag{5}$$

$$x_{ij}^p \geq 0 \quad (i, j) \in E, v_p \in T \tag{6}$$

$$y_{ij} \in \{0, 1\} \quad (i, j) \in E. \tag{7}$$

where y_{ij} is a binary variable indicating whether or not edge (i, j) is included in the solution and $N(v)$ denotes the neighbors of v . The variable x_{ij}^p is the amount of commodity p (the amount of flow between vertices v_1 and v_p) on edge (i, j) . Constraint 6 indicates that flow on an edge is allowed only if the edge is included in the solution. Constraint 4 dictates that one unit of commodity p be routed between vertices v_1 and v_p . In constraint 6 we can view y_{ij} as the flow capacity for commodity p on edge (i, j) . Constraints 4–7 indicate that a feasible solution must have a path of

edges (i.e., $y_{ij} = 1$) between vertex v_1 and every vertex belonging to L . Thus we represent the connectivity of the Steiner tree problem via an embedded multi-commodity network flow problem.

In MLST, if an edge is selected on level ℓ , it must be selected in all lower levels $\ell + 1, \dots, k$. We introduce binary variables y_{uv}^ℓ , where $y_{uv}^\ell = 1$ if edge uv is selected on level ℓ , and additional flow variables x_{uv}^ℓ for each level ℓ . We constrain that the graph on level ℓ is a subgraph of the graph on level $\ell + 1$ as follows:

$$y_{uv}^{\ell+1} \geq y_{uv}^\ell \quad \text{for } \ell \in \{1, 2, \dots, k-1\} \text{ and } (u, v) \in E \quad (8)$$

We also modify the objective function in the natural way:

$$\text{Minimize } \sum_{\ell=1}^k \sum_{uv \in E} c(u, v) \cdot y_{uv}^\ell \quad (9)$$

3.3 ILP based on Single Flow

We now model the problem using single flow instead of multi-commodity flow which reduces the number of variables:

$$\text{Minimize } \sum_{(u,v) \in E} c(u, v) \cdot y_{uv} \quad (10)$$

$$\text{subject to } \sum_{vw \in E} x_{vw} - \sum_{uv \in E} x_{uv} = \begin{cases} |T| - 1 & \text{if } v = s \\ -1 & \text{if } v \in T \setminus \{s\} \\ 0 & \text{else} \end{cases} \quad \text{for } v \in V \quad (11)$$

$$x_{uv} \leq (|T| - 1) \cdot y_{uv} \quad (12)$$

$$x_{uv} \geq 0 \quad (13)$$

$$y_{uv} \in \{0, 1\} \quad (14)$$

We can extend this formulation for MLST as we have done in the ILP based on multi-commodity flow. Note that while the ILP based on cuts is very easy to understand, it considers all $O(2^{|V|})$ subsets of V and there is a constraint for every subset. Hence with that formulation, the number of constraints is exponential in the size of the graph. In the other two formulations, the number of constraints is polynomial to the size of the graph. We select this single-flow variant as it uses only one flow per level, while the multi-commodity variant requires multiple flows, one for each terminal. Since the number of constraints increases with the number of flows, that formulation is more expensive.

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 [9], random geometric [17], Watts–Strogatz [21], and Barabási–Albert [3]. These networks are very well studied in the literature [16]. A brief overview of each network generation model is as the following.

The Erdős–Renyi model, $ER(n, p)$, assigns an edge to every possible pair among n nodes with probability p . It is well-known that an instance of $ER(n, p)$ with $p = (1 + \epsilon) \frac{\ln n}{n}$ is almost surely connected for $\epsilon > 0$ [9]. For our experiment n vary from 10 to 100 and p is equal to .25.

In the random geometric model, $\text{RG}(n, r_c)$, n points are uniformly randomly distributed in the unit square $[0, 1]^2$. In the output graph, each node corresponds to a single point, and nodes are connected to each other if their Euclidean distance is not larger than a threshold r_c . For $r_c = \sqrt{\frac{(1+\epsilon)\ln n}{\pi n}}$ with $\epsilon > 0$, the synthesized graph is almost surely connected [17]. The value of r_c in our experiment is 1.62.

