Tighter Approximation for the Uniform Cost-Distance Steiner Tree Problem

Josefine Foos
Research Institute for Discrete Mathematics and Hausdorff Center for Mathematics,
University of Bonn, Germany

Stephan Held
Research Institute for Discrete Mathematics and Hausdorff Center for Mathematics,
University of Bonn, Germany

Yannik Kyle Dustin Spitzley
Research Institute for Discrete Mathematics and Hausdorff Center for Mathematics,
University of Bonn, Germany

Abstract

Uniform cost-distance Steiner trees minimize the sum of the total length and weighted path lengths from a dedicated root to the other terminals. They are applied when the tree is intended for signal transmission, e.g. in chip design or telecommunication networks. They are a special case of general cost-distance Steiner trees, where different distance functions are used for total length and path lengths.

We improve the best published approximation factor for the uniform cost-distance Steiner tree problem from 2.39 [15] to 2.05. If we can approximate the minimum-length Steiner tree problem arbitrarily well, our algorithm achieves an approximation factor arbitrarily close to $1 + \frac{1}{\sqrt{2}}$. This bound is tight in the following sense. We also prove the gap $1 + \frac{1}{\sqrt{2}}$ between optimum solutions and the lower bound which we and all previous approximation algorithms for this problem use.

Similarly to previous approaches, we start with an approximate minimum-length Steiner tree and split it into subtrees that are later re-connected. To improve the approximation factor, we split it into components more carefully, taking the cost structure into account, and we significantly enhance the analysis.

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

Keywords and phrases cost-distance Steiner tree, approximation algorithm, uniform

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2023.19

Category APPROX

Related Version *Full Version*: https://arxiv.org/abs/2305.03381

1 Introduction

Steiner trees can be found in numerous applications, in particular in chip design and telecommunications. In these applications, both the total tree length and the signal speed are important. We consider Steiner trees that do not only minimize the total cost, but also the weighted path lengths from a dedicated root r to the other terminals. Formally, the problem is defined as follows.

An instance (M, c, T, r, p, w) consists of a metric space (M, c), a root r, a finite set T of sinks, a map $p : T \dot{\cup} \{r\} \to M$, and sink delay weights $w \colon T \to \mathbb{R}_{\geq 0}$. The task is to compute a Steiner tree A for $T \cup \{r\}$ with an extension $p : V(A) \setminus (T \cup \{r\}) \to M$ minimizing

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2023).
Editors: Nicole Megow and Adam D. Smith; Article No. 19; pp. 19:1–19:16
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

$$\sum_{\{x,y\}\in E(A)} c(p(x),p(y)) + \sum_{t\in T}\left(w(t)\sum_{\{x,y\}\in E(A_{[r,t]})} c(p(x),p(y))\right), \tag{1}$$

where $A_{[r,t]}$ is the unique r-t-path in A. We call (1) the **(total) cost** of (A,p).

Given a Steiner tree A, we call

$$\sum_{\{x,y\}\in E(A)} c(p(x),p(y)) \quad \text{and} \quad \sum_{t\in T}\left(w(t)\sum_{\{x,y\}\in E(A_{[r,t]})} c(p(x),p(y))\right)$$

its **connection cost** and its **delay cost**.

Usually the position p of vertices is clear from the context. Then, we simply write $c(x,y)$ instead of $c(p(x),p(y))$ and $c(e)$ instead of $c(x,y)$ for edges $e=\{x,y\}$. To shorten the notation, we often also omit the underlying metric space from the notation and write only (T,r,w) to denote an instance. A simple lower bound for the objective function, is given by

$$C_{SMT}(T\cup\{r\}) + D(T,r,w), \tag{2}$$

where $C_{SMT}(T\cup\{r\})$ is the connection cost of a minimum-length Steiner tree for $T\cup\{r\}$, i.e. a Steiner tree A for $T\cup\{r\}$ minimizing $\sum_{e\in E(A)} c(e)$, and $D(T,r,w) := \sum_{t\in T} w(t)c(r,t)$ is the sum of weighted root-sink distances.

The Uniform Cost-Distance Steiner Tree Problem was first mentioned by [17], who considered the (general) cost-distance Steiner tree problem, where the connection cost may be unrelated to the delay cost. Cost-distance Steiner trees are heavily used in VLSI routing and interconnect optimization [11, 7]. Here, the weights arise as Lagrangean multipliers when optimizing global signal delay constraints on an integrated circuit [11]. Uniform cost-distance Steiner trees are computed as a first step of a Steiner tree oracle in global routing [11, 7].

The general cost-distance Steiner tree problem does not permit an approximation factor better than $\Omega(\log\log|T|)$ unless NP $\subseteq$ DTIME$(|T|^{\mathcal{O}(\log\log\log|T|)})$ [6], while a randomized $\mathcal{O}(\log|T|)$-factor approximation algorithm was given by [17] and [4]. Meyerson, Munagala and Plotkin [17] observed that a constant factor approximation algorithm for the Uniform Cost-Distance Steiner Tree Problem can be obtained using the shallow-light spanning tree algorithm from [16]. The resulting factor is 3.57. Using shallow-light Steiner trees [12] instead of spanning trees, the factor was improved to 2.87 independently by [10] and [18]. The first algorithm using the delay weights algorithmically was given by Khazraei and Held [15]. They achieve an approximation factor of $1+\beta$, where β is the approximation factor for computing a minimum-length Steiner tree. All these approaches compare against the lower bound in (2).

Similarly to algorithms for shallow-light trees, the algorithm in [15] starts from an approximately minimum Steiner tree, which is cut into a forest whose components are connected to the root r individually. While [16] cut the tree whenever the path length is too large, [15] cut off a subtree if its delay weight exceeds a certain threshold. Each cut-off tree is later re-connected through a direct connection from the root through one of its terminals, minimizing the resulting objective function.

The special case where we require a spanning tree instead of a Steiner tree and $w(t)$ is identical for all $t\in T$ is known as the *cable-trench problem.* It does not permit a PTAS unless P = NP [2].

The Uniform Cost-Distance Steiner Tree Problem is related to the *single-sink buy-at-bulk problem* where a set of demands needs to be routed from a set of sources to a single sink using a pipe network that has to be constructed from a finite set of possible pipe types with different costs and capacities [9, 19, 14]. The best known approximation factor for this problem is 40.82 due to [8], who also achieve a factor of 20.41 for the splittable case. If there is only one pipe type this problem is equivalent to the Uniform Cost-Distance Steiner Tree Problem. In fact, the threshold-based tree cutting used in the proof of [15] is similar to the algorithm in [9], but the re-connect to the root/sink differs.

1.1 Our contribution

In this paper, we improve the approximation algorithm in [15] for the Uniform Cost-Distance Steiner Tree Problem.

▶ **Theorem 1.** *The Uniform Cost-Distance Steiner Tree Problem can be approximated in polynomial time with an approximation factor of*

$$\beta + \frac{\beta}{\sqrt{\beta^2+1}+\beta-1},$$

where $\beta \geq 1$ is the approximation guarantee for the minimum-length Steiner tree problem.

