Hypergraphic LP Relaxations for Steiner Trees*

Deeparnab Chakrabarty
University of Pennsylvania

Jochen Könemann
University of Waterloo†

David Pritchard
École Polytechnique Fédérale de Lausanne‡

August 24, 2011

Abstract

In this paper we prove new properties of hypergraphic linear programming relaxations for the Steiner tree problem. In particular, we show that a partition-based relaxation has the same value as other relaxations based on subtours and directed cuts. Additionally, we establish structural properties of basic solutions by using uncrossing methods. For quasibipartite instances we show that these hypergraphic relaxations have the same value as the well-studied graphic bidirected cut relaxation. We show how to analyze several approximation algorithms relative to the hypergraphic LPs; one gives an approximation ratio and integrality gap of at most $\sqrt{3} \simeq 1.729$ for the Steiner tree problem when the full components arrive online.

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 *Steiner tree*, a 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. Approximation algorithms are interesting for the Steiner tree problem since it is NP-hard. The best approximation algorithm for the Steiner tree problem is a recent result of Byrka, Grandoni, Rothvoß and Sanità [4], which for any fixed $\epsilon > 0$, achieves a performance ratio of $\ln(4)+\epsilon < 1.39$ in polynomial time. Their result is based on one of the linear programming (LP) relaxations we will later discuss. The best lower bound on the approximability is 96/95 (assuming P=NP), due to Chlebík and Chlebíková [8].

Numerous LP formulations are known for the Steiner tree problem (e.g., see [1, 10, 11, 13, 15, 22, 28, 29, 41, 43]), and they have led to impressive running time improvements for integer programming based methods. The *integrality gap* of a relaxation — a common measure of its strength — is the maximum ratio of the cost of integral and fractional optima, over all instances. Byrka et al. [4] prove an integrality gap upper bound of 1.55, breaking a long-standing barrier of $2 - o(1)$. The well-studied class of *quasibipartite* instances (e.g. [32, 24]) are those where Steiner vertices form an independent set; it is straightforward to extend [4] to give an integrality gap upper bound of 1.28 for this class. A simpler algorithm and proof is also known to yield both of these bounds [7], building on the framework of [4].

*Preliminary versions of this work appeared at IPCO 2010 and on the arXiv [6].

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

‡Partially supported by an NSERC post-doctoral fellowship.

A Steiner tree relaxation of particular interest is the *bidirected cut relaxation* [15, 43] (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. An integrality gap upper bound of 2 is known but it is a well-known open question to determine if this constant is tight, e.g. see [5, 32, 40]. The largest known lower bound on the integrality gap is $\frac{36}{31}$ [4]; for quasibipartite instances the integrality gap is so far known to be between $\frac{8}{7}$ (due to M. Skutella but reported in [27]) and $\frac{4}{3}$ [5].

In this paper we study a class of formulations 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 1.2.2) of a certain hypergraph on the terminals. They are known to be stronger than the bidirected cut relaxation [30].

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 show that the *uncrossing* technique applies to the partition formulation [27] of the hypergraphic LP, and we use it to prove structural properties. For example, 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 [41], and a *directed hypergraph relaxation* of Polzin and Vahdati Daneshmand [30] (also used by Byrka et al. [4]); these two are known to be equivalent [30]. For each of the two relaxations we prove it is *equivalent to the partition LP* (that is, they have the same objective value for any Steiner tree instance). By [30] these are two alternate proofs of the same fact, but we give both for completeness and to highlight the variety of techniques: one uses partition uncrossing, and the other uses hypergraph orientation results of Frank et al. [18].

We also show that, on *quasibipartite instances*, the hypergraphic and the bidirected cut LP relaxations are equivalent. By the 1.28 integrality gap bound [4] mentioned earlier, this improves the integrality gap of bidirected cut on quasibipartite instances from 4/3 to 1.28. We find this equivalence surprising for the following reasons. Firstly, some instances are known where the hypergraph relaxations are *strictly* stronger than the bidirected cut relaxation [30]. Secondly, the bidirected cut relaxations seems to resist uncrossing techniques; e.g. even in quasibipartite graphs extreme points for bidirected cut can have as many as $\Omega(|V|^2)$ positive variables [31, Sec. 4.9]. Thirdly, the known approaches to exploiting the bidirected cut relaxation (mostly primal-dual and local search algorithms [32, 5]) are very different from the combinatorial hypergraphic algorithms for the Steiner tree problem (almost all of them employ greedy strategies). In short, there is 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 both 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 $73/60 \simeq 1.216$. Our proof uses the approximation algorithm of Gröpl et al. [24] 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 Steiner tree instances where the best currently known approximation ratio and integrality gap upper bound are the same. (Interestingly, an approximation ratio of $73/60 + \epsilon$ can be obtained for *all* quasibipartite instances [4].)

For general graphs, we give an approximation algorithm with ratio $\sqrt{3} \simeq 1.729$ that also proves a matching integrality gap. (This is preceded by a slightly simpler version giving ratio $2\sqrt{2}-1 \simeq 1.83$.) Neither the approximation ratio nor the integrality gap is as good as what is already known [4], however the proof reveals a couple of interesting techniques worth reporting. Call a graph *gainless* if the minimum *spanning* tree of the terminals is also an optimal *Steiner* tree. To obtain these integrality gap upper bounds, we use the following key property of the hypergraphic relaxation which was implicit in [27]: on gainless instances, the LP value equals the minimum spanning tree cost, so the integrality gap is 1. Such a theorem was known for quasibipartite instances and the bidirected cut relaxation (implicitly in [32], explicitly in [5]). Our algorithms, extending ideas of [5], iterate through all of the full components in any order, and commit to include or exclude each one in the final solution as they are examined. In this sense the algorithm fits in paradigm of online algorithms.

1.2 Bidirected Cut and Hypergraphic Relaxations

1.2.1 The Bidirected Cut Relaxation

The first bidirected LP was given by Edmonds [15] as an exact formulation for the spanning tree problem. Wong [43] 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 $\mathrm{valid}(V)$ 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, $\delta^{\mathrm{out}}(U)$ 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 [15] implies this relaxation is exact.

$$\min \sum_{a \in A} c_a x_a : \quad x \in \mathbf{R}^A_{\geq 0} \qquad (\mathcal{B})$$

$$\sum_{a \in \delta^{\mathrm{out}}(U)} x_a \geq 1, \quad \forall U \in \mathrm{valid}(V) \qquad (1)$$

$$\max \sum_{U} z_U : \quad z \in \mathbf{R}^{\mathrm{valid}(V)}_{\geq 0} \qquad (\mathcal{B}_D)$$

$$\sum_{U : a \in \delta^{\mathrm{out}}(U)} z_U \leq c_a, \quad \forall a \in A \qquad (2)$$

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

Goemans & Myung [22] 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 [21] showed that if the graph is series-parallel, the relaxation is exact. Chakrabarty et al. give an alternate "embedding" formulation for ($\mathcal{B}$) in [5].

