Approximation Algorithms for Priority Steiner Tree Problems

Faryad Darabi Sahneh, Stephen Kobourov, and Richard Spence

University of Arizona, Tucson, AZ 85721, USA **

Abstract. In the Priority Steiner Tree (PST) problem, we are given an undirected graph $G = (V, E)$ with a source $s \in V$ and terminals $T \subseteq V \setminus \{s\}$, where each terminal $v \in T$ requires a nonnegative priority $P(v)$. The goal is to compute a minimum weight Steiner tree containing edges of varying rates such that the path from s to each terminal v consists of edges of rate greater than or equal to $P(v)$. The PST problem with k priorities admits a $\min\{2\ln|T|+2, k\rho\}$-approximation [Charikar et al., 2004], and is hard to approximate with ratio $c\log\log n$ for some constant c [Chuzhoy et al., 2008]. In this paper, we first strengthen the analysis provided by [Charikar et al., 2004] for the $(2\ln|T|+2)$-approximation to show an approximation ratio of $\lceil\log_2|T|\rceil + 1 \leq 1.443\ln|T|+2$, then provide a very simple, parallelizable algorithm which achieves the same approximation ratio. We then consider a more difficult node-weighted version of the PST problem, and provide a $(2\ln|T|+2)$-approximation using extensions of the spider decomposition by [Klein & Ravi, 1995]. This is the first result for the PST problem in node-weighted graphs. Moreover, the approximation ratios for all above algorithms are tight.

Keywords: priority Steiner tree · approximation algorithms · network design

1 Introduction

We consider generalizations of the Steiner tree and node-weighted Steiner tree (NWST) problems in graphs where the terminals T possess varying priority or quality of service (QoS) requirements, in which we seek to connect the terminals using edges of the appropriate rate or better. These problems have applications in multimedia and electric power distribution [4, 21, 27], multi-level graph visualization [1], and other network design problems where a source or root is to be connected to a set of heterogeneous receivers possessing different bandwidth or priority requests. We define a Priority Steiner Tree (PST) as follows:

Definition 1 (Priority Steiner Tree (PST)). *Given an undirected graph $G = (V, E)$, a source $s \in V$, and terminals $T \subseteq V \setminus \{s\}$, where each terminal $v \in T$ requires a nonnegative priority $P(v)$, a* PST *is a tree $\mathcal{T} \subseteq G$ rooted at s containing edges of varying rates such that for all terminals $v \in T$, the s–v path in $\mathcal{T}$ consists of edges of rate $P(v)$ or higher.*

** Supported in part by NSF grants CCF-1740858, CCF-1712119, and DMS-1839274.

We denote by k the number of distinct priorities. Vertices in $V \setminus (T \cup \{s\})$ have zero priority but may be included in $\mathcal{T}$. Let $w(e, r)$ denote the weight of edge e at rate r. We assume $w(e, 0) = 0$ and $w(e, r_1) \leq w(e, r_2)$ for all $0 \leq r_1 \leq r_2$ and edges e (i.e., higher-rate edges weigh at least as much as lower-rate edges). The weight of a PST $\mathcal{T}$ is the sum of the weights of the edges in $\mathcal{T}$ at their respective rates, namely $w(\mathcal{T}) := \sum_{e \in E(\mathcal{T})} w(e, R(e))$.

*Problem 1 (*Priority Steiner Tree *problem).* Given a graph $G = (V, E)$, source s, terminals $T \subseteq V$, priorities $P(\cdot)$, and edge weights $w : E \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, compute a PST $\mathcal{T}$ with minimum weight.

While Problem 1 in the case where edge weights are proportional to rate (i.e., $w(e, r) = r \cdot w(e, 1)$ for all $e \in E$ and $r \geq 0$) admits $O(1)$–approximations [1,6,18], the best known approximation ratio for Priority Steiner tree with arbitrary weights is $\min\{2 \ln |T| + 2, k\rho\}$ by Charikar et al. [6] (see Section 2). On the other hand, Chuzhoy et al. [9] show that Priority Steiner tree cannot be approximated with ratio $c \log \log n$ for some constant c unless NP $\subseteq$ DTIME($n^{O(\log \log \log n)}$), even with unit edge weights[1].

In Section 3, we introduce a node-weighted variant of Priority Steiner Tree, called Priority NWST (Definition 2). Here we assume edges have zero weight, as an instance with edge and vertex weights can be converted to an instance with only vertex weights by subdividing each edge uv into two edges uw, wv and assigning the weight of edge uv to vertex w.