With the best known approximation factor for the minimum Steiner tree problem $\beta = \ln(4) + \epsilon$ [3, 20], this results in an approximation factor < 2.05 and for $\beta = 1$ this gives the factor $1 + \frac{1}{\sqrt{2}} < 1.71$, clearly improving upon the previously best factors 2.39 and 2.0 in [15]. The polynomial-time approximation scheme by [1] allows choosing β arbitrarily close to one in the Euclidean and the Manhattan planes. However, general metric spaces do not allow $\beta \leq \frac{96}{95}$ unless $\mathrm{P} = \mathrm{NP}$ [5].

Assuming an ideal Steiner tree approximation factor of $\beta = 1$, our new approximation factor is tight with respect to the lower bound (2). We prove the following result

▶ **Theorem 2.**

$$\sup_{T,c,w} \frac{\mathrm{OPT}(T,r,w)}{C_{SMT}(T \cup \{r\}) + D(T,r,w)} = 1 + \frac{1}{\sqrt{2}},$$

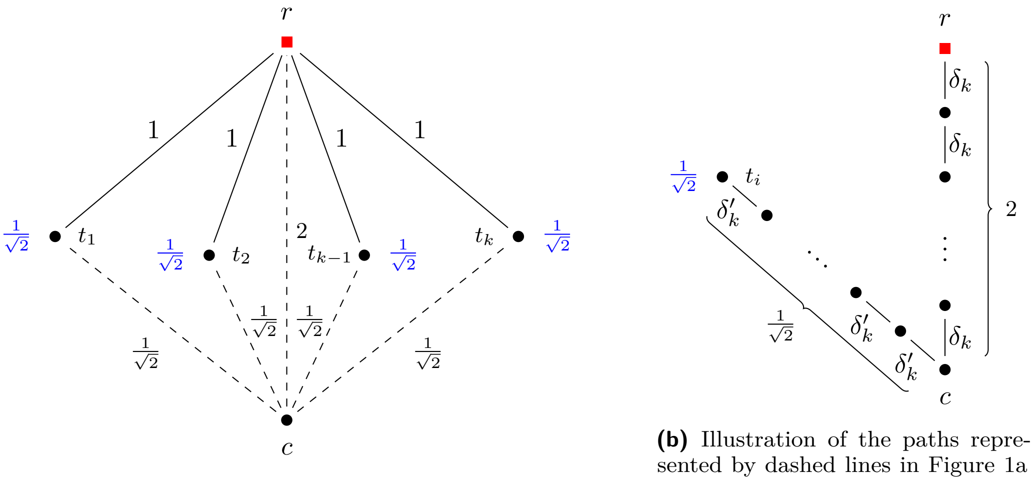
where $\mathrm{OPT}(T,r,w)$ denotes the optimum solution value for (T,r,w).

The algorithm in [15] starts from a short Steiner tree and iteratively splits off subtrees whose delay weight exceeds a given threshold. We proceed similarly, but we also take the structure of the subtrees into account and split off subtrees once they can be reconnected efficiently.

While [15] obtain a running time of $\mathcal{O}(\Lambda + |T|^2)$, where Λ is the time to compute an initial β-approximate minimum Steiner tree, our running time is $\mathcal{O}(\Lambda + |T|)$ (assuming that the metric c can be evaluated in constant time). Thus, it is very fast and applicable for large chip design instances.

We would like to mention that in a preliminary unpublished paper we achieved worse factors of 2.15 ($\beta = \ln(4)$) and 1.80 ($\beta = 1$) [13] using a more complicated algorithm and analysis. That paper also shows that the factor $1 + \beta$ in [15] is tight for that algorithm.

The remainder of this paper is structured as follows. In Section 2, we show that the supremum in Theorem 2 is at least $1 + \frac{1}{\sqrt{2}}$. Then, in Section 3 we will briefly summarize the algorithm and proof from [15], as our work enhances it.



(a) Illustration of the complete instance.

(b) Illustration of the paths represented by dashed lines in Figure 1a by a single t_i.

Figure 1 Instance defined in the proof of Theorem 3. Solid lines represent edges, each dashed line represents a path. Black figures denote edge/path lengths, blue figures denote terminal weights.

Our improved splitting algorithm and analysis is presented in Section 4. The proof of Theorem 1 is presented in Section 4.2. It also shows that the supremum in Theorem 2 is at most $1 + \frac{1}{\sqrt{2}}$. We finish with conclusions in Section 5.

2 Optimality Gap of Lower Bound

In this section, we will show that the gap between an optimum solution and the lower bound in (2) can be as large as $1+\frac{1}{\sqrt{2}}$. Together with the approximation factor of our new algorithm for $\beta = 1$ (Theorem 1), the gap is asymptotically $1 + \frac{1}{\sqrt{2}}$.

▶ **Theorem 3.** *There are instances $I^{(k)}_{k\in\mathbb{N}}$ with $I^{(k)} = (T^{(k)}, r^{(k)}, w^{(k)})$ $(k \in \mathbb{N})$ for the uniform cost-distance Steiner tree problem such that*

$$\lim_{k\to\infty} \frac{\mathrm{OPT}^{(k)}}{C^{(k)} + D^{(k)}} = 1 + \frac{1}{\sqrt{2}},$$

where $\mathrm{OPT}^{(k)}$ is the optimum value for the instance $I^{(k)}$, while $C^{(k)} = C_{SMT}(T^{(k)} \cup \{r\})$, $D^{(k)} := D(T^{(k)}, r^{(k)}, w^{(k)})$ denote minimum possible connection cost and the minimum possible delay cost of $I^{(k)}$.

Proof. We will construct instances with underlying graph metrics induced by graphs indicated in Figure 1a. For $k \in \mathbb{N}$, we define the graph $G^{(k)} = (V^{(k)}, E^{(k)})$ by

$$V^{(k)} = \{r^{(k)}, c^{(k)}, t_1^{(k)}, \dots, t_k^{(k)}, v_1^{(k)}, \dots, v_q^{(k)}\}$$
$$\text{and } E^{(k)} = E_{r,c}^{(k)} \,\dot\cup\, E_{c,t}^{(k)} \,\dot\cup\, E_{r,t}^{(k)},$$

where q is chosen sufficiently large to provide sufficiently many inner path vertices $v_i^{(k)}$ $(1 \le i \le q)$ in the following definitions of $E_{r,c}^{(k)}$, $E_{c,t}^{(k)}$ and $E_{r,t}^{(k)}$: For $0 < \delta_k < \delta'_k < \frac{1}{k}$, $E_{r,c}^{(k)}$ contains edges of length δ_k forming a path of total length 2 between $r^{(k)}$ and $c^{(k)}$, and $E_{c,t}^{(k)}$ contains edges of length δ'_k, forming paths of length $\frac{1}{\sqrt{2}}$ between $c^{(k)}$ and each $t_i^{(k)}$. Lastly,

$$E_{r,t}^{(k)} = \{\{r^{(k)}, t_i^{(k)}\} \mid i = 1, \dots, k\}$$

connects each $t_i^{(k)}$ directly to $r^{(k)}$ with an edge of length 1. The terminals are given by $T^{(k)} = V^{(k)} \setminus \{r^{(k)}\}$, and the delay weights $w^{(k)} \colon T^{(k)} \to \mathbb{R}_{\geq 0}$ are defined as

$$w^{(k)}(t) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } t \in \{t_1^{(k)}, \dots, t_k^{(k)}\} \\ 0 & \text{else.} \end{cases}$$

