On the Bidirected Cut Relaxation for Steiner Forest

Jarosław Byrka ⋆[1], Fabrizio Grandoni ⋆⋆[2], and Vera Traub ⋆⋆⋆[3]

[1] University of Wrocław, jaroslaw.byrka@cs.uni.wroc.pl,[0000−0002−3387−0913]
[2] IDSIA, USI-SUPSI, fabrizio.grandoni@idsia.ch,[0000−0002−9676−4931]
[3] Research Institute for Discrete Mathematics and Hausdorff Center for Mathematics, University of Bonn, traub@dm.uni-bonn.de,[0000−0001−9749−2600]

Abstract. The Steiner Forest problem is an important generalization of the Steiner Tree problem. We are given an undirected graph with nonnegative edge costs and a collection of pairs of vertices. The task is to compute a cheapest forest with the property that the elements of each pair belong to the same connected component of the forest. The current best approximation factor for Steiner Forest is 2, which is achieved by the classical primal-dual algorithm; improving on this factor is a big open problem in the area.

Motivated by this open problem, we study an LP relaxation for Steiner Forest that generalizes the well-studied Bidirected Cut Relaxation for Steiner Tree. We prove that this relaxation has several promising properties. Among them, it is possible to round any half-integral LP solution to a Steiner Forest instance while increasing the cost by at most a factor $\frac{16}{9}$. To prove this result we introduce a novel recursive densest-subgraph contraction algorithm.

Keywords: network design, Steiner forest, bidirected cut relaxation

1 Introduction

In the *Steiner Forest* problem we are given an undirected graph $G = (V, E)$ with nonnegative edge costs $c : E \to \mathbb{R}_{\geq 0}$. Furthermore, we are given a collection $\mathcal{P}$ of pairs of vertices. The task is to compute a subset of edges $F \subseteq E$ of minimum total cost $c(F) := \sum_{e \in F} c(e)$ such that, for each pair $P = \{s, t\} \in \mathcal{P}$, the graph (V, F) contains an s-t path. There is always an optimal solution F where (V, F) is a forest. We will refer to vertices $R \subseteq V$ that are present in at least one pair from $\mathcal{P}$ as *terminals*, and call the remaining vertices *Steiner vertices*.

Steiner Forest generalizes the famous Steiner tree problem, where all terminals need to be connected to each other. Because Steiner Tree is NP-hard, indeed

⋆ Supported by NCN grant number 2020/39/B/ST6/01641.
⋆⋆ Partially supported by the SNF Grant 200021_200731/1
⋆⋆⋆ Supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 537750605

APX-hard [19], the same applies to Steiner Forest. The current-best approximation factor for Steiner Forest is 2, achieved in a classical paper by Agrawal, Klein, and Ravi [1] using the primal-dual method (see also [24]). Incidentally, this result is also one of the earliest and most famous examples of the primal-dual technique. Improving on the 2-approximation for Steiner Forest is probably among the most prominent open problems in the area of approximation algorithms (see e.g. [39]).

The main motivation for our work is a very recent result [10] about the integrality gap of the Bidirected Cut Relaxation for Steiner Tree (following a long line of research [14,15,27,21]). To define this relaxation, we first *bidirect* all edges, i.e., we replace each undirected edge $\{u, v\}$ with two directed edges (u, v) and (v, u), inheriting the same cost as $\{u, v\}$. We let $\overrightarrow{E}$ be the resulting set of directed edges. We also select an arbitrary terminal $r \in R$ as a root.

The idea behind the relaxation is to consider the equivalent problem of finding a directed Steiner Tree solution $F \subseteq \overrightarrow{E}$ where F is the edge set of a tree oriented towards the root r. Any such directed solution $F \subseteq \overrightarrow{E}$ has to contain a directed path from each terminal to r. Equivalently, for every vertex set U containing some terminal but not r, the solution F must contain at least one outgoing edge of U. We write $\delta^+(U)$ to denote the set of of outgoing edges of U. This leads to the Bidirected Cut Relaxation for Steiner Tree, which is defined as follows:

$$
\begin{aligned}
\min \sum_{e \in \overrightarrow{E}} c(e) \cdot x_e & & \text{(Tree-BCR)} \\
\text{s.t.} \sum_{e \in \delta^+(U)} x_e \geq 1 & \qquad \forall U \subseteq V \setminus \{r\}, U \cap R \neq \emptyset. &
\end{aligned}
$$

Given a solution x to Tree-BCR, we denote its cost by $c(x) := \sum_{e \in \overrightarrow{E}} c(e) \cdot x_e$. Tree-BCR is one of the oldest and best-studied relaxations for Steiner Tree (see e.g. [22]). A classical result by Edmonds [20] shows that it is integral in the special case when $R = V$. In this case, Steiner Tree is equivalent to the Minimum Spanning Tree problem. The best-known lower bound on the integrality gap of Tree-BCR is only $\frac{6}{5}$ (see [38], improving on [9]). This gives hope that Tree-BCR might have a quite small integrality gap. Until very recently however, the best-known upper bound was only 2. In [10] this was improved for the first time, by showing an upper bound of 1.9988 on the integrality gap of Tree-BCR.

