Combination algorithms for Steiner Tree variants

Gruia Călinescu[1*] and Xiaolang Wang[2]

[1*]Department of Computer Science, Illinois Institute of Technology, Chicago, 60616, IL, USA.
[2]Department of Computer Science, Illinois Institute of Technology, Chicago, 60616, IL, USA.

*Corresponding author(s). E-mail(s): calinescu@iit.edu;
Contributing authors: xwang122@hawk.iit.edu;

Abstract

We give better approximation ratios for two Steiner Tree variants by combining known algorithms: the optimum $\mathbf{3}$-decomposition and iterative randomized rounding.

The first problem is Steiner Tree with minimum number of Steiner points and bounded edge length problem ($\boldsymbol{SMT-MSP}$). The input consists of a set of terminals $\boldsymbol{R}$ in the Euclidean space $\mathbb{R}^{\mathbf{2}}$. A feasible solution is a Steiner tree $\boldsymbol{\tau}$ spanning $\boldsymbol{R}$ with Steiner points $\boldsymbol{S}$ such that every edge in $\boldsymbol{\tau}$ has length at most $\mathbf{1}$. The objective is to minimize S. Previously, the best approximation ratio for $\boldsymbol{SMT-MSP}$ was $\mathbf{1}+\ln(\mathbf{4})+\boldsymbol{\epsilon}\approx\mathbf{2.386}$. We present a polynomial time algorithm with ratio $\mathbf{2.277}$. The second problem is Steiner Tree in quasi-bipartite graphs. It is a Steiner Tree problem on graph $\boldsymbol{G}=(\boldsymbol{V},\boldsymbol{E},\boldsymbol{c})$ with terminal set $\boldsymbol{R}$ when the edge set $\boldsymbol{E}$ does not include any edge between two vertices in $\boldsymbol{V}\setminus\boldsymbol{R}$. The best-known approximation ratio for this problem is $\frac{\mathbf{73}}{\mathbf{60}}$, We improve this ratio to $\frac{\mathbf{298}}{\mathbf{245}}$.

Keywords: approximation ratio, Steiner tree, Euclidean space, quasi-bipartite graph

1 Introduction

The Steiner Tree problem is a classic optimization problem defined by Gauss in a letter he wrote to Schumacher: given a weighted undirected graph and a subset of terminal nodes, find a minimum-cost tree spanning the terminals. Note that this tree can include non-terminal nodes of the graph; these are called Steiner points. Later in 1934, this problem was first formally formulated in a paper by Vojtěch Jarník and Miloš Kössler [1]. Steiner Tree was among the first problems proven NP-hard [2]. Since then, a series of papers provided approximation algorithms for the Steiner Tree problem [3–12] and other related problems [13–18]. In this paper, we combine some previously developed methods to achieve better approximation ratios in two Steiner Tree variants.

The Steiner Tree problem and many of its variants can be reduced (sometimes with an ϵ loss in the approximation ratio) to a more general problem that we call $MCSH$ (Minimum Connected Spanning Sub-hypergraph). It is described as follows: given a complete hypergraph $\mathcal{H} = (R, \mathcal{E})$, $\mathcal{E} = \{A \mid A \subseteq R\}$ with hyperedge cost $\{c(A) : A \in \mathcal{E}\}$, find a minimum-cost connected spanning sub-hypergraph of $\mathcal{H}$. Usually, the reduction is, the terminal set becomes the vertex set $R(\mathcal{H})$ in $MCSH$, and the collection of terminal subsets becomes the hyperedge set $\mathcal{E}(\mathcal{H})$. The cost $c(A)$ for a hyperedge $A \in \mathcal{E}(\mathcal{H})$ comes from the cost of a solution to a smaller Steiner Tree problem instance on terminal set A. Most Steiner Tree variants including the ones we study allow solving optimally in polynomial time any instance with A of bounded size.

The first method used in this paper is the 3-decomposition method (short for 3-restricted decomposition) [6, 9]. Consider a special class of hypergraphs:

Definition 1 A k-hypergraph $\mathcal{H}_k = (R, \mathcal{E}_k)$ is a (complete) hypergraph with hyperedge set $\mathcal{E}_k = \{A \subseteq R : 2 \leq |A| \leq k\}$. The k-restricted hypergraph of the complete hypergraph $\mathcal{H}$ is the k-hypergraph $\mathcal{H}_k$ obtained from $\mathcal{H}$ by keeping only the edges of size at most k, with the same cost as in $\mathcal{H}$.

Definition 2 A k-decomposition of $\mathcal{H} = (R, \mathcal{E})$ is a collection of hyperedges $\mathcal{A}_k = \{A \mid A \in \mathcal{E}_k\}$, such that $(R, \mathcal{A}_k)$ is connected. The cost of a k-decomposition is the total cost of all hyperedges in the k-decomposition.

We can see that a k-decomposition of $\mathcal{H} = (R, \mathcal{E})$ is a connected spanning sub-hypergraph in $\mathcal{H}_k = (R, \mathcal{E}_k)$. Recall that the problem of finding a minimum-cost connected spanning sub-hypergraph in $\mathcal{H}$ is $MCSH$; denote the same problem in $\mathcal{H}_k$ as $MCSH_k$.

The 3-decomposition method reduces a Steiner Tree problem, with some loss in the approximation ratio, to $MCSH_3$. The reason we consider 3-restricted hypergraphs is that an instance of $MCSH_3$ can be solved optimally in polynomial time from the recent work of Iwata and Kobayashi [19] (an $(1+\epsilon)$-approximation is known since [20], see also [9]). As an aside, for $k > 3$,

$MCSH_k$ becomes NP-hard and $(1 + 1/2 + ... + 1/(k-1))$-approximations are known [21, 22].

Minimum cost 2-decompositions also play a major role in the Steiner Tree problem, as they approximate the minimum Steiner tree within a ratio of 2, they can be computed optimally (it is the Minimum Spanning Tree problem), and the tree has a matroid structure that is used by further algorithms.

The second method is Relative Greedy [23]. This method starts with a spanning tree on the terminal set, or as previously defined, a 2-decomposition of the hypergraph $\mathcal{H} = (R, \mathcal{E})$. In each iteration, it tries to replace a set of spanning tree edges by a hyperedge in $\mathcal{E}_k$ with smaller cost while the connectivity of the decomposition is kept. Precisely, it chooses the hyperedge that minimizes its cost divided by the best (highest cost) set of spanning tree edges that can be removed once this hyperedge is used for connectivity. It iterates until no more replacement of edges can reduce the total cost of the decomposition.

The third and last method used in this paper is iterative randomized rounding (IRR) [12, 24]. This method starts with an LP relaxation of the problem based on directed hyperedges. It samples one hyperedge with probability proportional to the value of the associated variable in a fractional solution, then the hyperedge becomes contracted and the LP is updated consequently. It iterates this process until all terminals are connected. The analyses of IRR and Relative Greedy have similarities and, in all the problems we have seen, IRR can be used instead of Relative Greedy to get the approximation ratio one normally gets when analyzing Relative Greedy, while for some problems, notably Steiner Tree, IRR appears to be more powerful and its analysis gives a better approximation ratio at the moment of this writing.