Definition 2 (Priority Node-Weighted Steiner Tree (PNWST)). *Given an undirected graph $G = (V, E)$, source s, and terminals $T \subseteq V \setminus \{s\}$, where each terminal $v \in T$ requires a nonnegative priority $P(v)$, a* priority node-weighted Steiner tree *(PNWST) is a tree $\mathcal{T}$ rooted at s containing vertices of varying rates $R(v)$ such that for all terminals $v \in T$, the s–v path in $\mathcal{T}$ consists of vertices of rate $P(v)$ or higher.*

In particular, we require $R(v) \geq P(v)$ for all $v \in T$. Further, we can assume w.l.o.g. that the path from s to each terminal uses vertices of non-increasing rate (see Definition 3). As in the NWST problem, it is conventional to also assume terminals have zero weight, as they must be included in any feasible solution; thus, we assume $w(v, r) = 0$ for $0 \leq r \leq P(v)$ and $w(v, r_1) \leq w(v, r_2)$ for all $0 \leq r_1 \leq r_2$. The weight of a PNWST $\mathcal{T}$ with *vertex* rates $R(\cdot)$ is $w(\mathcal{T}) := \sum_{v \in V(\mathcal{T})} w(v, R(v))$.

*Problem 2 (*Priority NWST *problem).* Given a graph $G = (V, E)$, source s, terminals $T \subseteq V \setminus \{s\}$, vertex priorities $P(\cdot)$, and vertex weights $w : V \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, compute a PNWST $\mathcal{T}$ with minimum weight.

[1] We remark that the formulation of Priority Steiner Tree given in [9] is slightly more specific; each edge has a single weight c_e as well as a quality of service (priority) $Q(e)$ on input, and the goal is to compute a Steiner tree such that the path from root to each terminal v uses edges of quality of service greater than or equal to $P(v)$.

The Priority NWST problem generalizes the NWST problem, and hence cannot be approximated with ratio $(1-o(1))\ln|T|$ unless P = NP [12,13,19], via a reduction from the set cover problem. In Section 3, we show that the Priority NWST problem admits a $2\ln(|T|+1)$–approximation (Theorem 2) using extensions of the spider decomposition given by Klein and Ravi [19] to accommodate the priority constraints of the Priority NWST problem. The generalization is not immediately obvious; in particular it is not immediate whether an instance of Priority NWST can be formulated as an instance of NWST. However, NWST and Priority NWST can be easily reduced to Steiner arborescence (or directed Steiner tree), which admits a quasi-polynomial $O\left(\frac{\log^2|T|}{\log\log|T|}\right)$-approximation [14].

Notation. A graph $G=(V,E)$ with $n=|V|$ and $m=|E|$ is undirected and connected, unless stated otherwise. Given terminals $u,v\in T$ for the PST problem, denote by $\sigma(u,v)$ the weight of a minimum weight u–v path in G using edges of rate $\min\{P(u),P(v)\}$, and let p_{uv} denote such a path. For terminals $u,v\in T$ in the Priority NWST problem, we define $\sigma(u,v)$ to be the weight of a minimum u–v path using vertices of rate $\min\{P(u),P(v)\}$ not including the endpoints u and v, and similarly define $\sigma_b(u,v)$ to be the weight of a minimum weight vertex-weighted path using vertices of rate b, so that $\sigma(u,v)=\sigma_{\min\{P(u),P(v)\}}(u,v)$. In particular, we have $\sigma_b(v,v)=0$. Note that σ is symmetric but does not satisfy the triangle inequality, and is not a metric. Let ρ denote an approximation ratio for the (edge-weighted) Steiner tree problem, and let STEINER(n) denote the running time of such an approximation algorithm on an n-vertex graph. We denote by OPT the weight of a min-weight PST or PNWST. Lastly, for $n\in\mathbb{Z}^+$, we denote by $[n]$ the set $\{1,2,\ldots,n\}$.

1.1 Related work

The Steiner tree problem in graphs has been studied in a wide variety of contexts; see the compendium [16]. The (edge-weighted) Steiner tree problem admits a folklore $2\left(1-\frac{1}{|T|}\right)$–approximation, and is approximable with ratio $\rho=\ln 4+\varepsilon\approx 1.387$ [5], but NP-hard to approximate with ratio $\frac{96}{95}\approx 1.01$ [8]. As stated previously, NWST cannot be approximated with ratio $(1-o(1))\ln|T|$ unless P = NP [12,13,19], but algorithms with logarithmic approximation ratio exist. Klein and Ravi [19] give a $2\ln|T|$–approximation for NWST, which was improved to $1.61\ln|T|$ and a less practical $(1.35+\varepsilon)\ln|T|$ by Guha and Khuller [15]. Demaine et al. [11] give an $O(1)$–approximation for NWST when the input graph G is H–minor free, and a 6-approximation when G is planar. Naor et al. [23] give a randomized $O(\log n\log^2|T|)$-approximation algorithm for the online version.

