A Partition-Based Relaxation For Steiner Trees

Jochen Könemann* David Pritchard* Kunlun Tan†

April 26, 2009

Abstract

The Steiner tree problem is a classical NP-hard optimization problem with a wide range of practical applications. In an instance of this problem, we are given an undirected graph $G = (V,E)$, a set of *terminals* $R \subseteq V$, and non-negative costs c_e for all edges $e \in E$. Any tree that contains all terminals is called a *Steiner tree*; the goal is to find a minimum-cost Steiner tree. The vertices $V \setminus R$ are called *Steiner vertices*.

The best approximation algorithm known for the Steiner tree problem is a *greedy* algorithm due to Robins and Zelikovsky (SIAM J. Discrete Math, 2005); it achieves a performance guarantee of $1 + \frac{\ln 3}{2} \approx 1.55$. The best known *linear programming* (LP)-based algorithm, on the other hand, is due to Goemans and Bertsimas (Math. Programming, 1993) and achieves an approximation ratio of $2 - 2/|R|$. In this paper we establish a link between greedy and LP-based approaches by showing that Robins and Zelikovsky's algorithm can be viewed as an iterated primal-dual algorithm with respect to a novel LP relaxation. The LP used in the first iteration is stronger than the well-known *bidirected cut relaxation*.

An instance is *b-quasi-bipartite* if each connected component of $G \setminus R$ has at most b vertices. We show that Robins' and Zelikovsky's algorithm has an approximation ratio better than $1 + \frac{\ln 3}{2}$ for such instances, and we prove that the integrality gap of our LP is between $\frac{8}{7}$ and $\frac{2b+1}{b+1}$.

1 Introduction

The Steiner tree problem is a classical problem in combinatorial optimization which owes its practical importance to a host of applications in areas as diverse as VLSI design and computational biology. The problem is NP-hard [24], and Chlebík and Chlebíková show in [7] that it is NP-hard even to *approximate* the minimum-cost Steiner tree within any ratio better than $\frac{96}{95}$. They also show that it is NP-hard to obtain an approximation ratio better than $\frac{128}{127}$ in *quasi-bipartite* instances of the Steiner tree problem. These are instances in which no two Steiner vertices are adjacent in the underlying graph G.

1.1 Greedy algorithms and r-Steiner trees

One of the first approximation algorithms for the Steiner tree problem is the well-known *minimum-spanning tree heuristic* which is widely attributed to Moore [16]. Moore's algorithm has a performance ratio of 2 for the Steiner tree problem and this remained the best known until the 1990s, when Zelikovsky [48] suggested computing Steiner trees with a special structure, so called *r-Steiner trees*. Nearly all of the Steiner tree algorithms developed since then use r-Steiner trees. We now provide a formal definition.

A *full Steiner component* (or *full component* for short) is a tree whose internal vertices are Steiner vertices, and whose leaves are terminals. The edge set of any Steiner tree can be partitioned into full components

*Department of Combinatorics and Optimization, University of Waterloo, 200 University Avenue West, Waterloo, ON N2L 3G1, Canada. Email: {jochen,dagpritc}@math.uwaterloo.ca

†Microsoft Corp., Redmond, WA. Email: kunlun.tan@gmail.com

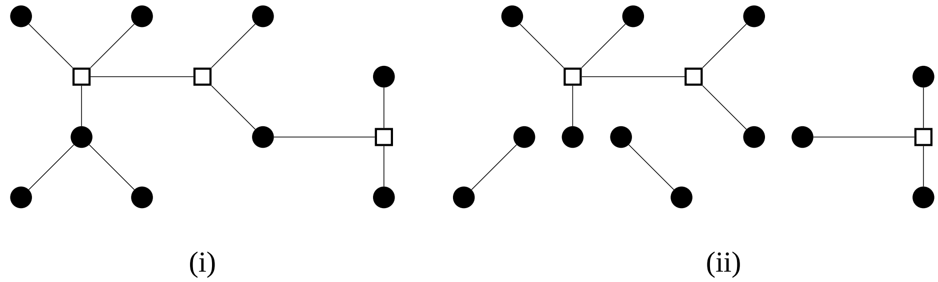


Figure 1: The figure shows a Steiner tree in (i) and its decomposition into full components in (ii). Square and round vertices correspond to Steiner and terminal vertices, respectively. This particular tree is 5-restricted.

by splitting the tree at terminals: see Figure 1 for an example. An *r-(restricted)-Steiner tree* is defined to be a Steiner tree all of whose full components have at most r terminals.

An r-restricted Steiner tree does not always exist; for example, if G is a star with a Steiner vertex at its center and more than r terminals at its tips. To avoid this problem, we *clone* each Steiner vertex v many times and connect these clones to all of v's neighbours in the graph. Copies of an edge have the same cost as the corresponding original edge in G. This cloning does not affect the cost of the optimal Steiner tree but ensures a relatively cheap r-Steiner tree exists, as follows. Let $\mathtt{opt}$ and $\mathtt{opt}_r$ be the cost of an optimal Steiner tree and of an optimal r-Steiner tree, respectively, for the cloned instance. The *r-Steiner ratio ρ_r* is defined to be the supremum of $\mathtt{opt}_r/\mathtt{opt}$ over all instances of the Steiner tree problem. Borchers and Du [5] computed ρ_r for every r; in particular, $\rho_r = 1 + \Theta(1/\log r)$ so ρ_r tends to 1 as r goes to infinity.

The prevailing strategy of all modern Steiner tree algorithms is to compute a cheap r-Steiner tree of the cloned graph, since this corresponds naturally to a Steiner tree of the original graph of equal cost or less. Computing minimum-cost r-Steiner trees is NP-hard for $r \geq 4$ [15], even if the underlying graph is quasi-bipartite. The complexity status for $r = 3$ is unresolved, and the case $r = 2$ reduces to the minimum-cost spanning tree problem.

In [48], Zelikovsky used 3-restricted full components to obtain an $11/6$-approximation for the Steiner tree problem. Subsequently, a series of papers (e.g., [4, 22, 25, 36]) improved upon this result. These efforts culminated in a recent paper by Robins and Zelikovsky [40] in which the authors presented a $\left(1 + \frac{\ln 3}{2}\right) \approx 1.55$-approximation (subsequently referred to as $\mathtt{RZ}$) for the r-Steiner tree problem. They hence obtain, for each fixed $r \geq 2$, a $1.55\rho_r$ approximation algorithm for the (unrestricted) Steiner tree problem. We refer the reader to two surveys in [21, 37].

1.2 Approaches based on linear programs

Many approximation algorithms in combinatorial optimization are based on LP relaxations. The general approach is to jointly design an algorithm and a relaxation so that the algorithm produces a feasible integral solution whose cost is close to the cost of the optimal LP solution. The *primal-dual method* (e.g., [20]) is one paradigm of this sort, whereby the algorithm jointly develops a dual and integral primal solution, the growth of each one guiding the other.

Numerous LP relaxations for the Steiner tree problem have been investigated in depth (e.g., [3, 9, 10, 11, 13, 19, 33, 45, 46]), and this in turn has helped to achieve vast improvements in the area of *integer programming*-based exact algorithms (e.g., see Warme [45] and Polzin [31, 34]). Despite the sizeable body of work on Steiner tree relaxations, the best LP-based algorithms for the Steiner tree problem do not perform as well as $\mathtt{RZ}$ in terms of approximation ratio.

For general graphs, the *classical* LP-based approximation algorithms for Steiner trees [18] and forests [2] use the *undirected cut relaxation* [3] and have a performance guarantee of $2 - \frac{2}{|R|}$. This relaxation has an integrality gap of $2 - \frac{2}{|R|}$ and the analysis of these algorithms is therefore tight. Slightly improved algorithms have since been designed for other LPs [26, 32] but do not achieve any constant approximation factor better than 2. Similarly, no LP relaxation for the Steiner tree problem is known with integrality gap any constant less than 2.

For quasi-bipartite graphs, Chakrabarty, Devanur, and Vazirani [6] considered the *bidirected cut relaxation* [13, 46] and obtained a $\frac{4}{3}$ approximation algorithm and integrality gap bound, improving an earlier ratio of $\frac{3}{2}$ [38, 39]. This yields the best known bound on the integrality gap of any LP relaxation for quasi-bipartite graphs; nonetheless, RZ achieves an approximation ratio better than $\frac{4}{3}$ for these graphs. On general graphs, the bidirected cut relaxation is conjectured (e.g. in [42]) to have a smaller integrality gap than 2; the worst known example shows a gap of only $\frac{8}{7}$ (see Section 5).

1.3 Contribution of this paper

In this paper we provide algorithmic evidence that the primal-dual method is useful for the Steiner tree problem. We first present a novel LP relaxation for the Steiner tree problem. It uses full components to strengthen a formulation based on *Steiner partition* inequalities [9]. We then show that the algorithm RZ of Robins and Zelikovsky can be analyzed as a primal-dual algorithm using this relaxation.

In [40], Robins and Zelikovsky showed that, for a fixed r, the performance ratio of RZ is $1.279\rho_r$ in quasi-bipartite graphs, and it is $1.55\rho_r$ in general graphs. We prove a natural interpolation of these two results. For a Steiner vertex v, define its *Steiner neighbourhood* $\mathtt{sn}(v)$ to be the collection of vertices that are in the same connected component as v in $G \setminus R$. A graph is *b-quasi-bipartite* if all of its Steiner neighbourhoods have cardinality at most b. We prove:

Theorem 1. *Given an undirected, b-quasi-bipartite graph $G = (V, E)$, terminals $R \subseteq V$, and a fixed constant $r \geq 2$, Algorithm* RZ *returns a feasible Steiner tree T s.t.*

$$c(T) \leq \begin{cases} 1.279 \cdot \mathtt{opt}_r & : \quad b = 1 \\ (1 + \frac{1}{e}) \cdot \mathtt{opt}_r & : \quad b \in \{2, 3, 4\} \\ \left(1 + \frac{1}{2}\ln\left(3 - \frac{2}{b}\right)\right) \mathtt{opt}_r & : \quad b \geq 5. \end{cases}$$

Note that b-quasi-bipartite graphs are a natural interpolation between quasi-bipartite graphs ($b = 1$) and general graphs ($b \leq |V \setminus R|$), hence Theorem 1 interpolates the two main results of Robins and Zelikovsky [40].

Unfortunately, Theorem 1 does not imply that our new relaxation has a small integrality gap. Nonetheless, we obtain the following bounds, when G is b-quasi-bipartite:

Theorem 2. *Our new relaxation has an integrality gap between $\frac{8}{7}$ and $\frac{2b+1}{b+1}$.*

We remark that the concept of *filtering*, due to Chakrabarty et al. [6], can be applied to improve the gap upper bound to $\frac{2b-1}{b}$ for $b \geq 2$ [28].