1.2.2 Hypergraphic Relaxations

We follow the *full component* approach used in virtually all recent Steiner tree algorithms, which leads naturally to hypergraphs where we identify full components as hyperedges. 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. For any Steiner tree (assuming without loss of generality that it has no Steiner node of degree 1) its edge set can be partitioned in a *unique* way into full components by splitting at internal terminals; see Figure 2(i)–(ii) for an example.

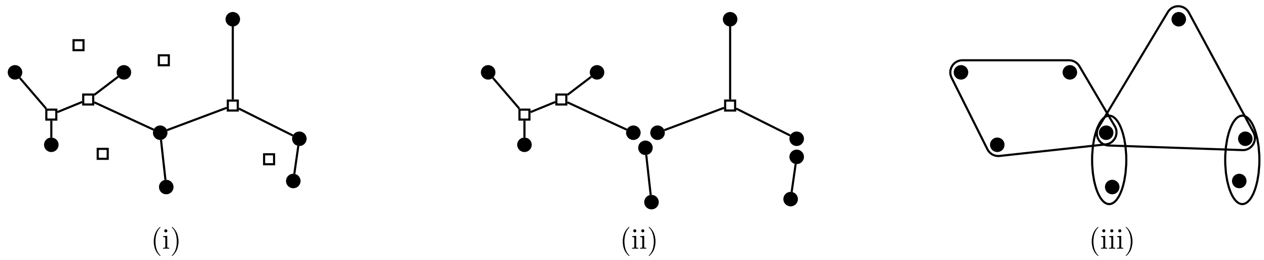

Figure 2: Black nodes are terminals and white nodes are Steiner nodes. (i): a Steiner tree for this instance. (ii): the Steiner tree's edges are partitioned into full components; there are four full components. (iii): the hyperedges corresponding to these full components form a hyper-spanning tree.

Fix an integer r, needed for technical reasons that we explain presently. Let $\mathcal{K}$ be the set of all nonempty subsets of terminals — *hyperedges* — which have size at most r. The resulting number $|\mathcal{K}|$ of full components is bounded by $|R|^r$, which is polynomial. We associate with each $K \in \mathcal{K}$ a minimum-cost full component spanning the terminals set K, and let C_K be its cost (if no full component spans K we let C_K be infinity). By [14] these full components and C_K values can be found in polynomial time for any constant r. The central observation in this approach is that a collection of full components forms a Steiner tree iff its corresponding terminal sets form a hypergraph in which each pair of terminals are joined by a unique hyper-path, a so-called *hyper-spanning tree*; see Figure 2(iii) for an example. The Steiner tree problem thus reduces to that of finding a minimum-cost hyper-spanning tree in the hypergraph $(R, \mathcal{K})$. Constraining all full components of a Steiner tree to have size at most r may increase the optimal cost, but provably by at most a $(1+\Theta(1/\log r))$ factor [3] (the *r-Steiner ratio*), so this effect is negligible for approximation purposes as r grows.

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 $K \in \mathcal{K}$ up to some fixed size, 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 [41] which is inspired by Edmonds' subtour elimination LP relaxation [16] 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 as $(\mathcal{S})$, is called the *subtour elimination* LP relaxation.

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

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 [30], and inspired by the bidirected cut relaxation. It was independently rediscovered in the work of Byrka et al. [4], where they use it to give a 1.39-approximation algorithm based on iterated randomized rounding. 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 a subset $U\subseteq R$ of terminals is called *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 set of all directed hyperedges. We show the directed hypergraph relaxation and its dual in Figure 3.

$$\min\Big\{\sum_{K\in\mathcal{K}, i\in K} C_K x_{K^i} : x\in\mathbf{R}^{\overrightarrow{\mathcal{K}}}_{\geq 0} \qquad (\mathcal{D})$$
$$\sum_{K^i\in\Delta^{\text{out}}(U)} x_{K^i}\geq 1,\quad \forall \text{ valid } \ U\subseteq R\Big\} \qquad (3)$$

$$\max\Big\{\sum_{U} z_U : \qquad z\in\mathbf{R}^{\text{valid}(R)}_{\geq 0} \qquad (\mathcal{D}_D)$$
$$\sum_{U:K\cap U\neq\varnothing, i\notin U} z_U\leq C_K,\quad \forall K\in\mathcal{K},\forall i\in K\Big\} \qquad (4)$$

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

Polzin & Vahdati Daneshmand [30] showed that $\text{OPT}(\mathcal{D})=\text{OPT}(\mathcal{S})$. Moreover they observed that this directed hypergraphic relaxation strengthens the bidirected cut relaxation.

Lemma 1.1 ([30])**.** *For any instance,* $\text{OPT}(\mathcal{D})\geq\text{OPT}(\mathcal{B})$.

Proof sketch. It suffices to show that any solution x of ($\mathcal{D}$) can be converted to a feasible solution x' of ($\mathcal{B}$) of the same cost. For each arc a, let x'_a be the sum of x_{K^i} over all directed full components K^i that (when viewed as an arborescence) contain a. Now for any valid subset U of V, it is not hard to see that every directed full component leaving $R\cap U$ has at least one arc leaving U, hence $\sum_{a\in\delta^{\text{out}}(U)} x'_a\geq\sum_{K^i\in\Delta^{\text{out}}(R\cap U)} x_{K^i}\geq 1$ and x' is feasible as needed. ∎

See [30] for an example where the strict inequality $\text{OPT}(\mathcal{D})>\text{OPT}(\mathcal{B})$ holds.

Könemann et al. [27], inspired by the work of Chopra [9], showed that the Robins-Zelikovsky algorithm [33] could be viewed through the lens of iterated primal-dual analysis. They described a partition-based relaxation which captures the fact that for any partition of the terminals, any hyper-spanning tree must have sufficiently many "cross hyperedges." To formally describe the LP, we say 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 is shown in Figure 4. (Technically, in [27] this LP was used for the *first* iteration, and modified in each further iteration.)

$$\min \Big\{ \sum_{K \in \mathcal{K}} C_K x_K : \quad x \in \mathbf{R}_{\geq 0}^{\mathcal{K}} \qquad (\mathcal{P})$$

$$\sum_{K \in \mathcal{K}} x_K \mathtt{rc}_K^\pi \geq r(\pi) - 1, \quad \forall \pi \in \Pi_R \Big\} \qquad (5)$$

$$\max \Big\{ \sum_{\pi} (r(\pi) - 1) \cdot y_\pi : \quad y \in \mathbf{R}_{\geq 0}^{\Pi_R} \qquad (\mathcal{P}_D)$$

$$\sum_{\pi \in \Pi_R} y_\pi \mathtt{rc}_K^\pi \leq C_K, \quad \forall K \in \mathcal{K} \Big\} \qquad (6)$$

Figure 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 an equality constraint that holds for all hyper-spanning trees to the LP.

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

1.2.3 Complexity of Solving the LPs

The bidirected cut relaxation is very attractive from a perspective of computational implementation. Although the formulation given in Section 1.2.1 has an exponential number of constraints, an equivalent compact flow formulation with $O(|E||R|)$ variables and constraints is well-known.