The (edge-weighted) Priority Steiner tree problem and variants thereof have been studied under various other names including Hierarchical Network Design [10], Multi-Level (or k-Level) Network Design [4], Multi-Tier Tree [22], Grade of Service Steiner Tree [28], Quality of Service Multicast Tree [6,18], and Multi-Level Steiner Tree [1,2]. Earlier results on this problem typically consider

a small number of priorities or restricted definition of weight [4, 10]. In the special case where edge weights are proportional to rate, Charikar et al. [6] give the first $O(1)$–approximations with approximation ratios 4ρ and $e\rho \approx 4.214$ (with $\rho \approx 1.55$ [25]) independent of the number of priorities k. Karpinski et al. [18] give a slightly stronger variant of the $e\rho$–approximation [6] which achieves approximation ratio 3.802. Ahmed et al. [1] give an approximation ratio of $2.351\rho \approx 3.268$ for $k \leq 100$. Xue et al. [28] consider this problem where the terminals are embedded in the Euclidean plane, and give $\frac{4}{3}\rho$ (resp. $\frac{5+4\sqrt{2}}{7}\rho \approx 1.522\rho$)–approximations for two (resp. three) different priorities. Integer programming formulations have been proposed and evaluated over realistic problem instances [1, 24].

If edge weights are not necessarily proportional to rate, Charikar et al. [6] give a simple $\min\{2\ln|T|+2, k\rho\}$-approximation (see Section 2), which remains the best known to date. Recently, Ahmed et al. [2] proposed an approximation based on Kruskal's MST algorithm which achieves the same approximation ratio, and provided an experimental study comparing the two methods. Chuzhoy et al. [9] show that Priority Steiner tree cannot be approximated with ratio $c\log\log n$ for some constant c unless NP $\subseteq$ DTIME$(n^{O(\log\log\log n)})$. Angelopoulos [3] showed that every deterministic online algorithm for online Priority Steiner tree has ratio $\Omega(\min\{k\log\frac{|T|}{k}, |T|\})$. Interestingly, no node-weighted variant of Priority Steiner tree has been studied in existing literature. However, a related problem is the (single-source) node-weighted buy-at-bulk problem (NSS-BB) studied by Chekuri et al. [7], who show a $3H_{|T|} = O(\log|T|)$–approximation for NSS-BB by giving a randomized algorithm then derandomizing it using an LP relaxation, where $H_n = \frac{1}{1} + \frac{1}{2} + \ldots + \frac{1}{n}$ is the n^{th} harmonic number.

1.2 Our results

In Section 2, we strengthen the analysis of the simple $(2\ln|T|+2)$-approximation (Algorithm 1) by Charikar et al. [6] to show that it is a $\lceil\log_2|T|\rceil + 1 \leq (1.443\ln|T|+2)$-approximation. We then give a parallelizable algorithm (Algorithm 2) with the same approximation ratio that does not require that terminals be connected sequentially or in a particular order. This contrasts with the inherently serial Algorithm 1 [6], where the shortest path for each terminal depends on the partial PST computed at the previous iteration.

Theorem 1. *Algorithm 1 [6] is a* $(\lceil\log_2|T|\rceil+1)$*-approximation for* Priority Steiner tree *with running time* $O(nm+n^2\log n)$*, and there is a parallelizable algorithm for* Priority Steiner tree *with the same approximation ratio.*

Moreover, the approximation ratio is tight up to a factor of 2, as there exists an input graph in which Algorithms 1–2 may output a PST with weight $\frac{1}{2}\log_2|T|+1$ times the optimum [17]. In Section 3, we show the following result for Priority NWST:

Theorem 2. *There exists a* $2\ln(|T|+1)$*–approximation algorithm for* Priority NWST *with running time* $O(n^4k\log n)$.

This is the first known approximation algorithm for Priority NWST, and is the main technical contribution of this paper. The analysis extends the spider decomposition of Klein and Ravi [19] in their greedy $(2\ln|T|)$–approximation for the NWST problem, to accommodate priority constraints in the Priority NWST problem. Note the additional $+1$ arises as we do not consider the source s a terminal. Proofs omitted for space are in the arXiv version [26].

2 Priority Steiner Tree: Two logarithmic approximations

We first review the greedy $\min\{2\ln|T|+2, k\rho\}$ approximation for Priority Steiner tree given by Charikar et al. [6]. This returns the better solution of two sub-algorithms; we focus primarily on the $(2\ln|T|+2)$-approximation (Algorithm 1). This algorithm sorts the terminals T from highest to lowest priority. Then for $i = 1, \ldots, |T|$, the i^{th} terminal v_i in the sorted list is connected to the existing tree (containing the source s) using a minimum weight path of rate $P(v_i)$. The weight of this path is the *connection cost* of v_i. Cycles can be removed in the end by removing an edge from each cycle with the lowest rate.