1.4 Overview

In Section 2 we give some LP background on spanning trees and define our new LP relaxation. In Section 3 we show that RZ can be interpreted as an iterated primal-dual algorithm using the new LP. Section 4 contains some analysis of b-quasi-bipartite graphs and the proof of Theorem 1. In Section 5 we prove Theorem 2 and compare the new LP to existing ones. Finally, Section 6 contains deferred technical details including a short proof of the *contraction lemma*, which appears in the analysis of many approximation algorithms for the Steiner tree problem. We also remark that the contraction lemma holds not just in the graphic setting, but more generally for matroids.

2 Spanning trees and a new LP relaxation for Steiner trees

Our work is strongly motivated by linear programming formulations for the spanning tree polyhedron due to Fulkerson [14] and Chopra [8]. In this section, we first discuss Chopra's formulation, and we describe a primal-dual interpretation of Kruskal's spanning tree algorithm [30] based on this LP. Finally we extend ideas in [9, 10] to derive a new LP relaxation for the Steiner tree problem.

2.1 The spanning tree polyhedron

To formulate the minimum-cost spanning tree (MST) problem as an LP, we associate a variable x_e with every edge $e \in E$. Each spanning tree T corresponds to its *incidence vector* x^T, which is defined by $x_e^T = 1$ if T contains e and $x_e^T = 0$ otherwise. Let Π denote the set of all partitions of the vertex set V, and suppose that $\pi \in \Pi$. The *rank* $r(\pi)$ of π is the number of parts of π. Let E_π denote the set of edges whose ends lie in different parts of π. Consider the following LP.

$$
\begin{aligned}
\min \quad & \sum_{e\in E} c_e x_e && \text{(P}_{SP}\text{)}\\
\text{s.t.} \quad & \sum_{e\in E_\pi} x_e \geq r(\pi) - 1 \quad \forall \pi \in \Pi, \\
& x \geq 0.
\end{aligned}
$$

Chopra [8] showed that the feasible region of (P_{SP}) is the dominant of the convex hull of all incidence vectors of spanning trees, and hence each basic optimal solution corresponds to a minimum-cost spanning tree. Its dual LP is

$$
\begin{aligned}
\max \quad & \sum_{\pi\in\Pi} (r(\pi) - 1) \cdot y_\pi && \text{(D}_{SP}\text{)}\\
\text{s.t.} \quad & \sum_{\pi : e\in E_\pi} y_\pi \leq c_e \quad \forall e \in E, && (1)\\
& y \geq 0. && (2)
\end{aligned}
$$

2.2 A primal-dual interpretation of Kruskal's MST algorithm

Kruskal's algorithm, which we will denote by `MST`, can be viewed as a continuous process over *time*: we start with an empty tree at time 0 and add edges as time increases. The algorithm terminates at time τ^* with a spanning tree of the input graph G. In this section we show that Kruskal's method can be interpreted as a primal-dual algorithm (see also [20]). At any time $0 \leq \tau \leq \tau^*$ we keep a pair (x^τ, y^τ), where x^τ is a (not necessarily feasible) 0-1 primal solution for (P_{SP}) and y^τ is a feasible dual solution for (D_{SP}).

The initial primal and dual values x^0 and y^0 are the all-zero vectors. Let $G^\tau = (V, E^\tau)$ denote the forest corresponding to x^τ, i.e., $E^\tau = \{e \in E \mid x_e^\tau = 1\}$. Let $\pi(\tau)$ denote the partition induced by the connected components of G^τ. At time τ, the algorithm increases $y_{\pi(\tau)}$ until a constraint of type (1) becomes tight for some edge $e \in E_{\pi(\tau)}$. (If more than one such constraint becomes tight simultaneously, we pick any such e arbitrarily.) Let $\tau' \geq \tau$ be the time at which this happens. The dual update is

$$y_{\pi(\tau)}^{\tau'} := \tau' - \tau.$$

We then include e in our solution, i.e., the primal update is $x_e^{\tau'} := 1$. We terminate at time τ^* such that G^{τ^*} is a spanning tree. Chopra [8] showed that the final primal and dual solutions have the same objective value (and are hence optimal), and we give a proof of this fact for completeness.

In what follows, let G^* be shorthand for G^{τ^*} and similarly for x^*, etc.

Theorem 3. *Algorithm* `MST` *finishes with a pair* (x^*, y^*) *of primal and dual feasible solutions to* (P_{SP}) *and* (D_{SP}), *respectively, such that*

$$\sum_{e \in E} c_e x_e^* = \sum_{\pi \in \Pi} (r(\pi) - 1) \cdot y_\pi^*.$$

Proof. Checking feasibility is straightforward. For each edge $e \in E^*$, the constraint (1) holds with equality. Hence, rearranging, we can express the cost of the final tree as follows:

$$\sum_{e \in E} c_e x_e^* = \sum_{e \in E^*} \sum_{\pi : e \in E_\pi} y_\pi^* = \sum_{\pi \in \Pi} |E^* \cap E_\pi| \cdot y_\pi^*. \tag{3}$$

Note that for each τ, the final tree G^* has exactly $|V| - r(\pi(\tau))$ edges not in $E_{\pi(\tau)}$; hence for all π with $y_\pi^* > 0$, we have $|E^* \cap E_\pi| = |V| - 1 - (|V| - r(\pi)) = r(\pi) - 1$. This fact, combined with Equation (3), completes the proof. □

Observe that the above primal-dual algorithm is indeed Kruskal's algorithm: if the algorithm adds an edge e at time τ, then e has cost exactly equal to τ, and e is a minimum-cost edge connecting two connected components of G^τ.

2.3 A new LP relaxation for Steiner trees

In an instance of the Steiner tree problem, a partition π of V is defined to be a *Steiner partition* when each part of π contains at least one terminal. Chopra and Rao [9] introduced this notion and proved that, when x is the incidence vector of a Steiner tree and π is a Steiner partition, the inequality

$$\sum_{e \in E_\pi} x_e \geq r(\pi) - 1. \tag{4}$$

holds. These *Steiner partition inequalities* motivate our approach. In order to fully describe and analyze our approach we need a *preprocessing* step; it essentially replaces the graph by the union of its full components, where the union is disjoint for edges and Steiner nodes.

In the following we use $G[U]$ to denote the subgraph of G induced by vertex set U, i.e., the graph with vertices U and edges $E(U) = \{uv \in E(G) \mid u \in U, v \in U\}$. We make the following assumptions:

A1. $G[R]$ is a complete graph and, for any two terminals $u, v \in R$, c_{uv} is the cost of a minimum-cost u,v-path in G.

A2. For every Steiner vertex v and every vertex $u \in \mathtt{sn}(v) \cup R$, uv is an edge of G, and c_{uv} is the cost of a minimum-cost u,v-path in G.

It is a well-known fact that these assumptions are without loss of generality, i.e., any given instance can be transformed into an equivalent instance that satisfies these assumptions (e.g., see [43]). Note that b-quasi-bipartiteness is preserved by these assumptions.

Recall from Section 1.1 that a full component is a tree whose internal vertices are Steiner vertices and all of whose leaves are terminals. Also recall that a full component K is r-restricted if it contains at most r terminals. Further, the edge-set of any r-restricted Steiner tree T can be partitioned into r-restricted full components. From now on, let $r \geq 2$ be an arbitrary fixed constant. Define

$$\mathscr{K}_r := \{K \subseteq R : 2 \leq |K| \leq r \text{ and there exists a full component whose terminal set is } K\}.$$

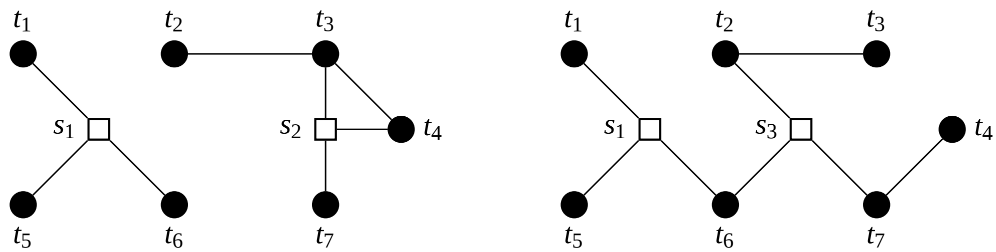


Figure 2: Left: a collection $\mathscr{S} = \{\{t_1,t_5,t_6\},\{t_3,t_4,t_7\},\{t_2,t_3\},\{t_3,t_4\}\}$ of 4 full components. Right: a Steiner tree with $\mathscr{S}$-decomposition $(\{t_1s_1,t_5s_1,t_6s_1,t_2t_3\},\{\{t_2,t_6,t_7\},\{t_4,t_7\}\})$.

We note that, for each $K \in \mathscr{K}_r$, we can determine a minimum-cost full component with terminal set K in polynomial time (e.g., by using the dynamic programming algorithm of Dreyfus and Wagner [12]). Thus, we can compute $\mathscr{K}_r$ in polynomial time as well.

For brevity we will abuse notation slightly and use $K \in \mathscr{K}_r$ interchangeably for a subset of the terminal set and for a particular min-cost full component spanning K. Given any r-restricted Steiner tree, we may assume that all of its full components are from $\mathscr{K}_r$, without increasing its cost.

For each full component K, we use $E(K)$ to denote its edges, $V(K)$ to denote its vertices (including Steiner vertices), and c_K to denote its cost. For a set $\mathscr{S}$ of full components we define $E(\mathscr{S}) := \cup_{K\in\mathscr{S}} E(K)$ and similarly $V(\mathscr{S}) := \cup_{K\in\mathscr{S}} V(K)$. By assumption A1 we may assume that the full component for a terminal pair is just the edge linking those terminals, and by assumption A2 we may assume that any Steiner vertex has degree at least 3. We will also assume that any two distinct full components $K_1, K_2 \in \mathscr{K}_r$ are edge disjoint and internally vertex disjoint. This assumption is without loss of generality as each Steiner vertex in G can be cloned a sufficient number of times to ensure this property. Finally, we redefine G to be $(V(\mathscr{K}_r), E(\mathscr{K}_r))$; as a result, the Steiner trees of the new graph correspond to the r-restricted Steiner trees of the original graph. This completes the preprocessing.

Let $\mathscr{K}_r(T)$ denote the set of all full components of a Steiner tree T. For an arbitrary subfamily $\mathscr{S}$ of the full components $\mathscr{K}_r$, our new LP uses the following canonical decomposition of a Steiner tree into elements of $E(\mathscr{S})$ and $\mathscr{K}_r \backslash \mathscr{S}$.

Definition 4. *If T is an r-restricted Steiner tree, its* $\mathscr{S}$-decomposition *is the pair*

$$(E(T) \cap E(\mathscr{S}), \mathscr{K}_r(T) \backslash \mathscr{S}).$$

Figure 2 illustrates the $\mathscr{S}$-decomposition of a Steiner tree. Observe that after $\mathscr{S}$-decomposing a Steiner tree T we have

$$\sum_{e \in E(T) \cap E(\mathscr{S})} c_e + \sum_{K \in \mathscr{K}_r(T) \backslash \mathscr{S}} c_K = c(T).$$

We hence obtain a new higher-dimensional view of the Steiner tree polyhedron. Define

$$\mathtt{ST}^{\mathscr{S}}_{G,R} := \mathtt{conv}\{x \in \{0,1\}^{E(\mathscr{S})} \times \{0,1\}^{\mathscr{K}_r \backslash \mathscr{S}} \quad : \quad \exists T \in \mathtt{ST}_{G,R} \text{ s.t. } x \text{ is the incidence vector of the } \mathscr{S}\text{-decomposition of } T\}.$$

The following definitions are used to generalize Steiner partition inequalities to use full components. We use $\Pi^{\mathscr{S}}$ to denote the family of all partitions of $V(\mathscr{S}) \cup R$.

