Hypergraphic LP Relaxations for Steiner Trees

Deeparnab Chakrabarty, Jochen Könemann, and David Pritchard

University of Waterloo, Waterloo, Ontario N2L 3G1, Canada*

Abstract. We investigate hypergraphic LP relaxations for the Steiner tree problem, primarily the partition LP relaxation introduced by Könemann et al. [Math. Programming, 2009]. Specifically, we are interested in proving upper bounds on the integrality gap of this LP, and studying its relation to other linear relaxations. Our results are the following.

Structural results: We extend the technique of uncrossing, usually applied to families of sets, to families of partitions. As a consequence we show that any basic feasible solution to the partition LP formulation has sparse support. Although the number of variables could be exponential, the number of positive variables is at most the number of terminals.

Relations with other relaxations: We show the equivalence of the partition LP relaxation with other known hypergraphic relaxations. We also show that these hypergraphic relaxations are equivalent to the well studied bidirected cut relaxation, if the instance is quasibipartite.

Integrality gap upper bounds: We show an upper bound of $\sqrt{3} \doteq 1.729$ on the integrality gap of these hypergraph relaxations in general graphs. In the special case of uniformly quasibipartite instances, we show an improved upper bound of $73/60 \doteq 1.216$. By our equivalence theorem, the latter result implies an improved upper bound for the bidirected cut relaxation as well.

1 Introduction

In the *Steiner tree* problem, we are given an undirected graph $G = (V, E)$, non-negative costs c_e for all edges $e \in E$, and a set of *terminal* vertices $R \subseteq V$. The goal is to find a minimum-cost tree T spanning R, and possibly some *Steiner vertices* from $V \setminus R$. We can assume that the graph is complete and that the costs induce a metric. The problem takes a central place in the theory of combinatorial optimization and has numerous practical applications. Since the Steiner tree problem is NP-hard[1] we are interested in approximation algorithms for it. The best published approximation algorithm for the Steiner tree problem is due to Robins and Zelikovsky [20], which for any fixed $\epsilon > 0$, achieves a performance ratio of $1 + \frac{\ln 3}{2} + \epsilon \doteq 1.55$ in polynomial time; an improvement is currently in press [2], see also Remark 1.

* Supported by NSERC grant no. 288340 and by an Early Research Award

[1] Chlebík and Chlebíková show that no $(96/95-\epsilon)$-approximation algorithm can exist for any positive ϵ unless P=NP [5].

In this paper, we study linear programming (LP) relaxations for the Steiner tree problem, and their properties. Numerous such formulations are known (e.g., see [7, 11, 16, 17, 24, 25]), and their study has led to impressive running time improvements for integer programming based methods. Despite the significant body of work in this area, none of the known relaxations is known to exhibit an *integrality gap* provably smaller than 2. The integrality gap of a relaxation is the maximum ratio of the cost of integral and fractional optima, over all instances. It is commonly regarded as a measure of strength of a formulation. One of the contributions of this paper are improved bounds on the integrality gap for a number of Steiner tree LP relaxations.

A Steiner tree relaxation of particular interest is the *bidirected cut relaxation* [7, 25] (precise definitions will follow in Section 1.2). This relaxation has a flow formulation using $O(|E||R|)$ variables and constraints, which is much more compact than the other relaxations we study. Also, it is also widely believed to have an integrality gap significantly smaller than 2 (e.g., see [3, 19, 23]). The largest lower bound on the integrality gap known is 8/7 (by Martin Skutella, reported in [15]), and Chakrabarty et al. [3] prove an upper bound of 4/3 in so called *quasi-bipartite* instances (where Steiner vertices form an independent set).

Another class of formulations are the so called *hypergraphic* LP relaxations for the Steiner tree problem. These relaxations are inspired by the observation that the minimum Steiner tree problem can be encoded as a minimum cost hyper-spanning tree (see Section 1.2) of a certain hypergraph on the terminals. They are known to be stronger than the bidirected cut relaxation [18], and it is therefore natural to try to use them to get better approximation algorithms, by drawing on the large corpus of known LP techniques. In this paper, we focus on one hypergraphic LP in particular: the *partition* LP of Könemann et al. [15].

1.1 Our Results and Techniques

There are three classes of results in this paper: structural results, equivalence results, and integrality gap upper bounds.

Structural results, Section 2: We extend the powerful technique of *uncrossing*, traditionally applied to families of sets, to families of *partitions*. Set uncrossing has been very successful in obtaining exact and approximate algorithms for a variety of problems (for instance, [9, 14, 21]). Using partition uncrossing, we show that any basic feasible solution to the partition LP has at most $(|R|-1)$ positive variables (even though it can have an exponentially large number of variables and constraints).

Equivalence results, Section 3: In addition to the partition LP, two other hypergraphic LPs have been studied before: one based on *subtour elimination* due to Warme [24], and a *directed hypergraph relaxation* of Polzin and Vahdati Daneshmand [18]; these two are known to be equivalent [18]. We prove that in fact *all three hypergraphic relaxations are equivalent* (that is, they have the same objective value for any Steiner tree instance).

We also show that, on *quasibipartite instances*, the hypergraphic and the bidirected cut LP relaxations are equivalent. This result is surprising since we are aware of no qualitative similarity to suggest why the two relaxations should be equivalent. We believe a better understanding of the bidirected cut relaxation is important because it is central in theory *and* practical for implementation.