The hypergraphic LPs can be solved in polynomial time: using a separation oracle, Warme showed [41] that for any chosen family $\mathcal{K}$ of full components, the subtour LP can be optimized in time poly$(|V|, |\mathcal{K}|)$. Note that to get relative error $1+\epsilon$, considering the r-Steiner ratio, this LP has $|R|^{\exp(\Theta(1/\epsilon))}$ variables. The ellipsoid algorithm can be circumvented, by taking ($\mathcal{D}$) and rewriting its cut constraints as a compact flow formulation, which has $|R|^{\exp(\Theta(1/\epsilon))}$ variables and constraints.

1.2.4 Other Related Work

Suppose that we want to find a spanning connected hypersubgraph in a hypergraph with *arbitrary* costs, where every hyperedge has size at most r. This is a generalization of the hypergraphic spanning tree approach to the Steiner tree problem. An LP-based approximation factor equal to the harmonic number $H(r - 1)$ is known for this problem, implicitly in [42], and explicitly in [2]. (One may use [42, Thm. 1(iv)] as follows. For a hyperedge E let K_E be the complete graph on vertex set E. Let $f(\mathcal{E})$ be the graphic matroid rank of $\bigcup_{E \in \mathcal{E}} K_E$. Then f is submodular and maximizers correspond to spanning connected hypersubgraphs.) This approximation ratio is also nearly best possible: a lower bound of $\ln(r - 1) - O(\ln \ln r)$ follows by a reduction from $(r - 1)$-set cover [38] when $r \geq 4$.

In the special case $r = 3$, the partition hypergraphic LP is essentially a special case of an LP introduced by Vande Vate [39] for matroid matching, which is totally dual half-integral [20]. Additional formulations and properties about hypergraphic LPs appear in the third author's thesis [31], such as an equivalent *gainless tree* formulation similar to the "1-tree" formulation for the Held-Karp TSP relaxation.

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 from combinatorial lattice theory.

Definition 2.1. *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 $\pi^1, \pi^2, \ldots, \pi^t$ such that π^i refines π^{i-1} for each $1 < i \leq t$.*

The family Π_R of all partitions of R has a *meet operator* $\wedge : \Pi_R^2 \to \Pi_R$ and a *join operator* $\vee : \Pi_R^2 \to \Pi_R$, defined as follows and illustrated in Figure 5. The meet refines both operands, and the join coarsens both operands.

Definition 2.2 (Meet of partitions)**.** *Let the parts of π be $\pi_1, \ldots, \pi_t$ and let the parts of π' be $\pi'_1, \ldots, \pi'_u$. Then the parts of the meet $\pi \wedge \pi'$ are the nonempty intersections of parts of π with parts of π',*

$$\pi \wedge \pi' = (\pi_i \cap \pi'_j \,:\, 1 \leq i \leq t, 1 \leq j \leq u \text{ and } \pi_i \cap \pi'_j \neq \varnothing).$$

Informally, whereas the meet $\wedge$ corresponds to intersecting parts, the *join* $\vee$ corresponds to the transitive closure of the union of parts. Formally, given a graph G and a partition π of $V(G)$, say that G *induces* π if the parts of π are the vertex sets of the connected components of G.

Definition 2.3 (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 the join $\pi \vee \pi'$.*

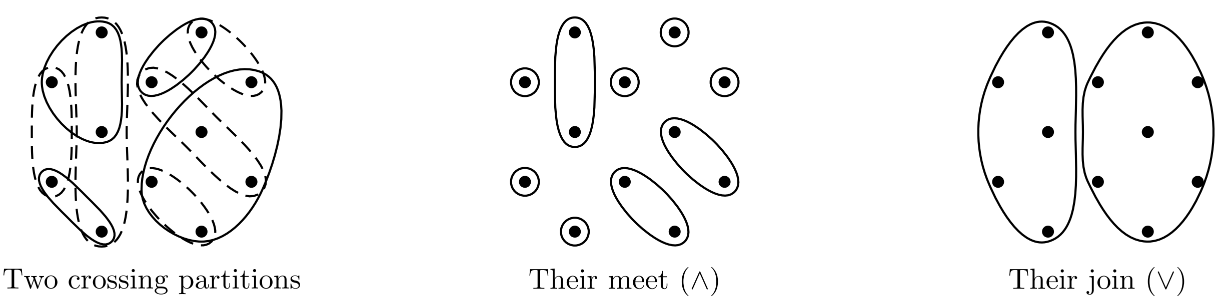

Figure 5: Several partitions of the terminal set R (the black dots).

It it straightforward to see $\vee$ and $\wedge$ are both symmetric and associative. Moreover, these operators form a *lattice* under refinement [37], meaning the following.

Fact 2.4. *For any partitions π, π', σ, the meet $\pi \wedge \pi'$ coarsens σ iff both π and π' coarsen σ. Likewise, $\pi \vee \pi'$ refines σ iff both π and π' refine σ.*

Given a feasible solution x to ($\mathcal{P}$), a partition π is *tight* if $\sum_{K \in \mathcal{K}} x_K \mathtt{rc}_K^\pi = r(\pi) - 1$, i.e. if the partition inequality (5) holds with equality. 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 2.5. *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 [12, 17, 25, 36] 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 "submodularity" of the coefficients in the LHS. In particular, if the following hypothetical inequalities were true,

$$(?) \quad \forall \pi, \pi' : \; r(\pi \vee \pi') + r(\pi \wedge \pi') \quad \geq \quad r(\pi) + r(\pi') \tag{7}$$

$$(?) \quad \forall K, \pi, \pi' : \; \mathtt{rc}_K^{\pi} + \mathtt{rc}_K^{\pi'} \quad \geq \quad \mathtt{rc}_K^{\pi \vee \pi'} + \mathtt{rc}_K^{\pi \wedge \pi'} \tag{8}$$

then Property 2.5 can be proved easily by writing a string of inequalities.[1]

Inequality (7) is indeed true (see, for example, [37]), but unfortunately inequality (8) is not true in general, as the following example shows; here we use $12\cdots$ as an abbreviation for $\{1, 2, \ldots\}$.

Example 2.6. *Let $R = \{1, 2, 3, 4\}$, $\pi = (12, 34)$ and $\pi' = (13, 24)$. 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 2.5 is true; its correct proof is given in Section 2.1. Given tight partitions π and π', we apply a procedure of Schrijver [35, §48.2,49.6] to obtain two other tight partitions. Applying this procedure iteratively leads us to deduce that the joins and meets are tight. We use this to obtain the following theorem, whose full proof appears in Section 2.2:

Theorem 1. *Let x^* be a basic feasible solution of ($\mathcal{P}$), let $\mathcal{C}$ be an inclusion-wise maximal chain in $\mathtt{tight}(x^*)$, and let $\mathcal{Z} = \{K \mid x_K^* = 0\}$. Then x^* is the unique solution to the linear system*

$$x_K^* = 0 \quad \forall K \in \mathcal{Z}; \qquad\qquad \sum_{K \in \mathcal{K}} \mathtt{rc}_K^{\pi} x_K^* = r(\pi) - 1 \quad \forall \pi \in \mathcal{C}. \tag{9}$$

Corollary 2.7. *Any basic solution x^* of ($\mathcal{P}$) has at most $|R| - 1$ non-zero coordinates.*

To prove this, we let $\underline{\pi}$ denote $\{R\}$, the unique partition with minimal rank 1; later we use $\overline{\pi}$ to denote $\{\{r\} : r \in R\}$, the unique partition with maximal rank $|R|$.