Definition 5. *Let $\pi = \{V_1, \dots, V_p\} \in \Pi^{\mathscr{S}}$ be a partition of the set $R \cup V(\mathscr{S})$. The* rank contribution *of full component $K \in \mathscr{K}_r \backslash \mathscr{S}$ is defined as*

$$\mathtt{rc}_K^\pi := |\{i : K \text{ contains a terminal in } V_i\}| - 1.$$

The Steiner rank $\overline{r}(\pi)$ *of π is defined as*

$$\overline{r}(\pi) := \{\text{the number of parts of } \pi \text{ that contain terminals}\}.$$

For example, where $\mathscr{S}$ denotes the collection of full components on the left side of Figure 2, consider the partition $\pi = \{\{t_1, t_5, s_1\}, \{s_2\}, \{t_6, t_7\}, \{t_2, t_3\}, \{t_4\}\} \in \Pi^{\mathscr{S}}$. Its rank is $r(\pi) = 5$ but its Steiner rank is $\overline{r}(\pi) = 4$. The rank contribution of full component $K = \{t_2, t_6, t_7\}$ is $\mathtt{rc}_K^\pi = 1$.

We describe below a new LP relaxation ($\mathrm{P}_{ST}^{\mathscr{S}}$) of $\mathtt{ST}_{G,R}^{\mathscr{S}}$. The relaxation has a variable x_e for each $e \in E(\mathscr{S})$ and a variable x_K for each $K \in \mathscr{K}_r \backslash \mathscr{S}$. For a partition $\pi \in \Pi^{\mathscr{S}}$, we define $E_\pi(\mathscr{S})$ to be the edges of $\mathscr{S}$ whose endpoints lie in different parts of π, i.e., $E_\pi(\mathscr{S}) = E(\mathscr{S}) \cap E_\pi$.

$$\min \quad \sum_{e \in E(\mathscr{S})} c_e \cdot x_e + \sum_{K \in \mathscr{K}_r \backslash \mathscr{S}} c_K \cdot x_K \tag{$\mathrm{P}_{ST}^{\mathscr{S}}$}$$

$$\text{s.t} \quad \sum_{e \in E_\pi(\mathscr{S})} x_e + \sum_{K \in \mathscr{K}_r \backslash \mathscr{S}} \mathtt{rc}_K^\pi \cdot x_K \geq \overline{r}(\pi) - 1 \qquad \forall \pi \in \Pi^{\mathscr{S}} \tag{5}$$

$$x_e, x_K \geq 0 \qquad \forall e \in E(\mathscr{S}), K \in \mathscr{K}_r \backslash \mathscr{S} \tag{6}$$

Its LP dual has a variable y_π for each partition $\pi \in \Pi^{\mathscr{S}}$:

$$\max \quad \sum_{\pi \in \Pi^{\mathscr{S}}} (\overline{r}(\pi) - 1) \cdot y_\pi \tag{$\mathrm{D}_{ST}^{\mathscr{S}}$}$$

$$\text{s.t} \quad \sum_{\pi \in \Pi^{\mathscr{S}} : e \in E_\pi(\mathscr{S})} y_\pi \leq c_e \qquad \forall e \in E(\mathscr{S}) \tag{7}$$

$$\sum_{\pi \in \Pi^{\mathscr{S}}} \mathtt{rc}_K^\pi \cdot y_\pi \leq c_K \qquad \forall K \in \mathscr{K}_r \backslash \mathscr{S} \tag{8}$$

$$y_\pi \geq 0 \qquad \forall \pi \in \Pi^{\mathscr{S}} \tag{9}$$

We conclude this section with a proof that the (primal) LP is indeed a relaxation of the convex hull of $\mathscr{S}$-decompositions for r-restricted Steiner trees. The inequalities (6) are obviously valid for $\mathtt{ST}_{G,R}^{\mathscr{S}}$.

Lemma 6. *The inequalities* (5) *are valid for* $\mathtt{ST}_{G,R}^{\mathscr{S}}$.

Proof. Let T be a Steiner tree with $\mathscr{S}$-decomposition $(E(T) \cap E(\mathscr{S}), \mathscr{K}_r(T) \backslash \mathscr{S})$, and let $x \in \mathtt{ST}_{G,R}^{\mathscr{S}}$ be the corresponding incidence vector. Fix an arbitrary partition $\pi \in \Pi^{\mathscr{S}}$; we will now argue that the left-hand side of (5) for π is at least $\overline{r}(\pi) - 1$.

In order to do that we successively modify the given partition π by merging some of its parts. Initially, let $\hat{\pi} = \pi$. For each each edge uv of $E(T) \cap E(\mathscr{S})$, merge the part of $\hat{\pi}$ containing u and that containing v; if both endpoints lie in the same part of $\hat{\pi}$, the partition remains unchanged. Subsequently, consider each $K \in \mathscr{K}_r(T) \backslash \mathscr{S}$, and merge all parts of $\hat{\pi}$ meeting any terminal of K.

Initially, $\hat{\pi}$ has Steiner rank $\overline{r}(\pi)$, and its final Steiner rank is 1 since T connects all terminals. The Steiner rank drop of $\hat{\pi}$ due to any edge $e \in E_\pi(\mathscr{S})$ with $x_e = 1$ is clearly at most 1. For any other edge $e \in E(T) \cap E(\mathscr{S})$, since the endpoints of e are in the same part of π, the Steiner rank drop of $\hat{\pi}$ due to e is 0. Similarly, the Steiner rank drop of $\hat{\pi}$ due to $K \in \mathscr{K}_r(T) \backslash \mathscr{S}$ is at most $\mathtt{rc}_K^\pi$. This shows that x satisfies constraint (5). As T and π were chosen arbitrarily, the lemma follows. $\square$

3 An iterated primal-dual algorithm for Steiner trees

As described in Section 2.2, $\texttt{MST}(G,c)$ denotes a call to Kruskal's minimum-spanning tree algorithm on graph G with cost-function c. It returns a minimum-cost spanning tree T and an optimal feasible dual solution y for (D_{SP}). Let $\texttt{mst}(G,c)$ denote the cost of $\texttt{MST}(G,c)$. Since c is fixed, in the rest of the paper we omit c where possible for brevity. Let us also abuse notation and identify each set $\mathscr{S} \subset \mathscr{K}_r$ of full components with the graph $(V(\mathscr{S}), E(\mathscr{S}))$. In particular when $\mathscr{S} = (V(\mathscr{S}), E(\mathscr{S}))$ is connected and spans all terminals, $\texttt{MST}(\mathscr{S})$ is a Steiner tree; namely, the one produced by running the MST heuristic on the instance wherein the full component set is $\mathscr{S}$ and all other full components from the original instance are not present.

The main idea of the greedy algorithms in [40, 47, 48] is to find a set $\mathscr{S} \subset \mathscr{K}_r$ of full components such that $\texttt{MST}(\mathscr{S})$ is a Steiner tree with small cost relative to $\texttt{opt}_r$. Let $\binom{R}{2}$ denote the collection of all pairs of terminals. The algorithms all start with $\mathscr{S} = \binom{R}{2}$ and then grow $\mathscr{S}$, so for the rest of the paper we assume that $\binom{R}{2} \subseteq \mathscr{S}$; hence $E(G[R]) \subseteq E(\mathscr{S})$ and $R \subseteq V(\mathscr{S})$.

The reason that MST is useful in our primal-dual framework is that we can relate the dual (D_{SP}) on graph $\mathscr{S}$ to the dual ($\mathrm{D}_{ST}^{\mathscr{S}}$). Let y be the dual returned by a call to $\texttt{MST}(\mathscr{S})$. We treat y as a dual solution of ($\mathrm{D}_{ST}^{\mathscr{S}}$); note that constraints (1) and (2) of (D_{SP}) imply that y also meets constraints (7) and (9) of ($\mathrm{D}_{ST}^{\mathscr{S}}$). If K is a full component such that (8) does not hold for y, we say that K is *violated* by y.

The primal-dual algorithm finds such a set $\mathscr{S}$ in an iterative fashion. Initially, $\mathscr{S}$ is equal to $\binom{R}{2}$. In each iteration, we compute a minimum-cost spanning tree T of the graph $\mathscr{S}$. The dual solution y corresponding to this tree is converted to a dual for ($\mathrm{D}_{ST}^{\mathscr{S}}$), and if y is feasible for ($\mathrm{D}_{ST}^{\mathscr{S}}$), we stop. Otherwise, we add a violated full component to $\mathscr{S}$ and continue. The algorithm clearly terminates (as $\mathscr{K}_r$ is finite) and at termination, it returns the final tree T as an approximately-optimum Steiner tree.

Algorithm 1 summarizes the above description. The greedy algorithms in [40, 47, 48] differ only in how K is selected in each iteration, i.e., in how the selection function $f_i : \mathscr{K}_r \to \mathbb{R}$ is defined (see also [21, §1.4] for a well-written comparison of these algorithms).

Algorithm 1 A general iterative primal-dual framework for Steiner trees.

```
Given: Undirected graph G = (V,E), non-negative costs c_e for all edges e ∈ E, constant r ≥ 2.
S^0 := (R choose 2), i := 0
repeat
    (T^i, y^i) := MST(S^i)
    if y^i is not feasible for (D_ST^{S^i}) then
        Choose a violated full component K^i ∈ K_r \ S^i such that f_i(K^i) is minimized
        S^{i+1} := S^i ∪ {K^i}
    end if
    i := i + 1
until y^{i-1} is feasible for (D_ST^{S^{i-1}})
Let p = i - 1 and return (T^p, y^p).
```

In the typical primal-dual approach [20, 43] dual feasibility is maintained and primal feasibility happens only at the end. This is true in MST relative to (D_{SP}), however if you consider the entirety of Algorithm 1 relative to our new LPs, we obtain a primal feasible solution in each iteration but attain dual feasibility only in the final iteration; more specifically the objective value of y^i decreases as i increases (see Lemma 21). We remark that the recent $\frac{4}{3}$-approximation algorithm of Chakrabarty et al. [6] for quasi-bipartite instances uses the same generic approach, with the addition of an initial *filtering* step, and using any possible selection function.

The following lemma is at the heart of our proof, and explains why our LP can be used to find cheap Steiner trees. We use $\mathscr{S}/K$ to denote the graph obtained from $\mathscr{S}$ by identifying the terminals in K, and by deleting loops created in this process.