In this paper, we consider the following two Steiner Tree variants. The first problem is Steiner Tree with minimum number of Steiner points and bounded edge length problem ($SMT-MSP$). $SMT-MSP$ is described as follows. The input consists of a set of terminals R in the Euclidean space $\mathbb{R}^2$. A feasible solution is a Steiner tree τ spanning R with Steiner points S such that every edge in τ has length at most 1. The objective is to minimize $|S|$. As an aside, it has been observed that, as in the Steiner Tree problem, the challenge consists of selecting the Steiner points of degrees at least three, after which a Minimum Spanning Tree algorithm can be used for optimally connecting the terminals and those Steiner points.

This problem was first formally formulated in a paper by Lin and Xue [25] in the year 1999, where they showed that the $SMT-MSP$ problem is NP-complete. They also showed that an algorithm based on Minimum Spanning Tree is a 5-approximation of $SMT-MSP$. The approximation ratio of this algorithm was improved to 4 in [26] and [27] separately at the same time, but [27] also gave a 3-approximation algorithm. Later, Cheng *et al.* [28] gave a $2.5+\epsilon$-approximation algorithm on $SMT-MSP$ using the 3-decomposition method ([19] removes the ϵ in the approximation ratio). Cohen and Nutov [29] improved the approximation ratio to $1+\ln(4)+\epsilon \approx 2.386$ using Relative Greedy.

Researchers are also working on the $SMT-MSP$ problem in other metric spaces. Let Δ be the maximum number of points in a unit ball such that the distance between any two of them is larger than 1. A series of papers ([26], [30], [29]) gave approximation algorithms on the $SMT-MSP$ problem in any metric space with approximation ratio $\Delta - 1$, $\lfloor(\Delta+1)/2\rfloor + 1 + \epsilon$ (this ϵ is also removed by [19]) and 1+ln $(\Delta - 1) + \epsilon$, respectively. Here, we only look at the $SMT-MSP$ problem in the Euclidean space $\mathbb{R}^2$, where $\Delta = 5$. In $\mathbb{R}^3$ or higher dimensions, the ratio $\lfloor(\Delta+1)/2\rfloor$, even another hopeful ratio using 3-decomposition $\lceil(\Delta-1)/2\rceil$, can never beat the ratio 1+ln $(\Delta - 1) + \epsilon$ that comes from IRR or Relative Greedy. Here, $\lceil(\Delta-1)/2\rceil$ is the best possible approximation ratio using the 3-decomposition method on an instance with Δ non-adjacent terminals and exactly one Steiner point in its optimal solution.

For $SMT - MSP$, we combine the Relative Greedy and the 3-decomposition methods and get a polynomial time algorithm with approximation ratio 2.277.

The other problem we consider is Steiner Tree in quasi-bipartite graphs. It is a Steiner Tree problem on graph $G = (V, E, c)$ with terminal set R, where the edge set does not include any edge between two vertices in $V \setminus R$. This problem was first considered by Rajagopalan and Vazirani [31] who obtain a $3/2+\epsilon$-approximation using primal dual methods. Then Robins and Zelikovsky [11] improved the approximation ratio to 1.28.

Byrka *et al.*[12] invented IRR and showed a $\left(\frac{73}{60} + \epsilon\right) \leq 1.2167$-approximation for this problem. Later, Goemans *et al.* [24] showed that the ϵ in the approximation ratio can be removed, also based on IRR.

For Steiner Tree problem in quasi-bipartite graphs, we combine the IRR and the 3-decomposition methods and get an approximation algorithm with ratio $\frac{298}{245} < \frac{73}{60}$. The difference is only 1/2940, but our contribution is in breaching the "natural" approximation ratio $\frac{73}{60}$ of IRR.

2 The Combination Algorithm for SMT-MSP

Denote an optimum Steiner point set as S_{opt}, denote $opt = |S_{opt}|$. We call a spanning tree over $S_{opt} \cup R$ with minimum total length a shortest optimum Steiner tree, and we denote it by τ_{opt}. Here, length means the Euclidean distance between two points.

Lemma 1 *(Lemma 1 in [27]) There exists a shortest optimum Steiner tree τ_{opt} for $SMT - MSP$ such that every vertex in τ_{opt} has degree at most five.*

A Steiner tree is called *full* if every terminal is a leaf. Given τ_{opt} such that every vertex in τ_{opt} has degree at most five, by breaking at terminals with degree more than 1, we can break τ_{opt} into several full Steiner trees. We call each full Steiner tree obtained this way a *full component*, and we call a full component *trivial* if it contains two terminals and no Steiner points. We use $d(a, b)$ to denote the Euclidean distance between points a and b in the Euclidean plane.

Lemma 2 *For every full component C, there is a Steiner tree C^* over the same terminal set and a Steiner point set included in the Steiner point set of C, with the following properties:*

- *Every Steiner point has degree at most five.*
- *Every pair of terminals in the same non-trivial full component in tree C^* have mutual Euclidean distance greater than 1.*

Proof: First replace C by a shortest (Euclidean) tree on the same vertex set (remind that, every vertex in C has degree at most 5). Then remove un-needed Steiner points, and break the tree into full components. Given a non-trivial full component C' of the resulting tree, we can get a C^* as follows. For each terminal t in C', let s be the Steiner point t is adjacent to; if there exists another terminal t' in C' such that $d(t, t') \leq 1$, then the replacement of edge (t, s) by edge (t, t') does not disconnect the terminals. Do this replacement; note that no Steiner vertex has its degree increased. If s becomes a leaf of this Steiner tree, remove it, and continue removing un-needed Steiner points.

As a result, the full component C' is broken into two: the trivial full component with terminals t and t', and what is left of C' without terminal t. Repeat if needed. □

We call each full component obtained this way a *cheerful component*. Thus, a trivial component is cheerful and a non-trivial cheerful component has the following properties:

- Every terminal is leaf.
- Every Steiner point has degree at most five.
- Every pair of terminals have mutual Euclidean distance greater than 1.

Based on the discussion above (also implicit in [27]), there exists an optimum solution such that each of its non-trivial full component is cheerful. We assume from now on that every full component of the optimum solution is a cheerful component.

For an $SMT-MSP$ instance, a *bead solution* is a solution obtained from a spanning tree over R by inserting $\lceil d(a, b) \rceil - 1$ Steiner points (called *beads*) to break each edge (a, b) into multiple pieces of length at most 1. The following lemma is implicit in the proof of Theorem 1 in [27].

Lemma 3 *Let R' be the set of terminals in a cheerful component with $m \geq 1$ Steiner points. Then R' has a bead solution with at most $3m + 1$ beads.*

As an aside, note that this lemma implies a 4-approximation for the algorithm based on computing a minimum (Euclidean length) spanning tree over the set of terminals, and subdividing every edge as in a bead solution. This is so because the tree spanning R with minimum number of beads is the same as the tree spanning R of minimum length (Kruskal's algorithm returns the same solution), and a tree spanning R with $4opt$ total number of beads can be obtained by replacing every non-trivial cheerful component with m Steiner points by a bead solution with $4m$ beads.