Now, the lower bound becomes:

$$C^{(k)} + D^{(k)} = 2 + \frac{k}{\sqrt{2}} + \sum_{i=1}^{k} \frac{1}{\sqrt{2}} \operatorname{dist}_{G^{(k)}}(r^{(k)}, t_i^{(k)}) = 2 + 2\,\frac{k}{\sqrt{2}} = 2 + \sqrt{2}\,k. \tag{3}$$

We claim that every optimum solution contains all edges of the form $\{r^{(k)}, t_i^{(k)}\}$. Additionally, we claim that all optimum solutions contain all edges of length δ_k and all but k edges of length δ'_k. This determines the structure of an optimum solution up to the choice of the k ommitted edges. The length of an optimum solution is $(1 + \frac{1}{\sqrt{2}})k + 2 - \delta'_k\, k$, and its objective is

$$\mathrm{OPT}^{(k)} = \left(1 + \frac{1}{\sqrt{2}}\right) k + 2 - \delta'_k k + \frac{k}{\sqrt{2}} = (1 + \sqrt{2})k + 2 - \delta'_k k. \tag{4}$$

Combining (3) and (4), we see that

$$\lim_{k\to\infty} \frac{\mathrm{OPT}^{(k)}}{C^{(k)} + D^{(k)}} = \lim_{k\to\infty} \frac{(1+\sqrt{2})k + 2 - \delta'_k k}{2 + \sqrt{2}\,k} = \frac{1+\sqrt{2}}{\sqrt{2}} = 1 + \frac{1}{\sqrt{2}}$$

as stated in the theorem.

To prove the first claim, assume there is an optimum solution Y^* not containing an edge $\{r^{(k)}, t_i^{(k)}\}$ for some $i \in \{1, \dots, k\}$. First, observe that any path from $r^{(k)}$ to $t_i^{(k)}$ not using the edge $\{r^{(k)}, t_i^{(k)}\}$ contains c, so we have

$$\operatorname{dist}_{Y^*}(r^{(k)}, t_i^{(k)}) \geq \operatorname{dist}_{G^{(k)}}(r^{(k)}, c^{(k)}) + \operatorname{dist}_{G^{(k)}}(c^{(k)}, t_i^{(k)}) = 1 + \frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} = 1 + \sqrt{2}.$$

Let e be the edge of length δ'_k adjacent to $t_i^{(k)}$. Then $e \in E(Y^*)$, as otherwise $t_i^{(k)}$ would be isolated in Y^*. Now define Y' from Y^* by adding $\{r^{(k)}, t_i^{(k)}\}$ and removing e. This increases the connection cost by $1 - \delta'_k$. The delay cost decreases by at least

$$w(t_i^{(k)}) \left(\operatorname{dist}_{Y^*}(r^{(k)}, t_i^{(k)}) - \operatorname{dist}_{Y'}(r^{(k)}, t_i^{(k)}) \right) \geq \frac{1 + \sqrt{2} - 1}{\sqrt{2}} = 1,$$

where we use $\operatorname{dist}_{Y'}(r^{(k)}, t_j^{(k)}) \leq \operatorname{dist}_{Y^*}(r^{(k)}, t_j^{(k)})$ for $j \neq i$. Thus, the total cost decreases by at least $1 - (1 - \delta'_k) = \delta'_k$, a contradiction to the optimality of Y^*.

Now we prove the second claim: By the first claim all optimum solutions have the same delay cost $\frac{k}{\sqrt{2}}$. Hence, only the connection cost for the remaining terminals is relevant. From each maximal path ending in c consisting only of short edges of length either δ_k or δ'_k, any solution must contain either all edges or all but one. Furthermore, there must be such a path from which the solution contains all edges, otherwise there would be no r-c-path. Since $\delta_k < \delta'_k$, the shortest such configuration is to take all edges of length δ_k and all but k edges of length δ'_k (namely all but one from each path). ◀

Algorithm 1 $(1+\beta)$-approximation algorithm by [15] using a parameter $\mu > 0$.

Step 1 (initial arborescence):
First, compute a β-approximate minimum cost Steiner r-arborescence A_0 for $T \cup \{r\}$ with outdegree 0 at all sinks in T and outdegree 2 at all Steiner vertices in $V(A_0) \setminus (T \cup \{r\})$.

Step 2 (split into branching):
Traverse A_0 bottom-up. For each traversed edge $(x, y) \in E(A_0)$, if $W_{(A_0)_y} > \mu$, remove the edge (x, y) creating a new arborescence $(A_0)_y$ in the branching.

Let $\mathcal{A}$ denote the set of all arborescences that were cut off from A_0 this way.

Step 3 (Re-connect arborescences):
Re-connect each sub-arborescence A' that was cut off in Step 2 as follows: Select a vertex $t \in T' := T_{A'}$ that minimizes the cost for serving the sinks in T' through the r-arborescence $A' + (r, t)$, i.e. select a vertex $t \in T'$ as a **port** for T' that minimizes

$$c(r,t) + C_{A'} + \sum_{t' \in T'} w(t') \cdot (c(r,t) + c(E(A'_{[t,t']}))).$$

Let $t_1, \ldots, t_{|\mathcal{A}|} \in T$ be the set of selected port vertices. Return the union of the final branching and the port connections $A_0 + \{(r, t_i) : i \in \{1, \ldots, |\mathcal{A}|\}\ \}$.

3 The $(1 + \beta)$-approximation algorithm

For shorter formulas, we will use the following notation in the remainder of this paper. Let A be an arborescence. By A_v we denote the sub-arborescence rooted at v. Furthermore, $T_A := V(A) \cap T$ is the set of terminals in A, $W_A := w(T_A)$ is the sum of delay weights in A, $C_A := c(E(A))$ is the **connection cost** of A and $D_A := D_{T_A} := \sum_{t \in T_A} w(t)c(r,t)$ the **minimum possible delay cost** for connecting the sinks in T_A (independent of the structure of A).

Recall that $\beta \geq 1$ is the approximation guarantee for the minimum-length Steiner tree problem. The algorithm in [15] is described in Algorithm 1. After orienting its edges, we can consider any solution A as an r-arborescence. We use arborescences instead of trees to simplify the algorithmic notation.