Proof. First, note that when $\pi = \underline{\pi}$, the partition constraint $\sum_{K \in \mathcal{K}} \mathtt{rc}_K^{\pi} x_K^* = r(\pi) - 1$ is vacuous (all terms are zero). The remaining partition constraints correspond to the partitions $\mathcal{C} \backslash \{\underline{\pi}\}$. There are at most $|R| - 1$ such partitions, since their ranks are integers between 2 and $|R|$, and the ranks are distinct since $\mathcal{C}$ is a chain. Finally, the size of $\{K \mid K \notin \mathcal{Z}\}$ must be at most $|\mathcal{C} \backslash \{\underline{\pi}\}|$ or else the system would have more free variables than partition constraints, contradicting that x^* is the unique solution. $\square$

2.1 Partition Uncrossing

In this section we prove Property 2.5. We need the following lemma that relates the rank of sets and the rank contribution of partitions. Recall $\rho(X) := \max(0, |X| - 1)$.

Lemma 2.8. *For a partition $\pi = (\pi_1, \ldots, \pi_t)$ of R, where $t = r(\pi)$, and for any $K \subseteq R$, we have*

$$\mathtt{rc}_K^{\pi} = \rho(K) - \sum_{i=1}^{t} \rho(K \cap \pi_i).$$

[1] 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.

Proof. By definition, $K \cap \pi_i \neq \varnothing$ for exactly $1 + \mathtt{rc}_K^\pi$ values of i. Also, $\rho(K \cap \pi_i) = 0$ for all other i. Hence

$$\sum_{i=1}^{t} \rho(K \cap \pi_i) = \sum_{i:K\cap\pi_i\neq\varnothing} (|K \cap \pi_i| - 1) = \left(\sum_{i:K\cap\pi_i\neq\varnothing} |K \cap \pi_i| \right) - (\mathtt{rc}_K^\pi + 1). \tag{10}$$

Observe that $\sum_{i:K\cap\pi_i\neq\varnothing} |K \cap \pi_i| = |K| = \rho(K) + 1$; using this fact together with Equation (10) we obtain

$$\sum_{i=1}^{t} \rho(K \cap \pi_i) = \left(\sum_{i:K\cap\pi_i\neq\varnothing} |K \cap \pi_i| \right) - (\mathtt{rc}_K^\pi + 1) = \rho(K) + 1 - (\mathtt{rc}_K^\pi + 1).$$

Rearranging, the proof of Lemma 2.8 is complete. □

Now we describe a procedure of Schrijver [35, §48.2,49.6] that takes as input two tight partitions $\pi = (\pi_1, \dots, \pi_p)$, and $\pi' = (\pi'_1, \dots, \pi'_q)$ of the ground set R. Schrijver's algorithm computes two new tight partitions ϕ and γ, no two of whose parts *cross*. Here two sets A and B *cross* if $A \cap B, A^c \cap B^c, A \cap B^c, A^c \cap B$ are all nonempty (A^c and B^c are the complements of A and B, respectively).

Schrijver's algorithm first computes the set multi-family

$$\pi \uplus \pi' := \{\pi_1, \dots, \pi_p, \pi'_1, \dots, \pi'_q\},$$

that contains the parts of π and π' possibly with multiplicity. This set multi-family is *2-regular*, meaning that every $v \in R$ lies in exactly two sets of the multi-family. The algorithm then applies *uncrossing* steps as long as this family contains a pair of crossing sets. 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 $A \cap B = \varnothing$. After the uncrossing phase ends, we are left with a laminar 2-regular family. It is well-known (e.g. [18, Lemma 1.3]) that such families decompose into partitions: in this example it is not hard to see the inclusion-maximal sets form a partition and that its complement is also a partition, consisting of inclusion-minimal sets. We output these two partitions.

Procedure Uncross-and-Decompose(π, π')

1: Create a 2-regular family $\mathcal{F} := \pi \uplus \pi'$.
2: As long as $\mathcal{F}$ has two crossing sets S and T, replace them with $S \cap T$ and $S \cup T$.
3: Decompose the 2-regular laminar $\mathcal{F}$ into two partitions ϕ, γ with ϕ refining γ. (I.e., for any x, of the two sets of $\mathcal{F}$ containing x, ϕ gets the smaller and γ the larger.)
4: Output ϕ and γ.

The following example shows that the output of the above algorithm depends on the choice of sets to uncross.

Example 2.9. *Consider input partitions* $\pi = (12, 34)$ *and* $\pi' = (13, 24)$*. Initially* $\mathcal{F} = \{12, 34, 13, 24\}$*. Say that in the first iteration we uncross* 12 *and* 13*, yielding* $\{123, 1, 34, 24\}$*. Then in the second iteration, two of the possible uncrossing operations are: (a) uncross* 123 *and* 34*, yielding* $\mathcal{F} = \{1234, 1, 3, 24\}, \phi = (1, 24, 3), \gamma = (1234)$*; or (b) uncross* 123 *and* 24*, yielding* $\mathcal{F} = \{1234, 1, 2, 34\}, \phi = (1, 34, 2), \gamma = (1234)$*. We remark in either case that* γ *is the join* $\pi \vee \pi'$ *of the two input partitions, but* ϕ *is* not *the meet* $\pi \wedge \pi'$*.*

Regardless of the choices, we make the following observations.

(i) The algorithm terminates. To see this observe that $\sum_{S\in\mathcal{F}} |S|^2$ increases in each iteration.

(ii) Both ϕ and γ coarsen $\pi \wedge \pi'$. To see this, suppose u, v are in the same part of π and also in the same part of π'; then all sets of $\mathcal{F}$ containing u also contain v throughout the algorithm.

(iii) We have that γ equals $\pi \vee \pi'$. First, consider u and v which are in the same part of π, and observe that $\mathcal{F}$ always has some set containing both u and v. So γ coarsens π and symmetrically γ coarsens π'; by Fact 2.4 γ coarsens $\pi \vee \pi'$. Second, note that all sets in $\mathcal{F}$ always are subsets of some part of $\pi \vee \pi'$, so γ refines $\pi \vee \pi'$.

(iv) We have that $r(\pi)+r(\pi') = r(\phi)+r(\gamma)$. To see this, note that the first is the initial cardinality of $\mathcal{F}$ and the last is the final cardinality of $\mathcal{F}$, and that this cardinality is invariant.

(v) If π and π' cross, then $r(\phi) > \max\{r(\pi), r(\pi')\}$. To see this, first note that $\pi \vee \pi'$ is not equal to π or π' when they cross, and since it coarsens both, $r(\pi \vee \pi') < \min\{r(\pi), r(\pi')\}$. Then use (iii), (iv), and the fact that $p + q = \min\{p, q\} + \max\{p, q\}$ for all p, q.