The Watts–Strogatz model, $\text{WS}(n, K, \beta)$, initially creates a ring lattice of constant degree K , and then rewires each edge with probability $0 \leq \beta \leq 1$ while avoiding self-loops or duplicate edges. Interestingly, the Watts–Strogatz model generates graphs that have the small-world property while having high clustering coefficient [21]. In our experiment, the values of K and β are equal to 6 and 0.2 respectively.

The Barabási–Albert model, $\text{BA}(m_0, m)$, uses a preferential attachment mechanism to generate a growing scale-free network. The model starts with a graph of m_0 nodes. Then, each new node connects to $m \leq m_0$ existing nodes with probability proportional to their instantaneous degree. The BA model generates networks with power-law degree distribution, i.e. a few nodes becomes hubs with extremely large degree [3]. This model is a network growth model. In our experiments, we let the network grow until a desired network size n is attained. We vary m_0 from 10 to 100 in our experiment. We keep the value of m equal to 5.

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 graph generated, we solved the MLST for $k = 2, 3, 4, 5$. We adopted two strategies for selecting the terminals for each k -level MLST problem: *linear* and *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 $|T_{i+1}|/|T_i| \approx |T_i|/|T_{i-1}|$.

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

Algorithms and Outputs. We implemented the bottom-up, top-down, hybrid, and composite approaches described in Section 2. Finally, we implemented the simple 4-approximation algorithm by Charikar et al. [6] for the QoS Multicast Tree problem and the ILP described in Section 3 to evaluate the heuristics.

We used 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. However for larger graphs, the maximum time limit for a single task is not enough if the number of graphs is 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 (CMP), the guaranteed performance heuristic ($\text{CMP}(\mathcal{Q}^*)$) heuristic, the hybrid heuristic (HYBRID), and the simple 4-approximation Quality-of-Service heuristic (QoS) given by Charikar et al. [6]. For the STP computation we used the 2-approximation algorithm of Gilbert and Pollak [10].

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 , $\text{CMP}(\mathcal{Q}^*)$: $2k$, HYBRID: 2^k , and QoS: k .

Results.

Figures 7–10 show how the heuristics perform with regards to the graph size $|V|$, for the ER, RG, WS, and BA network models respectively. Since multiple experiment input instances share the same network size, we have shown minimum, maximum, and mean values. In most of the cases, the average ratio increases as the number of vertices increases. The composite, hybrid, and composite-optimum heuristics perform very well in practice.