Lemma 7. *Let* $(T,y) = \mathtt{MST}(\mathscr{S})$*. Then* K *is violated by* y *if and only if*

$$c_K < c(T) - \mathtt{mst}(\mathscr{S}/K).$$

Proof. Let us adopt the notation from the proof of Theorem 3, and assume that $\mathtt{MST}(\mathscr{S})$ finishes at time τ^*. Consider how the rank contribution of K changes with respect to $\pi(\tau)$ over time. Clearly, $\mathtt{rc}_K^{\pi(0)} = |K|-1$ and $\mathtt{rc}_K^{\pi(\tau^*)} = 0$. Whenever an edge is added to E^τ in $\mathtt{MST}$, the value $\mathtt{rc}_K^{\pi(\tau)}$ either stays the same or drops by 1; hence there are edges $e_1,\ldots,e_{|K|-1} \in T$ such that, for $1 \le i \le |K|-1$, $\mathtt{rc}_K^{\pi(\tau)}$ drops from $|K|-i$ to $|K|-i-1$ when edge e_i is added. Let $\tau(i)$ denote the time at which edge e_i is added, then by the definition of the e_i,

$$\int_0^{\tau^*} \mathtt{rc}_K^{\pi(\tau)} d\tau = \sum_{i=1}^{|K|-1} \tau(i). \tag{10}$$

Notice that due to the definition of $\mathtt{MST}$, the following two facts hold: first, $\tau(i) = c_{e_i}$ for each i; second, the left hand side of Equation (10) is $\sum_\pi \mathtt{rc}_K^\pi y_\pi$. Hence we obtain

$$\sum_\pi \mathtt{rc}_K^\pi y_\pi = \sum_{i=1}^{|K|-1} c_{e_i} \tag{11}$$

Let the partition maintained by $\mathtt{MST}$ on input G at time τ be denoted by $\pi_G(\tau)$. An easy inductive argument shows that for all τ, we obtain $\pi_{\mathscr{S}/K}(\tau)$ from $\pi_{\mathscr{S}}(\tau)$ by first merging all parts that meet K, and by subsequently identifying the vertices of K. It follows that $T \backslash \{e_1,\ldots,e_{|K|-1}\}$ is a minimum spanning tree of $\mathscr{S}/K$. With Equation (11) this yields

$$\sum_\pi \mathtt{rc}_K^\pi y_\pi = c(T) - \mathtt{mst}(\mathscr{S}/K).$$

By the definition of violating full component, the proof is complete. □

Corollary 8. *Let* $(T,y) = \mathtt{MST}(\mathscr{S})$*. If* K *is violated by* y*, then adding* K *to* $\mathscr{S}$ *produces a cheaper spanning tree, i.e.,*

$$\mathtt{mst}(\mathscr{S} \cup \{K\}) < c(T).$$

Proof. $\mathtt{MST}(\mathscr{S}/K) \cup K$ is a spanning tree of $\mathscr{S} \cup \{K\}$, and by Lemma 7 its cost is less than $c(T)$. □

3.1 Cutting losses: the RZ selection function

A potential weak point in Algorithm 1 is that once a full component is added to $\mathscr{S}$, it is never removed. On the other hand, if some cheap subgraph H connects all Steiner vertices of $\mathscr{S}$ to terminals, then adding H to any Steiner tree gives us a tree that spans $V(\mathscr{S})$, i.e., we have so far *lost* at most $c(H)$ in the final answer. This leads to the concept of the *loss* of a Steiner tree which was first introduced by Karpinski and Zelikovsky in [25].

Definition 9. *Let* $G' = (V',E')$ *be a subgraph of* G*. The* loss $\mathtt{L}(G')$ *is a minimum-cost set* $E'' \subseteq E'$ *such that every connected component of* (V',E'') *contains a terminal. Let* $\mathtt{l}(G')$ *denote the cost of* $\mathtt{L}(G')$.

See Figure 3 for an example of the loss of a graph. The above discussion amounts to saying that $\min\{\mathtt{mst}(\mathscr{S}') \mid \mathscr{S}' \supseteq \mathscr{S}\} \le \mathtt{opt}_r + \mathtt{l}(\mathscr{S})$. Consequently, our selection function f_i in step 6 of the algorithm should try to keep the loss small. The following fact holds because full components in $\mathscr{K}_r$ meet only at terminals.

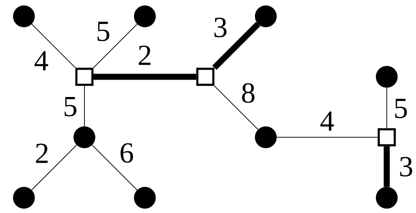


Figure 3: The figure shows the Steiner tree instance from Figure 1 with costs on the edges. The loss of the Steiner tree in this figure is shown in thick edges. Its cost is 8.

Fact 10. *If $\mathscr{S} \subseteq \mathscr{K}_r$, then $\mathtt{L}(\mathscr{S}) = \cup_{K \in \mathscr{S}} \mathtt{L}(K)$ and so $\mathtt{l}(\mathscr{S}) = \sum_{K \in \mathscr{S}} \mathtt{l}(K)$.*

For a set $\mathscr{S}$ of full components, where y is the dual solution returned by $\mathtt{MST}(\mathscr{S})$, define

$$\overline{\mathtt{mst}}(\mathscr{S}) := \sum_{\pi \in \Pi^{\mathscr{S}}} (\overline{r}(\pi) - 1) y_\pi. \tag{12}$$

If y is feasible for $(\mathrm{D}_{ST}^{\mathscr{S}})$ then by weak LP duality, $\overline{\mathtt{mst}}(\mathscr{S})$ provides a lower bound on $\mathtt{opt}_r$. If y is infeasible for $(\mathrm{D}_{ST}^{\mathscr{S}})$, then which full component should we add? Robins and Zelikovsky propose minimizing the ratio of the added loss to the change in potential lower bound (12). Their selection function f_i is defined by

$$f_i(K) := \frac{\mathtt{l}(K)}{\overline{\mathtt{mst}}(\mathscr{S}^i) - \overline{\mathtt{mst}}(\mathscr{S}^i \cup \{K\})} = \frac{\mathtt{l}(\mathscr{S}^i \cup \{K\}) - \mathtt{l}(\mathscr{S}^i)}{\overline{\mathtt{mst}}(\mathscr{S}^i) - \overline{\mathtt{mst}}(\mathscr{S}^i \cup \{K\})}, \tag{13}$$

where the equality uses Fact 10.

4 Analysis

Fix an optimum r-Steiner tree T^*. There are several steps in proving the performance guarantee of Robins and Zelikovsky's algorithm, and they are encapsulated in the following result, whose complete proof appears in Section 6.

Lemma 11. *The cost of the tree T^p returned by Algorithm 1 is at most*

$$\mathtt{opt}_r + \mathtt{l}(T^*) \cdot \ln\left(1 + \frac{\overline{\mathtt{mst}}(G[R], c) - \mathtt{opt}_r}{\mathtt{l}(T^*)}\right).$$

The main observation in the proof of the above lemma can be summarized as follows: from the discussion in Section 2, we know that the tree T^p returned by Algorithm 1 has cost

$$\mathtt{mst}(\mathscr{S}^p) = \sum_{\pi \in \Pi^{\mathscr{S}^p}} (r(\pi) - 1) y_\pi^p$$

and the corresponding lower-bound on $\mathtt{opt}_r$ returned by the algorithm is

$$\overline{\mathtt{mst}}(\mathscr{S}^p) = \sum_{\pi \in \Pi^{\mathscr{S}^p}} (\overline{r}(\pi) - 1) y_\pi^p.$$

We know that $\overline{\mathtt{mst}}(\mathscr{S}^p) \leq \mathtt{opt}_r$ but how large is the difference between $\mathtt{mst}(\mathscr{S}^p)$ and $\overline{\mathtt{mst}}(\mathscr{S}^p)$? We show that the difference

$$\sum_{\pi \in \Pi^{\mathscr{S}^p}} (r(\pi) - \overline{r}(\pi)) y_\pi^p$$

is exactly equal to the loss $\mathtt{l}(T^p)$ of tree T^p — this is proved in Lemma 18. We then bound the loss of each selected full component K^i, and putting everything together finally yields Lemma 11.

The following lemma states the performance guarantee of Moore's minimum-spanning tree heuristic as a function of the optimum loss and the maximum cardinality b of any Steiner neighbourhood in G.

Lemma 12. *Fix an arbitrary optimum r-restricted Steiner tree T^*. Given an undirected, b-quasi-bipartite graph $G = (V,E)$, a set of terminals $R \subseteq V$, and non-negative costs c_e for all $e \in E$, we have*

$$\mathtt{mst}(G[R],c) \leq 2\mathtt{opt}_r - \frac{2}{b}\mathtt{l}(T^*)$$

for any $b \geq 1$.

Proof. Recall that $\mathscr{K}_r(T^*)$ is the set of full components of tree T^*. Now consider a full component $K \in \mathscr{K}_r(T^*)$. We will now show that there is a minimum-cost spanning tree of $G[K]$ whose cost is at most $2c_K - \frac{2}{b}\mathtt{l}(K)$. By repeating this argument for all full components $K \in \mathscr{K}_r(T^*)$, adding the resulting bounds, and applying Fact 10, we obtain the lemma.

For terminals $r,s \in K$, let P_{rs} denote the unique r,s-path in K. Pick $u,v \in K$ such that $c(P_{uv})$ is maximal. Define the *diameter* $\Delta(K) := c(P_{uv})$. Do a depth-first search traversal of K starting in u and ending in v. The resulting walk in K traverses each edge not on P_{uv} twice while each edge on P_{uv} is traversed once. Hence the walk has cost $2c_K - \Delta(K)$. Using standard short-cutting arguments it follows that the minimum-cost spanning tree of $G[K]$ has cost at most

$$2c_K - \Delta(K) \tag{14}$$

as well.

Each Steiner vertex $s \in V(K) \backslash R$ can connect to some terminal $v \in K$ at cost at most $\frac{\Delta(K)}{2}$. Hence, the cost $\mathtt{l}(K)$ of the loss of K is at most $b\frac{\Delta(K)}{2}$. In other words we have $\Delta(K) \geq \frac{2}{b}\mathtt{l}(K)$. Plugging this into (14) yields the lemma. □

For small values of b we can obtain additional improvements via case analysis.