(vi) For any K, $\mathtt{rc}_K^{\pi} + \mathtt{rc}_K^{\pi'} \geq \mathtt{rc}_K^{\phi} + \mathtt{rc}_K^{\gamma}$. To see this, it suffices from Lemma 2.8 to show $\sum_{P \in \phi \uplus \gamma} \rho(K \cap P) \geq \sum_{P \in \pi \uplus \pi'} \rho(K \cap P)$, which follows from the supermodularity of the $\rho()$ function.

Proposition 2.10. *Let x be feasible for ($\mathcal{P}$) and let π, π' be tight for x. Then the outputs ϕ, γ of* Uncross-and-Decompose(π, π') *are tight for x.*

Proof. We have

$$r(\pi) + r(\pi') - 2 = \sum_K x_K(\mathtt{rc}_K^{\pi} + \mathtt{rc}_K^{\pi'}) \geq \sum_K x_K(\mathtt{rc}_K^{\phi} + \mathtt{rc}_K^{\gamma}) \geq r(\phi) + r(\gamma) - 2 \tag{11}$$

where the first inequality follows from (vi), and the second uses the feasibility of x for ($\mathcal{P}$). By (iv) we have equality throughout, so ϕ and γ are tight. ☐

Now we prove that tight partitions are closed under meet and join.

Proof of Property 2.5. Proposition 2.10 and (iii) immediately give that tight partitions are closed under the join $\vee$. Suppose for the sake of contradiction there exist tight partitions π and π' such that their meet $\pi \wedge \pi'$ is not tight; pick such a pair with $r(\pi)+r(\pi')$ maximal. Let ϕ be the partition obtained via the above procedure. Note that π and π' cross, so by (v), $r(\phi) > \max\{r(\pi), r(\pi')\}$. Due to choosing π, π' with maximal rank-sum, we know $\phi \wedge \pi$ is tight (since ϕ is tight), and $(\phi \wedge \pi) \wedge \pi'$ is tight. However, the latter partition equals $(\pi \wedge \pi') \wedge \phi$ which equals $\pi \wedge \pi'$ by (ii). So $\pi \wedge \pi'$ is tight. ☐

In previous versions of this work [6] we gave an alternate "direct" partition uncrossing methodology: rather than uncrossing pairs of sets, it always worked with partitions.

2.2 Structure of Basic Solutions: Proof of Theorem 1

Proof. Let $\mathtt{supp}(x^*)$ be the set of full components K with $x_K^* > 0$ (i.e. the complement of $\mathcal{Z}$). Consider the constraint submatrix of ($\mathcal{P}$) with rows corresponding to the tight partitions and columns corresponding to the full components in $\mathtt{supp}(x^*)$. Since x^* is a basic feasible solution, any full-rank subset of rows uniquely defines x^*. We now show that any maximal chain $\mathcal{C}$ in $\mathtt{tight}(x^*)$ corresponds to such a subset.

Let $\mathtt{row}(\pi) \in \mathbf{R}^{\mathtt{supp}(x^*)}$ denote the row corresponding to partition π of this matrix, i.e., $\mathtt{row}(\pi)_K = \mathtt{rc}_K^{\pi}$, and given a collection $\mathcal{R}$ of partitions (rows), let $\mathtt{span}(\mathcal{R})$ denote the linear span of the rows in

$\mathcal{R}$. We now prove that for any tight partition $\pi \notin \mathcal{C}$, we have $\mathtt{row}(\pi) \in \mathtt{span}(\mathcal{C})$; this will complete the proof of the theorem.

For sake of contradiction, suppose $\mathtt{row}(\pi) \notin \mathtt{span}(\mathcal{C})$ and choose such a tight π with $r(\pi)$ maximal. Then, since $\mathcal{C}$ is inclusion-maximal, π must cross some partition σ in $\mathcal{C}$; let σ be a partition in $\mathcal{C}$ which crosses π, with $r(\sigma)$ minimal. Run UNCROSS-AND-DECOMPOSE(π, σ), obtaining outputs ϕ, γ where $\gamma = \pi \vee \sigma$.

Claim 2.11. *We have* $\mathtt{row}(\pi) + \mathtt{row}(\sigma) = \mathtt{row}(\phi) + \mathtt{row}(\gamma)$.

Proof. Due to (v), it is enough to show that there is no column whose corresponding K has $\mathtt{rc}_K^{\pi} + \mathtt{rc}_K^{\sigma} > \mathtt{rc}_K^{\phi} + \mathtt{rc}_K^{\gamma}$. Then note, such a K has $x_K > 0$, and contradicts the fact that (11) holds with equality (as remarked in the proof of Proposition 2.10). ☐

Since $\mathtt{row}(\pi) \notin \mathtt{span}(\mathcal{C})$ and $\mathtt{row}(\sigma) \in \mathtt{span}(\mathcal{C})$, Claim 2.11 puts us in one of the following two cases.

Case 1: $\mathtt{row}(\phi) \notin \mathtt{span}(\mathcal{C})$. This contradicts our choice of π since we could have picked the tight partition ϕ instead and $r(\phi) > r(\pi)$.

Case 2: $\mathtt{row}(\gamma) \notin \mathtt{span}(\mathcal{C})$. By the maximality of $\mathcal{C}$, there is a partition $\sigma' \in \mathcal{C}$ which crosses $\gamma = \pi \vee \sigma$. Since σ, σ' are in a chain they do not cross; σ' does not refine σ since otherwise $\pi \vee \sigma$ coarsens σ which coarsens σ', contradicting that $\pi \vee \sigma$ and σ' cross; so σ refines σ'. Moreover since $\sigma \neq \sigma'$, $r(\sigma') < r(\sigma)$. Now we claim π and σ' cross: on the one hand, if π coarsens σ' then π coarsens σ whereas we know the latter pair cross; on the other hand, if π refines σ' then by Fact 2.4, $\pi \vee \sigma$ refines σ' whereas we know the latter pair cross. Thus σ' contradicts our choice of σ.

This completes the proof of Theorem 1. ☐

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.

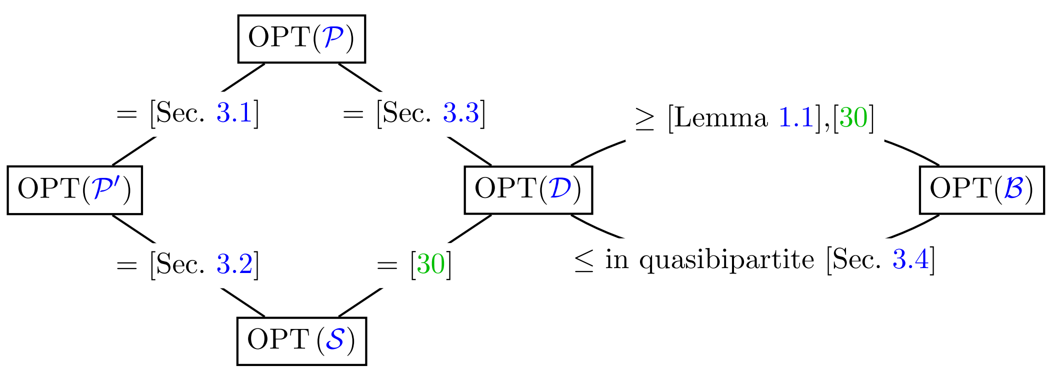

Figure 6: Summary of relations among various LP relaxations

3.1 Bounded and Unbounded Partition Relaxations

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

We actually prove a stronger statement.