3.1 Essential steps for a $1 + \beta$ approximation

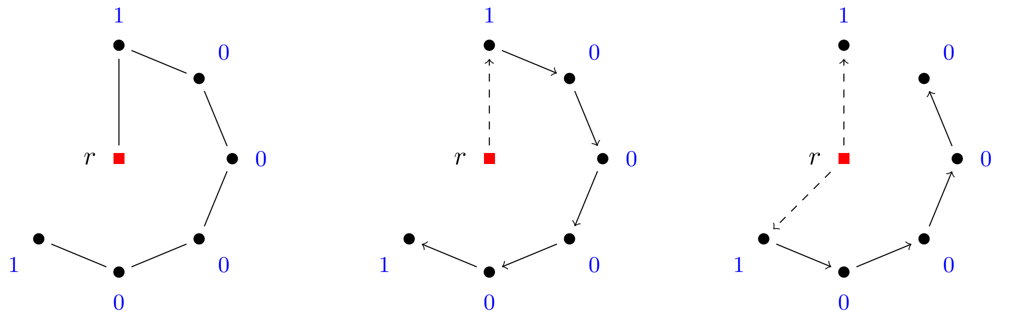
We quickly recap the essential steps in the analysis of [15], which we will use in our analysis. The cost to connect an arborescence $A' \in \mathcal{A}$ to the root r can be estimated as follows:

▶ **Lemma 4** (Khazraei and Held [15], Lemma 1). *Let $A' \in \mathcal{A}$ with corresponding terminal set T'. By the choice of the port $t \in T'$, the r-arborescence $(A' + \{r, t\})$ has a total cost at most*

$$C_{A'} + \sum_{e=(x,y) \in E(A')} \frac{2W_{A'_y}(W_{A'} - W_{A'_y})}{W_{A'}} c(e) + \left(1 + \frac{1}{W_{A'}}\right) D_{T'} \tag{5}$$

$$\leq \left(1 + \frac{W_{A'}}{2}\right) C_{A'} + \left(1 + \frac{1}{W_{A'}}\right) D_{T'} \tag{6}$$

$$\leq (1 + \mu)\, C_{A'} + \left(1 + \frac{1}{\mu}\right) D_{T'}. \tag{7}$$



(a) Minimum-length Steiner tree. **(b)** Cost: $6 + (1 + 6) = 13$. **(c)** Cost: $6 + (1 + 1) = 8$.

Figure 2 Weakness of Algorithm 1: (M, c) is induced by a complete graph with seven vertices and unit weights. Delay weights are indicated by the blue node labels and $\mu = 1$. Algorithm 1 might start with the minimum-length Steiner tree on the left. Then the algorithm will cut the edge incident to r and reconnect the sub-arborescence resulting possibly in the solution in the middle. On the right a better splitting is shown.

We sketch the proof in Appendix A, because we use an inequality that was not stated explicitly in [15], Lemma 1. Note (from the proof) that only the bound (7) uses how the arborescences $A' \in \mathcal{A}$ were cut off during Step 2. In particular the bounds (5)-(6) will also apply for our improvement.

A similar cost bound can be shown easily for the arborescence A_r containing the root r after Step 2. Summing up the resulting cost bounds and choosing $\mu = \frac{1}{\beta}$, [15] obtain the approximation factor $(1 + \beta)$.

4 Improving the approximation ratio

Algorithm 1 suffers from the following weakness. Assume that after splitting we are given a sub-arborescence $A' \in \mathcal{A}$ with a high delay weight $W_{A'}$, a high connection cost $C_{A'}$, but a low minimum possible delay cost $D_{A'}$, e.g. as shown in Figure 2.

In this section, we propose a refined splitting criterion that provides a better approximation ratio. Instead of using a fixed threshold μ, we allow to split off sub-arborescences earlier if their expected re-connection cost (5) is sufficiently cheap. The precise criterion is specified in (8) (inside Algorithm 2). Observe that (8) provides cheaper solutions than (7), as one occurrence of μ is replaced by $\frac{\mu}{2}$.

Then we show in Lemma 8 that every sub-arborescence of the remaining root component has delay weight at most μ. This allows us to prove a similar improved cost bound for the root component in Lemma 9.

In Section 4.2, we simply combine all sub-arborescences and choose μ to prove Theorem 1. Theorem 2 follows as an immediate consequence.

4.1 Improving the splitting routine

Algorithm 2 shows our improved splitting step, which cuts off a sub-arborescence if we can re-connect it cheaply, i.e. if (8) holds. With Lemma 4 we immediately get the following result for the cut-off sub-arborescences:

Algorithm 2 Modifying Step 2 of Algorithm 1.

Step 2 (split into branching):
Traverse A_0 bottom-up. For each traversed edge $(v,z) \in E(A_0)$ consider $A_z := (A_0)_z$:
If $W_{A_z} > 0$ and

$$\sum_{e=(p,q)\in E(A_z)} \frac{2W_{(A_z)_q}(W_{A_z} - W_{(A_z)_q})}{W_{A_z}} c(e) + \frac{D_{A_z}}{W_{A_z}} \le \frac{\mu}{2}\left(C_{A_z} + c(v,z)\right) + \frac{D_{A_z}}{\mu}, \quad (8)$$

remove (v,z) creating a new arborescence A_z.

▶ **Lemma 5.** *Let $A' \in \mathcal{A}$ be an arborescence that was cut off in Algorithm 2 and let $e_{A'}$ be the incoming edge in the root of the arborescence A' which was deleted during this step. Then the corresponding terminals in $T_{A'}$ can be connected to the root r with total cost at most*

$$\left(1+\frac{\mu}{2}\right)\left(C_{A'} + c(e_{A'})\right) + \left(1+\frac{1}{\mu}\right) D_{A'}.$$

After the original Step 2 of Algorithm 1, it is clear that for all edges $(r,x) \in \delta^+_{A_0}(r)$ of the root component the total delay weight $W_{(A_0)_x}$ is at most μ. We show that this also holds after the modified Step 2 in Algorithm 2. However, the analysis is more complicated and uses the following two functions.

▶ **Definition 6.** *Let $\mu > 0$ and $X^\mu := \{(a,b,c) \in (\mu, 2\mu) \times (0,\mu)^2 : c \le a-b < \mu\}$. We define the functions $f, g \colon X^\mu \to \mathbb{R}$ as*

$$f(a,b,c) := \frac{2(a-c)c}{a} - \frac{\mu}{2} + \left(\frac{1}{a} - \frac{1}{\mu}\right) \cdot \frac{1}{\frac{1}{a-b} - \frac{1}{\mu}} \cdot \left(\frac{\mu}{2} - \frac{2((a-b)-c)c}{a-b}\right)$$

$$g(a,b,c) := \frac{2(a-c)c}{a} - \frac{\mu}{2} + \left(\frac{1}{a} - \frac{1}{\mu}\right) \cdot \frac{1}{\frac{1}{a-b} - \frac{1}{\mu}} \cdot \frac{\mu}{2}.$$

▶ **Lemma 7.** *For all $(a,b,c) \in X^\mu$, $f(a,b,c) \le 0$ and $g(a,b,c) \le 0$.*

A proof of Lemma 7 based on algebraic transformations can be found in Appendix B.