Denote the number of non-trivial cheerful components in τ_{opt} as c, and denote the number of cheerful components with m Steiner points as c_m. Then, $c = \sum_{m\geq 1} c_m$, and $opt = \sum_{m\geq 1}(m \cdot c_m)$. Following Lemma 3,

Lemma 4 *For an $SMT-MSP$ instance with an optimum solution with c non-trivial cheerful components, there is a bead solution with at most $3opt + c$ beads.*

Recall that for hypergraph $\mathcal{H} = (R, \mathcal{E})$, $\mathcal{E} = \{A \mid A \subseteq R\}$ with hyperedge cost $\{c(A) : A \in \mathcal{E}\}$, we use $\mathcal{H}_k = (R, \mathcal{E}_k)$ to denote the k-restricted hypergraph of $\mathcal{H}$. Also recall that $MCSH_k$ is the problem of finding a minimum-cost connected spanning sub-hypergraph in $\mathcal{H}_k$. Let α_k be the ratio between the optimum cost of $MCSH_k$ and the optimum cost of $MCSH$ for a given instance. It is more common to define α_k as being the supremum, over instances, of this ratio; we however allow α_k to depend on the instance.

Lemma 5 *(implicit in [23], and in Corollary 3 in [29]) For any constant k, there is algorithm for $MCSH$ with running time polynomial in the number of vertices of the $MCSH$ instance that returns for every instance a solution of cost at most $\alpha_k(1 + \ln \alpha_2)opt$.*

This ratio comes from an algorithm using the Relative Greedy method. As an aside, we could use IRR instead of Relative Greedy here and get the same ratio; we chose Relative Greedy since it is faster and we can directly cite [29].

Given an $SMT-MSP$ instance on terminal set R, consider the hypergraph $\mathcal{H} = (R, \mathcal{E})$, where $\mathcal{E} = \{A \mid A \subseteq R\}$. For each $A \in \mathcal{E}$, let $c(A)$ equal to the size of the optimum Steiner set for $SMT - MSP$ on A. [27] showed that for any constant k, the cost $c(A)$ for $A \in \mathcal{E}_k$ can be calculated in polynomial time. Note that for any $SMT - MSP$ instance, the $MCSH$ instance we create has the same optimum value, and that for $\mathcal{H}_2$, the cost of an edge A that consists of two terminals $a, b \in R$ is the number of beads needed to connect a and b, i.e. $\lceil d(a, b)\rceil - 1$.

Lemma 6 *(as a corollary of Theorem 5 in [29]) For $SMT - MSP$, α_k converges to 1 as k gets larger, with a bound on the rate of convergence that does not depend on the instance.*

As an aside, the equivalent of Lemma 6 for the Steiner Tree problem is known from the pioneering work [32] and [33]. Let $\chi = \frac{(1/2)c_1+c_2+(1/2)c_3}{opt}$, then

Lemma 7 *For every positive ϵ, there exists a polynomial time algorithm that computes a solution to $SMT - MSP$ of cost at most $\Big(1 + \ln(3.25 + (3/2) \cdot \chi) + \epsilon\Big) opt$.*

Proof: Lemma 6 showed that as k gets larger, α_k converges to 1. From Lemma 4 we can deduce that $\alpha_2 \leq (3opt + c)/opt$.

Since $c = \sum_{m \geq 1} c_m \leq c_1 + c_2 + c_3 + \frac{opt - c_1 - 2c_2 - 3c_3}{4}$ (recall that $opt = \sum_{m \geq 1}(m \cdot c_m)$), with Lemma 5, we have a solution to $SMT - MSP$ of cost at most

$$\alpha_k \Big(1 + \ln(\frac{3opt + c}{opt})\Big) \cdot opt = \Big(1 + \ln(3 + c/opt) + \epsilon\Big) \cdot opt.$$

We show that $3 + c/opt \leq 3.25 + (3/2)\chi$, and we finish the proof of Lemma 7.

$$\begin{aligned} 3 + c/opt &\leq 3 + \frac{c_1 + c_2 + c_3 + \frac{opt - c_1 - 2c_2 - 3c_3}{4}}{opt} \\ &= 3.25 + \frac{(3/4)c_1 + (1/2)c_2 + (1/4)c_3}{opt} \\ &\leq 3.25 + (3/2)\chi \end{aligned}$$

□

The second method we use to approximate $SMT - MSP$ is 3-decomposition. Theorem 3.6 in [28] can be rephrased as, "there is a 3-decomposition with cost 2.5 times the optimum". If an $SMT - MSP$ instance consists of only the terminals of exactly one cheerful component, we get:

Lemma 8 *For each cheerful component with m Steiner points, there exists a connected 3-decomposition spanning the same terminals with cost at most $2.5m$.*

We prove later the stronger statement:

Lemma 9 *The 3-decomposition method computes a solution to $SMT - MSP$ of cost $(2.5 - \chi) \cdot opt$ in polynomial time.*

Note that, after proving Lemma 9, we have two polynomial time approximation algorithms for $SMT - MSP$. The Relative Greedy algorithm computes a solution of cost at most $\big(1 + \ln(3.25 + (3/2)\chi) + \epsilon\big) \cdot opt$. The algorithm that computes an optimum solution to $MCSH_3$ [19] obtains a solution of cost $(2.5 - \chi) \cdot opt$. When $\chi \leq 0.223$, we look at the solution given by the first algorithm, it has cost smaller than $(2.277) \cdot opt$. When $\chi > 0.223$, we look at the solution given by the second algorithm, its cost is at most $(2.277) \cdot opt$. Thus, by returning the better solution between two algorithms,

Theorem 1 *There is a polynomial time algorithm that is a (2.277)-approximation of $SMT-MSP$.*

Proof of Lemma 9: Recall that, $\chi = \frac{(1/2)c_1+c_2+(1/2)c_3}{opt}$. When a cheerful component has 1 Steiner point (3 Steiner points), Lemma 8 shows that a 3-decomposition with cost at most 2 (7) exists, since the number of Steiner points is an integer. For the case of 2 Steiner points, we will show in Lemma 10 below that its cost is at most 4. Thus, there exists a 3-decomposition of cost at most:

$$\begin{aligned}
& 2c_1 + 4c_2 + 7c_3 + \sum_{m>3} 2.5m \cdot c_m \\
&= 2c_1 + 4c_2 + 7c_3 + 2.5(opt - c_1 - 2c_2 - 3c_3) \\
&= 2.5 \cdot opt - (1/2)c_1 - c_2 - (1/2)c_3 \\
&= 2.5 \cdot opt - \chi \cdot opt \\
&= (2.5 - \chi) \cdot opt
\end{aligned}$$

Since the 3-decomposition method can find the optimum 3-decomposition of all terminals in polynomial time, it computes a solution to an $SMT-MSP$ instance of cost at most $(2.5 - \chi) \cdot opt$. □

Lemma 10 *If a cheerful component has exactly 2 Steiner points, there is a 3-decomposition of cost at most 4 for the terminals of this component. Here, as before, the cost of a hyperedge A (A is a set of terminals) is the size of the optimum Steiner set for $SMT-MSP$ on A.*

Proof: Since each Steiner point has degree at most 5, there are at most 8 terminals in this cheerful component. When there are at most 7 terminals, at most one Steiner point has 4 terminals adjacent to it. So one can decompose this cheerful component by grouping 3 (or fewer) terminals adjacent to the same Steiner point with cost 1 for each Steiner point, and grouping the possible left alone terminal with one terminal in each of the first two groups with cost 2; thus, this decomposition has cost at most 4. When there are 8 terminals, if one of the terminals (say t, which is adjacent to the Steiner point s_1 in this cheerful component) is within Euclidean distance 1 to both Steiner points (s_1 and s_2), then one can decompose this component as follows:

- Group 3 arbitrary terminals adjacent to the Steiner point s_2 with cost 1.
- Group terminal t, an arbitrary terminal in the above group, and the left alone terminal adjacent to the Steiner point s_2 with cost 1.
- Group 3 terminals other than t that are adjacent to the Steiner point s_1 with cost 1.
- Group terminal t with an another arbitrary terminal adjacent to s_1 with cost 1.