The latter result motivated us to study the Bidirected Cut Relaxation for Steiner Forest (Forest-BCR). To obtain this relaxation, we bidirect the edges as before. Consider any Steiner Forest solution F where each connected component of (V, F) is a tree. Then we can orient the edges of each such component towards an arbitrary root vertex. We consider the linear programming relaxation with variables z_P^r for $r \in V$ and $P \in \mathcal{P}$ specifying whether the vertices of the pair P belong to a connected component with root r. Moreover, we have variables x_e^r specifying whether a directed edge $e \in \overrightarrow{E}$ is used as part of a connected component with root r. This leads to the following relaxation:

$$\begin{aligned}
\min \ & \sum_{r\in V}\sum_{e\in \overrightarrow{E}} c(e)\cdot x_e^r & & \text{(Forest-BCR)}\\
\text{s.t.} \ & \sum_{r\in V} z_P^r = 1 & & \text{for all } P\in\mathcal{P}\\
& \sum_{e\in\delta^+(U)} x_e^r \ \ge\ z_P^r & & \text{for all } r\in V, P\in\mathcal{P}, \text{ and } U\subseteq V\setminus\{r\} \text{ with } P\cap U\neq\emptyset\\
& x_e^r \ \ge\ 0 & & \text{for all } r\in V \text{ and } e\in\overrightarrow{E}\\
& z_P^r \ \ge\ 0 & & \text{for all } r\in V \text{ and } P\in\mathcal{P}.
\end{aligned}$$

Given a solution (x, z) to Forest-BCR, we use $c(x) := \sum_{r\in V}\sum_{e\in\overrightarrow{E}} c(e)\cdot x_e^r$ as a shortcut for its cost.

The integer program corresponding to Forest-BCR turned out to perform well in experimental studies [34], [4] but to the best of our knowledge, from a theoretical viewpoint Forest-BCR has not been studied before. In this work, we investigate the properties of this LP relaxation, which in the long run might help to improve on the longstanding 2-approximation algorithm for Steiner Forest.

1.1 Our Contributions

One of our main results is an algorithm to round a half-integral solution[5] to Forest-BCR while loosing a factor smaller than 2 in the cost. Half-integral solutions often play an important role in developing rounding algorithms for network design problems, see e.g. [12,7,29]. While it remains open whether the integrality gap of Forest-BCR is less than 2, the result from the next theorem (proof in Section 2) gives some supporting evidence that this might be the case.

Theorem 1. *Given any half-integral solution (x, z) to Forest-BCR, we can compute in polynomial time a feasible solution to the associated Steiner Forest instance of cost at most $\frac{16}{9}\cdot c(x)$.*

To prove the above result, we provide a simple rounding algorithm which is applicable beyond half-integral solutions. The half-integrality of (x, z) will be used only in the analysis of the algorithm.

The basic idea of our algorithm is a recursive contraction of densest subgraphs with respect to x. First, we turn the given LP solution (x, z) into a more structured one, applying splitting-off operations as long as this maintains feasibility of the solution. (Splitting-off means that we reduce the value of variables $x^r_{(u,v)}$ and $x^r_{(v,w)}$ while increasing the value of $x^r_{(u,w)}$ by the same amount.) Next, we identify a densest subgraph with respect to x. More precisely, we compute a subset of vertices W that maximizes the sum of the values $x^r_{(u,v)}$ with $u, v\in W$,

[4] Forest-BCR is equivalent to the LP underlying the best performing IP in [34].
[5] I.e., a solution where the value of each variable is an integer multiple of $\frac{1}{2}$.

divided by $|W|-1$[6]. Then we add a minimum spanning tree $\mathrm{MST}(W)$ on vertex set W to the solution under construction, contract W (updating $\mathcal{P}$ and (x,z) in the natural way), and continue recursively until each pair is connected.

We prove that our algorithm yields a solution of cost less than $2 \cdot c(x)$ if the maximum density of a set W with respect to x is always larger than $\frac{1}{2}$. Moreover, we prove that in the half-integral case, we can achieve a density of at least $\frac{9}{16}$, leading to Theorem 1.

Next, we study further basic properties of Forest-BCR. Recall that the pairs $\mathcal{P}$ do not need to be disjoint and a vertex can appear in multiple pairs. Often the same Steiner Forest instance can be represented in multiple equivalent ways. For example, the pairs $\{\{a,b\},\{b,c\}\}$ can be replaced by the pairs $\{\{a,b\},\{a,c\}\}$ without changing the set of feasible solutions. We investigate the impact of such representation changes on Forest-BCR.

Define the *demand graph* of an instance to be the graph with vertex set V that contains an edge between two vertices s and t if and only if the pair $\{s,t\}$ belongs to $\mathcal{P}$. All vertices that belong to the same connected component of the demand graph must be connected in any feasible Steiner Forest solution. Thus, if for two instances $((V,E),c,\mathcal{P}_1)$ and $((V,E),c,\mathcal{P}_2)$ the vertex sets of the connected components of the demand graphs are the same, then the two instances are equivalent. In this case, we also say that $\mathcal{P}_1$ is a different *representation* of $\mathcal{P}_2$ and vice versa.

Even though the feasible solutions of the two instances are identical (and the cost function is the same), it turns out that the value of Forest-BCR can depend on the representation.

Theorem 2. *There exists an instance $((V,E),c,\mathcal{P}_1)$ of the Steiner Forest problem and a different representation $\mathcal{P}_2$ of $\mathcal{P}_1$ such that the value of Forest-BCR is not the same for the two instances $((V,E),c,\mathcal{P}_1)$ and $((V,E),c,\mathcal{P}_2)$.*

We remark that, on the instances we consider in the proof of Theorem 2 (see full version [11]), Forest-BCR has optimal half-integral solutions. Hence already in this case, where we show that a good rounding algorithm exists, the representation matters.

Given Theorem 2, one might ask what the best way is to represent a given instance. For a given fixed instance, one could simply add all pairs $\{s,t\}$ for which s and t are in the same connected component of the demand graph. Alternatively, one could also generalize Forest-BCR by allowing $\mathcal{P}$ to contain arbitrary subsets of V rather than only subsets of size two. Then one can simply include the vertex set of every connected component of the demand graph in $\mathcal{P}$. The latter construction yields a relaxation that is at least as strong as any representation using only pairs.[7]

[6] Notice that we do not divide by $|W|$ as it is done in other definitions of density. This difference is crucial for us.