▶ **Lemma 8.** *After cutting off sub-arborescences with Algorithm 2, every child $x \in \Gamma^+_{A_r}(r)$ of r in the remaining root component $A_r := (A_0)_r$ satisfies $W_{(A_r)_x} \le \mu$.*

Proof. Assume the opposite would be true. Let z be a vertex in $A_r - r$ such that the weight of the sub-arborescence $A_z := (A_r)_z$ exceeds μ and the weight of every child arborescence $(A_z)_x$ is at most μ for all edges $(z,x) \in \delta^+_{A_z}(z)$. We distinguish between two cases:

Case 1. z is a terminal. Then z is also a leaf and the left-hand side of (8) simplifies to

$$\frac{1}{W_{A_z}} D_{A_z} \le \frac{1}{\mu} D_{A_z}$$

since A_z does not contain any edges. But then A_z would have been cut-off in Step 2, a contradiction.

Case 2. z is a Steiner vertex. Then z has two outgoing edges $e_x := (z,x), e_y := (z,y) \in \delta^+_{A_z}(z)$ as shown in Figure 3. A single outgoing edge would contradict the choice of z. With $A_x := (A_z)_x$ or $A_y := (A_z)_y$ this implies $0 < W_{A_x}, W_{A_y} \le \mu$. If $W_{A_x} = \mu$, Lemma 4, (6)

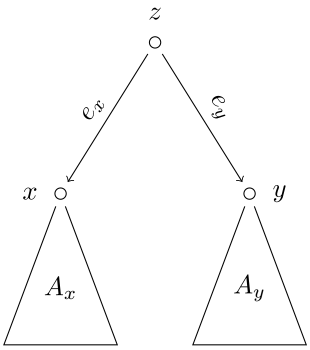


Figure 3 Setting in the proof of Lemma 8 if z is not a terminal.

shows that A_x satisfied the bound (8) when it was considered in Step 2 and would have been cut off. Analogously, $W_{A_y} \neq \mu$. Thus, $W_{A_x}, W_{A_y} < \mu$. Since (8) does not hold for A_x, we get (by transforming its negation)

$$\underbrace{\left(\frac{1}{W_{A_x}} - \frac{1}{\mu}\right)}_{>0} D_{A_x} > \sum_{e=(u,v)\in E(A_x)} \left(\frac{\mu}{2} - \frac{2(W_{A_x} - W_{(A_x)_v})W_{(A_x)_v}}{W_{A_x}}\right) c(e) + \frac{\mu}{2} c(e_x)$$

Combining this with the analogue inequality for A_y and using $D_{A_z} = D_{A_x} + D_{A_y}$, we get

$$\begin{aligned}
&\underbrace{\left(\frac{1}{W_{A_z}} - \frac{1}{\mu}\right)}_{<0} D_{A_z} \\
&\quad < \left(\frac{1}{W_{A_z}} - \frac{1}{\mu}\right) \Bigg(\sum_{e=(u,v)\in E(A_x)} \frac{1}{\frac{1}{W_{A_x}} - \frac{1}{\mu}} \left(\frac{\mu}{2} - \frac{2(W_{A_x} - W_{(A_x)_v})W_{(A_x)_v}}{W_{A_x}}\right) c(e) \\
&\qquad\qquad + \frac{\frac{\mu}{2}}{\frac{1}{W_{A_x}} - \frac{1}{\mu}} c(e_x) \\
&\qquad\qquad + \sum_{e=(u,v)\in E(A_y)} \frac{1}{\frac{1}{W_{A_y}} - \frac{1}{\mu}} \left(\frac{\mu}{2} - \frac{2(W_{A_y} - W_{(A_y)_v})W_{(A_y)_v}}{W_{A_y}}\right) c(e) \\
&\qquad\qquad + \frac{\frac{\mu}{2}}{\frac{1}{W_{A_y}} - \frac{1}{\mu}} c(e_y) \Bigg).
\end{aligned}$$

This inequality together with

$$\begin{aligned}
&\sum_{e=(u,v)\in E(A_z)} \left(\frac{2(W_{A_z} - W_{(A_z)_v})W_{(A_z)_v}}{W_{A_z}} - \frac{\mu}{2}\right) c(e) \\
&= \sum_{e=(u,v)\in E(A_x)} \left(\frac{2(W_{A_z} - W_{(A_z)_v})W_{(A_z)_v}}{W_{A_z}} - \frac{\mu}{2}\right) c(e) \\
&\quad + \left(\frac{2(W_{A_z} - W_{(A_z)_x})W_{(A_z)_x}}{W_{A_z}} - \frac{\mu}{2}\right) c(e_x) \\
&\quad + \sum_{e=(u,v)\in E(A_y)} \left(\frac{2(W_{A_z} - W_{(A_z)_v})W_{(A_z)_v}}{W_{A_z}} - \frac{\mu}{2}\right) c(e) \\
&\quad + \left(\frac{2(W_{A_z} - W_{(A_z)_y})W_{(A_z)_y}}{W_{A_z}} - \frac{\mu}{2}\right) c(e_y)
\end{aligned}$$

yields

$$\begin{aligned}&\sum_{e=(u,v)\in E(A_z)} \left(\frac{2(W_{A_z}-W_{(A_z)_v})W_{(A_z)_v}}{W_{A_z}}-\frac{\mu}{2}\right)c(e)+\left(\frac{1}{W_{A_z}}-\frac{1}{\mu}\right)D_{A_z}\\ &\quad< \sum_{e=(u,v)\in E(A_x)} f(W_{A_z},W_{A_y},W_{(A_z)_v})c(e)+\sum_{e=(u,v)\in E(A_y)} f(W_{A_z},W_{A_x},W_{(A_z)_v})c(e)\\ &\quad+g(W_{A_z},W_{A_y},W_{(A_z)_v})c(e_x)+g(W_{A_z},W_{A_x},W_{(A_z)_v})c(e_y).\end{aligned}$$

By Lemma 7 and $0 < W_{A_x}, W_{A_y} < \mu$, the last term is non-positive. Therefore A_z satisfied the bound (8) when it was considered in Step 2 and would have been cut off, a contradiction. ◀

In [15] the final root arborescence A_r, which was not cut off in Step 2 of Algorithm 1, was kept unaltered. Using Lemma 8, we show how to connect it in a better way.

▶ **Lemma 9.** *Let A_r be the sub-arborescence of A_0 rooted at r after the modified Step 2 of Algorithm 1. The terminal set T_{A_r} can be connected to the root r with total cost at most*

$$\left(1+\frac{\mu}{2}\right)C_{A_r}+\left(1+\frac{1}{\mu}\right)D_{A_r}.$$

Proof. Let $(r,x)\in\delta^+_{A_r}(r)$ be arbitrary and A_x the arborescence of $A_r - r$ rooted at x. We show that the terminal set T_{A_x} can be connected to the root r with total cost at most

$$\left(1+\frac{\mu}{2}\right)(C_{A_x}+c(r,x))+\left(1+\frac{1}{\mu}\right)D_{A_x}.$$

Adding this cost for all edges in $\delta^+_{A_r}(r)$, we obtain the claim.