Improved integrality gap upper bounds, Section 4: For *uniformly quasibipartite instances* (quasibipartite instances where for each Steiner vertex, all incident edges have the same cost), we show that the integrality gap of the hypergraphic LP relaxations is upper bounded by $73/60 \doteq 1.216$. Our proof uses the approximation algorithm of Gröpl et al. [13] which achieves the same ratio with respect to the (integral) optimum. We show, via a simple dual fitting argument, that this ratio is also valid with respect to the LP value. To the best of our knowledge this is the only nontrivial class of instances where the best currently known approximation ratio and integrality gap upper bound are the same.

For general graphs, we give simple upper bounds of $2\sqrt{2} - 1 \doteq 1.83$ and $\sqrt{3} \doteq 1.729$ on the integrality gap of the hypergraph relaxation. Call a graph *gainless* if the minimum *spanning* tree of the terminals is the optimal Steiner tree. To obtain these integrality gap upper bounds, we use the following key property of the hypergraphic relaxation which was implicit in [15]: on gainless instances (instances where the optimum terminal spanning tree is the optimal Steiner tree), the LP value equals the minimum spanning tree and the integrality gap is 1. Such a theorem was known for quasibipartite instances and the bidirected cut relaxation (implicitly in [19], explicitly in [3]); we extend techniques of [3] to obtain improved integrality gaps on all instances.

Remark 1. The recent independent work of Byrka et al. [2], which gives an improved approximation for Steiner trees in general graphs, also shows an integrality gap bound of 1.55 on the hypergraphic directed cut LP. This is stronger than our integrality gap bounds and was obtained prior to the completion of our paper; yet we include our bounds because they are obtained using fairly different methods which might be of independent interest in certain settings.

The proof in [2] can be easily modified to show an integrality gap upper bound of 1.28 in quasibipartite instances. Then using our equivalence result, we get an integrality gap upper bound of 1.28 for the bidirected cut relaxation on quasibipartite instances, improving the previous best of 4/3.

1.2 Bidirected Cut and Hypergraphic Relaxations

The Bidirected Cut Relaxation The first bidirected LP was given by Edmonds [7] as an exact formulation for the spanning tree problem. Wong [25] later extended this to obtain the bidirected cut relaxation for the Steiner tree problem, and gave a dual ascent heuristic based on the relaxation. For this relaxation, introduce two arcs (u, v) and (v, u) for each edge $uv \in E$, and let both of their costs be c_{uv}. Fix an arbitrary terminal $r \in R$ as the root. Call a subset $U \subseteq V$ *valid* if it contains a terminal but not the root, and let $\text{valid}(V)$ be the family of all valid sets. Clearly, the in-tree rooted at r (the directed tree with all vertices but the root having out-degree exactly 1) of a Steiner tree T must have at least one arc with tail in U and head outside U, for all valid U. This leads to the bidirected cut relaxation ($\mathcal{B}$) (shown in Figure 1 with dual) which has a variable for each arc $a \in A$, and a constraint for every valid set U. Here and later, $\delta^{\text{out}}(U)$ denotes the set of arcs in A whose tail is in U and whose head lies in $V \setminus U$. When there are no Steiner vertices, Edmonds' work [7] implies this relaxation is exact.

$$\min \sum_{a \in A} c_a x_a : \quad x \in \mathbf{R}^A_{\geq 0} \qquad (\mathcal{B})$$
$$\sum_{a \in \delta^{\text{out}}(U)} x_a \geq 1, \quad \forall U \in \text{valid}(V)$$

$$\max \sum_{U} z_U : \quad z \in \mathbf{R}^{\text{valid}(V)}_{\geq 0} \qquad (\mathcal{B}_D)$$
$$\sum_{U : a \in \delta^{\text{out}}(U)} z_U \leq c_a, \quad \forall a \in A$$

Fig. 1. The bidirected cut relaxation ($\mathcal{B}$) and its dual ($\mathcal{B}_D$).

Goemans & Myung [11] made significant progress in understanding the LP, by showing that the bidirected cut LP has the same value independent of which terminal is chosen as the root, and by showing that a whole "catalogue" of very different-looking LPs also has the same value; later Goemans [10] showed that if the graph is series-parallel, the relaxation is exact. Rajagopalan and Vazirani [19] were the first to show a non-trivial integrality gap upper bound of 3/2 on quasibipartite graphs; this was subsequently improved to 4/3 by Chakrabarty et al. [3], who gave another alternate formulation for ($\mathcal{B}$).

Hypergraphic Relaxations Given a Steiner tree T, a *full component* of T is a maximal subtree of T all of whose leaves are terminals and all of whose internal nodes are Steiner nodes. The edge set of any Steiner tree can be partitioned in a *unique* way into full components by splitting at internal terminals; see Figure 2 for an example.

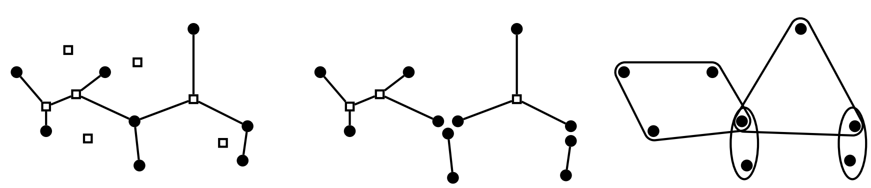

Fig. 2. Black nodes are terminals and white nodes are Steiner nodes. Left: a Steiner tree for this instance. Middle: the Steiner tree's edges are partitioned into full components; there are four full components. Right: the hyperedges corresponding to these full components.

Let $\mathcal{K}$ be the set of all nonempty subsets of terminals (*hyperedges*). We associate with each $K \in \mathcal{K}$ a fixed full component spanning the terminals in K, and let C_K be its cost[2]. The problem of finding a minimum-cost Steiner tree spanning R now reduces to that of finding a minimum-cost hyper-spanning tree in the hypergraph $(R, \mathcal{K})$.