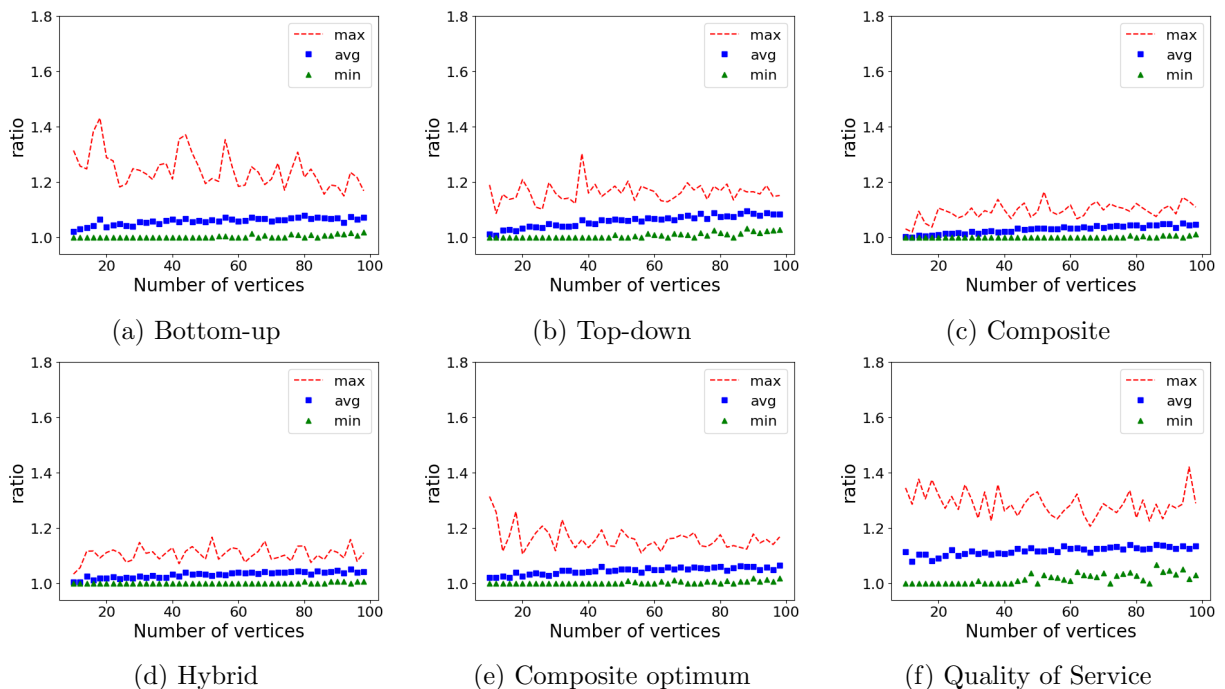


Figure 7: Erdős–Rényi

Next, we examined how the performance ratio of the heuristics changes with the number k of levels. In our experiments, $k = 2, \dots, 5$. We show the results using box plots in Figure 11. As the number of levels increases, the ratio increases in general.

Finally, we examined how the performance of the heuristics compare with the exact solution as the terminal selection method in levels changes. In our experiments, the node distribution in levels vary between LINEAR and EXPONENTIAL. We show the results using box plots in Figure 12. In most of the cases the ratio is higher when the node distribution is EXPONENTIAL.