We distinguish between two cases:

Case 1.

$$W_{A_x}(C_{A_x}+c(r,x))\le\frac{\mu}{2}(C_{A_x}+c(r,x))+\frac{1}{\mu}D_{A_x}.$$

By keeping the arborescence A_x connected through (r,x), the connection cost is $C_{A_x}+c(r,x)$. In particular, for each terminal $t\in T_{A_x}$, the r-t-path in $A_x+(r,x)$ has a length of at most $C_{A_x}+c(r,x)$. We therefore obtain a total cost of at most

$$(1+W_{A_x})(C_{A_x}+c(r,x))\le\left(1+\frac{\mu}{2}\right)(C_{A_x}+c(r,x))+\frac{1}{\mu}D_{A_x}.$$

Case 2.

$$W_{A_x}(C_{A_x}+c(r,x))>\frac{\mu}{2}(C_{A_x}+c(r,x))+\frac{1}{\mu}D_{A_x}. \tag{9}$$

Therefore we have $W_{A_x}>0$ and obtain from (9) an upper bound on the minimum possible delay cost of A_x

$$D_{A_x}<\left(W_{A_x}-\frac{\mu}{2}\right)\mu(C_{A_x}+c(r,x)). \tag{10}$$

We remove the edge (r, x) and connect the arborescence A_x to the root r. By Lemma 8, $W_{A_x} \leq \mu$. As in Lemma 4 we obtain total cost of at most

$$\begin{aligned}
&\left(1+\frac{W_{A_x}}{2}\right) C_{A_x} + \left(1+\frac{1}{W_{A_x}}\right) D_{A_x} \\
&= \left(1+\frac{W_{A_x}}{2}\right) C_{A_x} + \left(1+\frac{1}{\mu}\right) D_{A_x} + \underbrace{\frac{\mu - W_{A_x}}{\mu W_{A_x}}}_{\geq 0} D_{A_x} \\
&\overset{(10)}{\leq} \left(1+\frac{W_{A_x}}{2}\right) C_{A_x} + \left(1+\frac{1}{\mu}\right) D_{A_x} + \frac{\mu - W_{A_x}}{\mu W_{A_x}} \left(W_{A_x} - \frac{\mu}{2}\right) \mu (C_{A_x} + c(r,x)) \\
&\leq \left(1 - \frac{W_{A_x}}{2} + \frac{3}{2}\mu - \frac{\mu^2}{2W_{A_x}}\right) (C_{A_x} + c(r,x)) + \left(1+\frac{1}{\mu}\right) D_{A_x}.
\end{aligned}$$

With the following estimation we obtain the claimed bound

$$-\frac{W_{A_x}}{2} + \frac{3}{2}\mu - \frac{\mu^2}{2W_{A_x}} = \frac{\mu}{2} - \frac{1}{2}\left(\sqrt{W_{A_x}} - \frac{\mu}{\sqrt{W_{A_x}}}\right)^2 \leq \frac{\mu}{2}.$$ ◀

▶ **Theorem 10.** *Algorithm 2 and the re-connect in Step 3 as well as of the root component can be implemented to run in time $\mathcal{O}(|T|)$.*

Proof. A naïve implementation would immediately result in a quadratic running time. We can achieve a linear running time by computing all relevant information incrementally in constant time per node during the bottom-up traversal. Details can be found in Appendix C. ◀

4.2 Proving Theorem 1 and Theorem 2

We start by analyzing the combination of all sub-arborescences.

▶ **Theorem 11.** *Given an instance (T, r, w) of the* Uniform Cost-Distance Steiner Tree Problem*, we can compute in $\mathcal{O}(\Lambda + |T|)$ time a Steiner tree with objective value at most*

$$\left(1+\frac{\mu}{2}\right) C + \left(1+\frac{1}{\mu}\right) D, \tag{11}$$

where C is the cost of a β-approximate minimum-length Steiner tree and $D := D(T, r, w)$. Here, Λ is the running time for computing a β-approximate minimum Steiner tree for $T \cup \{r\}$.

Proof. We run Algorithm 1 with two modifications:

1. The cut-off routine (Step 2) is modified according to Algorithm 2.
2. The arborescence A_r containing the root r after Step 2 may be re-connected to the root r according to Lemma 9.

The total cost of the computed solution is upper bounded by the sum of the cost bounds for these r-arborescences, which is (11). For the running time analysis, we consider the individual steps of the algorithm:

In Step 1, a β-approximate minimum Steiner tree for $T \cup \{r\}$ is computed in time $\mathcal{O}(\Lambda)$ and transformed into the arborescence A_0 obeying the degree constraints in linear time as in [15]. The linear running time of Step 2 and Step 3 follows from Theorem 10. ◀

Finally, we choose the threshold μ based on the quantities C and D to prove Theorem 1:

Table 1 Comparison of approximation factors for the Uniform Cost-Distance Steiner Tree Problem with different approximation factors β for the minimum-length Steiner tree problem.

Parameter β	1	$\ln(4)+\epsilon$	$\frac{3}{2}$	2
Algorithm 1 [15]	2.00000	2.38630	2.50000	3.00000
Theorem 1	1.70711	2.04782	2.15139	2.61804

Proof of Theorem 1. We make the following modification of the algorithm in Theorem 11:

If $C = c(E(A_0)) = 0$, each r-t-path, $t \in T$, has length 0 in A_0. So this is already an optimal solution and we just return A_0.

Otherwise, set $\mu := \sqrt{\frac{2D}{C}}$ and the algorithm from Theorem 11 provides us with a solution with total cost at most

$$C + D + \sqrt{2}\sqrt{CD} \leq \beta C_{SMT}(T \cup \{r\}) + D + \sqrt{2}\sqrt{\beta C_{SMT}(T \cup \{r\}) \cdot D}.$$

We divide this by the lower bound $C_{SMT}(T \cup \{r\}) + D$ in (2). Now, the approximation factor is at most the maximum of the function $h\colon \mathbb{R}_{>0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}$ given by

$$h(x,y) := \frac{\beta x + y + \sqrt{2}\sqrt{\beta x y}}{x + y}.$$

As we proof in Appendix D using algebraic reformulations,

$$h(x,y) \leq \beta + \frac{\beta}{\sqrt{\beta^2+1} + \beta - 1},$$

proving the claimed approximation ratio. ◀

Using Theorem 1 we obtain the approximation factors shown in Table 1 (rounded to five decimal digits) for some interesting values of β in Table 1.

Proof of Theorem 2. This is a direct consequence of Theorem 3 and Theorem 1 for $\beta = 1$. ◀

5 Conclusion

We significantly improve the approximation factor for the Uniform Cost-Distance Steiner Tree Problem. For the lower bound (2) it is best possible if the minimum-length Steiner tree problem can be solved optimally.

This is achieved by an enhancement of the cut-off routine, where we do not simply cut off by delay weight, but take the (cost) structure of the sub-arborescences into account. Furthermore, the root component will be reconnected in a smarter way.

Our algorithm is very fast. After computing an approximate minimum-length Steiner tree, the remaining cutting and re-assembling takes linear time, which previously took a quadratic running time.

Based on the lower bound gap result, further attempts to improve the approximation ratio should rather focus on improving the lower bound. However, this does not appear to be easy.