Spanning trees in (normal) graphs are well understood and there are many different exact LP relaxations for this problem. These exact LP relaxations for spanning trees in graphs inspire the *hypergraphic relaxations* for the Steiner tree problem. Such relaxations have a variable x_K for every[3] $K \in \mathcal{K}$, and the different relaxations are based on the constraints used to capture a hyper-spanning tree, just as constraints on edges are used to capture a spanning tree in a graph.

The oldest hypergraphic LP relaxation is the subtour LP introduced by Warme [24] which is inspired by Edmonds' subtour elimination LP relaxation [8] for the spanning tree polytope. This LP relaxation uses the fact that there are no hypercycles in a hyper-spanning tree, and that it is spanning. More formally, let $\rho(X) := \max(0, |X| - 1)$ be the *rank* of a set X of vertices. Then a sub-hypergraph $(R, \mathcal{K}')$ is a hyper-spanning tree iff $\sum_{K \in \mathcal{K}'} \rho(K) = \rho(R)$ and $\sum_{K \in \mathcal{K}'} \rho(K \cap S) \leq \rho(S)$ for every subset S of R. The corresponding LP relaxation, denoted below as $(\mathcal{S})$, is called the *subtour elimination* LP relaxation.

$$\min \Big\{ \sum_{K \in \mathcal{K}} C_K x_K : \ x \in \mathbf{R}^{\mathcal{K}}_{\geq 0}, \ \sum_{K \in \mathcal{K}} x_K \rho(K) = \rho(R), \qquad (\mathcal{S})$$

$$\sum_{K \in \mathcal{K}} x_K \rho(K \cap S) \leq \rho(S), \ \forall S \subset R \Big\}$$

Warme showed that if the maximum hyperedge size r is bounded by a constant, the LP can be solved in polynomial time.

The next hypergraphic LP introduced for Steiner tree was a directed hypergraph formulation $(\mathcal{D})$, introduced by Polzin and Vahdati Daneshmand [18], and inspired by the bidirected cut relaxation. Given a full component K and a terminal $i \in K$, let K^i denote the arborescence obtained by directing all the edges of K towards i. Think of this as directing the hyperedge K towards i to get the directed hyperedge K^i. Vertex i is called the *head* of K^i while the terminals in $K \setminus i$ are the *tails* of K. The cost of each directed hyperedge K^i is the cost of the corresponding undirected hyperedge K. In the directed hypergraph formulation, there is a variable x_{K^i} for every directed hyperedge K^i. As in the bidirected cut relaxation, there is a vertex $r \in R$ which is a root, and as described above, a subset $U \subseteq R$ of terminals is valid if it does not contain the root but contains at least one vertex in R. We let $\Delta^{\text{out}}(U)$ be the set of directed full components coming out of U, that is all K^i such that $U \cap K \neq \varnothing$ but $i \notin U$. Let $\overrightarrow{\mathcal{K}}$ be the

[2] We choose the minimum cost full component if there are many. If there is no full component spanning K, we let C_K be infinity. Such a minimum cost component can be found in polynomial time, if $|K|$ is a constant.

[3] Observe that there could be exponentially many hyperedges. This computational issue is circumvented by considering hyperedges of size at most r, for some constant r. By a result of Borchers and Du [1], this leads to only a $(1 + \Theta(1/\log r))$ factor increase in the optimal Steiner tree cost.

set of all directed hyperedges. We show the directed hypergraph relaxation and its dual in Figure 3.

$$\min\Big\{\sum_{K\in\mathcal{K},i\in K} C_K x_{K^i} : x\in\mathbf{R}_{\geq 0}^{\overrightarrow{\mathcal{K}}} \qquad (\mathcal{D})$$
$$\sum_{K^i\in\Delta^{\mathrm{out}}(U)} x_{K^i}\geq 1,\quad \forall \text{ valid } \ U\subseteq R\Big\}$$

$$\max\Big\{\sum_{U} z_U : \qquad z\in\mathbf{R}_{\geq 0}^{\mathrm{valid}(R)} \qquad (\mathcal{D}_D)$$
$$\sum_{U:K\cap U\neq\varnothing,i\notin U} z_U\leq C_K,\quad \forall K\in\mathcal{K}, i\in K\Big\}$$

Fig. 3. The directed hypergraph relaxation $(\mathcal{D})$ and its dual $(\mathcal{D}_D)$.

Polzin & Vahdati Daneshmand [18] showed that $\mathrm{OPT}(\mathcal{D}) = \mathrm{OPT}(\mathcal{S})$. Moreover they observed that this directed hypergraphic relaxation strengthens the bidirected cut relaxation.

Lemma 1 ([18]). *For any instance,* $\mathrm{OPT}(\mathcal{D})\geq\mathrm{OPT}(\mathcal{B})$*. There are instances for which this inequality is strict.*

Könemann et al. [15], inspired by the work of Chopra [6], described a partition-based relaxation which captures that given any partition of the terminals, any hyper-spanning tree must have sufficiently many "cross hyperedges". More formally, a partition, π, is a collection of pairwise disjoint nonempty terminal sets $(\pi_1,\ldots,\pi_q)$ whose union equals R. The number of *parts* q of π is referred to as the partition's *rank* and denoted as $r(\pi)$. Let Π_R be the set of all partitions of R. Given a partition $\pi=\{\pi_1,\ldots,\pi_q\}$, define the *rank contribution* $\mathtt{rc}_K^{\pi}$ of hyperedge $K\in\mathcal{K}$ for π as the rank reduction of π obtained by merging the parts of π that are touched by K; i.e., $\mathtt{rc}_K^{\pi} := |\{i : K\cap\pi_i\neq\varnothing\}|-1$. Then a hyper-spanning tree $(R,\mathcal{K}')$ must satisfy $\sum_{K\in\mathcal{K}'}\mathtt{rc}_K^{\pi}\geq r(\pi)-1$. The partition based LP of [15] and its dual are given in Figure 4.