One can easily verify the above decomposition is connected and with cost 4. Thus here, we claim:

Lemma 11 *If a cheerful component in τ_{opt} has exactly 2 Steiner points and* 8 *terminals, then at least* 1 *terminal is within Euclidean distance* 1 *to both Steiner points.*

We will prove Lemma 11 in Appendix A, and with the proof of the lemma we finish the proof of Lemma 10.

3 The Combination Algorithm for Steiner Tree in quasi-bipartite graphs

The first method we use to approximate this problem is iterative randomized rounding (IRR). Let undirected graph $G(V, E, c)$ and terminal set $R \subseteq V$ be the input of a Steiner Tree problem.

Recall that a Steiner tree is full if every terminal is a leaf. Also, as before, a full component C is a set of terminals $R' \subseteq R$ together with a full Steiner tree on R' using edges in $E(G)$. In this section, the cost of a full component C is given by $c(C) = \sum_{e \in E(C)} c(e)$. Since G does not have an edge between two non-terminal vertices, a non-trivial full component is a star (tree of height 1) with the root a Steiner vertex and the leafs terminals. To avoid special cases, we also consider trivial components as stars with two leafs (duplicating a terminal into a Steiner point at distance 0 to the terminal).

Take a full component Q that is a star from OPT, which is the optimum solution. Its q edges are sorted in non-increasing order by cost: $c(e_1) \geq c(e_2) \geq \cdots \geq c(e_q)$. Note that $c(Q) = c(e_1) + c(e_2) + \ldots + c(e_q)$. [24] used the following potential $\Phi(Q)$ to achieve the $\frac{73}{60}$ approximation ratio in their paper:

$$\Phi(Q) = c(e_1) + c(e_2) + c(e_3) \cdots + c(e_{q-1}) + H_{q-1} c(e_q)$$

Here, H_{q-1} is the $(q-1)^{th}$ Harmonic number. [24] showed that IRR has expected output cost at most $\sum_{Q \in X} \Phi(Q)$, where X is the set of full components in the optimum solution OPT.

A second method we use for Steiner Tree in quasi-bipartite graphs is 3-decomposition. Following [19] one can obtain in polynomial time the optimum 3-decomposition. Take full component Q from OPT, which now is a star with $q \geq 3$ terminals and edges: $e_1, e_2, ..., e_q$. As defined above, e_q is the edge with minimum cost. 3-decomposition for Q gives a solution of cost at most:

$$\Psi(Q) = c(e_1) + c(e_2) + \ldots + c(e_{q-1}) + \lfloor \frac{q}{2} \rfloor c(e_q),$$

because one can 3-decompose Q as follows:

- duplicate the least-costing edge e_q $\lfloor \frac{q-2}{2} \rfloor$ times (so we have $\lfloor \frac{q}{2} \rfloor$ copies in total),
- arbitrarily pair the rest $q-1$ edges up with possibly one edge left alone,
- add a copy of e_q into each size 2 or 1 group.

Based on this, the [19] algorithm has an output of cost at most $\sum_{Q \in X} \Psi(Q)$, where as above X is the set of full components in the optimum solution OPT.

While the algorithm consists of producing the best output from the algorithms of [24] and [19] as adapted to $MCSH_3$, for the purpose of analysis, we take a convex combination of the two methods: the iterative randomized rounding algorithm with probability $\theta = 48/49$ and the 3-decomposition algorithm with probability $1-\theta = 1/49$. We have seen this idea previously in [34]. As an aside, the value of θ is obtained by solving the following two-dimensional linear program $LP1$:

$$\begin{aligned} &\text{minimize } \gamma \\ &\text{subject to } \frac{q-1+\theta \cdot H_{q-1} + (1-\theta) \cdot \lfloor \frac{q}{2} \rfloor}{q} \le \gamma, \forall q \in \{2, \ldots, 10\} \qquad (LP1) \\ &\qquad 0 \le \theta \le 1. \end{aligned}$$

For a full component Q, let $h(Q) = \frac{\theta \cdot \Phi(Q) + (1-\theta) \cdot \Psi(Q)}{c(Q)}$. $LP1$ is obtained from the definition of $h(Q)$: we consider only full components with at most 10 edges and we consider each edge has a uniform weight. $LP1$ has optimal solution $\theta = \frac{48}{49}, \gamma = \frac{298}{245}$. We show that,

Lemma 12 $h(Q) \le \frac{298}{245} < \frac{73}{60}$.

Proof: Let q be the number of terminals in Q, and let $r_j = \frac{j-1+\theta \cdot H_{j-1} + (1-\theta) \cdot \lfloor \frac{j}{2} \rfloor}{j}$. We have

$$\begin{aligned} h(Q) &= \frac{\sum_{i \in \{1,\ldots,q-1\}} c(e_i) + \theta \cdot H_{q-1} c(e_q) + (1-\theta) \cdot \lfloor \frac{q}{2} \rfloor c(e_q)}{\sum_{i \in \{1,\ldots,q\}} c(e_i)} \\ &\le \frac{q-1+\theta \cdot H_{q-1} + (1-\theta) \cdot \lfloor \frac{q}{2} \rfloor}{q} = r_q. \end{aligned}$$

The inequality above can be easily checked using $c(e_q) \le \frac{\sum_{i \in \{1,\ldots,q-1\}} c(e_i)}{q-1}$ and $\theta \cdot H_{q-1} + (1-\theta) \cdot \lfloor \frac{q}{2} \rfloor \ge 1$. Remind that $\theta = \frac{48}{49}$.

We will show next that $r_j \le \frac{298}{245}, \forall j \ge 2$. The proof proceeds by induction on j. First, let us consider the cases of $j = 2, 3, 4, 5, 6$.

- **When** $j = 2$**,** $H_{j-1} = \lfloor j/2 \rfloor = 1$, we have $r_j = 1 < \frac{298}{245}$.
- **When** $j = 3$**,** $H_{j-1} = \frac{3}{2}$, $\lfloor j/2 \rfloor = 1$, we have $r_j = \frac{2+72/49+1/49}{3} = \frac{57}{49} < \frac{298}{245}$.