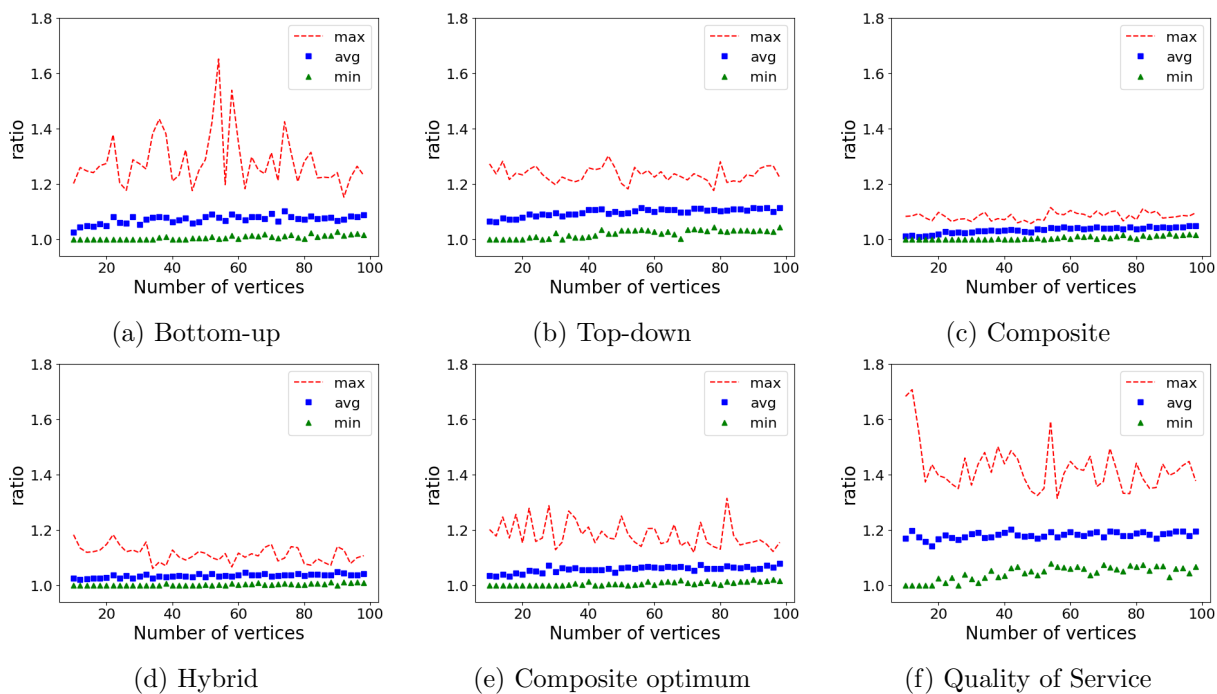


Figure 8: Geometric

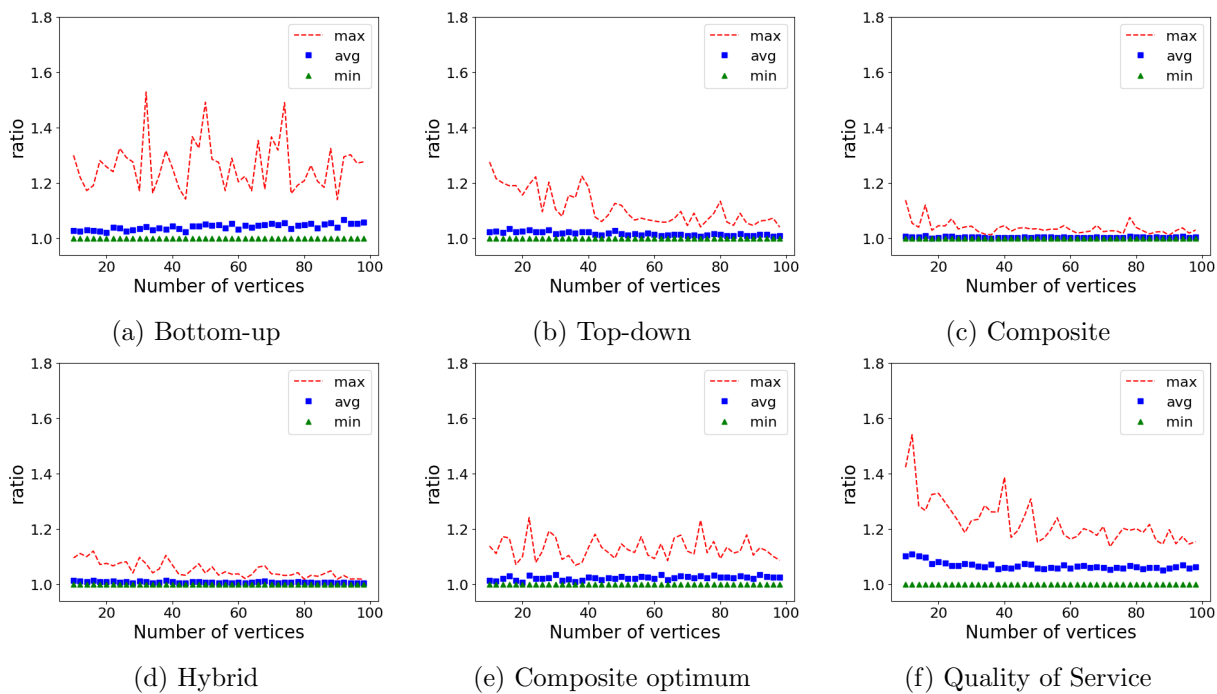