$$\min\Big\{\sum_{K\in\mathcal{K}} C_K x_K : \quad x\in\mathbf{R}_{\geq 0}^{\mathcal{K}} \quad (\mathcal{P})$$
$$\sum_{K\in\mathcal{K}} x_K\mathtt{rc}_K^{\pi}\geq r(\pi)-1,\quad \forall\pi\in\Pi_R\Big\}$$

$$\max\Big\{\sum_{\pi}(r(\pi)-1)y_\pi : \quad y\in\mathbf{R}_{\geq 0}^{\Pi_R} \quad (\mathcal{P}_D)$$
$$\sum_{\pi\in\Pi_R} y_\pi\mathtt{rc}_K^{\pi}\leq C_K,\quad \forall K\in\mathcal{K}\Big\}$$

Fig. 4. The unbounded partition relaxation $(\mathcal{P})$ and its dual $(\mathcal{P}_D)$.

The feasible region of $(\mathcal{P})$ is *unbounded*, since if x is a feasible solution for $(\mathcal{P})$ then so is any $x'\geq x$. We obtain a *bounded* partition LP relaxation, denoted by $(\mathcal{P}')$ and shown below, by adding a valid equality constraint to the LP.

$$\min\Big\{\sum_{K\in\mathcal{K}} C_K x_K : x\in(\mathcal{P}),\ \sum_{K\in\mathcal{K}} x_K(|K|-1)=|R|-1\Big\} \qquad (\mathcal{P}')$$

2 Uncrossing Partitions

In this section we are interested in *uncrossing* a minimal set of *tight partitions* that uniquely define a basic feasible solution to $(\mathcal{P})$. We start with a few preliminaries necessary to state our result formally.

2.1 Preliminaries

We introduce some needed well-known properties of partitions that arise in combinatorial lattice theory [22].

Definition 1. *We say that a partition π'* refines *another partition π if each part of π' is contained in some part of π. We also say π* coarsens *π'. Two partitions* cross *if neither refines the other. A family of partitions forms a* chain *if no pair of them cross. Equivalently, a chain is any family $\pi^1, \pi^2, \dots, \pi^t$ such that π^i refines π^{i-1} for each $1 < i \le t$.*

The family Π_R of all partitions of R forms a *lattice* with a *meet operator* $\wedge : \Pi_R^2 \to \Pi_R$ and a *join operator* $\vee : \Pi_R^2 \to \Pi_R$. The meet $\pi \wedge \pi'$ is the coarsest partition that refines both π and π', and the join $\pi \vee \pi'$ is the most refined partition that coarsens both π and π'. See Figure 5 for an illustration.

Definition 2 (Meet of partitions). *Let the parts of π be $\pi_1, \dots, \pi_t$ and let the parts of π' be $\pi'_1, \dots, \pi'_u$. Then the parts of the meet $\pi \wedge \pi'$ are the nonempty intersections of parts of π with parts of π',*

$$\pi \wedge \pi' = \{\pi_i \cap \pi'_j \mid 1 \le i \le t, 1 \le j \le u \text{ and } \pi_i \cap \pi'_j \neq \varnothing\}.$$

Given a graph G and a partition π of $V(G)$, we say that G *induces* π if the parts of π are the vertex sets of the connected components of G.

Definition 3 (Join of partitions). *Let (R, E) be a graph that induces π, and let (R, E') be a graph that induces π'. Then the graph $(R, E \cup E')$ induces $\pi \vee \pi'$.*

Given a feasible solution x to $(\mathcal{P})$, a partition π is *tight* if $\sum_{K \in \mathcal{K}} x_K \mathtt{rc}_K^{\pi} = r(\pi) - 1$. Let $\mathtt{tight}(x)$ be the set of all tight partitions. We are interested in *uncrossing* this set of partitions. More precisely, we wish to find a cross-free set of partitions (chain) which uniquely defines x. One way would be to prove the following.

Property 21 *If two crossing partitions π and π' are in $\mathtt{tight}(x)$, then so are $\pi \wedge \pi'$ and $\pi \vee \pi'$.*

This type of property is already well-used [9, 14, 21] for sets (with meets and joins replaced by unions and intersections respectively), and the standard approach is the following. The typical proof considers the constraints in $(\mathcal{P})$ corresponding to π and π' and uses the "supermodularity" of the RHS and the

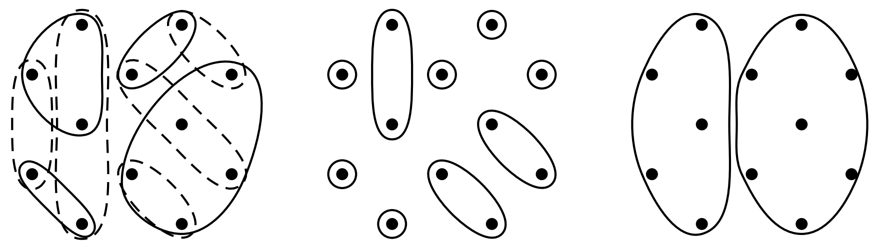

Fig. 5. Illustrations of some partitions. The black dots are the terminal set R. (a): two partitions; neither refines the other. (b): the meet of the partitions from (a). (c): the join of the partitions from (a).