Lemma 13. *Suppose $b \in \{3,4\}$. Fix an arbitrary optimum r-restricted Steiner tree T^*. Given an undirected, b-quasi-bipartite graph $G = (V,E)$, a set of terminals $R \subseteq V$, and non-negative costs c_e for all $e \in E$, we have*

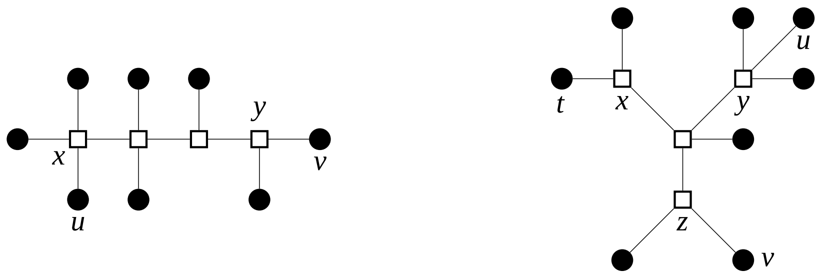
$$\mathtt{mst}(G[R],c) \leq 2\mathtt{opt}_r - \mathtt{l}(T^*).$$

Proof. As in the proof of Lemma 12 it suffices to prove that, for each full component $K \in \mathscr{K}_r(T^*)$, there is a minimum-cost spanning tree of $G[K]$ whose cost is at most $2c_K - \mathtt{l}(K)$, for then we can add the bound over all such K to get the desired result. For terminals $r,s \in K$, let P_{rs} again denote the unique r,s-path in K.

Notice that the Steiner vertices (there are at most b of them) in the full component K either form a path, or else there are 4 of them and they form a star.

Case 1: the Steiner vertices in K form a path. Let x and y be the Steiner vertices on the ends of this path. Let u (resp. v) be any terminal neighbour of x (resp. y); see Figure 4(i) for an example. Perform a depth-first search in K starting from u and ending at v; the cost of this search is $2c_K - c(P_{uv})$. By standard short-cutting arguments it follows that $2c_K - c(P_{uv})$ is an upper bound on $\mathtt{mst}(G[K])$. On the other hand, since $P_{uv} \backslash \{ux\}$ is a candidate for the loss of K, we know that $\mathtt{l}(K) \leq c(P_{uv} \backslash \{ux\}) \leq c(P_{uv})$. Therefore we obtain

$$\mathtt{mst}(G[K]) \leq 2c_K - c(P_{uv}) \leq 2c_K - \mathtt{l}(K). \tag{15}$$



(i) (ii)

Figure 4: The figure shows the two types of full components when $b \leq 4$. On the left is a full component where the Steiner vertices form a path, and on the right is a full component where the Steiner vertices form a star with 3 tips.

Case 2: the Steiner vertices in K form a star. Let the tips of the star be x,y,z and let t,u,v be any terminal neighbours of x,y,z respectively; see Figure 4(ii) for an example. Without loss of generality, we may assume that $c_{xt} \leq c_{yu} \leq c_{zv}$. As before, a depth-first search in K starting from u and ending at v has cost $2c_K - c(P_{uv})$ and this is an upper bound on $\mathtt{mst}(G[K])$. On the other hand, $P_{uv} \backslash \{yu\} \cup \{xt\}$ is a candidate for the loss of K and so $\mathtt{l}(K) \leq c(P_{uv}) - c_{yu} + c_{xt} \leq c(P_{uv})$. We hence obtain Equation (15) as in the previous case. □

We are ready to prove our main theorem. We restate it using the notation introduced in the last two sections.

Theorem 1. *Given an undirected, b-quasi-bipartite graph $G = (V,E)$, terminals $R \subseteq V$, and a fixed constant $r \geq 2$, Algorithm 1 returns a feasible Steiner tree T^p with*

$$c(T^p) \leq \begin{cases} 1.279 \cdot \mathtt{opt}_r & : \quad b = 1 \\ (1+1/e) \cdot \mathtt{opt}_r & : \quad b \in \{2,3,4\} \\ \left(1 + \frac{1}{2}\ln\left(3 - \frac{2}{b}\right)\right) \mathtt{opt}_r & : \quad b \geq 5. \end{cases}$$

Proof. Using Lemma 11 we see that

$$\begin{aligned} c(T^p) &\leq \mathtt{opt}_r + \mathtt{l}(T^*) \cdot \ln\left(1 + \frac{\overline{\mathtt{mst}}(G[R],c) - \mathtt{opt}_r}{\mathtt{l}(T^*)}\right) \\ &= \mathtt{opt}_r + \mathtt{l}(T^*) \cdot \ln\left(1 + \frac{\mathtt{mst}(G[R],c) - \mathtt{opt}_r}{\mathtt{l}(T^*)}\right). \end{aligned} \tag{16}$$

The second equality above holds because $G[R]$ has no Steiner vertices. Applying the bound on $\mathtt{mst}(G[R],c)$ from Lemma 12 yields

$$c(T^p) \leq \mathtt{opt}_r \cdot \left[1 + \frac{\mathtt{l}(T^*)}{\mathtt{opt}_r} \cdot \ln\left(1 - \frac{2}{b} + \frac{\mathtt{opt}_r}{\mathtt{l}(T^*)}\right)\right]. \tag{17}$$

Karpinski and Zelikovsky [25] show that $\mathtt{l}(T^*) \leq \frac{1}{2}\mathtt{opt}_r$. We can therefore obtain an upper-bound on the right-hand side of (17) by bounding the maximum value of function $x\ln(1 - 2/b + 1/x)$ for $x \in [0,1/2]$. We branch into cases:

$b = 1$**:** The maximum of $x\ln(1/x - 1)$ for $x \in [0,1/2]$ is attained for $x \approx 0.2178$. Hence, $x\ln(1/x - 1) \leq 0.279$ for $x \in [0,1/2]$.

$b = 2$**:** The maximum of $x\ln(1/x)$ is attained for $x = 1/e$ and hence $x\ln(1/x) \le 1/e$ for $x \in [0, 1/2]$.

$b \in \{3, 4\}$**:** We use Equation (16) together with Lemma 13 in place of Lemma 12; the subsequent analysis and result are the same as in the previous case.

$b \ge 5$**:** The function $x\ln(1-2/b+1/x)$ is increasing in x and its maximum is attained for $x = 1/2$. Thus, $x\ln(1-2/b+1/x) \le \frac{1}{2}\ln(3-2/b)$ for $x \in [0, 1/2]$.

The four cases above conclude the proof of the theorem. ☐

We remark that under the original analysis of Robins and Zelikovsky, for RZ to achieve an approximation ratio better than the MST heuristic requires $(1+\frac{1}{2}\ln(3))\rho_r < 2$ which occurs for $r \ge 12$. Note the graph resulting from preprocessing under a given choice of r is $(r-2)$-quasi-bipartite; hence, Theorem 1 shows that for $r = 5$, RZ achieves ratio $\rho_5 \cdot (1+\frac{1}{e}) = \frac{13}{9} \cdot (1+\frac{1}{e}) < 2$ and does better than the MST heuristic.

5 Properties of $(\mathrm{P}_{ST}^{\mathscr{S}})$

In this section, we first prove that the linear program $(\mathrm{P}_{ST}^{\mathscr{S}})$ is gradually weakened as the algorithm progresses (i.e., as more full components are added to $\mathscr{S}$). Then we describe bounds on the integrality gap of the new LP, and its strength compared to other LPs for the Steiner tree problem.

Lemma 14. *If $\mathscr{S} \subset \mathscr{S}'$, then the integrality gap of $(\mathrm{P}_{ST}^{\mathscr{S}})$ is at most the integrality gap of $(\mathrm{P}_{ST}^{\mathscr{S}'})$.*

Proof. We consider only the case where $\mathscr{S}' = \mathscr{S} \cup \{J\}$ for some full component J; the general case then follows by induction on $|\mathscr{S}' \backslash \mathscr{S}|$.

Let x be any feasible primal point for $(\mathrm{P}_{ST}^{\mathscr{S}})$ and define the *extension* x' of x to be a primal point of $(\mathrm{P}_{ST}^{\mathscr{S}'})$, with $x'_e = x_J$ for all $e \in E(J)$ and $x'_Z = x_Z$ for all $Z \in (\mathscr{K}_r \backslash \mathscr{S}') \cup E(\mathscr{S})$. We claim that x' is feasible for $(\mathrm{P}_{ST}^{\mathscr{S}'})$. Since x and x' have the same objective value, this will prove Lemma 14.

It is clear that x' satisfies constraints (6), so now let us show that x' satisfies the partition inequality (5) in $(\mathrm{P}_{ST}^{\mathscr{S}'})$. Fix an arbitrary partition π' of $V(\mathscr{S}')$, and let π be the restriction of π' to $V(\mathscr{S})$. We get

$$\sum_{e \in E_{\pi'}(\mathscr{S}')} x'_e + \sum_{K \in \mathscr{K}_r \backslash \mathscr{S}'} \mathtt{rc}_K^{\pi'} x'_K = \left(\sum_{e \in E_{\pi}(\mathscr{S})} x_e + \sum_{K \in \mathscr{K}_r \backslash \mathscr{S}} \mathtt{rc}_K^{\pi} x_K \right) + |E_{\pi'} \cap E(J)| x_J - \mathtt{rc}_J^{\pi} x_J. \tag{18}$$

Now J spans at least $\mathtt{rc}_J^{\pi} + 1$ parts of π', and it follows that $|E_{\pi'} \cap E(J)| \ge \mathtt{rc}_J^{\pi}$. Hence, using Equation (18), the fact that x satisfies constraint (5) for π, and the fact that $\overline{r}(\pi) = \overline{r}(\pi')$, we have

$$\sum_{e \in E_{\pi'}(\mathscr{S}')} x'_e + \sum_{K \in \mathscr{K}_r \backslash \mathscr{S}'} \mathtt{rc}_K^{\pi'} x'_K \ge \sum_{e \in E_{\pi}(\mathscr{S})} x_e + \sum_{K \in \mathscr{K}_r \backslash \mathscr{S}} \mathtt{rc}_K^{\pi} x_K \ge \overline{r}(\pi) - 1 = \overline{r}(\pi') - 1.$$

So x' satisfies (5) for π'. ☐

In 1997, Warme [44] introduced a new linear program for the Steiner tree problem. He observed (as did the authors of [36] in the same year) that full components allow a reduction from the Steiner tree problem to the *spanning-tree-in-hypergraph* problem. Warme also gave an LP relaxation for spanning trees in hypergraphs. That LP turns out to be exactly as strong as our own LP; see [27, Corollary 3.19] for a proof. Now, Polzin et al. [35] proved that Warme's relaxation is stronger than the bidirected cut relaxation, and Goemans [17] proved that the (graph) Steiner partition inequalities are valid for the bidirected cut formulation. Hence, using full components as in $(\mathrm{P}_{ST}^{\mathscr{S}})$ strengthens the Steiner partition inequalities.

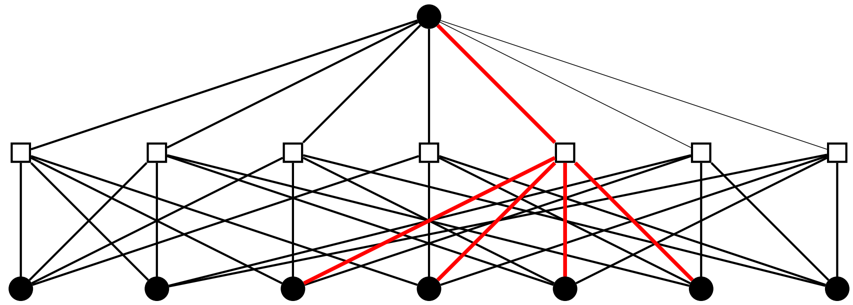


Figure 5: Skutella's example, which shows that the bidirected cut formulation and our new formulation both have a gap of at least $\frac{8}{7}$. The shaded edges denote one of the quasi-bipartite full components on 5 terminals.