Figure 9: Watts–Strogatz

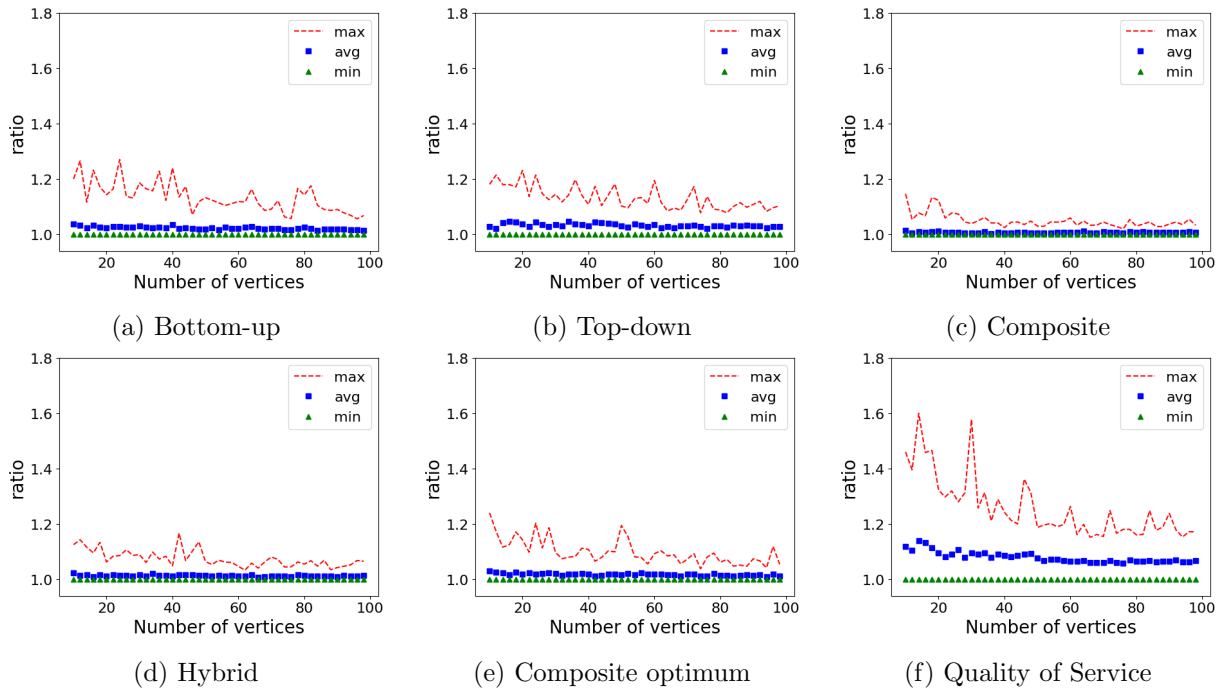
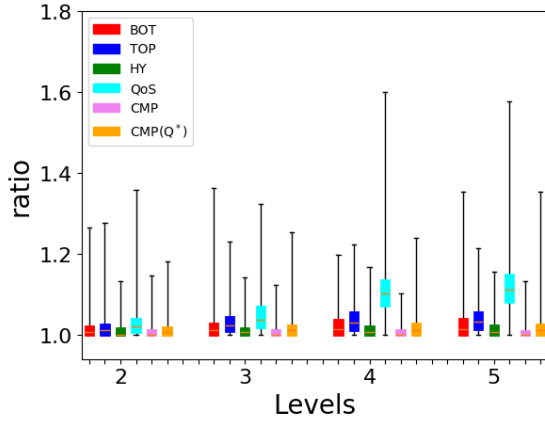