Algorithm 1 $R(\cdot) = \text{QoSMT}(\text{graph } G, \text{ priorities } P, \text{ edge weights } w, \text{ source } s)$ [6]

1: Sort terminals T by decreasing priority $P(\cdot)$
2: Initialize $V' = \{s\}$, $R(e) = 0$ for $e \in E$
3: **for** $i = 1, 2, \ldots, |T|$ **do**
4: Connect i^{th} terminal v_i to V' using minimum weight path p_i of rate $P(v_i)$
5: $R(e) = P(v_i)$ for $e \in p_i$
6: $V' = V' \cup V(p_i)$
7: Remove lowest-rate edge from each cycle
8: **return** edge rates $R(\cdot)$

Algorithm 1 is based on a $(\log_2|T|)$-approximation for an online Steiner tree problem analyzed by Imase and Waxman [17]; however, Charikar et al. [6] give a simpler analysis which proves a weaker approximation ratio of $2\ln|T|+2$, based on the following lemma:

Lemma 1 ([6]). *For $1 \le x \le |T|$, the x^{th} most expensive connection cost incurred by Algorithm 1 is at most $\frac{2\text{OPT}}{x}$.*

Lemma 1 implies the weight of the PST is at most $2\text{OPT}\left(\frac{1}{1}+\frac{1}{2}+\ldots+\frac{1}{|T|}\right) = 2\text{OPT}H_{|T|} \le (2\ln|T|+2)\text{OPT}$. Line 4 can be executed by running Dijkstra's algorithm from v_i with edge weights $w(\cdot, P(v_i))$ until reaching a vertex in V'; hence Algorithm 1 runs in $O(nm + n^2\log n)$ time.

We strengthen the analysis by Charikar et al. [6] to prove an approximation ratio of $\lceil\log_2|T|\rceil + 1$, thus matching the result for the online Steiner tree

problem [17]. Instead of an upper bound on the x^{th} most expensive connection cost, we establish a bound on the $\frac{|T|}{2}$ *least* expensive connection costs; a similar technique was used in [20] for a bicriteria diameter-constrained Steiner tree problem. For simplicity, we assume w.l.o.g. $|T|$ is a power of 2; this can be done by adding up to one dummy terminal of priority 1 to each terminal, connected with a zero-weight edge.

Lemma 2. *The sum of the* $\frac{|T|}{2}$ *least expensive connection costs incurred by Algorithm 1 is at most* OPT.

Theorem 3. *Algorithm 1 is a* $(\lceil \log_2 |T| \rceil + 1)$*-approximation for* Priority Steiner tree.

Lemma 2 is proved by considering pairs of consecutive terminals in a depth-first traversal of the optimum PST $\mathcal{T}^*$, and Theorem 3 is proved by applying Lemma 2 $\lceil \log_2 |T| \rceil + 1$ times.

In the following, we give a simpler, parallelizable algorithm for Priority Steiner tree which achieves the same approximation ratio of $\lceil \log_2 |T| \rceil + 1$. For simplicity we assume $P(s) = \infty$ and every (non-source) terminal has a different priority; ties between terminals of the same priority can be broken arbitrarily. The idea is to connect each terminal v to the "closest" terminal or source with a greater priority than v. Specifically, for $v \in T$, find a vertex $u \in T \cup \{s\}$ with $P(u) > P(v)$ which minimizes $\sigma(u, v)$, and connect v to u with edges of rate $P(v)$. This can be done by executing Dijkstra's algorithm from v using edge weights $w(\cdot, P(v))$ and stopping once we find a vertex with a greater priority than v. Moreover, this algorithm is parallelizable as the corresponding path for each terminal can be found in parallel. The weight of connecting v to its parent u is the *connection cost* of v. As before, cycles can be removed in the end by removing an edge from each cycle with the lowest rate.

Algorithm 2 $R(\cdot) = \text{PST}(\text{graph } G,\ \text{priorities } P,\ \text{edge weights } w,\ \text{source } s)$

Initialize $R(e) = 0$ for $e \in E$
for $v \in T$ **do**
    Find $u \in T \cup \{s\}$ with $P(u) > P(v)$ such that $\sigma(u, v)$ is minimized
    $R(e) = \max\{R(e), P(v)\}$ for $e \in p_{vu}$
Remove lowest-rate edge from each cycle
return edge rates $R(\cdot)$

Algorithm 2 produces a valid PST which spans all terminals, since there is a path from each terminal v to the source using edges of rate $P(v)$ or higher. Moreover, Lemma 1 and Theorem 3 extend easily:

Lemma 3. *The sum of the* $\frac{|T|}{2}$ *least expensive connection costs incurred by Algorithm 2 is at most* OPT.

Theorem 4. *Algorithm 2 is a* $(\lceil \log_2 |T| \rceil + 1)$*-approximation for* Priority Steiner tree.

One main difference compared to Algorithm 1 [6] is that Algorithm 2 is not required to connect the terminals sequentially, or even by order of priority. Further, unlike Algorithm 1, Algorithm 2 is not dependent on the solution computed at the previous iteration. If $k \ll |T|$, a simple $k\rho$-approximation given by Charikar et al. [6] is to compute a ρ-approximate Steiner tree over the terminals of each priority separately, taking $O(k \cdot \text{STEINER}(n))$ time. Executing both approximations and taking the better of the two solutions yields a $\min\{\lceil \log_2 |T| \rceil + 1, k\rho\}$-approximation as desired.