Definition 3.1. *The collection $\mathcal{K}$ of hyperedges is* down-closed *if whenever $S \in \mathcal{K}$ and $\varnothing \neq T \subset S$, then $T \in \mathcal{K}$. For down-closed $\mathcal{K}$, the cost function $C : \mathcal{K} \to \mathbf{R}_+$ is* non-decreasing *if $C_S \leq C_T$ whenever $S \subset T$.*

Theorem 3. *If the set of hyperedges is down-closed and the cost function is non-decreasing, then ($\mathcal{P}'$) and ($\mathcal{P}$) have the same optimal value.*

Theorem 3 implies Theorem 2 since the hypergraph and cost function derived from instances of the Steiner tree problem are down-closed and non-decreasing (e.g. $C_{\{k\}} = 0$ for every $k \in R$; we remark that the variables $x_{\{k\}}$ act just as placeholders). Our proof of Theorem 2 relies on the following operation which we call *shrinking*.

Definition 3.2. *Given an assignment $x : \mathcal{K} \to \mathbf{R}_+$ to the full components, suppose $x_K > 0$ for some K. The operation* $\texttt{Shrink}(x, K, K', \delta)$*, where $K' \subseteq K$, $|K'| = |K| - 1$ and $0 < \delta \leq x_K$, changes x to x' by decreasing $x'_K := x_K - \delta$ and increasing $x'_{K'} := x_{K'} + \delta$.*

Note that shrinking is well-defined for down-closed hypergraphs. Also note that on performing a shrinking operation, the cost of the solution cannot increase, if the cost function is non-decreasing. The theorem is proved by taking the optimum solution to ($\mathcal{P}$) which minimizes the sum $\sum_{K \in \mathcal{K}} x_K |K|$, and then showing that this must satisfy the equality in ($\mathcal{P}'$), or a shrinking operation can be performed. Now we give the details.

Proof of Theorem 3. It suffices to exhibit an optimum solution of ($\mathcal{P}$) which satisfies the equality in ($\mathcal{P}'$). Let x be an optimal solution to ($\mathcal{P}$) which minimizes the sum $\sum_{K \in \mathcal{K}} x_K |K|$. Say that a partition π *isolates r from K* if the part of π containing r does not contain any element of $K \backslash \{r\}$.

Claim 3.3. *For every K with $x_K > 0$ and for every $r \in K$, there exists a tight (w.r.t. x) partition π that isolates r from K.*

Proof. Let $K' = K \setminus \{r\}$. Pick δ maximal such that $\texttt{Shrink}(x, K, K', \delta)$ is feasible for ($\mathcal{P}$). Observe that for a partition π, if it isolates r from K then $\texttt{rc}^\pi_{K'} = \texttt{rc}^\pi_K - 1$, and otherwise $\texttt{rc}^\pi_{K'} = \texttt{rc}^\pi_K$. Therefore, δ is given by the explicit formula

$$\delta := \min\{x_K, \min_{\pi : \pi \text{ isolates } r \text{ from } K} \big(\sum_L \texttt{rc}^\pi_L x_L - r(\pi) + 1\big)\}$$

(note the term in parentheses is the slack for x in the constraint for partition π).

Since we chose x with minimal $\sum_K |K| x_K$, no shrinking can be feasible, so $\delta = 0$. But as $x_K > 0$, this means some isolating partition is tight. $\square$

Let $\texttt{tight}(x)$ be the set of tight partitions, and $\pi^* := \bigwedge\{\pi : \pi \in \texttt{tight}(x)\}$ the meet of all tight partitions. By Property 2.5, π^* is tight. By Claim 3.3, for any K with $x_K > 0$, we have $\texttt{rc}^{\pi^*}_K = |K| - 1$. Thus, $r(\pi^*) - 1 = \sum_{K \in \mathcal{K}} x_K \texttt{rc}^{\pi^*}_K = \sum_{K \in \mathcal{K}} x_K (|K| - 1) \geq r(\overline{\pi}) - 1$. But since $\overline{\pi}$ is the unique maximal-rank partition, this implies $\pi^* = \overline{\pi}$. Thus $\overline{\pi}$ is tight. This implies $x \in (\mathcal{P}')$. $\square$

3.2 Partition and Subtour Elimination Relaxations

Theorem 4. *The feasible regions of* $(\mathcal{P}')$ *and* $(\mathcal{S})$ *are the same.*

Proof. Let x be any feasible solution to the LP $(\mathcal{S})$. Note that the equality constraint of $(\mathcal{P}')$ is the same as that of $(\mathcal{S})$. We now show that x satisfies (5). Fix a partition $\pi = (\pi_1, \dots, \pi_t)$, so $t = r(\pi)$. For each $1 \le i \le t$, subtract the inequality constraint in $(\mathcal{S})$ with $S = \pi_i$, from the equality constraint in $(\mathcal{S})$ to obtain

$$\sum_{K \in \mathcal{K}} x_K \Big(\rho(K) - \sum_{i=1}^{t} \rho(K \cap \pi_i)\Big) \ge \rho(R) - \sum_{i=1}^{t} \rho(\pi_i). \tag{12}$$

From Lemma 2.8, $\rho(K) - \sum_{i=1}^{t} \rho(K \cap \pi_i) = \mathtt{rc}_K^{\pi}$. We also have $\rho(R) - \sum_{i=1}^{t} \rho(\pi_i) = |R| - 1 - (|R| - r(\pi)) = r(\pi) - 1$. Thus x is a feasible solution to the LP $(\mathcal{P}')$.

Now, let x be a feasible solution to $(\mathcal{P}')$ and it suffices to show that it satisfies the inequality constraints of $(\mathcal{S})$. Fix a set $S \subseteq R$. Note when $S = \varnothing$ that inequality constraint is vacuously true so we may assume $S \neq \varnothing$. Let $R \backslash S = \{r_1, \dots, r_u\}$. Consider the partition $\pi = (\{r_1\}, \dots, \{r_u\}, S)$. Subtract (5) for this π from the equality constraint in $(\mathcal{P}')$, to obtain

$$\sum_{K \in \mathcal{K}} x_K (\rho(K) - \mathtt{rc}_K^{\pi}) \le \rho(R) - r(\pi) + 1. \tag{13}$$

Using Lemma 2.8 and the fact that $\rho(K \cap \{r_j\}) = 0$ (the set is either empty or a singleton), we get $\rho(K) - \mathtt{rc}_K^{\pi} = \rho(K \cap S)$. Finally, as $\rho(R) - r(\pi) + 1 = |R| - 1 - (|R \backslash S| + 1) + 1 = \rho(S)$, the inequality (13) is the same as the constraint needed. Thus x is a feasible solution to $(\mathcal{S})$, proving the theorem. ∎

3.3 Directed Hypergraph LP Relaxation

Theorem 5. *For any Steiner tree instance,* $\mathrm{OPT}\,(\mathcal{P}) = \mathrm{OPT}\,(\mathcal{D})$.