- **When** $j = 4$**,** $H_{j-1} = \frac{11}{6}, \lfloor j/2 \rfloor = 2$, we have $r_j = \frac{3+88/49+2/49}{4} = \frac{237}{196} < \frac{298}{245}$.
- **When** $j = 5$**,** $H_{j-1} = \frac{25}{12}, \lfloor j/2 \rfloor = 2$, we have $r_j = \frac{4+100/49+2/49}{5} = \frac{298}{245}$.
- **When** $j = 6$**,** $H_{j-1} = \frac{137}{60}, \lfloor j/2 \rfloor = 3$, we have $r_j = \frac{5+548/245+3/49}{6} = \frac{298}{245}$.
- **For** $j > 6$**,** we use induction on j.

$$\begin{aligned}
r_j &= \frac{j - 1 + \theta \cdot H_{j-1} + (1-\theta) \cdot \lfloor \frac{j}{2} \rfloor}{j} \\
&= \frac{j - 3 + 2 + \theta \cdot (H_{j-3} + \frac{1}{j-2} + \frac{1}{j-1}) + (1-\theta)(\lfloor \frac{j-2}{2} \rfloor + 1)}{j-2} \cdot \frac{j-2}{j} \\
&\leq \Big(\frac{298}{245} + \frac{2 + \theta(\frac{1}{j-2} + \frac{1}{j-1}) + (1-\theta)}{j-2} \Big) \cdot \frac{j-2}{j} \\
&= \frac{298}{245} \cdot \frac{j-2}{j} + \frac{2 + \theta(\frac{1}{j-2} + \frac{1}{j-1}) + (1-\theta)}{j} \\
&\leq \frac{298}{245} + \frac{-\frac{298}{245} \cdot 2 + 2 + \frac{48}{49}(\frac{1}{5} + \frac{1}{6}) + \frac{1}{49}}{j} \\
&= \frac{298}{245} - \frac{13}{245 \cdot j}.
\end{aligned}$$

The first inequality follows from the induction hypothesis and the second inequality follows from $j > 6$. This finishes the proof of Lemma 12. □

Using linearity of expectation and the lemma above, we have:

Theorem 2 *There is a randomized polynomial-time algorithm with expected approximation ratio* $\frac{298}{245} < \frac{73}{60}$ *for Steiner Tree in quasi-bipartite graphs.*

4 Conclusion

We improved the approximation ratio of $SMT-MSP$ in the Euclidean metric space from 2.386 to 2.277 by taking the best output of two previously proposed approximation algorithms: [28] and [29]. None of these algorithm is known to be tight, but maybe improving the ratio of one would still lead to the combination algorithm being even better. In particular, our hope comes from the fact that we cannot find any instance with an approximation ratio strictly greater than 2 using 3-decomposition.

While IRR is a powerful method, it fails to find the optimum for Minimum Weight Edge Cover. One can find this optimum employing matching methods, using the algorithm of Lawler [35], (see also Gabows thesis [36] and Edmonds and Johnson [37]). 3-decomposition is an extension of Matching [19, 38] and this explains why it sometimes works when IRR fails. Our results take advantage of this situation.

There are limits on our combination method. For $SMT - MSP$ in $\mathbb{R}^3$ or higher dimensions, as mentioned in detail in the introduction, the 3-decomposition method can never beat IRR or Relative Greedy.

For the Steiner Tree problem, we have examples where the (5/3) ratio given by the 3-decomposition [6] is tight and the $(\ln 4 + \epsilon)$-approximation analysis of IRR as given by [24] is also tight.

For Minimum Power Spanning Tree in graphs with edge weights in $\{0, 1\}$ where both IRR as used by Grandoni [39] and the 3-decomposition as used by Nutov and Yaroshevitch [30] give ratios of $3/2 + \epsilon$ and $3/2$ respectively, we have examples where both analyses get the same 3/2 ratio.

Acknowledgments. Gruia would like to thank Neil Olver for discussions on the IRR method.

5 Compliance with Ethical Standards

No funding was received for conducting this study. The authors declare that they have no financial interests.

References

[1] Jarník, V., Kössler, M.: O minimálních grafech, obsahujících n danỳch bodů. Časopis pro pěstování matematiky a fysiky **63**(8), 223–235 (1934)

[2] Karp, R.M.: Reducibility among combinatorial problems. In: Complexity of Computer Computations, pp. 85–103. Springer, New York (1972)

[3] Dreyfus, S.E., Wagner, R.A.: The Steiner problem in graphs. Networks **1**(3), 195–207 (1971)

[4] Hakimi, S.L.: Steiner's problem in graphs and its implications. Networks **1**(2), 113–133 (1971)

[5] Kou, L., Markowsky, G., Berman, L.: A fast algorithm for Steiner trees. Acta informatica **15**(2), 141–145 (1981)

[6] Zelikovsky, A.: An 11/6-approximation algorithm for the network Steiner problem. Algorithmica **9**(5), 463–470 (1993). https://doi.org/10.1007/BF01187035

[7] Berman, P., Ramaiyer, V.: Improved approximations for the Steiner tree problem. JALG **17**, 381–408 (1994)

[8] Karpinski, M., Zelikovsky, A.: New approximation algorithms for the Steiner tree problems. Electronic Colloquium on Computational Complexity (ECCC) **2**(030) (1995)

[9] Prömel, H.J., Steger, A.: A new approximation algorithm for the Steiner tree problem with performance ratio 5/3. Journal of Algorithms **36**, 89–101 (2000)

[10] Hougardy, S., Prömel, H.J.: A 1.598 approximation algorithm for the Steiner problem in graphs. In: Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 448–453 (1999)

[11] Robins, G., Zelikovsky, A.: Tighter bounds for graph Steiner tree approximation. SIAM Journal on Discrete Mathematics **19**(1), 122–134 (2005)

[12] Byrka, J., Grandoni, F., Rothvoss, T., Sanità, L.: Steiner tree approximation via iterative randomized rounding. J. ACM **60**(1), 6–1633 (2013). https://doi.org/10.1145/2432622.2432628

[13] Zelikovsky, A.: A series of approximation algorithms for the acyclic directed Steiner tree problem. Algorithmica **18**(1) (1997)

[14] Charikar, M., Chekuri, C., Cheung, T.-Y., Dai, Z., Goel, A., Guha, S., Li, M.: Approximation algorithms for directed Steiner problems. Journal of Algorithms **33**(1), 73–91 (1999)

[15] Klein, P., R.Ravi: A nearly best-possible approximation algorithm for node-weighted Steiner trees. Journal of Algorithms **19**, 104–115 (1995)

[16] Guha, S., Khuller, S.: Improved Methods for Approximating Node Weighted Steiner Trees and Connected Dominating Sets. Information and Computation **150**, 57–74 (1999)

[17] Garg, N., Konjevod, G., Ravi, R.: A polylogarithmic approximation algorithm for the group Steiner tree problem. J. Algorithms **37(1)**, 66–84 (2000)

[18] Chekuri, C., Even, G., Kortsarz, G.: A greedy approximation algorithm for the group Steiner problem. Discrete Applied Mathematics **154**(1), 15–34 (2006)

[19] Iwata, S., Kobayashi, Y.: A weighted linear matroid parity algorithm. SIAM Journal on Computing (0), 17–238 (2021)

[20] Camerini, P.M., Galbiati, G., Maffioli, F.: Random pseudo-polynomial algorithms for exact matroid problems. JALG **13**, 258–273 (1992)

[21] Wolsey, L.A.: Analysis of the greedy algorithm for the submodular set covering problem. Combinatorica **2**, 385–392 (1982)

[22] Baudis, G., Gropl, C., Hougardy, S., Nierhoff, T., Prömel, H.J.: Approximating minimum spanning sets in hypergraphs and polymatroids. In:

ICALP (2000)

[23] Zelikovsky, A.: Better approximation bounds for the network and Euclidean Steiner tree problems. Technical Report CS-96-06, Department of Computer Science, University of Virginia (1996)