References

1 Sanjeev Arora. Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems. *Journal of the ACM*, 45(5):753–782, 1998.

2 Marcelo P.L. Benedito, Lehilton L.C. Pedrosa, and Hugo K.K. Rosado. On the inapproximability of the cable-trench problem. *Procedia Computer Science*, 195:39–48, 2021.

3 Jarosław Byrka, Fabrizio Grandoni, Thomas Rothvoss, and Laura Sanità. Steiner tree approximation via iterative randomized rounding. *Journal of the ACM*, 60(1):article 6, 2013.

4 Chandra Chekuri, Sanjeev Khanna, and Joseph Naor. A deterministic algorithm for the cost-distance problem. In *Proc. ACM-SIAM symposium on Discrete Algorithms (SODA '01), 2001*, pages 232–233. SIAM, 2001.

5 Miroslav Chlebík and Janka Chlebíková. The steiner tree problem on graphs: Inapproximability results. *Theoretical Computer Science*, 406(3):207–214, 2008.

6 Julia Chuzhoy, Anupam Gupta, Joseph Naor, and Amitabh Sinha. On the approximability of some network design problems. *ACM Transactions on Algorithms*, 4(2):article 23, 2008.

7 Siad Daboul, Stephan Held, Bento Natura, and Daniel Rotter. Global interconnect optimization. *ACM Transactions on Design Automation of Electronic Systems*, 2023. (to appear, conference paper in Proc. ICCAD '19). doi:10.1145/3587044.

8 Fabrizio Grandoni and Thomas Rothvoß. Network design via core detouring for problems without a core. In *International Colloquium on Automata, Languages, and Programming*, pages 490–502, 2010.

9 Sudipto Guha, Adam Meyerson, and Kamesh Munagala. A constant factor approximation for the single sink edge installation problem. *SIAM Journal on Computing*, 38(6):2426–2442, 2009.

10 Longkun Guo, Nianchen Zou, and Yidong Li. Approximating the shallow-light steiner tree problem when cost and delay are linearly dependent. In *Proc. International Symposium on Parallel Architectures, Algorithms and Programming*, pages 99–103, 2014.

11 Stephan Held, Dirk Müller, Daniel Rotter, Rudolf Scheifele, Vera Traub, and Jens Vygen. Global routing with timing constraints. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37(2):406–419, 2018.

12 Stephan Held and Daniel Rotter. Shallow-light steiner arborescences with vertex delays. In *Proc. International Conference on Integer Programming and Combinatorial Optimization (IPCO '13)*, pages 229–241, 2013.

13 Stephan Held and Yannik.K.D. Spitzley. Further improvements on approximating the uniform cost-distance steiner tree problem. Technical report, Research Institute for Discrete Mathematics, University of Bonn, 2022. arXiv:2211.03830.

14 Raja Jothi and Balaji Raghavachari. Improved approximation algorithms for the single-sink buy-at-bulk network design problems. *Journal of Discrete Algorithms*, 7(2):249–255, 2009.

15 Ardalan Khazraei and Stephan Held. An improved approximation algorithm for the uniform cost-distance Steiner tree problem. In *Proc. Workshop on Approximation and Online Algorithms (WAOA 2020)*, pages 189–203. Springer, 2021.

16 Samir Khuller, Balaji Raghavachari, and Neal E. Young. Balancing minimum spanning trees and shortest-path trees. *Algorithmica*, 14(4):305–321, 1995.

17 Adam Meyerson, Kamesh Munagala, and Serge Plotkin. Cost-distance: Two metric network design. *SIAM Journal on Computing*, 38(4):1648–1659, 2008.

18 Daniel Rotter. *Timing-constrained global routing with buffered Steiner trees.* PhD thesis, Universitäts-und Landesbibliothek Bonn, 2017.

19 Kunal Talwar. The single-sink buy-at-bulk lp has constant integrality gap. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 475–486, 2002.

20 Vera Traub and Rico Zenklusen. Local search for weighted tree augmentation and steiner tree. In *Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA '22)*, pages 3253–3272, 2022.

A Proof of Lemma 4

Proof (Lemma 4). Note that Lemma 1 in [15] states only the bound (7). The bounds (5) and (6) follow immediately from their proof, which we will briefly sketch: We choose a terminal $t \in T'$ randomly with probability $p_t := \frac{w(t)}{W_{A'}}$ as the "port" vertex (only for the analysis). Then we obtain:

- The expected cost of (r,t) is $\mathbb{E}(c(r,t)) = \sum_{t \in T'} p_t c(r,t) = \frac{1}{W_{A'}} D_{T'}$.
- The (deterministic) connection cost within A' is $C_{A'}$.
- The expected effective delay cost of (r,t) is

$$\mathbb{E}(W_{A'} \cdot c(r,t)) = W_{A'} \cdot \sum_{t \in T'} p_t c(r,t) = D_{T'}.$$

- The expected delay weight served by an edge $(x,y) \in E(A')$ is

$$\frac{W_{A'_y}}{W_{A'}} \cdot (W_{A'} - W_{A'_y}) + \frac{W_{A'} - W_{A'_y}}{W_{A'}} \cdot W_{A'_y} = \frac{2W_{A'_y}(W_{A'} - W_{A'_y})}{W_{A'}} \leq \frac{W_{A'}}{2},$$

where A'_y is the sub-arborescence of $A' - (x,y)$ containing y. The formula reflects the expected component of $(A' - (x,y))$ in which the port vertex is located. Summation over all edges in A' yields the following expected delay cost contribution of $E(A')$:

$$\sum_{e=(x,y) \in E(A')} \frac{2W_{A'_y}(W_{A'} - W_{A'_y})}{W_{A'}} c(e) \leq \frac{W_{A'}}{2} C_{A'}.$$

The addition of these four terms gives the expected total cost of connecting A' to the root r, and provides the bound in (5). The bounds (6)-(7) follow as (5) is maximized for $W_{A'_y} = \frac{1}{2} W_{A'}$ and $W_{A'} \leq 2\mu$ or A' is a (heavy) singleton. The deterministic best choice of the "port" vertex in Algorithm 1 cannot be more expensive. ◀

B Proof of Lemma 7

Proof. Note that the functions f and g differ only in the last factor. Actually, because of $\frac{1}{a} - \frac{1}{\mu} < 0$, $\frac{1}{\frac{1}{a-b} - \frac{1}{\mu}} > 0$ and $\frac{\mu}{2} - \frac{2((a-b)-c)c}{a-b} \leq \frac{\mu}{2}$ we get $f(a,b,c) \geq g(a,b,c)$ for all $(a,b,c) \in X^\mu$, so it is sufficient to show $f(a,b,c) \leq 0$.