"submodularity" of the coefficients in the LHS. In particular, if the following is true,

$$\forall \pi, \pi' : \; r(\pi \vee \pi') + r(\pi \wedge \pi') \quad \geq \quad r(\pi) + r(\pi') \tag{1}$$

$$\forall K, \pi, \pi' : \; \mathtt{rc}_K^{\pi} + \mathtt{rc}_K^{\pi'} \quad \geq \quad \mathtt{rc}_K^{\pi \vee \pi'} + \mathtt{rc}_K^{\pi \wedge \pi'} \tag{2}$$

then Property 21 can be proved easily by writing a string of inequalities.[4]

Inequality (1) is indeed true (see, for example, [22]), but unfortunately inequality (2) is not true in general, as the following example shows.

Example 1. Let $R = \{1, 2, 3, 4\}$, $\pi = \{\{1, 2\}, \{3, 4\}\}$ and $\pi' = \{\{1, 3\}, \{2, 4\}\}$. Let K denote the full component $\{1, 2, 3, 4\}$. Then $\mathtt{rc}_K^{\pi} + \mathtt{rc}_K^{\pi'} = 1 + 1 < 0 + 3 = \mathtt{rc}_K^{\pi \vee \pi'} + \mathtt{rc}_K^{\pi \wedge \pi'}$.

Nevertheless, Property 21 is true; its correct proof is given in the full version of this paper [4] and depends on a simple though subtle extension of the usual approach. The crux of the insight needed to fix the approach is not to consider *pairs* of constraints in $(\mathcal{P})$, but rather multi-sets which may contain more than two inequalities. Using this uncrossing result, we can prove the following theorem (details are given in [4]). Here, we let $\underline{\pi}$ denote $\{R\}$, the unique partition with (minimal) rank 1; later we use $\overline{\pi}$ to denote $\{\{r\} \mid r \in R\}$, the unique partition with (maximal) rank $|R|$.

Theorem 1. *Let x^* be a basic feasible solution of $(\mathcal{P})$, and let $\mathcal{C}$ be an inclusion-wise maximal chain in $\mathtt{tight}(x^*)\backslash\underline{\pi}$. Then x^* is uniquely defined by*

$$\sum_{K \in \mathcal{K}} \mathtt{rc}_K^{\pi} x_K^* = r(\pi) - 1 \quad \forall \pi \in \mathcal{C}. \tag{3}$$

[4] In this hypothetical scenario we get $r(\pi) + r(\pi') - 2 = \sum_K x_K(\mathtt{rc}_K^{\pi} + \mathtt{rc}_K^{\pi'}) \geq \sum_K x_K(\mathtt{rc}_K^{\pi \wedge \pi'} + \mathtt{rc}_K^{\pi \vee \pi'}) \geq r(\pi \wedge \pi') + r(\pi \vee \pi') - 2 \geq r(\pi) + r(\pi') - 2$; thus the inequalities hold with equality, and the middle one shows $\pi \wedge \pi'$ and $\pi \vee \pi'$ are tight.

Any chain of distinct partitions of R that does not contain $\underline{\pi}$ has size at most $|R|-1$, and this is an upper bound on the rank of the system in (3). Elementary linear programming theory immediately yields the following corollary.

Corollary 1. *Any basic solution x^* of $(\mathcal{P})$ has at most $|R|-1$ non-zero coordinates.*

3 Equivalence of Formulations

In this section we describe our equivalence results. A summary of the known and new results is given in Figure 6.

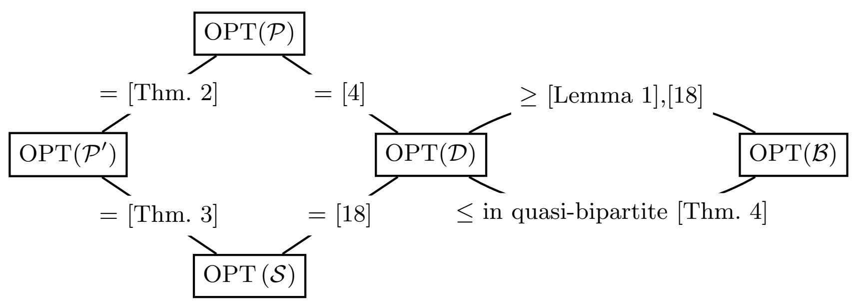

Fig. 6. Summary of relations among various LP relaxations

For lack of space, we present only sketches for our main equivalence results in this extended abstract, and refer the reader to [4] for details.

Theorem 2. *The LPs $(\mathcal{P}')$ and $(\mathcal{P})$ have the same optimal value.*

Proof sketch. To show this, it suffices to find an optimum solution of $(\mathcal{P})$ which satisfies the equality in $(\mathcal{P}')$; i.e., we want to find a solution for which the maximal-rank partition $\overline{\pi}$ is tight. We pick the optimum solution to $(\mathcal{P})$ which minimizes the sum $\sum_{K\in\mathcal{K}} x_K|K|$. Using Property 21, we show that either $\overline{\pi}$ is tight or there is a *shrinking* operation which decreases $\sum_{K\in\mathcal{K}} x_K|K|$ without increasing the cost. Since the latter is impossible, the theorem is proved.

Theorem 3. *The feasible regions of $(\mathcal{P}')$ and $(\mathcal{S})$ are the same.*

Proof sketch. We show that the inequalities defining $(\mathcal{P}')$ are valid for $(\mathcal{S})$, and vice-versa. Note that both have the same equality and non-negativity constraints. To show that the partition inequality of $(\mathcal{P}')$ for π holds for any $x \in (\mathcal{S})$, we use the subtour inequalities in $(\mathcal{S})$ for every part of π. For the other direction, given any subset $S \subseteq R$, we invoke the inequality in $(\mathcal{P}')$ for the partition $\pi := \{\{S\}$ as one part and the remaining terminals as singletons$\}$.