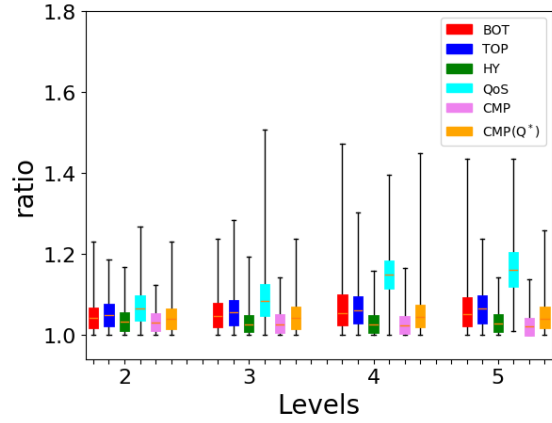
Figure 10: Barabási-Albert

5 Conclusions

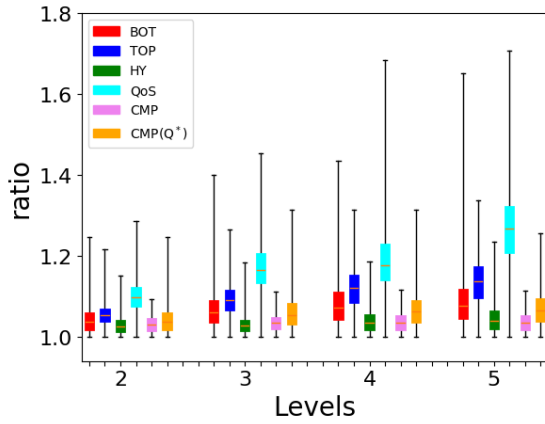
We presented multiple heuristics for the MLST problem and analyzed them both theoretically and experimentally. Natural open problems include determining inapproximability results for the k -level MLST, determining a closed-form expression for the approximation ratio t given by the composite heuristic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