Proof. First, we show $\mathrm{OPT}\,(\mathcal{P}) \le \mathrm{OPT}\,(\mathcal{D})$. Consider a feasible solution x to $(\mathcal{D})$, and define a solution x' to $(\mathcal{P})$ by $x'_K = \sum_{i \in K} x_{K^i}$; informally, x' is obtained from x by ignoring the orientation of the hyperedges. Clearly x' and x have the same objective value. Further, x' is feasible for $(\mathcal{P})$; to see this, for any partition π, note that (5) is implied by the sum of constraints (3) over U set to those parts of π not containing the root — any orientation of a full component with rank contribution t must leave at least t parts.

To obtain the reverse direction $\mathrm{OPT}\,(\mathcal{D}) \le \mathrm{OPT}\,(\mathcal{P})$, we use a similar strategy. We require some notation and a hypergraph orientation theorem of Frank et al. [18]. For any $U \subseteq R$ we say that a directed hyperedge K^i *lies in* $\Delta^{\mathrm{in}}(U)$ if $i \in U$ and $K \backslash U \neq \varnothing$, i.e. if $K^i \in \Delta^{\mathrm{out}}(R \backslash U)$. Two subsets U and W of R are called *crossing* if all four sets $U \setminus W$, $W \setminus U$, $U \cap W$, and $R \setminus (U \cup W)$ are non-empty. A set-function $p : 2^R \to \mathbb{Z}$ is a *crossing supermodular* function if

$$p(U) + p(W) \le p(U \cap W) + p(U \cup W)$$

for all crossing sets U and W. A directed hypergraph is said to *cover* p if $|\Delta^{\mathrm{in}}(U)| \ge p(U)$ for all $U \subseteq R$. Here is the needed result.

Theorem 6 (Frank, Király & Király [18])**.** *Given a hypergraph* $H = (R, \mathcal{X})$*, and a crossing supermodular function* p*, the hypergraph has an orientation covering* p *if and only if for every partition* π *of* R*,*

(a) $\sum_{K \in \mathcal{X}} \min\{1, \mathtt{rc}_K^{\pi}\} \ge \sum_{\pi_i \in \pi} p(\pi_i)$, *and,* (b) $\sum_{K \in \mathcal{X}} \mathtt{rc}_K^{\pi} \ge \sum_{\pi_i \in \pi} p(R \setminus \pi_i)$.

We will show every rational solution x to ($\mathcal{P}$) can be fractionally oriented to get a feasible solution for ($\mathcal{D}$), which will complete the proof of Theorem 5. Let M be the smallest integer such that the vector Mx is integral. Let $\mathcal{X}$ be a multi-set of hyperedges which contains Mx_K copies of each K. Define the function p by $p(U) = M$ if $r \in U \neq R$, and $p(U) = 0$ otherwise; i.e. $p(U) = M$ iff $R \backslash U$ is valid. It is not hard to check that p is crossing supermodular.

Claim 3.4. *$H = (R, \mathcal{X})$ satisfies conditions (a) and (b).*

Proof. Note $\sum_{\pi_i \in \pi} p(R \setminus \pi_i) = M(r(\pi) - 1)$ since all parts of π are valid except the part containing the root r. Thus condition (b), upon scaling by $\frac{1}{M}$, is a restatement of constraint (5), which holds since x is feasible for ($\mathcal{P}$).

For this p, condition (a) follows from (b) in the following sense. Fix a partition π, and let π_1 be the part of π containing r. If $\pi_1 = R$ then (a) is vacuously true, so assume $\pi_1 \neq R$. Let σ be the rank-2 partition $(\pi_1, R \setminus \pi_1)$. Then it is easy to check that $\min\{1, \mathtt{rc}_K^\pi\} \geq \mathtt{rc}_K^\sigma$ for all K, and consequently $\sum_{K \in \mathcal{X}} \min\{1, \mathtt{rc}_K^\pi\} \geq \sum_{K \in \mathcal{X}} \mathtt{rc}_K^\sigma$ and $\sum_{\pi_i \in \sigma} p(R \setminus \pi_i) = M = \sum_{\pi_i \in \pi} p(\pi_i)$. Thus, (a) for π follows from (b) for σ. ☐

Now using Theorem 6, take an orientation of $\mathcal{X}$ that covers p. For each $K \in \mathcal{K}$ and each $i \in K$, let n_{K^i} denote the number of the Mx_K copies of K that are oriented as K^i, i.e. directed towards i. So, $\sum_{i \in K} n_{K^i} = Mx_K$. Let $x'_{K^i} := \frac{n_{K^i}}{M}$ for all K^i. Hence $\sum_i x'_{K^i} = x_K$ and x' has the same objective value as x.

To complete the proof, we show x' is feasible for ($\mathcal{D}$). Fix a valid subset U and consider condition (3) for a valid set U. Note that $p(R \backslash U) = M$. Therefore, since the orientation covers p, we get

$$\sum_{K^i \in \Delta^{\text{out}}(U)} x'_{K^i} = \frac{1}{M} \sum_{K^i \in \Delta^{\text{out}}(U)} n_{K^i} = \frac{1}{M} \sum_{K^i \in \Delta^{\text{in}}(R \backslash U)} n_{K^i} \geq \frac{1}{M} p(R \backslash U) = \frac{1}{M} M = 1$$

as needed. ☐

3.4 Partition and Bidirected Cut Relaxations in Quasibipartite Instances

Theorem 7. *On quasibipartite Steiner tree instances,* $\mathrm{OPT}(\mathcal{B}) \geq \mathrm{OPT}(\mathcal{D})$.

To prove Theorem 7, 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. The following is a straightforward application of uncrossing; we remark that it does not depend on quasibipartiteness.

Lemma 3.5. *There exists an optimal solution to ($\mathcal{D}_D$) whose support is a laminar family of sets.*

Proof. Choose an optimal solution z to ($\mathcal{D}_D$) which maximizes $\sum_U z_U |U|^2$ among all optimal solutions. We claim that the support of this solution is laminar. Suppose not and there exists U and U' with $U \cap U' \neq \varnothing$ and $z_U > 0$ and $z_{U'} > 0$. Define z' to be the same as z except $z'_U = z_U - \delta$, $z'_{U'} = z_{U'} - \delta$, $z'_{U \cup U'} = z_{U \cup U'} + \delta$ and $z'_{U \cap U'} = z_{U \cap U'} + \delta$; we will show for small $\delta > 0$, z' is feasible. Note that $U \cap U'$ is not empty and $U \cup U'$ doesn't contain r, and the objective value remains unchanged. Feasibility is preserved: for any K and $i \in K$, if $z_{U \cup U'}$ or $z_{U \cap U'}$ appears in the summand of a constraint, then at least one of z_U or $z_{U'}$ also appears; if both $z_{U \cup U'}$ and $z_{U \cap U'}$ appear, then both z_U and $z_{U'}$ appear. Thus z' is an optimal solution and $\sum_U z'_U |U|^2 > \sum_U z_U |U|^2$, contradicting the choice of z. ☐

Lemma 3.6. *For quasibipartite instances, given a solution of* ($\mathcal{D}_D$) *with laminar support, we can get a feasible solution to* ($\mathcal{B}_D$) *of the same value.*

Proof. This lemma is the heart of the theorem, and is a little technical to prove. We first give a sketch of how we convert a feasible solution z of ($\mathcal{D}_D$) 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 every valid subset of V *including* those with Steiner vertices. 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. To see this consider the Steiner vertex v and its neighbours depicted in Figure 7. 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.