[7] Suppose $\mathcal{P}$ contains a subset A, where possibly $|A|>2$, then we can turn any LP solution (x,z) into a feasible LP solution for the instance containing all pairs $P \subseteq A$

However, to prove an upper bound on the integrality gap for general instances of Steiner Forest, we need to prove an upper bound for any representation of the instance. To see this, note that whenever a vertex appears in multiple pairs, we can replace it by several collocated copies. This changes neither the value of Forest-BCR nor the cost of an optimum Steiner Forest solution. Of course, one could contract vertices at distance zero upfront, but a similar situation arises if the copies have a very small, but positive distance from each other. In this new instance, there is only one possible representation of the pairs but the integrality gap of the LP is essentially the same as for the original instance.

We say that a connected component of the demand graph is *trivial* if it contains only a single vertex and we call it *nontrivial* otherwise. Observe that Steiner Tree instances are instances of Steiner Forest where the demand graph has only one nontrivial connected component. In the full version of this paper [11], we prove that for Steiner Tree instances, the value of Forest-BCR does not depend on the representation of the pairs $\mathcal{P}$. Even more, this value is always equal to the value of Tree-BCR for the corresponding Steiner Tree instance. In particular, the result from [10] implies that such instances have integrality gap strictly smaller than 2, namely at most 1.9988.

Theorem 3. *Let $(G, c, \mathcal{P})$ be an instance of Steiner Forest where the demand graph has only one nontrivial connected component (with vertex set R). Then the value of Forest-BCR is equal to the value of Tree-BCR for the Steiner Tree instance (G, c, R).*

Finally, we provide a family of instances showing that the integrality gap of Forest-BCR is at least $\frac{3}{2}$ (see Section 3).

Theorem 4. *The integrality gap of Forest-BCR is at least $\frac{3}{2}$.*

Because Forest-BCR is a strengthening of the classical undirected LP relaxation of Steiner Forest,[8] which has an integrality gap of 2 [1,24,28], the integrality gap of Forest-BCR is at most 2. Whether the integrality gap of Forest-BCR is strictly less than 2, remains an interesting question for future work. We remark that for the instances we use to prove Theorem 4, the optimal solutions to Forest-BCR are not half-integral. Nevertheless, the rounding algorithm we use to prove Theorem 1 yields optimal integral solutions for these instances.

1.2 Related Work

The *Steiner Network* problem, also known as *Survivable Network Design*, is a generalization of Steiner Forest where we are given pairwise vertex connectivity

by setting $z^r_P := z^r_A$ for every such pair and each root $r \in V$. Hence, the LP relaxation where A is part of $\mathcal{P}$ is at least as strong as any equivalent representation containing pairs $P \subseteq A$ instead of A.

[8] To see this, observe that for every feasible solution (x, z) to Forest-BCR, the vector $\tilde{x} \in \mathbb{R}^E$ defined by $\tilde{x}_{\{u,v\}} := \sum_{r \in V}(x^r_{(u,v)} + x^r_{(v,u)})$ is a feasible solution to the undirected LP relaxation.

requirements $\lambda_{u,v} \geq 0$, and the task is to compute a cheapest subgraph of G such that each such pair of vertices u, v is $\lambda_{u,v}$-edge connected. In a celebrated result, Jain [28] obtained a 2-approximation for this problem using the iterative rounding technique. Since then, improving on the 2-approximation barrier, even just for special cases of Steiner Network, became an important open problem. This was recently achieved for some problems in this family, such as Connectivity Augmentation [8,13,35,36,37] and Forest Augmentation [25]. Among the special cases for which 2 is still the best-known factor, we already mentioned the Steiner Forest problem. Another interesting special case is the Minimum-Weight 2-Edge Connected Spanning Subgraph problem.

For the Steiner Tree problem several better than 2 approximation algorithms are known. A series of works based on the relative-greedy approach [41,40,31,32] culminated in a 1.55 approximation [33]. The current-best approximation algorithm achieves an approximation guarantee of $(\ln 4 + \varepsilon) \approx 1.39$ [9] and is based on an iterative randomized rounding approach, applied to the so-called Hypergraphic Cut Relaxation for Steiner tree. This LP relaxation has integrality gap at most $\ln 4$ [23]. Recently, the same approximation factor and a new proof of this integrality gap upper bound has been obtained using local search [36].

For Steiner Forest, better-than-2 approximation algorithms are only known in special cases. In particular, approximation schemes are known for planar and bounded tree-width graphs [4], and for Euclidean plane instances [6]. A better-than-2 approximation is also known for very dense unit weight graphs [30].

For the prize-collecting generalizations of Steiner Tree and Steiner Forest, where some terminals or terminal pairs might be left disconnected by paying a given associated cost, the currently best-known approximation factors are 1.79 for prize-collecting Steiner Tree [3] and 2 for prize-collecting Steiner Forest [2].

For the directed analogue of the Steiner Forest problem, approximation factors of $O(|R|^{1/2+\varepsilon})$ [17] and $O(|V|^{2/3+\varepsilon})$ [5] are known for every fixed $\varepsilon > 0$. For the special case of planar graphs, polylogarithmic approximation factors have been obtained in [18]. For the directed analogue of Steiner Tree, the current best approximation algorithms for it are $|R|^{\varepsilon}$ in polynomial time [16] and $O(\log^2 |R| / \log\log |R|)$ in quasi-polynomial time [26].

1.3 Notation

For a vertex v, we denote by $\delta^-(v)$ and $\delta^+(v)$ the sets of incoming and outgoing edges of v, respectively. Similarly, for a vertex set U, we denote by $\delta^-(U)$ and $\delta^+(U)$ the incoming and outgoing edges of U, respectively.

For a solution (x, z) to Forest-BCR and a vertex $r \in V$, we write $c(x^r) := \sum_{e \in \overrightarrow{E}} c(e) \cdot x^r_e$. Moreover, for an edge set $F \subseteq E$, we write $\overrightarrow{F}$ to denote the directed edge set obtained by bidirecting F and define $x^r(F) := \sum_{e \in \overrightarrow{F}} x^r_e$ and $x(F) := \sum_{r \in V} x^r(F)$.