3 An $O(\log |T|)$-approximation for Priority NWST

We remark that the analysis of Algorithms 1-2 does not extend to Priority NWST; one can construct an example input graph in which Algorithm 1 or 2 (considering minimum weight node-weighted paths) returns a poor NWST with weight $\Omega(|T|)\text{OPT}$. In this section, we extend the $(2 \ln |T|)$-approximation by Klein and Ravi [19] which maintains a collection of trees, and greedily merges a subset of these trees at each iteration to minimize a cost-to-connectivity ratio (Algorithm 3). For Priority NWST, we need to ensure that the priority constraint is always maintained throughout the construction process. To this end, we first define a *rate tree*:

Definition 3 (Rate tree). *Let* $G = (V, E)$*, and let* $\mathcal{T}_r$ *be a subtree of* G *(not necessarily a Steiner or spanning tree of* G*) which includes vertex* r*. Let* $R : V \to \mathbb{R}_{\geq 0}$ *be a function which assigns rates to the vertices in* G*. We say that* $\mathcal{T}_r$ *is a* rate tree *rooted at* r *if, for all* $v \in V(\mathcal{T}_r) \setminus \{r\}$*, the path from* r *to* v *in* $\mathcal{T}_r$ *consists of vertices of non-increasing rate.*

The main idea of Algorithm 3 is to maintain a set (not necessarily a forest) of rate trees. By simply connecting the *roots* of the rate trees with paths of appropriate vertex rates, we can satisfy the priority constraints.

Another challenge to tackle involves properly devising a definition of weight when greedily merging rate trees at each iteration. The greedy NWST algorithm by Klein and Ravi [19] simply sums the weights from a root vertex to each terminal. In our algorithm, we cannot simply connect the root of a rate tree to other roots of other rate trees of lower or equal priority and compute the weight similarly. This is due to a technical challenge needed for the analysis of the algorithm (see Section 3.2) that it is not possible, in general, to perform a spider decomposition (similar to [19]) on a rate tree such that paths from the center to leaves have non-increasing rates. To overcome this challenge, we introduce the notion of *rate spiders* and prove the existence of a *rate spider decomposition*, which further guides us to properly define weight computations at each iterative step.

3.1 Algorithm description

In the following, let $p_1 < p_2 < \ldots < p_k$ denote the k vertex priorities. Initialize a set $\mathcal{F}$ (not necessarily a forest) of $|T|+1$ rate trees so that each terminal $v \in T$, including the source s, is a singleton rate tree whose root is itself. Initialize vertex rates $R(v) = P(v)$ for $v \in T$, $R(s) = p_k$, and $R(v) = 0$ for $v \notin T \cup \{s\}$. While $|\mathcal{F}| > 1$, the construction proceeds iteratively as follows. Each iteration consists of greedily selecting the following:

- a rate tree $\mathcal{T}_r \in \mathcal{F}$ rooted at r, called the *root tree*
- a special vertex $v \in V$ called the *center* (note v could equal r)
- a real number $b \leq P(r)$ representing the rate which v is "upgraded" to
- a nonempty subset $\mathcal{S} = \{\mathcal{T}_{r_1}, \ldots, \mathcal{T}_{r_{|\mathcal{S}|}}\} \subset \mathcal{F}$ of rate trees where $\mathcal{T}_r \notin \mathcal{S}$, and $P(r_j) \leq b$ for all roots r_j associated with the rate trees in $\mathcal{S}$

By connecting r to the center v using vertices of rate b, upgrading $R(v)$ to b, then connecting v to the root of each rate tree $\mathcal{T}_{r_j} \in \mathcal{S}$ using vertices of rate $P(r_j)$, we can replace the $|\mathcal{S}|+1$ rate trees in $\mathcal{F}$ with a new rate tree $\mathcal{T}_r^{\text{new}}$ rooted at r (see Figure 1).

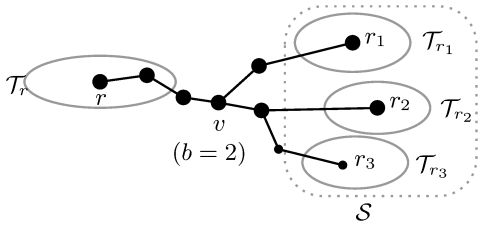

Fig. 1: Illustration of an iteration step in Algorithm 3 with $P(r) = 2$, $b = 2$, $P(r_1) = P(r_2) = 2$, and $P(r_3) = 1$. Vertices with larger circles (not necessarily terminals) have rate 2; vertices with smaller circles have rate 1.