Theorem 4. *On quasibipartite Steiner tree instances,* $\mathrm{OPT}(\mathcal{B}) \geq \mathrm{OPT}(\mathcal{D})$.

Proof sketch. We look at the duals of the two LPs and we show $\mathrm{OPT}(\mathcal{B}_D) \geq \mathrm{OPT}(\mathcal{D}_D)$ in quasibipartite instances. Recall that the support of a solution to $(\mathcal{D}_D)$ is the family of sets with positive z_U. A family of sets is called *laminar* if for any two of its sets A, B we have $A \subseteq B, B \subseteq A$, or $A \cap B = \varnothing$. The following fact follows along the standard line of "set uncrossing" argumentation.

Lemma 2. *There is an optimal solution to* $(\mathcal{D}_D)$ *with laminar support.*

Given the above result, we may now assume that we have a solution z to $(\mathcal{D}_D)$ whose support is laminar. The heart of the proof of Theorem 4 is to show that z can be converted into a feasible solution to $(\mathcal{B}_D)$ of the same value.

Comparing $(\mathcal{D}_D)$ and $(\mathcal{B}_D)$ one first notes that the former has a variable for every valid subset of the terminals, while the latter assigns values to all valid subsets of the entire vertex set. We say that an edge uv is *satisfied* for a candidate solution z, if both a) $\sum_{U:u\in U, v\notin U} z_U \leq c_{uv}$ and b) $\sum_{U:v\in U, u\notin U} z_U \leq c_{uv}$ hold; z is then feasible for $(\mathcal{B}_D)$ if *all* edges are satisfied.

Let z be a feasible solution to $(\mathcal{D}_D)$. One easily verifies that all terminal-terminal edges are satisfied. On the other hand, terminal-Steiner edges may initially not be satisfied; e.g., consider the Steiner vertex v and its neighbours depicted in Figure 7 below. Initially, none of the sets in z's support contains v, and the load on the edges incident to v is quite *skewed*: the left-hand side of condition a) above may be large, while the left-hand side of condition b) is initially 0.

To construct a valid solution for $(\mathcal{B}_D)$, we therefore *lift* the initial value z_S of each terminal subset S to supersets of S, by adding Steiner vertices. The lifting procedure processes each Steiner vertex v one at a time; when processing v, we change z by moving dual from some sets U to $U \cup \{v\}$. Such a dual transfer decreases the left-hand side of condition a) for edge uv, and increases the (initially 0) left-hand sides of condition b) for edges connecting v to neighbours other than v.

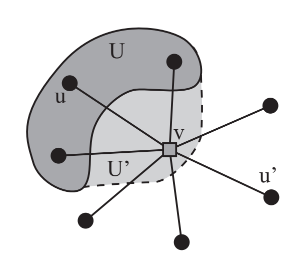

Fig. 7. Lifting variable z_U.

We are able to show that there is a way of carefully lifting duals around v that ensures that all edges incident to v become satisfied. The definition of our procedure will ensure that these edges remain satisfied for the rest of the lifting procedure. Since there are no Steiner-Steiner edges, all edges will be satisfied once all Steiner vertices are processed.

Throughout the lifting procedure, we will maintain that z remains unchanged, when projected to the terminals. The main consequence of this is that the objective value $\sum_{U\subseteq V} z_U$ remains constant throughout, and the objective value of z in $(\mathcal{B}_D)$ is not affected by the lifting. This yields Theorem 4.

4 Improved Integrality Gap Upper Bounds

In this extended abstract, we show the improved bound of 73/60 for uniformly quasibipartite graphs, and due to space restrictions, we only show the weaker $(2\sqrt{2}-1) \doteq 1.828$ upper bound on general graphs.

4.1 Uniformly Quasibipartite Instances

Uniformly quasibipartite instances of the Steiner tree problem are quasibipartite graphs where the cost of edges incident on a Steiner vertex are the same. They were first studied by Gröpl et al. [13], who gave a 73/60 factor approximation algorithm. We start by describing the algorithm of Gröpl et al. [13] in terms of full components. A collection $\mathcal{K}'$ of full components is acyclic if there is no list of $t > 1$ distinct terminals and hyperedges in $\mathcal{K}'$ of the form $r_1 \in K_1 \ni r_2 \in K_2 \cdots \ni r_t \in K_t \ni r_1$ — i.e. there are no *hypercycles.*

Procedure RatioGreedy

1: Initialize the set of acyclic components $\mathcal{L}$ to $\varnothing$.
2: Let L^* be a minimizer of $\frac{C_L}{|L|-1}$ over all full components L such that $|L| \geq 2$ and $L \cup \mathcal{L}$ is acyclic.
3: Add L^* to $\mathcal{L}$.
4: Continue until $(R, \mathcal{L})$ is a hyper-spanning tree and return $\mathcal{L}$.

Theorem 5. *On a uniformly quasibipartite instance* RatioGreedy *returns a Steiner tree of cost at most* $\frac{73}{60}\,\mathrm{OPT}\,(\mathcal{P})$.

Proof sketch. Let t denote the number of iterations and $\mathcal{L} := \{L_1, \ldots, L_t\}$ be the ordered sequence of full components obtained. We now define a dual solution y to $(\mathcal{P}_D)$. Let $\pi(i)$ denote the partition induced by the connected components of $\{L_1, \ldots, L_i\}$. Let $\theta(i)$ denote $C_{L_i}/(|L_i|-1)$ and note that θ is nondecreasing. Define $\theta(0) = 0$ for convenience. We define a dual solution y with

$$y_{\pi(i)} = \theta(i+1) - \theta(i)$$

for $0 \leq i < t$, and all other coordinates of y set to zero. It is straightforward to verify that the objective value $\sum_i y_{\pi(i)}(r(\pi(i)) - 1)$ of y in $(\mathcal{P}_D)$ equals $C(\mathcal{L})$.