Finally, for a graph $G = (V, E)$ and a set $W \subseteq V$, we write $G[W]$ to denote the subgraph of G induced by W, and we write $E[W]$ to denote the edge set of this subgraph, i.e., the set of all edges from E with both endpoints in W.

2 Rounding half-integral LP solutions

In this section we prove Theorem 1, which we restate here for convenience.

Theorem 1. *Given any half-integral solution (x, z) to Forest-BCR, we can compute in polynomial time a feasible solution to the associated Steiner Forest instance of cost at most $\frac{16}{9} \cdot c(x)$.*

To prove Theorem 1 we describe an algorithm that could be applied to round arbitrary solutions (x, z) to Forest-BCR and then prove that it yields a Steiner Forest solution of cost at most $\frac{16}{9}c(x)$ if the LP solution (x, z) is half-integral.

2.1 Description of the algorithm

We now describe our recursive algorithm, which is summarized in Algorithm 1. See Figure 1 for an example. If $\mathcal{P} = \emptyset$, we return the optimum solution, which is $F = \emptyset$.

Otherwise, we compute the metric closure $(\overline{G}, \overline{x})$ of the weighted graph (G, c). Next, we modify the given LP solution (x, z) into a more structured LP solution for the instance $(\overline{G}, \overline{c}, \mathcal{P})$. First, we ensure that whenever $z^r_P > 0$ for some vertex $r \in V$ and some pair $P \in \mathcal{P}$, then r is a terminal. This modification can be done without increasing the cost of the LP solution by a simple rerouting step (see full version [11]). We remark that we could have required this property in Forest-BCR, in which case it would be automatically satisfied at the beginning of the algorithm. However, the property will not be automatically satisfied for the instances on which we call the algorithm recursively.

Then we do splitting-off steps, as long as possible: If for a vertex $v \in V$, a root $r \in V$, and edges (u, v) and (v, w) we can reduce $x^r_{(u,v)}$ and $x^r_{(v,w)}$ by $\varepsilon > 0$ and increase $x^r_{(u,w)}$ by ε while maintaining the feasibility of our LP solution, then we do so for the largest possible such ε. Because the cost function $\overline{c}$ is metric, this does not increase the cost of our LP solution. We then obtain an LP solution (x, z) with the following two properties:

(a) We have $z^r_P = 0$ for all $r \in V \setminus R$ and all $P \in \mathcal{P}$.
(b) No splitting-off step is possible while maintaining the feasibility of (x, z).

We call an LP solution with these two properties *well structured.*

Having a well structured LP solution for the instance $(\overline{G}, \overline{c}, \mathcal{P})$, we compute a densest subgraph of $\overline{G} = (V, E)$ with respect to the following notion of density:

Definition 1. *For $W \subseteq V$ with $|W| \geq 2$, we define the density of W with respect to x as*

$$\text{density}_x(W) := \frac{x(E[W])}{|W| - 1}.$$

Algorithm 1 Recursive algorithm to round a solution (x, z) to Forest-BCR for a given instance $(G, c, \mathcal{P})$.

If $\mathcal{P} = \emptyset$, **return** $\emptyset$.

Let $(\overline{G}, \bar{c})$ be the metric closure of (G, c).

Replace (x, z) with a well structured solution for the instance $(\overline{G}, \bar{c}, \mathcal{P})$ without increasing the cost.

Compute a densest subgraph of G with respect to edge weights x, i.e., compute a set $W \subseteq V$ maximizing $\text{density}_x(W) := \frac{x(E[W])}{|W|-1}$.

Compute a minimum spanning tree $\text{MST}(W)$ of $\overline{G}[W]$ with respect to the costs $\bar{c}$.

Call the algorithm recursively on the instance and LP solution that arise by contraction of W and let F_W be the resulting edge set.

Let F result from $F_W \cup \text{MST}(W)$ by replacing every edge $\{v, w\}$ by a shortest v-w path in (G, c).

Return F.

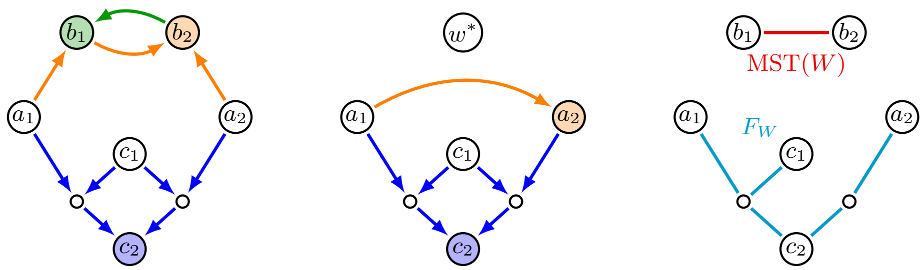

Fig. 1: The left part of figure illustrates an LP solution for an instance with pairs $\{a_1, a_2\}$, $\{b_1, b_2\}$, and $\{c_1, c_2\}$. Every edge e with $x_e^r = \frac{1}{2}$ is drawn in the same color as the vertex r. In the middle, we see the LP solution in the recursive call of Algorithm 1 (applied to the instance arising from the contraction of $W = \{b_1, b_2\}$) after rerouting and splitting-off. The solution returned by the algorithm is shown on the right.

We compute $W \subseteq V$ with $|W| \geq 2$ maximizing $\text{density}_x(W)$. Such a set W can be found in polynomial time by a reduction to submodular function minimization; details in the full version of the paper [11].

Next we compute a minimum cost spanning tree $\text{MST}(W)$ in the graph $\overline{G}[W]$ induced by W with respect to edge costs $\bar{c}$. Then we call the algorithm recursively on the instance obtained by the contraction of W into a vertex w^*. We identify the edges in the contracted instance with the corresponding edges in the original instance. If parallel edges would arise by the contraction, we keep only one edge with minimal cost. Edges with both endpoints in W are no longer present after contraction.