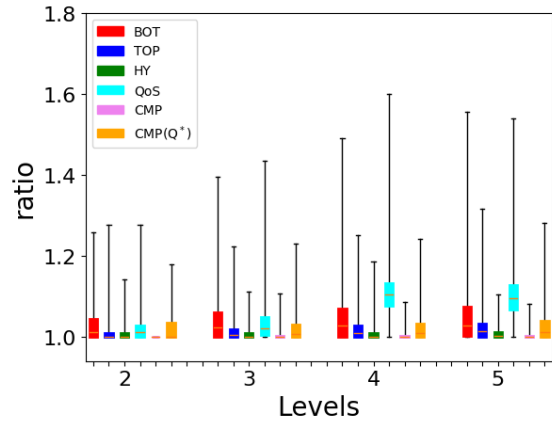
(a) Barabási-Albert



(b) Erdős-Rényi

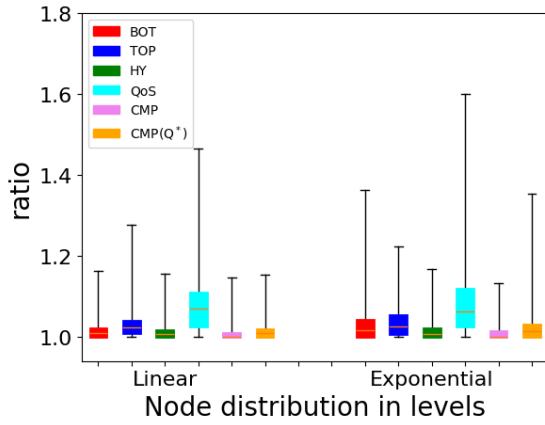


(c) Geometric

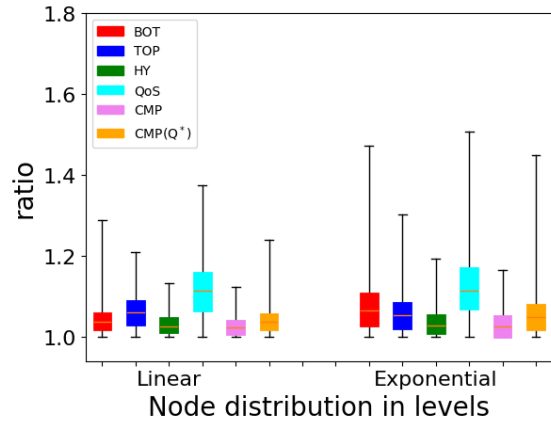


(d) Watts-Strogatz

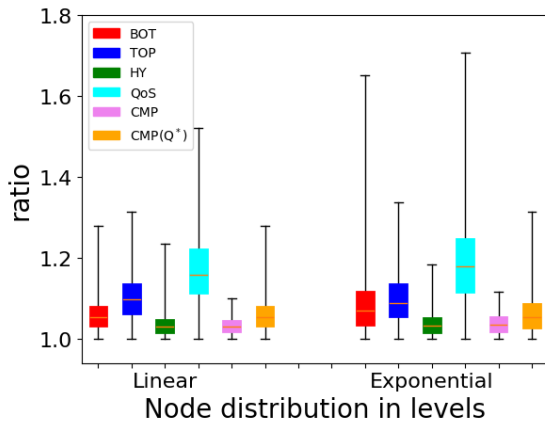
Figure 11: Performance as a function of the number k of levels.



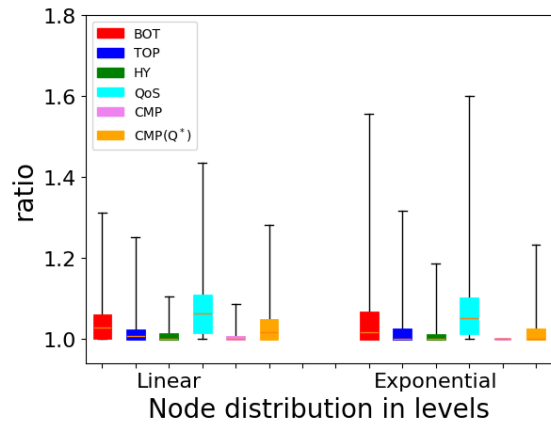
(a) Barabási-Albert



(b) Erdős-Rényi



(c) Geometric



(d) Watts-Strogatz

Figure 12: Heuristic performance as a function of the terminal selection method.

References

- [1] Sanjeev Arora. Polynomial time approximation schemes for Euclidean Traveling Salesman and other geometric problems. *J. ACM*, 45(5):753–782, 1998.
- [2] 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.
- [3] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.
- [4] Marshall Bern and Paul Plassmann. The Steiner problem with edge lengths 1 and 2. *Inform. Process. Lett.*, 32(4):171–176, 1989.
- [5] 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.
- [6] 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.
- [7] Miroslav Chlebík and Janka Chlebíková. The Steiner tree problem on graphs: Inapproximability results. *Theoret. Comput. Sci.*, 406(3):207–214, 2008.
- [8] 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.
- [9] Paul Erdős and Alfréd Rényi. On random graphs, i. *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959.
- [10] Edgar N. Gilbert and Henry O. Pollak. Steiner minimal trees. *SIAM J. Appl. Math.*, 16(1):1–29, 1968.
- [11] Mathias Hauptmann and Marek Karpinski (eds.). A compendium on Steiner tree problems, 2015.
- [12] 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.
- [13] 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.
- [14] Prakash Mirchandani. The multi-tier tree problem. *INFORMS J. Comput.*, 8(3):202–218, 1996.
- [15] 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.
- [16] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

- [17] Mathew Penrose. *Random geometric graphs*. Number 5. Oxford university press, 2003.
- [18] Tobias Polzin and Siavash Vahdati Daneshmand. A comparison of Steiner tree relaxations. *Discrete Appl. Math.*, 112(1):241–261, 2001.
- [19] Hans Jürgen Prömel and Angelika Steger. *The Steiner Tree Problem*. Vieweg and Teubner Verlag, 2002.
- [20] Gabriel Robins and Alexander Zelikovsky. Tighter bounds for graph Steiner tree approximation. *SIAM J. Discrete Math.*, 19(1):122–134, 2005.
- [21] Duncan J Watts and Steven H Strogatz. Collective dynamics of ‘small-world’ networks. *nature*, 393(6684):440, 1998.
- [22] Pawel Winter. Steiner problem in networks: A survey. *Networks*, 17(2):129–167, 1987.
- [23] Richard T. Wong. A dual ascent approach for Steiner tree problems on a directed graph. *Mathematical Programming*, 28(3):271–287, 1984.
- [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.