The root tree, center, b, and $\mathcal{S}$ are greedily chosen to minimize a cost-to-connectivity ratio γ, defined as follows:

$$\gamma := \frac{1}{|\mathcal{S}|+1}\left(\sigma_b(r,v) + w(v,b) + \sum_{j=1}^{|\mathcal{S}|} \sigma_{P(r_j)}(v, r_j)\right) \tag{1}$$

where r_j denotes the root of the j^{th} rate tree $\mathcal{T}_{r_j}$ in $\mathcal{S}$. The second expression $\sigma_b(r,v) + w(v,b) + \sum_{j=1}^{|\mathcal{S}|} \sigma_{P(r_j)}(v, r_j)$ gives an upper bound on the weight of connecting r to v, upgrading $R(v)$ to b, then connecting v to $|\mathcal{S}|$ roots, and the denominator $|\mathcal{S}|+1$ represents the "connectivity", or the number of connected rate trees. Lemma 6 shows how to execute this iteration step in polynomial time.

Once $\mathcal{T}_r$, v, b, and $\mathcal{S}$ are chosen, we "upgrade" the vertex rates $R(\cdot)$ along a shortest r–v path to b, then upgrade the vertex rates along each shortest v–r_j

path to $P(r_j)$. In the case that some vertex u is on multiple v–r_j paths, then $R(u)$ is upgraded to the maximum over all root priorities $P(r_j)$ for which u appears on the corresponding path. Pseudocode is shown in Algorithm 3.

Algorithm 3 $R(\cdot) = \text{PNWST}(G,\ \text{terminals } T,\ \text{priorities } P,\ \text{vertex weights } w)$

Initialize $\mathcal{F}$, $R(v) = P(v)$ if $v \in T \cup \{s\}$ and $R(v) = 0$ if $v \notin T \cup \{s\}$
while $|\mathcal{F}| > 1$ **do**
    Find $\mathcal{T}_r$, v, b, $\mathcal{S}$ which minimize γ (Lemma 6)
    $R(u) = \max\{R(u), b\}$ for u on r–v path
    $R(v) = \max\{R(v), b\}$
    for $j = 1, \ldots, |\mathcal{S}|$ **do**
        $R(u) = \max\{R(u), P(r_j)\}$ for u on v–r_j path
    $\mathcal{F} = \mathcal{F} \setminus (\{\mathcal{T}_r\} \cup \mathcal{S})$
    $\mathcal{F} = \mathcal{F} \cup \{\mathcal{T}_r^{\text{new}}\}$
return vertex rates $R(\cdot)$

3.2 Analysis of Algorithm 3

We show Theorem 2 by asserting that Algorithm 3 is a $2\ln(|T|+1)$–approximation for Priority NWST. We extend the spider decomposition given by Klein and Ravi [19] to account for the priority constraints in the Priority NWST problem.

Definition 4 (Spider). *A* spider *is a tree where at most one vertex has degree greater than 2. A* nontrivial spider *is a spider with at least 2 leaves.*

A spider is identified by its *center*, a vertex from which all paths from the center to the leaves of the spider are vertex-disjoint. A foot of a spider is a leaf; if the spider has at least three leaves, then its center is unique and is also a foot. Klein and Ravi [19] show that given a graph G and subset $M \subseteq V$ of vertices, G can be decomposed into vertex-disjoint nontrivial spiders such that the union of the feet of the nontrivial spiders contains M. We extend the notions of spider and spider decomposition to the Priority NWST problem.

Definition 5 (Rate spider). *A* rate spider *is a rate tree $\mathcal{X}$ which is also a nontrivial spider. It is identified by a* root *r as well as a* center *v such that:*

- *The root r is either the center or a leaf of $\mathcal{X}$, and the path from r to every vertex in $\mathcal{X}$ uses vertices of non-increasing rate $R(\cdot)$*
- *The paths from the center v to each* non-root *leaf of $\mathcal{X}$ are vertex-disjoint and use vertices of non-increasing rate $R(\cdot)$.*

In Figure 2, right, rate spiders $\mathcal{X}_2$ and $\mathcal{X}_3$ have centers distinct from their roots r_2, r_3 while $\mathcal{X}_1$ has center $v = r_1$. In Definition 6, we supply a notion of a "minimal" weight tree with respect to a subset M of vertices.

Definition 6 (M–optimized rate tree). *Let $\mathcal{T}_r$ be a rate tree rooted at r with vertex rates R. Let $M \subseteq V(\mathcal{T}_r)$ with $r \in M$. Then $\mathcal{T}_r$ is* M–optimized *if every leaf of $\mathcal{T}_r$ is in M, and if for every vertex $v \in V(\mathcal{T}_r) \setminus M$, we have $R(v) = \max R(w)$ over all vertices $w \in M$ in the subtree of $\mathcal{T}_r$ rooted at v.*

We show any M–optimized rate tree has a rate spider decomposition.