When contracting the set W, we adapt the set of pairs in the natural way: we remove every pair $P \in \mathcal{P}$ with $P \subseteq W$; in every pair $\{s, t\}$ with $s \in W$ and $t \notin W$, we replace s with w^*. We also transform the LP solution (x, z) in the canonical way as follows. We set $z_P^{w^*} := \sum_{w \in W} z_P^w$ for each $P \in \mathcal{P}$. For all vertices $r, v \in$

$V \setminus W$, we set $x^r_{(v,w^*)} \coloneqq \sum_{w \in W} x^r_{(v,w)}$ and similarly $x^r_{(w^*,v)} = \sum_{w \in W} x^r_{(w,v)}$. Moreover, for each vertex $v \in V \setminus W$, we set $x^{w^*}_{(v,w^*)} \coloneqq \sum_{w \in W} \sum_{u \in W} x^w_{(v,u)}$ and similarly $x^{w^*}_{(w^*,v)} \coloneqq \sum_{w \in W} \sum_{u \in W} x^w_{(u,v)}$.

Then we take the union of $\mathrm{MST}(W)$ and the edge set F_W obtained from the recursive call to obtain a solution to the instance $(\overline{G}, \overline{c}, \mathcal{P})$. Finally, we replace every edge $\{v, w\}$ by a shortest v-w path in the original graph G with respect to the cost function c. In this way, we obtain a Steiner Forest solution to the original instance $(G, c, \mathcal{P})$.

2.2 Analysis of the algorithm

For a proof that Algorithm 1 can be implemented to run in polynomial time we refer to the full version of the paper [11]. We next analyze the approximation ratio of the algorithm and prove that we obtain a solution of cost at most $\frac{16}{9} \cdot c(x)$ if (x, z) is half-integral.

First, we discuss how we can bound the cost $\mathrm{mst}(W)$ of the minimum spanning tree $\mathrm{MST}(W)$ in $\overline{G}[W]$. We denote by $x|_{E[W]}$ the vector x restricted to directed edges with both endpoints in W and therefore write
$c(x|_{E[W]}) = \sum_{v,w \in W} c(e) \cdot \sum_{r \in V} x^r_{(v,w)}$.

Lemma 1. *Let (x, z) be a solution to Forest-BCR and let $W \subseteq V$ with $|W| \geq 2$ maximizing* $\mathrm{density}_x(W)$. *Then*

$$\mathrm{mst}(W) \leq \frac{c(x|_{E[W]})}{\mathrm{density}_x(W)}.$$

To prove Lemma 1, we consider the vector $y \in \mathbb{R}^{E[W]}_{\geq 0}$ that is given by $y_{\{v,w\}} \coloneqq \frac{1}{\mathrm{density}_x(W)} \cdot \sum_{r \in R} \big(x^r_{(v,w)} + x^r_{(w,v)}\big)$ for every edge $\{v, w\} \in E[W]$. One can show that y is contained in the spanning tree polytope, using the description of the spanning tree polytope by partition constraints. Here, we crucially exploit the fact that W maximizes the density with respect to x. Then we can bound $\mathrm{mst}(W)$ by $c(y) = \frac{c(x|_{E[W]})}{\mathrm{density}_x(W)}$. For details of the proof, see full version [11].

The next step our analysis is to prove that for a well structured half-integral LP solution (x, z), a sufficiently dense subgraph with respect to x does always exist:

Lemma 2. *Let (x, z) be a well structured solution to Forest-BCR for an instance with $|\mathcal{P}| \geq 1$. If (x, z) is half-integral, then there exists a set $W \subseteq V$ with $|W| \geq 2$ and* $\mathrm{density}_x(W) \geq \frac{9}{16}$.

The proof of Lemma 2 is the only part of our analysis where we use the half-integrality of the LP solution. We first observe that we may assume that decreasing the value of any variable x^r_e will make the LP solution infeasible: otherwise, we can decrease x^r_e by $\frac{1}{2}$ and this can only decrease the density of any set W with respect to x. (The decrease of x^r_e will also maintain the property that (x, z) is well structured.)

Next, we consider the *projection multi-graph* $\hat{G} = (V, \hat{E})$ obtained from a half-integral solution (x, z) as follows. The graph $\hat{G}$ has vertex set V and its edge set contains $2 \cdot \sum_{r \in V}(x^r_{(v,w)} + x^r_{(w,v)})$ copies of each edge $\{v, w\}$. We say that a vertex is a *high-degree vertex* if it has degree at least three in $\hat{G}$ and call it a *low-degree vertex*, otherwise.

We now use the fact that (x, z) is well structured to show that every Steiner vertex has either degree zero in $\hat{G}$ or it is a high-degree vertex. Consider a Steiner vertex v of nonzero degree in $\hat{G}$. Because of our assumption that decreasing the value of any variable x^r_e makes the LP solution infeasible and because $z^v_P = 0$ for every pair $P \in \mathcal{P}$, we have $x(\delta^+(v)) > 0$ and $x(\delta^-(v)) > 0$, implying that v has degree at least two in $\hat{G}$. If $x(\delta^+(v)) = \frac{1}{2}$ and $x(\delta^-(v)) = \frac{1}{2}$, we could have applied a splitting-off step at v, contradicting the fact that (x, z) is well structured. Hence, at least one of $x(\delta^+(v))$ and $x(\delta^-(v))$ must be at least 1, implying that v has degree at least three in $\hat{G}$.

Then, using the feasibility of the LP solution (x, z) for Forest-BCR, we can show that every vertex v of nonzero degree in $\hat{G}$, which we will also call a *support vertex*, has degree at least two in $\hat{G}$ and satisfies in addition at least one of the following two properties:

(i) there is a vertex $w \in V$ such that $\hat{G}$ contains two parallel edges $\{v, w\}$;
(ii) either v itself is a high-degree vertex, or one of its neighbors in $\hat{G}$ is a high-degree vertex.