The key is to show that for all $K \in \mathcal{K}$,

$$\sum_i y_{\pi(i)} \mathtt{rc}_K^{\pi(i)} \leq (|K| - 1 + H(|K|-1))/|K| \cdot C_K, \tag{4}$$

where H denotes the harmonic series; this is obtained by using the greedy nature of the algorithm and the fact that, in uniformly quasi-bipartite graphs, $C_{K'} \leq C_K \frac{|K'|}{|K|}$ whenever $K' \subset K$. Now, $(|K| - 1 + H(|K|-1))/|K|$ is always at most $\frac{73}{60}$. Thus (4) implies that $\frac{60}{73} \cdot y$ is a feasible dual solution, which completes the proof.

4.2 General graphs

For conciseness we let a "graph" be a triple $G = (V, E, R)$ where $R \subset V$ are G's terminals. In the following, we let $\mathtt{mtst}(G; c)$ denote the minimum *terminal spanning tree*, i.e. the minimum spanning tree of the terminal-induced subgraph $G[R]$ under edge-costs $c : E \to \mathbf{R}$. We will abuse notation and let $\mathtt{mtst}(G; c)$ mean both the tree and its cost under c.

When contracting an edge uv in a graph, the new merged node resulting from contraction is defined to be a terminal iff at least one of u or v was a terminal; this is natural since a Steiner tree in the new graph is a minimal set of edges which, together with uv, connects all terminals in the old graph. Our algorithm performs contraction, which may introduce parallel edges, but one may delete all but the cheapest edge from each parallel class without affecting the analysis.

Our algorithm proceeds in stages. In each stage we apply the operation $G \mapsto G/K$ which denotes contracting all edges in some full component K. To describe and analyze the algorithm we introduce some notation. For a minimum terminal spanning tree $T = \mathtt{mtst}(G; c)$ define $\mathtt{drop}_T(K; c) := c(T) - \mathtt{mtst}(G/K; c)$. We also define $\mathtt{gain}_T(K; c) := \mathtt{drop}_T(K) - c(K)$, where $c(K)$ is the cost of full component K. A tree T is called *gainless* if for every full component K we have $\mathtt{gain}_T(K; c) \leq 0$. The following useful fact is implicit in [15] (see also [4]).

Theorem 6 (Implicit in [15]). *If* $\mathtt{mtst}(G; c)$ *is gainless, then* $\mathrm{OPT}(\mathcal{P})$ *equals the cost of* $\mathtt{mtst}(G; c)$*.*

We now give the algorithm and its analysis, which uses a reduced cost trick introduced by Chakrabarty et al.[3].

Procedure Reduced One-Pass Heuristic

1: Define costs c'_e by $c'_e := c_e/\sqrt{2}$ for all terminal-terminal edges e, and $c'_e = c_e$ for all other edges. Let $G_1 := G$, $T_i := \mathtt{mtst}(G_i; c')$, and $i := 1$.
2: The algorithm considers the full components in any order. When we examine a full component K, if $\mathtt{gain}_{T_i}(K; c') > 0$, let $K_i := K$, $G_{i+1} := G_i/K_i$, $T_{i+1} := \mathtt{mtst}(G_{i+1}; c')$, and $i := i + 1$.
3: Let f be the final value of i. Return the tree $T_{alg} := T_f \cup \bigcup_{i=1}^{f-1} K_i$.

Note that the full components are scanned in *any* order and they are not examined a priori. Hence the algorithm works just as well if the full components arrive "online," which might be useful for some applications.

Theorem 7. $c(T_{alg}) \leq (2\sqrt{2} - 1)\,\mathrm{OPT}(\mathcal{P})$.

Proof. First we claim that $\mathtt{gain}_{T_f}(K; c') \leq 0$ for all K. To see this there are two cases. If $K = K_i$ for some i, then we immediately see that $\mathtt{drop}_{T_j}(K) = 0$ for all $j > i$ so $\mathtt{gain}_{T_f}(K) = -c(K) \leq 0$. Otherwise (if for all i, $K \neq K_i$) K had nonpositive gain when examined by the algorithm; and the well-known *contraction lemma* (e.g., see [12, §1.5]) immediately implies that $\mathtt{gain}_{T_i}(K)$ is nonincreasing in i, so $\mathtt{gain}_{T_f}(K) \leq 0$.

By Theorem 6, $c'(T_f)$ equals the value of $(\mathcal{P})$ on the graph G_f with costs c'. Since $c' \leq c$, and since at each step we only contract terminals, the value of this optimum must be at most $\mathrm{OPT}\,(\mathcal{P})$. Using the fact that $c(T_f) = \sqrt{2}c'(T_f)$, we get

$$c(T_f) = \sqrt{2}c'(T_f) \leq \sqrt{2}\,\mathrm{OPT}\,(\mathcal{P}) \tag{5}$$

Furthermore, for every i we have $\mathtt{gain}_{T_i}(K_i; c') > 0$, that is, $\mathtt{drop}_{T_i}(K_i; c') > c'(K) = c(K)$. The equality follows since K contains no terminal-terminal edges. However, $\mathtt{drop}_{T_i}(K_i; c') = \frac{1}{\sqrt{2}}\mathtt{drop}_{T_i}(K_i; c)$ because all edges of T_i are terminal-terminal. Thus, we get for every $i = 1$ to f, $\mathtt{drop}_{T_i}(K_i; c) > \sqrt{2} \cdot c(K_i)$.