[24] Goemans, M.X., Olver, N., Rothvoß, T., Zenklusen, R.: Matroids and integrality gaps for hypergraphic Steiner tree relaxations. In: STOC, pp. 1161–1176 (2012)

[25] Lin, G.-H., Xue, G.: Steiner tree problem with minimum number of Steiner points and bounded edge-length. Information Processing Letters **69**(2), 53–57 (1999). https://doi.org/10.1016/S0020-0190(98)00201-4

[26] Mandoiu, I.I., Zelikovsky, A.Z.: A note on the MST heuristic for bounded edge-length Steiner trees with minimum number of Steiner points. Inf. Process. Lett. **75**(4), 165–167 (2000). https://doi.org/10.1016/S0020-0190(00)00095-8

[27] Chen, D., Du, D.-Z., Hu, X.-D., Lin, G.-H., Wang, L., Xue, G.: Approximations for Steiner trees with minimum number of Steiner points. Theoretical Computer Science **262**(1-2), 83–99 (2001). https://doi.org/10.1016/S0304-3975(00)00182-1

[28] Cheng, X., Du, D.-Z., Wang, L., Xu, B.: Relay sensor placement in wireless sensor networks. Wireless Networks **14**(3), 347–355 (2008). https://doi.org/10.1007/s11276-006-0724-8

[29] Cohen, N., Nutov, Z.: Approximating Steiner trees and forests with minimum number of Steiner points. Journal of Computer and System Sciences **98**, 53–64 (2018)

[30] Nutov, Z., Yaroshevitch, A.: Wireless network design via 3-decompositions. Inf. Process. Lett. **109**(19), 1136–1140 (2009). https://doi.org/10.1016/j.ipl.2009.07.013

[31] Rajagopalan, S., Vazirani, V.V.: On the bidirected cut relaxation for the metric Steiner tree problem. In: SODA, vol. 99, pp. 742–751 (1999)

[32] Du, D.-Z., Zhang, Y.: On better heuristics for Steiner minimum trees. Mathematical Programming **57**(1), 193–202 (1992). https://doi.org/10.1007/BF01581080

[33] Borchers, A., Du, D.-Z.: The k-Steiner ratio in graphs. SIAM Journal on Computing **26**(3), 857–869 (1997)

[34] Goemans, M.X., Williamson, D.P.: New 3/4-approximation algorithms for

the maximum satisfiability problem. SIAM J. on Discrete Mathematics **7**, 656–666 (1994)

[35] Lawler, E.L..: Combinatorial Optimization: Networks and Matroids. Holt, Rinehart and Winston, New York (1976)

[36] Gabow, H.N.: Implementation of algorithms for maximum matching on nonbipartite graphs. PhD thesis, Stanford, CA, USA (1974). AAI7413628

[37] Edmonds, J., Johnson, E.L.: Matching: A well-solved class of integer linear programs, Combinatorial Structures and their Applications (Proc. Calgary Internat., Calgary, Alta., 1969). Gordon and Breach, New York (1970)

[38] Lovász, L., Plummer, M.D.: Matching Theory. North-Holland, Amsterdam–New York (1986)

[39] Grandoni, F.: On min-power Steiner tree. In: European Symposium on Algorithms, pp. 527–538 (2012). Springer

Appendix A Proof of Lemma 11

We prove Lemma 11 by contradiction, and we want to show that there is always at least one terminal within Euclidean distance 1 to both Steiner points or within Euclidean distance 1 to another terminal (which makes this full component not cheerful anymore). For intuition, look at the following extreme case with 8 terminals and 2 Steiner points in Figure A1. Each terminal is at distance exactly 1 to the Steiner point it is connected to, and at distance exactly 1 to two other terminals (so it is not cheerful). Our proof modifies any given cheerful component with 8 terminals and 2 Steiner points towards this extreme case while keeping the component being cheerful.

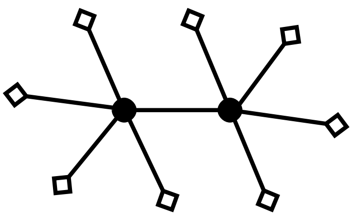

Fig. A1 An extreme case with 8 terminals and 2 Steiner points. Each terminal is at distance exactly 1 to the Steiner point it is connected to, and at distance exactly 1 to two other terminals.

Assume that we have a cheerful component as in Figure A2, it has 8 terminals (t_1 to t_8) and 2 Steiner points (s_1 and s_2). More precisely, if we traverse counterclockwise the embedding of this cheerful component we will have the following closed trail: $t_1, s_1, t_2, s_1, s_2, t_3, s_2, t_4, s_2, t_5, s_2, t_6, s_2, s_1, t_7, s_1, t_8, s_1, t_1$.

An edge between points a and b in Figure A2 represents that distance $d(a, b) \leq 1$; and distance $d(a, b) > 1$ if there is no edge between points a and b. For each pair of terminals t_i, t_j adjacent to the same Steiner point s, by applying law of cosines to triangle $\triangle t_i s t_j$ with $d(s, t_i) \leq 1$, $d(s, t_j) \leq 1$ and $d(t_1, t_j) > 1$, we have $\angle t_i s t_j > 60°$. Note that, any angle $\angle abc$ mentioned in this paper is referred to the one that formed by line segment $\overline{b, a}$ rotating around b to meet line segment $\overline{b, c}$. An angle is positive if the rotation of line segment $\overline{b, a}$ to $\overline{b, c}$ is clockwise, and is negative if this rotation is counter clockwise; and $|\angle abc| \leq 180°$. We say a point c is on the right (left) side of oriented line segment $\overline{b, a}$ if and only if $\angle abc > 0$ ($\angle abc < 0$).

For Steiner point s_1, if at least 3 terminals adjacent to it are on the same side of straight line $s_1 s_2$ and let t be the one closest to s_2 among these terminals, then $\angle s_2 s_1 t < 60°$; and by law of cosines applied to triangle $\triangle t s_2 s_1$ with $d(s_1, s_2) \leq 1$,$d(s_1, t) \leq 1$ and $\angle s_2 s_1 t < 60°$, we have $d(s_2, t) \leq 1$, then we finish the proof. So from here on, we only consider the situation that there are exactly 2 terminals adjacent to each Steiner point on each side of straight line $s_1 s_2$.

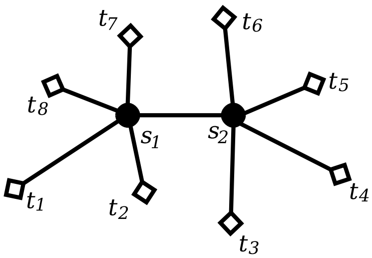

Fig. A2 Example of cheerful component with 2 Steiner points and 8 terminals.

We can modify the locations of some points by modifying some of the edges as follows, while the Euclidean distance between any pair of points without an edge is still greater than 1.

- Extend edge (s_1, s_2) so that $d(s_1, s_2) = 1$ while the location of s_1 is fixed and the embedding of subgraph induced by s_2 and all terminals adjacent to s_2 is fixed.
- Extend edges $(s_1, t_1), (s_1, t_8), (s_2, t_4)$, and (s_2, t_5) so that the length of each of these edges is exactly 1.