See Lemma 2.7 in the full version [11] for a proof of this fact.

To prove Lemma 2, we next lower bound the maximum density. If at least one terminal v satisfies (i), then $W = \{v, w\}$ satisfies $\text{density}_x(W) = 1 \geq \frac{9}{16}$. Otherwise every support vertex satisfies (ii) and we consider the set W consisting of all support vertices. In order to prove $\text{density}_x(W) \geq \frac{9}{16}$, we apply a token counting argument. Initially we distribute $2|\hat{E}|$ tokens by giving to every support vertex one token per incident edge in $\hat{G}$. Then we redistribute the tokens: every low-degree vertex in W keeps its 2 tokens and every high-degree vertex of degree $k \geq 3$ keeps $2 + \frac{k-2}{k+1} \geq \frac{9}{4}$ tokens for itself and gives $\frac{k-2}{k+1} \geq \frac{1}{4}$ tokens to each of its neighbors. Then (ii) implies that every support vertex has at least $\frac{9}{4}$ tokens, implying $2|\hat{E}[W]| = 2|\hat{E}| \geq \frac{9}{4} \cdot |W|$. From this we conclude $\text{density}_x(W) \geq \frac{9}{16}$. For full details of the proof of Lemma 2 we refer to the full version [11].

Having shown Lemma 1 and Lemma 2, we can prove that Algorithm 1 returns a Steiner Forest solution of cost at most $\frac{16}{9} \cdot c(x)$ whenever (x, z) is half-integral. To this end, we observe that the modifications of the solution (x, z) turning it into a well structured solution can only decrease the cost $c(x)$. Then we let (x_W, z_W) denote the LP solution that arises by contraction of W, i.e., the LP solution which we round to the edge set F_W by calling the algorithms recursively. Because the number of vertices decreases by at least one in every recursive call, we may assume by induction on the number of vertices, that $c(F_W) \leq \frac{16}{9} \cdot c(x_W)$. Moreover, by Lemma 1, we have $\text{mst}(W) \leq \frac{c(x|_{E[W]})}{\text{density}_x(W)}$ and thus by Lemma 2, $\text{mst}(W) \leq \frac{16}{9} \cdot c(x|_{E[W]})$. Using $c(x_W) + (x|_{E[W]}) \leq c(x)$, this implies the claimed upper bound on the cost of the returned edge set F.

3 Lower bound on the integrality gap

In this section we prove that the integrality gap of Forest-BCR is at least $\frac{3}{2}$. We consider for every $q \in \mathbb{Z}_{>0}$, the following instance of Steiner Forest. The graph G has vertices s_i, v_i, and t_i for $i \in \{1, \dots, q\}$. The edges of G consist of all edges between vertices s_i and v_j, and all edges between vertices v_i and t_j, i.e.,

$$E \; \coloneqq \; \big\{\{s_i, v_j\} : i, j \in \{1, \dots, q\}\big\} \cup \big\{\{v_i, t_j\} : i, j \in \{1, \dots, q\}\big\}.$$

See Figure 2. The set $\mathcal{P}$ contains the pairs $\{s_i, t_i\}$ for $i \in \{1, \dots, q\}$ and the pairs $\{v_i, v_{i+1}\}$ for $i \in \{1, \dots, q-1\}$. The cost of every edge is 1.

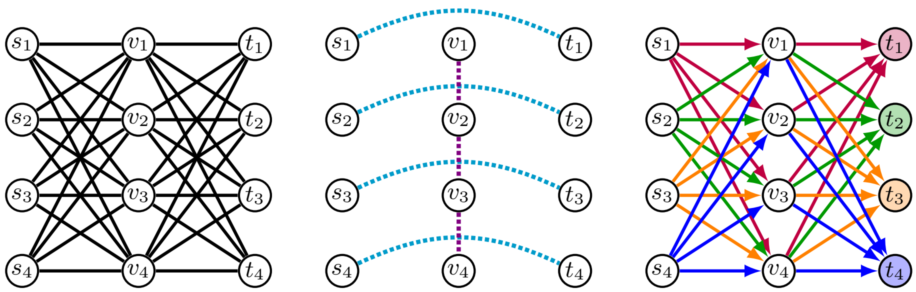

Fig. 2: The instance and LP solution we use to establish an integrality gap lower bound for $q = 4$. The figure shows the graph $G = (V, E)$ (left), the demand graph of the instance (middle), and an illustration of the values of x^r_e in the LP solution (right). For an edge $e \in \overrightarrow{E}$ we have $x^r_e = \frac{1}{q} = \frac{1}{4}$ if e is drawn in the same color as the vertex r. All other variables x^r_e are zero. In particular, for every non-colored vertex r, we have $x^r_e = 0$ for all $e \in \overrightarrow{E}$.

In any feasible solution F of this instance, the vertices $v_1, \dots, v_q$ must belong to the same connected component of (V, F). Moreover, every path in G from s_i to t_i contains some vertex v_j and thus the vertices s_i and t_i must also belong to this connected component. We conclude that (V, F) has only a single connected component and is therefore a spanning tree of cost $|V| - 1 = 3q - 1$.

We now construct a feasible LP solution of cost $2q$. For $i \in \{1, \dots, q\}$, we set

$$z^r_{\{s_i, t_i\}} \coloneqq \begin{cases} 1 & \text{if } r = t_i \\ 0 & \text{otherwise} \end{cases}$$

and for $i \in \{1, \dots, q-1\}$, we set

$$z^r_{\{v_i, v_{i+1}\}} \coloneqq \begin{cases} \frac{1}{q} & \text{if } r = t_j \text{ for } j \in \{1, \dots, q\} \\ 0 & \text{otherwise.} \end{cases}$$

For $i, j \in \{1, \dots, q\}$, we set $x^{t_i}_{(s_i, v_j)} = x^{t_i}_{(v_j, t_i)} = \frac{1}{q}$. All other variables x^r_e with $r \in V$ and $e \in \overrightarrow{E}$ are 0. This defines a feasible solution to Forest-BCR of value $2q$. Therefore, the integrality gap of the instance is at least $\frac{3q-1}{2q}$, which becomes arbitrarily close to $\frac{3}{2}$ for q sufficiently large. This proves Theorem 4.