We have

$$\begin{aligned}
f(a,b,c) &= \frac{4ca - 4c^2 - \mu a}{2a} + \frac{(\mu - a)(a-b)}{a(\mu + b - a)} \left(\frac{\mu(a-b) - 4c(a-b-c)}{2(a-b)} \right) \\
&= \frac{(\mu - a)(4ca - 4c^2 - \mu a) + b(4ca - 4c^2 - \mu a)}{2a(\mu + b - a)} \\
&\quad + \frac{(\mu - a)\left(\mu(a-b) - 4c(a-b-c)\right)}{2a(\mu + b - a)} \\
&= \frac{b(4ca - 4c^2 - \mu a) + b(\mu - a)(4c - \mu)}{2a(\mu + b - a)} \\
&= \frac{b(4ca - 4c^2 - \mu a + 4\mu c - \mu^2 - 4ca + \mu a)}{2a(\mu + b - a)} \\
&= -\frac{b(2c - \mu)^2}{2a(\mu + b - a)} \\
&\leq 0,
\end{aligned}$$

where $\mu + b - a > 0$ by $(a,b,c) \in X^\mu$. ◀

C Detailed Proof of Theorem 10

Proof (Theorem 10). We will proof two claims:

1. Checking whether a branch should be cut off at the traversed vertex can be done in $\mathcal{O}(1)$ time.
2. Choosing the ports can be done in linear time.

Then, we just observe that Step 2 traverses the initial tree once, which also needs linear time.

Proof of Claim 1. We keep track of five values for each node v and its corresponding sub-arborescence $A_v := (A_0)_v$: $W_v := W_{A_v}$, the weight inside A_v, $D_v := D_{A_v}$, the minimum possible delay cost of A_v, $C_v := C_{A_v}$, the connection cost of A_v, $S_v^1 := \sum_{e=(p,q)\in E(A_v)} W_q(W_v - W_q)c(e)$ and $S_v^2 := \sum_{e=(p,q)\in E(A_v)} W_q c(e)$.

For leaves, we can compute these in constant time. For a node v with only one child x (because the other has been cut off), we can compute the values as follows: $W_v = W_x$, $D_v = D_x$, $C_v = C_x + c(v,x)$, $S_v^1 = S_x^1$, and $S_v^2 = S_x^2 + W_x c(v,x)$.

Whenever we consider a node v with children x and y, and $A_x := (A_v)_x$ and $A_y := (A_v)_y$, we can compute the values for v like so: $W_v = W_x + W_y$, $D_v = D_x + D_y$, $C_v = C_x + c(v,x) + C_y + c(v,y)$,

and

$$\begin{aligned}
S_v^1 &= \sum_{e=(p,q)\in E(A_v)} W_q(W_v - W_q)c(e) \\
&= \left(\sum_{e=(p,q)\in E(A_x)} W_q(W_v - W_q)c(e)\right) + W_x(W_v - W_x)c(v,x) \\
&\quad + \left(\sum_{e=(p,q)\in E(A_y)} W_q(W_v - W_q)c(e)\right) + W_y(W_v - W_y)c(v,y) \\
&= \left(\sum_{e=(p,q)\in E(A_x)} W_q(W_x - W_q)c(e)\right) + \left(\sum_{e=(p,q)\in E(A_x)} W_q W_y c(e)\right) \\
&\quad + W_x(W_v - W_x)c(v,x) \\
&\quad + \left(\sum_{e=(p,q)\in E(A_y)} W_q(W_y - W_q)c(e)\right) + \left(\sum_{e=(p,q)\in E(A_y)} W_q W_x c(e)\right) \\
&\quad + W_y(W_v - W_y)c(v,y) \\
&= S_x^1 + W_y S_x^2 + W_x(W_v - W_x)c(v,x) + S_y^1 + W_x S_y^2 + W_y(W_v - W_y)c(v,y)
\end{aligned}$$

and

$$S_v^2 = S_x^2 + W_x c(v,x) + S_y^2 + W_y c(v,y).$$

Proof of Claim 2. For each node $v \in V(A')$, the cost when using v as the port is

$$\mathrm{cost}_v = c(r,v) + C_{A'} + \sum_{t\in T_{A'}} w(t)\cdot(c(r,v) + c(E(A'_{[v,t]}))).$$

So for an edge $e = (x, y) \in E(A')$ we have

$$\begin{aligned}\mathrm{cost}_x - \mathrm{cost}_y = c(r,x) - c(r,y) + \sum_{t\in T_{A'}} w(t)(c(r,x) - c(r,y)) + \sum_{t\in T_{A'_y}} w(t)c(e)\\ - \sum_{t\in T_{A'}\setminus T_{A'_y}} w(t)c(e)\\ = (c(r,x) - c(r,y))(1 + W_{A'}) + c(e)W_{A'_y} - c(e)(W_{A'} - W_{A'_y}).\end{aligned}$$

This allows us to compute in constant time the cost for choosing y as the port from the cost for choosing its parent x as the port. We take advantage of this property and first compute the cost for using the root of A' as the port in $\mathcal{O}(|E(A')| + |T_{A'}|)$. Then, we find the find the best "port" vertex in a top-down traversal in the claimed linear time. ◀

D Upper Bound on $h(x, y)$

We prove that for $\beta \geq 1$, $x, y \geq 0, x + y > 0$

$$\begin{aligned}h(x,y) &:= \frac{\beta x + y + \sqrt{2}\sqrt{\beta xy}}{x+y}\\ &= \beta + \frac{(1-\beta)y + \sqrt{2}\sqrt{\beta xy}}{x+y}\\ &\leq \beta + \frac{\beta}{\sqrt{\beta^2+1}+\beta-1}.\end{aligned}$$

Proof. For shorter notation, we set

$$a := \frac{\beta}{\sqrt{\beta^2+1}+\beta-1} > 0$$

and get

$$\begin{aligned}\frac{\beta}{2a} + 1 - \beta - a &= \frac{1}{2}\sqrt{\beta^2+1} - \frac{1}{2}(\beta-1) - \frac{\beta}{\sqrt{\beta^2+1}+\beta-1}\\ &= \frac{(\sqrt{\beta^2+1} - (\beta-1))(\sqrt{\beta^2+1} + (\beta-1)) - 2\beta}{2(\sqrt{\beta^2+1}+\beta-1)}\\ &= 0.\end{aligned}$$

Therefore,

$$\begin{aligned}h(x,y) &= \beta + \frac{(1-\beta)y + \sqrt{2}\sqrt{\beta xy}}{x+y} - \frac{\left(\frac{\beta}{2a} + 1 - \beta - a\right)y}{x+y}\\ &= \beta + \frac{\sqrt{2}\sqrt{\beta xy}}{x+y} - \frac{\left(\frac{\beta}{2a} - a\right)y}{x+y}\\ &= \beta + a - \frac{ax}{x+y} + \frac{\sqrt{2}\sqrt{\beta xy}}{x+y} - \frac{\frac{\beta}{2a}y}{x+y}\\ &= \beta + a - \frac{a}{x+y}\left(\sqrt{x} - \frac{\sqrt{\beta}}{\sqrt{2}a}\sqrt{y}\right)^2.\end{aligned}$$

As $a > 0$, we obtain

$$h(x,y) \leq \beta + a = \beta + \frac{\beta}{\sqrt{\beta^2+1}+\beta-1}.$$ ◀