5.1 A lower bound on the integrality gap of $(\mathrm{P}^{\varnothing}_{ST})$

Note that when $\mathscr{S} = \binom{R}{2}$, $(\mathrm{P}^{\varnothing}_{ST})$ and $(\mathrm{P}^{\mathscr{S}}_{ST})$ are equivalent LPs: for each terminal-terminal edge uv, the full component variable $x_{\{u,v\}}$ of the former corresponds to the edge variable x_{uv} of the latter. Hence although we consider the simpler LP $(\mathrm{P}^{\varnothing}_{ST})$ in this section, the results apply also to the LP used in the first iteration of $\mathtt{RZ}$.

As reported by Agarwal & Charikar [1], Goemans gave a family of graphs upon which, in the limit, the integrality gap of the bidirected cut relaxation is $\frac{8}{7}$. Interestingly, it can be shown that once you preprocess these graphs as described in Section 2.3, the gap completely disappears. Here we describe another example, due to Skutella [41]. It shows not only that the gap of the bidirected cut relaxation is at least $\frac{8}{7}$, but that the gap of our new formulation (including preprocessing) is at least $\frac{8}{7}$. The example is quasi-bipartite.

The Fano design is a well-known finite geometry consisting of 7 *points* and 7 *lines*, such that every point is on 3 lines, every line contains 3 points, any two lines meet in a unique point, and any two points lie on a unique common line. We construct Skutella's example by creating a bipartite graph, with one side consisting of one vertex n_p for each point p of the Fano design, and the other side consisting of one vertex n_ℓ for each line ℓ of the Fano design. Define n_p and n_ℓ to be adjacent in our graph if and only if p does *not* lie on ℓ. Then it is easy to see this graph is 4-regular, and that given any two vertices n_1, n_2 from one side, there is a vertex from the other side that is adjacent to neither n_1 nor n_2. Let one side be terminals, the other side be Steiner vertices, and then attach one additional terminal to all the Steiner vertices. We illustrate the resulting graph in Figure 5.

Each Steiner vertex is in a unique 5-terminal quasi-bipartite full component. There are 7 such full components. Denote the family of these 7 full components by $\mathscr{C}$.

Claim 15. *Let $x^*_K = \frac{1}{4}$ for each $K \in \mathscr{C}$, and $x^*_K = 0$ otherwise. Then x^* is feasible for $(\mathrm{P}^{\varnothing}_{ST})$.*

Proof. It is immediate that x^* satisfies constraints (6). It remains only to show that x^* meets constraint (5). Let $\pi = (V_0, V_1, \dots, V_m)$ be an arbitrary partition such that V_0 contains the extra *top* terminal. If we can show that $\sum_K x^*_K \mathtt{rc}^\pi_K \geq m$ then we will be done, since π was arbitrary. For each $i = 1, \dots, m$, let t_i be any terminal in V_i. Note that each t_i lies in exactly 4 full components from $\mathscr{C}$. Furthermore, every full component $K \in \mathscr{C}$ satisfies $\mathtt{rc}^\pi_K \geq |K \cap \{t_1, \dots, t_m\}|$, as K meets V_0 as well as each part V_j for which $t_j \in K$. Hence

$$\sum_{K\in\mathscr{C}} x^*_K \mathtt{rc}^\pi_K = \frac{1}{4}\sum_{K\in\mathscr{C}} \mathtt{rc}^\pi_K \geq \frac{1}{4}\sum_{K\in\mathscr{C}} |\{j : t_j \in K\}| = \frac{1}{4}\sum_{j=1}^{m} |\{K \in \mathscr{C} : t_j \in K\}| = \frac{1}{4}\cdot m \cdot 4 = m. \qquad \square$$

The objective value of x^* is $\frac{35}{4}$, but the optimal integral solution to the LP is 10, since at least 3 Steiner vertices need to be included. Hence, the gap of our new LP is no better than $\frac{10}{35/4} = \frac{8}{7}$.

5.2 A gap upper bound for b-quasi-bipartite instances

In [38] Rajagopalan and Vazirani show that the bidirected cut relaxation has a gap of at most $\frac{3}{2}$, if the graph is quasi-bipartite. Since $(\mathrm{P}_{ST}^{\varnothing})$ is stronger than the bidirected cut relaxation its gap is also at most $\frac{3}{2}$ for such graphs. We are able to generalize this result as follows.

Theorem 2. *On b-quasi-bipartite graphs,* $(\mathrm{P}_{ST}^{\varnothing})$ *has an integrality gap between* $\frac{8}{7}$ *and* $\frac{2b+1}{b+1}$ *in the worst case.*

Proof. The lower bound comes from Section 5.1. We assume G is b-quasi-bipartite, we let T^* be an optimal Steiner tree, and we let $\mathscr{S}^*$ be its set of full components. Since T^* is a minimum spanning tree for $\mathscr{S}^*$, there is a corresponding feasible dual y for (D_{SP}). When we convert y to a dual for $(\mathrm{D}_{ST}^{\mathscr{S}^*})$, we claim that y is feasible: indeed, by Corollary 8 a violated full component could be used to improve the solution, but T^* is already optimal. The next lemma is the cornerstone of our proof.

Lemma 16. *Let π be a partition of $V(\mathscr{S}^*)$ with $y_\pi > 0$. Then $(\overline{r}(\pi)-1) \geq \frac{b+1}{2b+1}(r(\pi)-1)$.*

Proof. For each part V_i of π, let us identify all of the vertices of V_i into a single pseudo-vertex v_i. We may assume by Theorem 3 that each $T^*[V_i]$ is connected, hence this identification process yields a tree T'. Let us say that v_i is *Steiner* if and only if all vertices of V_i are Steiner. Note that T' has $r(\pi)$ pseudo-vertices and $r(\pi)-\overline{r}(\pi)$ of these pseudo-vertices are Steiner. The full components of T' are defined analogously to the full components of a Steiner tree.

Consider any full component K' of T' and let K' contain exactly s Steiner pseudo-vertices. It is straightforward to see that $s \leq b$. Each Steiner pseudo-vertex in K' has degree at least 3 by Assumptions A1 and A2, and at most $s-1$ edges of K' join Steiner vertices to other Steiner vertices. Hence K' has at least $3s-(s-1) = 2s+1$ edges, and so

$$|E(K')| \geq \frac{2s+1}{s} \cdot s \geq \frac{2b+1}{b} \cdot s.$$

Now summing over all full components K', we obtain

$$|E(T')| \geq \frac{2b+1}{b} \cdot \#\{\text{Steiner pseudo-vertices of } T'\}.$$

But $|E(T')| = r(\pi)-1$ and T' has $r(\pi)-\overline{r}(\pi)$ Steiner pseudo-vertices, therefore

$$r(\pi)-1 \geq \frac{2b+1}{b}((r(\pi)-1)-(\overline{r}(\pi)-1)) \quad \Rightarrow \quad \frac{2b+1}{b}(\overline{r}(\pi)-1) \geq \frac{b+1}{b}(r(\pi)-1).$$

This proves what we wanted to show. ☐

It follows that the objective value of y in $(\mathrm{D}_{ST}^{\mathscr{S}^*})$ is

$$\sum_{\pi \in \Pi^{\mathscr{S}}} (\overline{r}(\pi)-1) y_\pi \geq \sum_{\pi \in \Pi^{\mathscr{S}}} \frac{b+1}{2b+1}(r(\pi)-1) y_\pi = \frac{b+1}{2b+1} c(T^*)$$

and since T^* is an optimum integer solution of $(\mathrm{P}_{ST}^{\mathscr{S}^*})$, it follows that the integrality gap of $(\mathrm{P}_{ST}^{\mathscr{S}^*})$ is at most $\frac{2b+1}{b+1}$. Then, finally, by applying Lemma 14 to $(\mathrm{P}_{ST}^{\varnothing})$ and $(\mathrm{P}_{ST}^{\mathscr{S}^*})$ we obtain Theorem 2. ☐

6 Proof of Lemma 11

In this section we present a proof of Lemma 11. The methodology follows that proposed by Gröpl et al. [21], see also the presentation of Korte & Vygen [29, Ch. 20] which corrects a small bug. The essential novelty of our approach is an integral-based interpretation of $\mathtt{mst}$, $\overline{\mathtt{mst}}$ and loss, which leads to the cornerstone $\mathtt{mst} = \overline{\mathtt{mst}} + \mathtt{l}$ (Lemma 18). This also results in a new, short proof of the ubiquitous *contraction lemma* (Lemma 22).

When G is a graph and τ is a real number, let $G_{\leq\tau}$ denote the subgraph of G obtained by deleting all edges with weight greater than τ. For a graph G, let $\kappa(G)$ denote the number of connected components of G.

Lemma 17. $\mathtt{mst}(G) = \int_{\tau=0}^{\infty}(\kappa(G_{\leq\tau}) - 1)d\tau$.

Proof. At time τ, Kruskal's primal-dual algorithm raises the objective function of (D_{SP}) at a rate of $r(\pi(\tau)) - 1$ per unit time. By Theorem 3,

$$\mathtt{mst}(G) = c(T) = \sum_{\pi} y^*_{\pi}(r(\pi) - 1) = \int_{\tau=0}^{\tau^*}(r(\pi(\tau)) - 1)d\tau.$$

Now, since $\pi(\tau)$ is the same as the partition induced by the connected components of $G_{\leq\tau}$, and since $\kappa(G_{\leq\tau}) = 1$ for $\tau \geq \tau^*$, we are done. $\square$

We first relate the cost of a minimum-cost spanning tree of $\mathscr{S}$ for some set $\mathscr{S}$ of full components to the (potential) lower-bound $\overline{\mathtt{mst}}(\mathscr{S})$ on $\mathtt{opt}_r$ that it provides.

Lemma 18. *For any graph G and terminal set $R \subset V(G)$,*

$$\mathtt{mst}(G) = \overline{\mathtt{mst}}(G) + \mathtt{l}(G).$$

Proof. Run MST on input G, obtaining output (T, y) Let us adopt the notation from the proof of Theorem 3. The difference $\mathtt{mst}(G) - \overline{\mathtt{mst}}(G)$ satisfies

$$\mathtt{mst}(G) - \overline{\mathtt{mst}}(G) = \sum_{\pi} y_{\pi} r(\pi) - \sum_{\pi} y_{\pi}\overline{r}(\pi) = \int_0^{\tau^*}(r(\pi(\tau)) - \overline{r}(\pi(\tau)))d\tau. \tag{19}$$

Let a *Steiner part* of a partition be a part which contains only Steiner vertices. The quantity $r(\pi(\tau)) - \overline{r}(\pi(\tau))$ counts the number of Steiner parts of $\pi(\tau)$. Recall from Section 2.2 that G^{τ} denotes the forest maintained by Kruskal's algorithm at time $\tau \geq 0$. We then obtain G^{τ}/R from G^{τ} by identifying the set of all terminals; G^{τ}/R has one connected component for each Steiner part of $\pi(\tau)$, and one additional connected component containing all other vertices. Therefore, the right-hand side of (19) is equal to

$$\int_0^{\tau^*}(\kappa(G^{\tau}/R) - 1)d\tau = \int_0^{\infty}(\kappa((G/R)_{\leq\tau}) - 1)d\tau = \mathtt{mst}(G/R),$$

where the last equality uses Lemma 17.