We get a graph as in Figure A3 after the modification. Denote two points on straight line $s_1 s_2$ as a and b such that line segment $\overline{s_1, s_2} \subset \overline{a, b}$. Since $\angle t_1 s_1 t_8$ and $\angle t_5 s_2 t_4$ are both greater than $60°$, at least one of the following two values is greater than $60°$: $\angle t_1 s_1 a + \angle b s_2 t_4$ and $\angle a s_1 t_8 + \angle t_5 s_2 b$. Without loss of generality, we assume that $\angle t_1 s_1 a + \angle b s_2 t_4 > 60°$. A special case is that $\angle t_1 s_1 a \geq 60°$ or $\angle b s_2 t_4 \geq 60°$. If $\angle t_1 s_1 a \geq 60°$, then $\angle s_2 s_1 t_1 \leq 120°$. Since

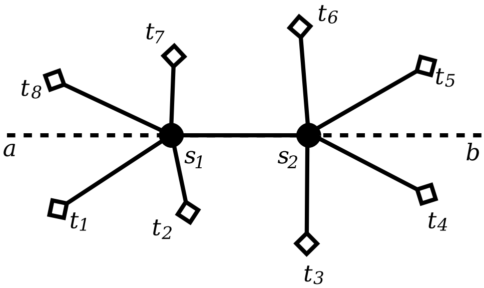

Fig. A3 Modified cheerful component with 2 Steiner points and 8 terminals. We have $d(s_1, s_2) = d(s_1, t_1) = d(s_1, t_8) = d(s_2, t_4) = d(s_2, t_5) = 1$.

point t_2 is in the angle $\angle s_2s_1t_1$, and $\angle t_2s_1t_1 > 60°$, we have that $\angle s_2s_1t_2 < 60°$. Applying law of cosines to triangle $\triangle s_2s_1t_2$ with $d(s_1, s_2) = 1$, $d(s_1, t_2) \leq 1$, we have $d(t_2, s_2) \leq 1$, and then we finish the proof. Similarly, if $\angle bs_2t_4 \geq 60°$, we can also finish the proof. So from here on, we only consider that case that $\angle t_1s_1a < 60°$ and $\angle bs_2t_4 < 60°$. Remind that, $\angle t_2s_1t_1 > 60°$ and $\angle t_4s_2t_3 > 60°$. Thus, in this case, we have that $\angle s_2s_1t_2 + \angle t_3s_2s_1 = (180° - \angle t_1s_1a - \angle t_2s_1t_1) + (180° - \angle bs_2t_4 - \angle t_4s_2t_3) = 360° - \angle t_2s_1t_1 - \angle t_4s_2t_3 - (\angle t_1s_1a + \angle bs_2t_4) < 360° - 60° - 60° - 60° = 180°$.

We further modify the graph by extending the length of one of the edges (s_1, t_2) and (s_2, t_3) to exactly 1. To make sure that we will not shorten the distance between t_2 and t_3 by this modification, we look at the relation between these two terminals. Note that, $\angle s_2s_1t_2 > 60°$ and $\angle t_3s_2s_1 > 60°$. Otherwise, applying law of cosines to triangles $\triangle s_2s_1t_2$ and $\triangle t_3s_2s_1$ respectively, we have that $d(s_2, t_2) \leq 1$ or $d(s_1, t_3) \leq 1$, then we finish the proof. With $d(s_1, t_2) \leq 1$ and $d(s_2, t_3) \leq 1$, we do not have a case that line segments $\overline{s_1, t_2}$ and $\overline{s_2, t_3}$ intersect.

Let p_2 be the projection of t_2 on straight line s_2t_3, and let p_3 be the projection of t_3 on straight line s_1t_2. If either p_2 and t_2 are on different sides of straight line s_1s_2 or p_3 and t_3 are on different sides of straight line s_1s_2, then we show that either $d(s_1, t_3) < 1$ or $d(s_2, t_2) < 1$, and we can finish the proof. Without loss of generality, let t_3 and p_3 be on different sides of straight line s_1s_2 (as shown in Figure A4), then we show that $d(s_1, t_3) < 1$. In this case, we have $\angle t_3s_1t_2 \geq 90°$ or else p_3 and t_3 will be on the same side of straight line s_1s_2. Since $\angle s_2s_1t_2 + \angle t_3s_2s_1 < 180°$, we have that $\angle t_3s_2s_1 + \angle s_2s_1t_3 < 180° - 90° = 90°$ and $\angle s_1t_3s_2 > 90°$. Applying law of sines to triangle $\triangle s_2s_1t_3$ with $\angle s_1t_3s_2 > \angle t_3s_2s_1$, we have $d(s_1, t_3) < d(s_1, s_2) = 1$. From here on, we only consider the case that both p_2, p_3 are on the same side of straight line s_1s_2 as t_2 and t_3.

If s_1, t_2 are on the same side of p_3 on straight line s_1t_2, and s_2, t_3 are on the same side of p_2 on straight line s_2t_3 (as shown in Figure A5), then we show that $d(t_2, t_3) \leq 1$, and we can finish the proof. Through each of point s_1 and point s_2 draw straight lines parallel to straight line t_2t_3. At least one of these two straight lines intersects both line segments $\overline{s_1, t_2}$ and $\overline{s_2, t_3}$. Without loss of

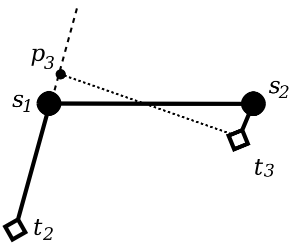

Fig. A4 The case that p_3 and t_3 are on the same side of p_3 on straight line s_1s_2.

generality, we assume that straight line ss_2 intersects $\overline{s_1, t_2}$ at s and intersects $\overline{s_2, t_3}$ at s_2. Since $\angle s_2s_1t_2 + \angle t_3s_2s_1 < 180°$ and line segments $\overline{s_1, t_2}$, $\overline{s_2, t_3}$ do not intersect, we have $d(s, s_2) \geq d(t_2, t_3)$. Let p be the projection of s_2 on straight line s_1t_2. Since line segments $\overline{s_2, p}$, $\overline{t_3, p_3}$ are parallel, and line segments $\overline{s_2, s}$, $\overline{t_3, t_2}$ are parallel, it is easy to see that s_1 and s are on the same side of p on straight line s_1t_2, and $d(p, s) \leq d(p, s_1)$. Therefore $d(s, s_2) \leq d(s_1, s_2)$. Thus, we have $1 = d(s_1, s_2) \geq d(s, s_2) \geq d(t_2, t_3)$. From here on, we only consider the case that either s_1 and t_2 are on different sides of p_3 on straight line s_1t_2 or s_2 and t_3 are on different sides of p_2 on straight line s_2t_3.

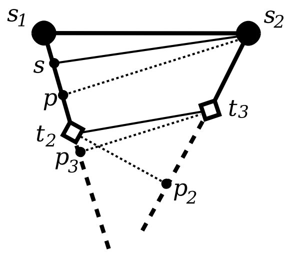

Fig. A5 The case that s_1, t_2 are on the same side of p_3 on straight line s_1t_2, and s_2, t_3 are on the same side of p_2 on straight line s_2t_3. Line segments $\overline{s_2, p}$ and $\overline{t_3, p_3}$ are parallel. Line segments $\overline{s_2, s}$ and $\overline{t_3, t_2}$ are parallel.