References

1. Agrawal, A., Klein, P.N., Ravi, R.: When trees collide: An approximation algorithm for the generalized Steiner problem on networks. SIAM J. Comput. **24**(3), 440–456 (1995), https://doi.org/10.1137/S0097539792236237
2. Ahmadi, A., Gholami, I., Hajiaghayi, M., Jabbarzade, P., Mahdavi, M.: 2-approximation for prize-collecting Steiner forest. In: Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms, (SODA). pp. 669–693 (2024). https://doi.org/10.1137/1.9781611977912.25, https://doi.org/10.1137/1.9781611977912.25
3. Ahmadi, A., Gholami, I., Hajiaghayi, M., Jabbarzade, P., Mahdavi, M.: Prize-collecting Steiner tree: A 1.79 approximation. In: Proceedings of the 56th Annual ACM Symposium on Theory of Computing, (STOC). pp. 1641–1652 (2024). https://doi.org/10.1145/3618260.3649789, https://doi.org/10.1145/3618260.3649789
4. Bateni, M., Hajiaghayi, M., Marx, D.: Approximation schemes for steiner forest on planar graphs and graphs of bounded treewidth. J. ACM **58**(5) (Oct 2011). https://doi.org/10.1145/2027216.2027219, https://doi.org/10.1145/2027216.2027219
5. Berman, P., Bhattacharyya, A., Makarychev, K., Raskhodnikova, S., Yaroslavtsev, G.: Approximation algorithms for spanner problems and directed steiner forest. Information and Computation **222**, 93–107 (2013)
6. Borradaile, G., Klein, P.N., Mathieu, C.: A polynomial-time approximation scheme for euclidean steiner forest. ACM Transactions on Algorithms (TALG) **11**(3), 1–20 (2015)
7. Boyd, S., Cheriyan, J., Cummings, R., Grout, L., Ibrahimpur, S., Szigeti, Z., Wang, L.: A 4/3-approximation algorithm for the minimum 2-edge connected multisubgraph problem in the half-integral case. SIAM Journal on Discrete Mathematics **36**(3), 1730–1747 (2022)
8. Byrka, J., Grandoni, F., Ameli, A.J.: Breaching the 2-approximation barrier for connectivity augmentation: A reduction to Steiner tree. SIAM J. Comput. **52**(3), 718–739 (2023). https://doi.org/10.1137/21M1421143, https://doi.org/10.1137/21m1421143
9. Byrka, J., Grandoni, F., Rothvoß, T., Sanità, L.: Steiner tree approximation via iterative randomized rounding. J. ACM **60**(1), 6:1–6:33 (2013). https://doi.org/10.1145/2432622.2432628, https://doi.org/10.1145/2432622.2432628
10. Byrka, J., Grandoni, F., Traub, V.: The bidirected cut relaxation for steiner tree has integrality gap smaller than 2. In: Proceedings of the 65th IEEE Annual Symposium on Foundations of Computer Science (FOCS) (2024)
11. Byrka, J., Grandoni, F., Traub, V.: On the bidirected cut relaxation for steiner forest. CoRR **abs/2412.06518** (2024). https://doi.org/10.48550/ARXIV.2412.06518, https://doi.org/10.48550/arXiv.2412.06518
12. Carr, R., Ravi, R.: A new bound for the 2-edge connected subgraph problem. In: International Conference on Integer Programming and Combinatorial Optimization. pp. 112–125. Springer (1998)
13. Cecchetto, F., Traub, V., Zenklusen, R.: Bridging the gap between tree and connectivity augmentation: unified and stronger approaches. In: Proceedings of the 53rd Annual ACM Symposium on Theory of Computing (STOC). pp. 370–383 (2021). https://doi.org/10.1145/3406325.3451086, https://doi.org/10.1145/3406325.3451086