Finally, note that $\mathtt{l}(G) = \mathtt{mst}(G/R)$, since the loss is the minimum-cost set of edges to connect every Steiner vertex to some terminal, which is the same as the minimum-cost set of edges to connect every Steiner vertex to the pseudo-vertex corresponding to R in G/R, which is in turn the minimum spanning tree of G/R. $\square$

We obtain the following immediate corollary:

Corollary 19. *In iteration i of Algorithm 1, adding full component $K \in \mathscr{K}_r$ to $\mathscr{S}$ reduces the cost of* $\mathtt{mst}(\mathscr{S})$ *if and only if $f_i(K) < 1$.*

Proof. By applying Lemma 18 we see that

$$\mathtt{mst}(\mathscr{S}^i) - \mathtt{mst}(\mathscr{S}^i \cup \{K\}) = \overline{\mathtt{mst}}(\mathscr{S}^i) + \mathtt{l}(\mathscr{S}^i) - \overline{\mathtt{mst}}(\mathscr{S}^i \cup \{K\}) - \mathtt{l}(\mathscr{S}^i \cup \{K\}).$$

Whereas the left-hand side is positive iff adding K to $\mathscr{S}^i$ causes a reduction in $\mathtt{mst}$, the right-hand side is positive iff $f_i(K) < 1$, due to the definition of f_i (equation (13)). □

Using Corollaries 8 and 19, we obtain the following.

Corollary 20. *For all* $1 \le i \le p$, $f_i(K^i) < 1$.

Fix an optimum r-Steiner tree T^*. The next two lemmas give bounds that are needed to analyze $\mathtt{RZ}$'s greedy strategy. Informally, the first says that $\overline{\mathtt{mst}}$ is non-increasing, while the second says that $\overline{\mathtt{mst}}$ is supermodular.

Lemma 21. *If* $\mathscr{S} \subseteq \mathscr{S}' \subseteq \mathscr{K}_r$, *then* $\overline{\mathtt{mst}}(\mathscr{S}') \le \overline{\mathtt{mst}}(\mathscr{S})$.

Proof. Using Lemma 18 and Fact 10 we see

$$\overline{\mathtt{mst}}(\mathscr{S}) - \overline{\mathtt{mst}}(\mathscr{S}') = \mathtt{mst}(\mathscr{S}) + \mathtt{l}(\mathscr{S}' \backslash \mathscr{S}) - \mathtt{mst}(\mathscr{S}').$$

However, the right hand side of the above equation is non-negative, as $\mathtt{MST}(\mathscr{S}) \cup \mathtt{L}(\mathscr{S}' \backslash \mathscr{S})$ is a spanning tree of $\mathscr{S}'$. Lemma 21 then follows. □

Lemma 22 (Contraction Lemma)**.** *Let* $\mathscr{S}^0, \mathscr{S}^1, \mathscr{S}^2 \subset \mathscr{K}_r$ *be disjoint collections of full components with* $\binom{R}{2} \subseteq \mathscr{S}^0$. *Then*

$$\overline{\mathtt{mst}}(\mathscr{S}^0) - \overline{\mathtt{mst}}(\mathscr{S}^0 \cup \mathscr{S}^2) \ge \overline{\mathtt{mst}}(\mathscr{S}^0 \cup \mathscr{S}^1) - \overline{\mathtt{mst}}(\mathscr{S}^0 \cup \mathscr{S}^1 \cup \mathscr{S}^2).$$

Proof. The statement to be proved is equivalent to

$$\mathtt{mst}(\mathscr{S}^0) - \mathtt{mst}(\mathscr{S}^0 \cup \mathscr{S}^2) \ge \mathtt{mst}(\mathscr{S}^0 \cup \mathscr{S}^1) - \mathtt{mst}(\mathscr{S}^0 \cup \mathscr{S}^1 \cup \mathscr{S}^2), \tag{20}$$

due to Lemma 18 and Fact 10. Our proof is centred around proving that for all $\tau \ge 0$,

$$\kappa(\mathscr{S}^0_{\le\tau}) - \kappa(\mathscr{S}^0_{\le\tau} \cup \mathscr{S}^2_{\le\tau}) \ge \kappa(\mathscr{S}^0_{\le\tau} \cup \mathscr{S}^1_{\le\tau}) - \kappa(\mathscr{S}^0_{\le\tau} \cup \mathscr{S}^1_{\le\tau} \cup \mathscr{S}^2_{\le\tau}). \tag{21}$$

If we prove Equation (21), then by adding $-1+1$ to each side, integrating along τ, and using Lemma 17, we obtain Equation (20) as needed.

Define a function μ on graphs by $\mu(G) = |V(G)| - \kappa(G)$. The crux is that μ is the rank function for graphic matroids, and is hence submodular. Similarly, the function $|V(G)|$ is modular, and so $\kappa(G) = |V(G)| - \mu(G)$ is supermodular, which proves Equation (21). □

Note that the proof of Lemma 22 easily generalizes to other matroids. This seems not to have been noticed before, and is not evident from early proofs of the Contraction Lemma (e.g. [4, Lemma 3.9], [21], [39, Lemma 2]) — although it is not hard to deduce from the presentation of Korte & Vygen [29].

We are finally near the end of the analysis, where the Contraction Lemma comes into play. We can now bound the value $f_i(K^i)$ for all $0 \le i \le p-1$ in terms of the cost of T^*'s loss. In the remainder of the section, let $\mathtt{l}^*$ denote $\mathtt{l}(T^*)$, let $\overline{\mathtt{mst}}^i$ denote $\overline{\mathtt{mst}}(\mathscr{S}^i)$ and let $\overline{\mathtt{mst}}^*$ denote $\overline{\mathtt{mst}}(T^*)$.

Lemma 23. *For all* $0 \le i \le p-1$, *if* $\overline{\mathtt{mst}}^i - \overline{\mathtt{mst}}^* > 0$, *then* $f_i(K^i) \le \mathtt{l}^*/(\overline{\mathtt{mst}}^i - \overline{\mathtt{mst}}^*)$.

Proof. Let the full components of T^* be $K^{*,1},\dots,K^{*,q}$. By the choice of K^i in Algorithm 1, we have $f_i(K^i) \le \min_j f_i(K^{*,j})$. A standard fraction averaging argument implies that

$$\begin{aligned} f_i(K^i) &\le \frac{\sum_{j=1}^{q} \mathtt{l}(K^{*,j})}{\sum_{j=1}^{q}\left(\overline{\mathtt{mst}}(\mathscr{S}^i) - \overline{\mathtt{mst}}(\mathscr{S}^i \cup \{K^{*,j}\})\right)} \\ &\le \frac{\mathtt{l}^*}{\sum_{j=1}^{q}\left(\overline{\mathtt{mst}}(\mathscr{S}^i \cup \{K^{*,1},\dots,K^{*,j-1}\}) - \overline{\mathtt{mst}}(\mathscr{S}^i \cup \{K^{*,1},\dots,K^{*,j}\})\right)} \end{aligned} \tag{22}$$

where the last inequality uses Fact 10 and Lemma 22. The denominator of the right-hand side of Equation (22) is a telescoping sum. Canceling like terms, and using Lemma 21 to replace $\overline{\mathtt{mst}}(\mathscr{S}^i \cup \{K^{*,1},\dots,K^{*,q}\})$ with $\overline{\mathtt{mst}}^*$, we are done. □

We can now bound the cost of T^p.

Proof of Lemma 11. We first bound the loss $\mathtt{l}(T^p)$ of tree T^p. Using Fact 10,

$$\mathtt{l}(T^p) = \sum_{i=0}^{p-1} \mathtt{l}(K^i) = \sum_{i=0}^{p-1} f_i(K^i)\cdot(\overline{\mathtt{mst}}^i - \overline{\mathtt{mst}}^{i+1}) \tag{23}$$

where the last equality uses the definition of f_i from (13). Using Corollary 20 and Lemma 23, the right hand side of Equation (23) is bounded as follows:

$$\sum_{i=0}^{p-1} f_i(K^i)\cdot(\overline{\mathtt{mst}}^i - \overline{\mathtt{mst}}^{i+1}) \le \sum_{i=0}^{p-1} \frac{\mathtt{l}^*}{\max\{\mathtt{l}^*, \overline{\mathtt{mst}}^i - \overline{\mathtt{mst}}^*\}}\cdot(\overline{\mathtt{mst}}^i - \overline{\mathtt{mst}}^{i+1}). \tag{24}$$

The right hand side of Equation (24) can in turn be bounded from above by the following integral:

$$\sum_{i=0}^{p-1} \frac{\mathtt{l}^*\cdot(\overline{\mathtt{mst}}^i - \overline{\mathtt{mst}}^{i+1})}{\max\{\mathtt{l}^*, \overline{\mathtt{mst}}^i - \overline{\mathtt{mst}}^*\}} \le \int_{\overline{\mathtt{mst}}^p}^{\overline{\mathtt{mst}}^0} \frac{\mathtt{l}^*}{\max\{\mathtt{l}^*, x - \overline{\mathtt{mst}}^*\}}dx = \int_{\overline{\mathtt{mst}}^p - \overline{\mathtt{mst}}^*}^{\overline{\mathtt{mst}}^0 - \overline{\mathtt{mst}}^*} \frac{\mathtt{l}^*}{\max\{\mathtt{l}^*, x\}}dx. \tag{25}$$

Notice that $\overline{\mathtt{mst}}^0 = \mathtt{mst}(G[R],c) \ge \mathtt{opt}_r = \mathtt{l}^* + \overline{\mathtt{mst}}^*$. The termination condition in Algorithm 1 and Lemma 6 imply that $\overline{\mathtt{mst}}^p \le \mathtt{opt}_r$. Hence the result of evaluating the integral in the right-hand side of Equation (25) is

$$\mathtt{l}^* - (\overline{\mathtt{mst}}^p - \overline{\mathtt{mst}}^*) + \mathtt{l}^*\cdot \int_{\mathtt{l}^*}^{\overline{\mathtt{mst}}^0 - \overline{\mathtt{mst}}^*} \frac{1}{x}dx = \mathtt{opt}_r - \overline{\mathtt{mst}}^p + \mathtt{l}^*\cdot \ln\left(\frac{\overline{\mathtt{mst}}^0 - \overline{\mathtt{mst}}^*}{\mathtt{l}^*}\right) \tag{26}$$

where the equality uses Lemma 18. Applying Lemma 18 two more times, and combining Equations (23)–(26), we obtain

$$\begin{aligned} c(T^p) = \overline{\mathtt{mst}}^p + \mathtt{l}(T^p) &\le \mathtt{opt}_r + \mathtt{l}^*\cdot\ln\left(\frac{\overline{\mathtt{mst}}^0 - \overline{\mathtt{mst}}^*}{\mathtt{l}^*}\right) \\ &= \mathtt{opt}_r + \mathtt{l}^*\cdot\ln\left(1 + \frac{\overline{\mathtt{mst}}^0 - (\overline{\mathtt{mst}}^* + \mathtt{l}^*)}{\mathtt{l}^*}\right) \\ &= \mathtt{opt}_r + \mathtt{l}^*\cdot\ln\left(1 + \frac{\overline{\mathtt{mst}}^0 - \mathtt{opt}_r}{\mathtt{l}^*}\right) \end{aligned}$$

as wanted. □

7 Conclusion and Future Directions

There is a large body of work on relaxations for the Steiner tree problem. While many of these formulations have lead to improved, integer-programming based exact algorithms, none of the relaxations has a known integrality gap smaller than 2. In this paper we propose a hypergraph-based relaxation, and we first showed that the best known approximation algorithm for the Steiner tree problem has a natural interpretation as a primal-dual algorithm for this LP. We then derived an upper-bound on the integrality gap of our LP which is nearly 2 for general graphs, but smaller than 2 for graphs with small Steiner neighbourhoods.