Lemma 4 (Rate spider decomposition). *Let $M \subseteq V(\mathcal{T}_r)$ with $|M| \geq 2$, and let $\mathcal{T}_r$ be an M–optimized rate tree where $r \in M$. Then $\mathcal{T}_r$ can be decomposed into vertex-disjoint rate spiders $\mathcal{X}_1, \ldots, \mathcal{X}_d$ rooted at $r_1, \ldots, r_d$ such that:*

- *the leaves and roots of the rate spiders are contained in M*
- *every vertex in M is a either a leaf, root, or center of some rate spider*

Figure 2, right, shows an example of an M–optimized rate tree $\mathcal{T}_r$ for $|M| = 10$ and a rate spider decomposition $\mathcal{X}_1$, $\mathcal{X}_2$, $\mathcal{X}_3$ over M.

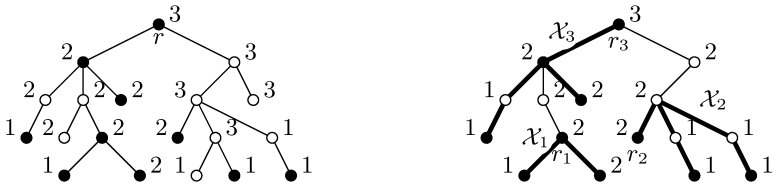

Fig. 2: *Left:* A rate tree rooted at r with rates $R(\cdot)$ indicated and vertices in M shown in black. *Right:* An M–optimized rate tree $\mathcal{T}_r$ and a rate spider decomposition $\mathcal{X}_1$, $\mathcal{X}_2$, $\mathcal{X}_3$ with roots r_1, r_2, r_3.

For $i \geq 1$, let $\mathcal{F}_i$ denote the set of rate trees at the beginning of iteration i of Algorithm 3, and let $h_i \geq 2$ denote the number of rate trees in $\mathcal{F}_i$ which are connected on iteration i (i.e., $h_i = |\mathcal{S}| + 1$). Let ΔC_i denote the actual weight incurred on iteration i by upgrading vertex rates in line 7. Let γ_i denote the minimum cost-to-connectivity ratio (Eq. (1)) computed by Algorithm 3 on iteration i. Lemma 4 (rate spider decomposition) yields the following lemma:

Lemma 5. *For each iteration i of Algorithm 3, we have* $\dfrac{\Delta C_i}{h_i} \leq \dfrac{\mathrm{OPT}}{|\mathcal{F}_i|}$.

Using Lemma 5, we can prove Theorem 2, by asserting that Algorithm 3 is a $2\ln(|T|+1)$–approximation for Priority NWST. The remainder of the proof can be completed by following the analysis by Klein and Ravi [19].

It is worth noting that the extension of the $(2\ln|T|)$-approximation by Klein and Ravi [19] to the Priority NWST problem is not immediately obvious, as we must be careful when merging multiple rate trees while simultaneously satisfying the priority and rate requirements.

Lemma 6. *On iteration i of Algorithm 3, a choice of $\mathcal{T}_r$, v, b, and $\mathcal{S}$ which minimizes γ can be found in $O(n^3 k \log n)$ time.*

Algorithm 3 runs for $I \leq |T|$ iterations, as the size of $|\mathcal{F}|$ decreases by at least 1 at each iteration. By Lemma 6, the running time of Algorithm 3 is $O(n^4 k \log n)$. The approximation ratio is tight as is the case for the Ravi-Klein algorithm [19].

4 Conclusions and future work

By strengthening the analysis of [6], we showed that Priority Steiner tree is approximable with ratio $\min\{\lceil \log_2 |T| \rceil + 1, k\rho\} \leq \min\{1.443 \ln |T| + 2, k\rho\}$, then provided a simple, parallelizable algorithm with the same approximation ratio. Second, we showed that a natural node-weighted generalization of Priority Steiner tree admits a $O(\log |T|)$-approximation using a generalization of the Ravi-Klein algorithm [19] and spider decomposition. It remains open whether the approximability gap between $c \log \log n$ [9] and $O(\log n)$ for Priority Steiner tree can be tightened, or whether a more efficient approximation algorithm for Priority NWST can be formed. As both problems can be reduced to directed Steiner tree, this suggests a hierarchy in terms of hardness of approximation.

Acknowledgments The authors wish to thank Alon Efrat and Spencer Krieger for their discussions related to the priority NWST problem.