14. Chakrabarty, D., Devanur, N.R., Vazirani, V.V.: New geometry-inspired relaxations and algorithms for the metric Steiner tree problem. Math. Program. **130**(1), 1–32 (2011). https://doi.org/10.1007/S10107-009-0299-0, https://doi.org/10.1007/s10107-009-0299-0
15. Chakrabarty, D., Könemann, J., Pritchard, D.: Hypergraphic LP relaxations for Steiner trees. In: Proceedings of the 14th International Conference on Integer Programming and Combinatorial Optimization, (IPCO). pp. 383–396 (2010). https://doi.org/10.1007/978-3-642-13036-6_29, https://doi.org/10.1007/978-3-642-13036-6_29
16. Charikar, M., Chekuri, C., Cheung, T., Dai, Z., Goel, A., Guha, S., Li, M.: Approximation algorithms for directed Steiner problems. J. Algorithms **33**(1), 73–91 (1999). https://doi.org/10.1006/jagm.1999.1042, https://doi.org/10.1006/jagm.1999.1042
17. Chekuri, C., Even, G., Gupta, A., Segev, D.: Set connectivity problems in undirected graphs and the directed steiner network problem. ACM Transactions on Algorithms (TALG) **7**(2), 1–17 (2011)
18. Chekuri, C., Jain, R.: A polylogarithmic approximation for directed steiner forest in planar digraphs. arXiv preprint arXiv:2410.17403 (2024)
19. Chlebík, M., Chlebíková, J.: The Steiner tree problem on graphs: Inapproximability results. Theor. Comput. Sci. **406**(3), 207–214 (2008). https://doi.org/10.1016/J.TCS.2008.06.046, https://doi.org/10.1016/j.tcs.2008.06.046
20. Edmonds, J.: Optimum branchings. Journal of Research of the National Bureau of Standards **B71**, 233–240 (1967)
21. Feldmann, A.E., Könemann, J., Olver, N., Sanità, L.: On the equivalence of the bidirected and hypergraphic relaxations for Steiner tree. Math. Program. **160**(1-2), 379–406 (2016). https://doi.org/10.1007/S10107-016-0987-5, https://doi.org/10.1007/s10107-016-0987-5
22. Goemans, M.X., Myung, Y.: A catalog of Steiner tree formulations. Networks **23**(1), 19–28 (1993). https://doi.org/10.1002/NET.3230230104, https://doi.org/10.1002/net.3230230104
23. Goemans, M.X., Olver, N., Rothvoß, T., Zenklusen, R.: Matroids and integrality gaps for hypergraphic Steiner tree relaxations. In: Proceedings of the 44th ACM Symposium on Theory of Computing Conference (STOC). pp. 1161–1176 (2012). https://doi.org/10.1145/2213977.2214081, https://doi.org/10.1145/2213977.2214081
24. Goemans, M.X., Williamson, D.P.: A general approximation technique for constrained forest problems. SIAM J. Comput. **24**(2), 296–317 (1995). https://doi.org/10.1137/S0097539793242618, https://doi.org/10.1137/S0097539793242618
25. Grandoni, F., Ameli, A.J., Traub, V.: Breaching the 2-approximation barrier for the forest augmentation problem. In: Proceedings of the 54th Annual ACM Symposium on Theory of Computing (STOC). pp. 1598–1611 (2022). https://doi.org/10.1145/3519935.3520035, https://doi.org/10.1145/3519935.3520035
26. Grandoni, F., Laekhanukit, B., Li, S.: $O(\log^2 k / \log\log k)$-approximation algorithm for directed Steiner tree: a tight quasi-polynomial-time algorithm. In: Proceedings of the 51st Annual ACM Symposium on Theory of Computing (STOC). pp. 253–264 (2019). https://doi.org/10.1145/3313276.3316349, https://doi.org/10.1145/3313276.3316349
27. Hyatt-Denesik, D., Jabal Ameli, A., Sanità, L.: Finding Almost Tight Witness Trees. In: 50th International Colloquium on Automata, Languages, and Programming (ICALP 2023). pp. 79:1–79:16 (2023)

28. Jain, K.: A factor 2 approximation algorithm for the generalized Steiner network problem. Combinatorica **21**(1), 39–60 (2001). https://doi.org/10.1007/S004930170004, https://doi.org/10.1007/s004930170004
29. Karlin, A.R., Klein, N., Oveis Gharan, S.: An improved approximation algorithm for tsp in the half integral case. In: Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing. p. 28–39. STOC 2020, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3357713.3384273, https://doi.org/10.1145/3357713.3384273
30. Karpinski, M., Lewandowski, M., Meesum, S.M., Mnich, M.: Dense Steiner problems: Approximation algorithms and inapproximability. CoRR **abs/2004.14102** (2020), https://arxiv.org/abs/2004.14102
31. Karpinski, M., Zelikovsky, A.: New approximation algorithms for the Steiner tree problems. J. Comb. Optim. **1**(1), 47–65 (1997). https://doi.org/10.1023/A:1009758919736, https://doi.org/10.1023/A:1009758919736
32. Prömel, H.J., Steger, A.: A new approximation algorithm for the Steiner tree problem with performance ratio 5/3. J. Algorithms **36**(1), 89–101 (2000). https://doi.org/10.1006/JAGM.2000.1086, https://doi.org/10.1006/jagm.2000.1086
33. Robins, G., Zelikovsky, A.: Tighter bounds for graph Steiner tree approximation. SIAM J. Discret. Math. **19**(1), 122–134 (2005). https://doi.org/10.1137/S0895480101393155, https://doi.org/10.1137/S0895480101393155
34. Schmidt, D.R., Zey, B., Margot, F.: Stronger MIP formulations for the steiner forest problem. Math. Program. **186**(1), 373–407 (2021). https://doi.org/10.1007/S10107-019-01460-6, https://doi.org/10.1007/s10107-019-01460-6
35. Traub, V., Zenklusen, R.: A better-than-2 approximation for weighted tree augmentation. In: Proceedings of the 62nd IEEE Annual Symposium on Foundations of Computer Science (FOCS). pp. 1–12 (2021). https://doi.org/10.1109/FOCS52979.2021.00010, https://doi.org/10.1109/FOCS52979.2021.00010
36. Traub, V., Zenklusen, R.: Local search for weighted tree augmentation and Steiner tree. In: Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms (SODA). pp. 3253–3272 (2022). https://doi.org/10.1137/1.9781611977073.128, https://doi.org/10.1137/1.9781611977073.128
37. Traub, V., Zenklusen, R.: A $(1.5+\varepsilon)$-approximation algorithm for weighted connectivity augmentation. In: Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC). pp. 1820–1833 (2023). https://doi.org/10.1145/3564246.3585122, https://doi.org/10.1145/3564246.3585122
38. Vicari, R.: Simplex based Steiner tree instances yield large integrality gaps for the bidirected cut relaxation. CoRR **abs/2002.07912** (2020), https://arxiv.org/abs/2002.07912
39. Williamson, D.P., Shmoys, D.B.: The design of approximation algorithms. Cambridge University Press (2011)
40. Zelikovsky, A.: Better approximation bounds for the network and Euclidean Steiner tree problems. Tech. rep., University of Virginia (1996), cS-96-06
41. Zelikovsky, A.: An 11/6-approximation algorithm for the network Steiner problem. Algorithmica **9**(5), 463–470 (1993). https://doi.org/10.1007/BF01187035, https://doi.org/10.1007/BF01187035