The obvious open question is whether there is a relaxation whose gap is a constant strictly smaller than 2 for general instances. The integrality gap of the bidirected cut relaxation, and therefore also the gap of our formulation, is widely conjectured to be bounded away from 2. We hope that the connection between greedy and LP-based algorithms developed in this paper proves useful in the quest for smaller integrality gaps.

Most primal-dual algorithms naïvely increase dual variables in a monotone way, and thus often find dual solutions of poor quality. In their recent paper [6], Chakrabarty et al. showed that a suitable preprocessing of a given quasi-bipartite Steiner tree instance may steer a primal-dual algorithm to higher value dual solutions. As mentioned, we can use the filtering technique from [6] in order to slightly improve the bound given in Theorem 2 to $(2b-1)/b$ for $b \geq 2$ [28]. Can this bound be decreased further by using more sophisticated filtering ideas?

Direct primal LP rounding techniques offer yet another way of proving upper bounds on the integrality gap of an LP. Hypergraph-based formulations may be useful in this approach as their basic solutions have an appealing *nested* structure. Extending known results for undirected-cut formulations, partitions corresponding to tight inequalities in basic solutions to our LP may be *uncrossed* [27]. This suggests an attack via *iterated rounding*, a technique pioneered by Jain [23] that produces an integral feasible solution for an instance of the survivable network design problem by rounding a fractional basic solution in multiple stages. However, one quickly realizes that a naïve implementation of Jain's strategy will not work as a folklore example similar to Skutella's shows that some extreme points of bidirected cut have support of size $\Omega(|V|^2)$. Developing more a sophisticated direct rounding strategy is a challenging open question.

References

[1] A. Agarwal and M. Charikar. On the advantage of network coding for improving network throughput. In *Proc. IEEE Information Theory Workshop*, pages 105–109, 2004.

[2] A. Agrawal, P. Klein, and R. Ravi. When trees collide: An approximation algorithm for the generalized Steiner problem in networks. *SIAM J. Comput.*, 24:440–456, 1995.

[3] Y. P. Aneja. An integer linear programming approach to the Steiner problem in graphs. *Networks*, 10:167–178, 1980.

[4] P. Berman and V. Ramaiyer. Improved approximations for the Steiner tree problem. *J. Algorithms*, 17(3):381–408, 1994.

[5] A. Borchers and D. Du. The *k*-Steiner ratio in graphs. *SIAM J. Comput.*, 26(3):857–869, 1997.

[6] D. Chakrabarty, N. R. Devanur, and V. V. Vazirani. New geometry-inspired relaxations and algorithms for the metric Steiner tree problem. In *Proc. 13th IPCO*, pages 344–358, 2008.

[7] M. Chlebík and J. Chlebíková. The Steiner tree problem on graphs: Inapproximability results. *Theor. Comput. Sci.*, 406(3):207–214, 2008. Preliminary version appeared in *Proc. 8th SWAT*, pages 170–179, 2002.

[8] S. Chopra. On the spanning tree polyhedron. *Operations Research Letters*, 8:25–29, 1989.

[9] S. Chopra and M. R. Rao. The Steiner tree problem 1: Formulations, compositions, and extension of facets. *Mathematical Programming*, 64:209–229, 1994.

[10] S. Chopra and M. R. Rao. The Steiner tree problem 2: Properties and classes of facets. *Mathematical Programming*, 64:231–246, 1994.

[11] M. Didi Biha, H. Kerivin, and A. R. Mahjoub. Steiner trees and polyhedra. *Discrete Applied Mathematics*, 112(1-3):101–120, 2001.

[12] S. E. Dreyfus and R. A. Wagner. The Steiner problem in graphs. *Networks*, 1:195–207, 1972.

[13] J. Edmonds. Optimum branchings. *J. Res. Nat. Bur. Standards*, B71:233–240, 1967.

[14] D. R. Fulkerson. Blocking and anti-blocking pairs of polyhedra. *Math. Programming*, 1:168–194, 1971.

[15] M. R. Garey and D. S. Johnson. The rectilinear Steiner tree problem is NP complete. *SIAM J. Appl. Math.*, 32:826–834, 1977.

[16] E. N. Gilbert and H. O. Pollak. Steiner minimal trees. *SIAM J. Appl. Math.*, 16(1):1–29, 1968.

[17] M. X. Goemans. The Steiner tree polytope and related polyhedra. *Math. Program.*, 63(2):157–182, 1994.

[18] M. X. Goemans and D. Bertsimas. Survivable networks, linear programming relaxations and the parsimonious property. *Math. Programming*, 60:145–166, 1993.

[19] M. X. Goemans and Y. Myung. A catalog of Steiner tree formulations. *Networks*, 23:19–28, 1993.

[20] M. X. Goemans and D. P. Williamson. The primal-dual method for approximation algorithms and its application to network design problems. In D. S. Hochbaum, editor, *Approximation Algorithms for NP-hard Problems*, chapter 4. PWS, Boston, 1997.

[21] C. Gröpl, S. Hougardy, T. Nierhoff, and H. J. Prömel. Approximation algorithms for the Steiner tree problem in graphs. In X. Cheng and D. Du, editors, *Steiner trees in industries*, pages 235–279. Kluwer Academic Publishers, Norvell, Massachusetts, 2001.

[22] S. Hougardy and H. J. Prömel. A 1.598 approximation algorithm for the Steiner problem in graphs. In *Proc. 10th SODA*, pages 448–453, 1999.

[23] K. Jain. A factor 2 approximation algorithm for the generalized Steiner network problem. *Combinatorica*, 21(1):39–60, 2001. Preliminary version appeared in *Proc. 39th FOCS*, pages 448–457, 1998.

[24] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Plenum Press, NY, 1972.

[25] M. Karpinski and A. Zelikovsky. New approximation algorithms for the Steiner tree problems. *J. Combinatorial Optimization*, 1(1):47–65, 1997.

[26] J. Könemann, S. Leonardi, G. Schäfer, and S. H. M. van Zwam. A group-strategyproof cost sharing mechanism for the Steiner forest game. *SIAM J. Comput.*, 37(5):1319–1341, 2008. Preliminary version appeared in *Proc. 16th SODA*, pages 612–619, 2005.

[27] J. Könemann and D. Pritchard. Uncrossing partitions. Technical Report CORR 2007-11, Department of Combinatorics & Optimization, University of Waterloo, 2007.

[28] J. Könemann, D. Pritchard, and Y. Wei. Filtering for the Steiner tree problem. Manuscript, 2008.

[29] B. Korte and J. Vygen. *Combinatorial Optimization*. Springer, New York, 2008.

[30] J. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings, American Mathematical Society*, 7:48–50, 1956.

[31] T. Polzin. *Algorithms for the Steiner Problem in Networks*. PhD thesis, Universität des Saarlandes, February 2003.

[32] T. Polzin and S. V. Daneshmand. Primal-dual approaches to the Steiner problem. In *Proc. 3rd APPROX*, pages 214–225, 2000.

[33] T. Polzin and S. Vahdati Daneshmand. A comparison of Steiner tree relaxations. *Discrete Applied Mathematics*, 112(1-3):241–261, 2001.

[34] T. Polzin and S. Vahdati Daneshmand. Improved algorithms for the Steiner problem in networks. *Discrete Applied Mathematics*, 112(1-3):263–300, 2001.

[35] T. Polzin and S. Vahdati Daneshmand. On Steiner trees and minimum spanning trees in hypergraphs. *Oper. Res. Lett.*, 31(1):12–20, 2003.

[36] H. J. Prömel and A. Steger. A new approximation algorithm for the Steiner tree problem with performance ratio 5/3. *J. Algorithms*, 36(1):89–101, 2000. Preliminary version appeared in *Proc. 14th STACS*, pages 559–570, 1997.

[37] H. J. Prömel and A. Steger. *The Steiner Tree Problem — A Tour through Graphs, Algorithms, and Complexity*. Vieweg Verlag, Braunschweig-Wiesbaden, 2002.

[38] S. Rajagopalan and V. V. Vazirani. On the bidirected cut relaxation for the metric Steiner tree problem. In *Proc. 10th SODA*, pages 742–751, 1999.

[39] R. Rizzi. On Rajagopalan and Vazirani's $3/2$-approximation bound for the Iterated 1-Steiner heuristic. *Information Processing Letters*, 86(6):335–338, 2003.

[40] G. Robins and A. Zelikovsky. Tighter bounds for graph Steiner tree approximation. *SIAM J. Discrete Math.*, 19(1):122–134, 2005. Preliminary version appeared in *Proc. 11th SODA*, pages 770–779, 2000.

[41] M. Skutella. Personal communication, 2006.

[42] V. Vazirani. Recent results on approximating the Steiner tree problem and its generalizations. *Theoret. Comput. Sci.*, 235(1):205–216, 2000.

[43] V. V. Vazirani. *Approximation Algorithms*. Springer, 2001.

[44] D. Warme. A new exact algorithm for rectilinear Steiner trees. In P. Pardalos and D.-Z. Du, editors, *Network Design: Connectivity and Facilities Location*, pages 357–395. American Mathematical Society, 1997.

[45] D. Warme. *Spanning Trees in Hypergraphs with Applications to Steiner Trees*. PhD thesis, University of Virginia, 1998.

[46] R. T. Wong. A dual ascent approach for Steiner tree problems on a directed graph. *Math. Programming*, 28:271–287, 1984.

[47] A. Zelikovsky. Better approximation bounds for the network and Euclidean Steiner tree problems. Technical Report CS-96-06, University of Virginia, Charlottesville, VA, USA, 1996.

[48] A. Z. Zelikovsky. An $11/6$-approximation algorithm for the network Steiner problem. *Algorithmica*, 9:463–470, 1993.