Without loss of generality, we assume that s_2 and t_3 are on different sides of p_2 on straight line s_2t_3 (as shown in Figure A6). We extend edge (s_2, t_3) so that $d(s_2, t_3) = 1$. Since we move t_3 farther from p_2 along line s_2t_3, the distance between t_2 and t_3 does not get smaller. Remind that, $\angle t_4s_2t_3 > 60°$ and $\angle t_3s_2s_1 > 60°$. Applying law of cosines to triangle $\triangle t_4s_2t_3$ with $d(s_2, t_3) = d(s_2, t_4) = 1$ and $\angle t_4s_2t_3 > 60°$, and applying law of cosines to triangle $\triangle t_3s_2s_1$ with $d(s_2, t_3) = d(s_1, t_2) = 1$ and $\angle t_3s_2s_1 > 60°$, we still have that $d(t_3, t_4) > 1$ and $d(s_1, t_3) > 1$.

In the rest of this proof, we will show that at least one of the three following distances is within 1: $d(t_2, s_2), d(t_1, t_2)$, and $d(t_2, t_3)$. Here we list some information of the graph that we already proved in the current case. We can use this information in the rest of the proof.

- $d(s_1, s_2) = d(s_1, t_1) = d(s_2, t_3) = d(s_2, t_4) = 1.$

- $d(s_1, t_2) \leq 1$.
- $\angle t_2 s_1 t_1 > 60°$ and $\angle t_4 s_2 t_3 > 60°$.
- $\angle s_2 s_1 t_2 > 60°$ and $\angle t_3 s_2 s_1 > 60°$.
- $\angle t_1 s_1 a + \angle b s_2 t_4 > 60°$.
- $\angle t_1 s_1 a < 60°$ and $\angle b s_2 t_4 < 60°$.
- $\angle s_2 s_1 t_2 + \angle t_3 s_2 s_1 < 180°$.

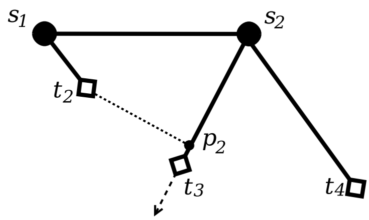

Fig. A6 The case that s_2 and t_3 are on different sides of p_2 on straight line s_2t_3. In this case, if we extend edge (s_2, t_3) so that $d(s_2, t_3) = 1$, we still have that $d(t_3, t_4) > 1$, and we will not shorten the distance between t_2 and t_3.

Denote the circle centered at point d with radius 1 as $C(d)$. Let $C(s_2)$ and $C(t_3)$ intersect on the left of line segment $\overline{t_3, s_2}$ at point z, then triangle $\triangle z t_3 s_2$ is an equilateral triangle. Let point x be on the right side of line segment $\overline{s_1, s_2}$ with $\overline{s_1, x}$ parallel with $\overline{s_2, t_3}$ and $d(s_1, x) = 1$. The locations of points z and x are illustrated in Figure A7. Since $d(s_1, x) = d(s_1, s_2) = d(s_2, t_3)$ and line segments $\overline{s_1, x}$ and $\overline{s_2, t_3}$ are parallel, the quadrilateral $s_1 x t_3 s_2$ is a rhombus. Remind that $\angle s_2 s_1 t_2 + \angle t_3 s_2 s_1 < 180°$. Since $\angle s_2 s_1 x + \angle t_3 s_2 s_1 = 180°$, we have that t_2 is in the angle $\angle s_2 s_1 x$. Since t_2 is outside of circles $C(t_3)$ and $C(s_2)$, we have that t_2 is inside of triangle $\triangle x z s_1$.

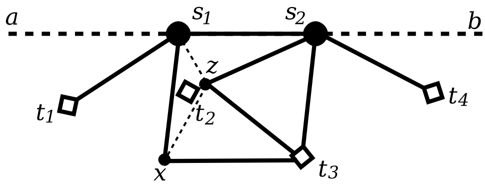

Fig. A7 Locations of points z and x. Each solid line segment in the figure has length exactly 1.

In the rest of this proof, we will show that $d(t_1, x) \leq 1$ and $d(t_1, z) \leq 1$. Since t_2 is a convex combination of s_1, x and z, after this proof, together with the fact that $d(t_1, s_1) = 1$, we have that $d(t_1, t_2) \leq 1$, and then we can finish the proof of Lemma 11.

Firstly, in this paragraph we show that $d(t_1, x) \leq 1$. Remind that, $\angle t_4 s_2 t_3 > 60°$, $\angle t_1 s_1 a + \angle b s_2 t_4 > 60°$ and $\angle s_2 s_1 x + \angle t_3 s_2 s_1 = 180°$. We have that, $\angle x s_1 t_1 = 180° - \angle t_1 s_1 a - \angle s_2 s_1 x = 360° - (\angle t_1 s_1 a + \angle b s_2 t_4) - (\angle s_2 s_1 x +$

$\angle t_3s_2s_1) - \angle t_4s_2t_3 < 360° - 60° - 180° - 60° = 60°$. Applying law of cosines to triangle $\triangle s_1t_1x$ with $d(s_1,t_1) = d(s_1,x) = 1$ and $\angle xs_1t_1 < 60°$, we have that $d(t_1,x) \le 1$.

Last, in this paragraph we show that $d(t_1,z) \le 1$. Let y be a point such that $d(s_1,y) = 1$ and $\angle xs_1y = 60°$. The location of point y is illustrated in Figure A8. Since $d(s_1,x) = d(s_1,y) = 1$ and $\angle xs_1y = 60°$, the triangle $\triangle s_1yx$ is an equilateral triangle. Remind that line segments $\overline{s_1,x}$ and $\overline{s_2,t_3}$ are parallel, and triangle $\triangle zt_3s_2$ is an equilateral triangle, so line segments $\overline{s_1,y}$ and $\overline{s_2,z}$ are parallel. Since $d(s_1,y) = d(s_1,s_2) = d(s_2,z) = 1$ and line segments $\overline{s_1,y}$, $\overline{s_2,z}$ are parallel, the quadrilateral s_1yzs_2 is a rhombus and $d(y,z) = 1$. Remind that $\angle xs_1t_1 < 60°$, we have $\angle zs_1t_1 < \angle zs_1y$. Since $f(x) = \cos x, x \in [0,\pi]$ is monotone decreasing, by law of cosines applied to triangles $\triangle s_1yz$ and $\triangle s_1t_1z$, we have that $d(t_1,z) = \sqrt{d(s_1,z)^2 + 1 - 2d(s_1,z)\cos\angle zs_1t_1} < \sqrt{d(s_1,z)^2 + 1 - 2d(s_1,z)\cos\angle zs_1y} = d(y,z) = 1$. Till here, we finish the proof that $d(t_1,t_2) \le 1$.

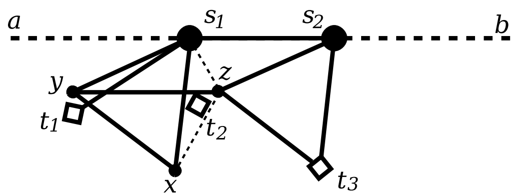

Fig. A8 Location of point y. Each solid line segment in the figure has length exactly 1.

$\square$