Since $\mathtt{drop}_{T_i}(K_i; c) := \mathtt{mtst}(G_i; c) - \mathtt{mtst}(G_{i+1}; c)$, we have

$$\sum_{i=1}^{f-1} \mathtt{drop}_{T_i}(K_i; c) = \mathtt{mtst}(G; c) - c(T_f).$$

Thus, we have

$$\begin{aligned} \sum_{i=1}^{f-1} c(K_i) \leq \frac{1}{\sqrt{2}} \sum_{i=1}^{f} \mathtt{drop}_{T_i}(K_i; c) &= \frac{1}{\sqrt{2}}(\mathtt{mtst}(G; c) - c(T_f)) \\ &\leq \frac{1}{\sqrt{2}}(2\,\mathrm{OPT}\,(\mathcal{P}) - c(T_f)) \end{aligned}$$

where we use the fact that $\mathtt{mtst}(G, c)$ is at most twice $\mathrm{OPT}\,(\mathcal{P})$[5]. Therefore

$$c(T_{alg}) = c(T_f) + \sum_{i=1}^{f-1} c(K_i) \leq \Big(1 - \frac{1}{\sqrt{2}}\Big)c(T_f) + \sqrt{2}\,\mathrm{OPT}\,(\mathcal{P}).$$

Finally, using $c(T_f) \leq \sqrt{2}\,\mathrm{OPT}\,(\mathcal{P})$ from (5), the proof of Theorem 7 is complete.

References

1. A. Borchers and D. Du. The k-Steiner ratio in graphs. *SIAM J. Comput.*, 26(3):857–869, 1997.
2. J. Byrka, F. Grandoni, T. Rothvoß, and L. Sanità. An improved LP-based approximation for Steiner tree. To appear in Proc. 42nd STOC, 2010.
3. D. Chakrabarty, N. R. Devanur, and V. V. Vazirani. New geometry-inspired relaxations and algorithms for the metric Steiner tree problem. In *IPCO*, pages 344–358, 2008.
4. D. Chakrabarty, J. Könemann, and D. Pritchard. Hypergraphic LP relaxations for Steiner trees. Technical Report 0910.0281, arXiv, 2009.

[5] This follows using standard arguments, and can be seen, for instance, by applying Theorem 6 to the cost-function with all terminal-terminal costs divided by 2, and using short-cutting.

5. M. Chlebík and J. Chlebíková. Approximation hardness of the Steiner tree problem on graphs. In *Proceedings, Scandinavian Workshop on Algorithm Theory*, pages 170–179, 2002.
6. S. Chopra. On the spanning tree polyhedron. *Operations Research Letters*, 8:25–29, 1989.
7. J. Edmonds. Optimum branchings. *Journal of Research of the National Bureau of Standards B*, 71B:233–240, 1967.
8. J. Edmonds. Matroids and the greedy algorithm. *Math. Programming*, 1:127–136, 1971.
9. J. Edmonds and R. Giles. A min-max relation for submodular functions on graphs. *Annals of Discrete Mathematics*, 1:185–204, 1977.
10. M. X. Goemans. The Steiner tree polytope and related polyhedra. *Math. Program.*, 63(2):157–182, 1994.
11. M. X. Goemans and Y. Myung. A catalog of Steiner tree formulations. *Networks*, 23:19–28, 1993.
12. 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.
13. C. Gröpl, S. Hougardy, T. Nierhoff, and H. J. Prömel. Steiner trees in uniformly quasi-bipartite graphs. *Inform. Process. Lett.*, 83(4):195–200, 2002. Preliminary version appeared as a Technical Report at TU Berlin, 2001.
14. 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.
15. J. Könemann, D. Pritchard, and K. Tan. A partition-based relaxation for Steiner trees. *Math. Programming*, 2009. [In press].
16. T. Polzin. *Algorithms for the Steiner Problem in Networks*. PhD thesis, Universität des Saarlandes, February 2003.
17. T. Polzin and S. Vahdati Daneshmand. A comparison of Steiner tree relaxations. *Discrete Applied Mathematics*, 112(1-3):241–261, 2001. Preliminary version appeared at COS 1998.
18. T. Polzin and S. Vahdati Daneshmand. On Steiner trees and minimum spanning trees in hypergraphs. *Oper. Res. Lett.*, 31(1):12–20, 2003.
19. S. Rajagopalan and V. V. Vazirani. On the bidirected cut relaxation for the metric Steiner tree problem. In *Proceedings, ACM-SIAM Symposium on Discrete Algorithms*, pages 742–751, 1999.
20. G. Robins and A. Zelikovsky. Tighter bounds for graph Steiner tree approximation. *SIAM J. Discrete Math.*, 19(1):122–134, 2005. Preliminary version appeared as "Improved Steiner tree approximation in graphs" at SODA 2000.
21. M. Singh and L. C. Lau. Approximating minimum bounded degree spanning trees to within one of optimal. In *Proc. 39th STOC*, pages 661–670, 2007.
22. R. P. Stanley. *Enumerative Combinatorics*, volume 1. Wadsworth & Brooks/Cole, 1986.
23. V. Vazirani. Recent results on approximating the Steiner tree problem and its generalizations. *Theoret. Comput. Sci.*, 235(1):205–216, 2000.
24. D. Warme. *Spanning Trees in Hypergraphs with Applications to Steiner Trees*. PhD thesis, University of Virginia, 1998.
25. R. T. Wong. A dual ascent approach for Steiner tree problems on a directed graph. *Math. Programming*, 28:271–287, 1984.