References

1. Ahmed, R., Angelini, P., Sahneh, F.D., Efrat, A., Glickenstein, D., Gronemann, M., Heinsohn, N., Kobourov, S., Spence, R., Watkins, J., Wolff, A.: Multi-level Steiner trees. Proceedings of the 17th International Symposium on Experimental Algorithms (2018)
2. Ahmed, R., Sahneh, F.D., Kobourov, S., Spence, R.: Kruskal-based approximation algorithm for the multi-level Steiner tree problem. Proceedings of the 28th Annual European Symposium on Algorithms (2020)
3. Angelopoulos, S.: Online priority Steiner tree problems. In: WADS (2009)
4. Balakrishnan, A., Magnanti, T.L., Mirchandani, P.: Modeling and heuristic worst-case performance analysis of the two-level network design problem. Management Sci. **40**(7), 846–867 (1994)
5. Byrka, J., Grandoni, F., Rothvoß, T., Sanità, L.: Steiner tree approximation via iterative randomized rounding. J. ACM **60**(1), 6:1–6:33 (2013)
6. Charikar, M., Naor, J.S., Schieber, B.: Resource optimization in QoS multicast routing of real-time multimedia. IEEE/ACM Trans. Networking **12**(2), 340–348 (2004)
7. Chekuri, C., Hajiaghayi, M.T., Kortsarz, G., Salavatipour, M.R.: Approximation algorithms for nonuniform buy-at-bulk network design. SIAM Journal on Computing **39**(5), 1772–1798 (2010)
8. Chlebík, M., Chlebíková, J.: The Steiner tree problem on graphs: Inapproximability results. Theoret. Comput. Sci. **406**(3), 207–214 (2008)
9. Chuzhoy, J., Gupta, A., Naor, J.S., Sinha, A.: On the approximability of some network design problems. ACM Trans. Algorithms **4**(2), 23:1–23:17 (2008)
10. Current, J.R., ReVelle, C.S., Cohon, J.L.: The hierarchical network design problem. European Journal of Operational Research **27**(1), 57 – 66 (1986)

11. Demaine, E.D., Hajiaghayi, M., Klein, P.N.: Node-weighted Steiner tree and group Steiner tree in planar graphs. In: Albers, S., Marchetti-Spaccamela, A., Matias, Y., Nikoletseas, S., Thomas, W. (eds.) Automata, Languages and Programming. pp. 328–340. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
12. Dinur, I., Steurer, D.: Analytical approach to parallel repetition. Proceedings of the Annual ACM Symposium on Theory of Computing (05 2013). https://doi.org/10.1145/2591796.2591884
13. Feige, U.: A threshold of $\ln n$ for approximating set cover. J. ACM **45**(4), 634–652 (1998)
14. Grandoni, F., Laekhanukit, B., Li, S.: $O(\log^2 k/\log\log k)$-approximation algorithm for directed Steiner tree: A tight quasi-polynomial-time algorithm. In: Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing. p. 253–264. STOC 2019, Association for Computing Machinery, New York, NY, USA (2019), https://doi.org/10.1145/3313276.3316349
15. Guha, S., Khuller, S.: Improved methods for approximating node weighted Steiner trees and connected dominating sets. J. Inform. Comput. **150**(1), 57–74 (1999)
16. Hauptmann, M., Karpinski, M.: A compendium on Steiner tree problems (2015), http://theory.cs.uni-bonn.de/info5/steinerkompendium/
17. Imase, M., Waxman, B.M.: Dynamic Steiner tree problem. SIAM Journal on Discrete Mathematics **4**(3), 369–384 (1991)
18. Karpinski, M., Măndoiu, I.I., Olshevsky, A., Zelikovsky, A.: Improved approximation algorithms for the quality of service multicast tree problem. Algorithmica **42**(2), 109–120 (2005)
19. Klein, P., Ravi, R.: A nearly best-possible approximation algorithm for node-weighted Steiner trees. J. Algorithms **19**(1), 104–115 (1995)
20. Marathe, M.V., Ravi, R., Sundaram, R., Ravi, S., Rosenkrantz, D.J., Hunt, H.B.: Bicriteria network design problems. Journal of Algorithms **28**(1), 142–171 (1998). https://doi.org/https://doi.org/10.1006/jagm.1998.0930, https://www.sciencedirect.com/science/article/pii/S0196677498909300
21. Maxemchuk, N.F.: Video distribution on multicast networks. IEEE Journal on Selected Areas in Communications **15**(3), 357–372 (1997)
22. Mirchandani, P.: The multi-tier tree problem. INFORMS J. Comput. **8**(3), 202–218 (1996)
23. Naor, J.S., Panigrahi, D., Singh, M.: Online node-weighted Steiner tree and related problems. In: 52nd Annual IEEE Symposium on Foundations of Computer Science FOCS 2011. pp. 210–219. IEEE (January 2011)
24. Risso, C., Robledo, F., Nesmachnow, S.: Mixed Integer Programming Formulations for Steiner Tree and Quality of Service Multicast Tree Problems. Programming and Computer Software **46**(8), 661–678 (2020)
25. Robins, G., Zelikovsky, A.: Improved Steiner tree approximation in graphs. In: Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms. p. 770–779. SODA '00, Society for Industrial and Applied Mathematics, USA (2000)
26. Sahneh, F.D., Kobourov, S., Spence, R.: Approximation algorithms for priority Steiner tree problems. arXiv preprint arXiv:2108.13544 (2021)
27. Turletti, T., chrysostome Bolot, J.: Issues with multicast video distribution in heterogeneous packet networks. In: In Proceedings of the Sixth International Workshop on Packet Video (1994)
28. Xue, G., Lin, G.H., Du, D.Z.: Grade of service Steiner minimum trees in the Euclidean plane. Algorithmica (New York) **31**(4), 479–500 (Jan 2001)