We will soon see 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. Formally, we maintain the following crucial *projection invariant*:

$$\text{The quantity } \sum_{U:S\subseteq U\subseteq S\cup(V\setminus R)} z_U \text{ remains constant, for all terminal sets } S. \tag{PI}$$

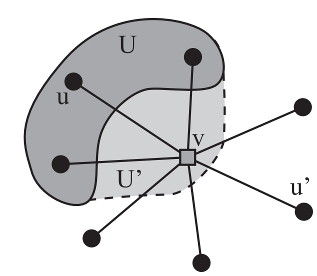

Figure 7: Lifting variable z_U.

This invariant leads to two observations: first, the constraint (4) is satisfied by z at all times, even when it is defined on subsets of all vertices; second, $\sum_{U\subseteq V} z_U$ is constant throughout, and the objective value of z in ($\mathcal{B}_D$) is not affected by the lifting. The existence of a lifting of duals around Steiner vertex v such that (PI) is maintained, and such that all edges incident to v are satisfied can be phrased as a feasibility problem for a linear system of inequalities. We will use Farkas' lemma and the feasibility of z for (4) to complete the proof.

We now fill in the proof details. Let $\Gamma(v)$ denote the set of neighbours of vertex v in the given graph G. In each iteration, where we process Steiner node v, let

$$\mathcal{U}_v := \{U : z_U > 0 \;\text{ and }\; U\cap\Gamma(v)\neq\varnothing\}$$

be the sets in z's support that contain neighbours of v. Note that $U\in\mathcal{U}_v$ could contain Steiner vertices on which the lifting procedure has already taken place. However, by (PI) and by Lemma 3.5 the multi-family $\{U\cap R : U\in\mathcal{U}_v\}$ is laminar. In the lifting process, we will transfer x_U units of the z_U units of dual of each set $U\in\mathcal{U}_v$ to the set $U' = U\cup\{v\}$; this decreases the dual load (LHS of (2)) on arcs from $U\cap\Gamma(v)$ to v (e.g. uv in Figure 7) and increases the dual load on arcs from v to $\Gamma(v)\backslash U$ (e.g. vu' in the figure). The following system of inequalities describes the set of feasible liftings.

$$\forall U \in \mathcal{U}_v: \qquad x_U \leq z_U \tag{L1}$$

$$\forall u \in \Gamma(v): \qquad \sum_{U: u \in U} (z_U - x_U) \leq c_{uv} \tag{L2}$$

$$\forall u \in \Gamma(v): \qquad \sum_{U: u \notin U} x_U \leq c_{uv} \tag{L3}$$

Claim 3.7. *If (L1), (L2), (L3) have a feasible solution $x \geq 0$, then the lifting procedure can be performed at Steiner vertex v, while maintaining the projection invariant property.*

Proof. Define the new solution to be $z_U := z_U - x_U$, and, $z_{(U \cup v)} := x_U$, for all $U \in \mathcal{U}_v$, and z_U remains unchanged for all other U. It is easy to check that all edges which were satisfied remain satisfied, and (L2) and (L3) imply that all edges incident to v are satisfied. Also note that the projection invariant property is maintained. □

By Farkas' lemma, if (L1), (L2), (L3) do *not* have a feasible solution $x \geq 0$, then there exist non-negative multipliers — λ_U for all $U \in \mathcal{U}_v$, and α_u, β_u for all $u \in \Gamma(v)$ — satisfying the following dual set of linear inequalities:

$$\sum_{U \in \mathcal{U}_v} \lambda_U z_U + \sum_{u \in \Gamma(v)} \alpha_u \big(c_{uv} - \sum_{U: u \in U} z_U\big) + \sum_{u \in \Gamma(v)} \beta_u c_{uv} \quad < \quad 0 \tag{D1}$$

$$\forall U \in \mathcal{U}_v: \lambda_U - \sum_{u \in U} \alpha_u + \sum_{u \notin U} \beta_u \quad \geq \quad 0 \tag{D2}$$

As a technicality, note that the sub-system $\{(\text{L1}), (\text{L2}), x \geq 0\}$ is feasible — take $x = z$. Thus any α, β, λ satisfying (D1) and (D2) has $\sum_u \beta_u > 0$, so by dividing all α, β, λ by $\sum_i \beta_i$, we may assume without loss of generality that

$$\sum_{u \in \Gamma(v)} \beta_u = 1. \tag{D3}$$

Subtracting (D3) from (D2) allows us to rewrite the latter set of constraints conveniently as

$$\forall U \in \mathcal{U}_v: \qquad \lambda_U - \sum_{u \in U} (\alpha_u + \beta_u) + 1 \geq 0. \tag{D2'}$$

The following claim shows that (L1), (L2), (L3) does have a feasible solution, and thus by Claim 3.7, lifting can be done, which completes the proof of Lemma 3.6.

Claim 3.8. *There exists no feasible solution to $\{\alpha, \beta, \lambda \geq 0 : (\text{D1}), (\text{D2'}), \text{and } (\text{D3})\}$.*

Proof. Consider the linear program which minimizes the LHS of (D1) subject to the constraints (D2') and (D3). We show that the LP has value at least 0, which will complete the proof.

Let $(\lambda^*, \alpha^*, \beta^*)$ be an optimal solution to the LP. In Lemma 3.9 we will show that the constraint matrix of the LP is totally unimodular; hence, since the right-hand side of the given system is integral, we may assume that λ^*, α^*, and β^* are non-negative and integral. From (D3) we infer

$$\text{There is a unique } \bar{u} \in \Gamma(v) \text{ for which } \beta^*_{\bar{u}} = 1; \text{ for all } u \neq \bar{u},\ \beta^*_u = 0. \tag{14}$$

Moreover, since each λ_U appears only in the two constraints (D2') and $\lambda_U \geq 0$, and since λ_U has nonnegative coefficient in the objective, we may assume

$$\lambda^*_U = \lambda^*_U(\alpha^*, \beta^*) := \max\{\sum_{u \in U}(\alpha^*_u + \beta^*_u) - 1, 0\} \tag{15}$$

for all U.

Next, we establish the following:

$$\alpha^*_u + \beta^*_u \in \{0, 1\} \text{ for all } u \in \Gamma(v). \tag{16}$$

Suppose for the sake of contradiction that property (16) does not hold for our solution. Let u be such that $\alpha^*_u + \beta^*_u \geq 2$. By (14), $\alpha^*_u \geq 1$. We propose the following update to our solution: decrease α^*_u by 1 (which by (15) will decrease λ^*_U by 1 for all $U \in \mathcal{U}_v$ with $u \in U$). This maintains the feasibility of (D2'), and the objective value decreases by

$$\sum_{U \in \mathcal{U}_v : u \in U} z_U + (c_{uv} - \sum_{u \in U} z_U)$$

which is non-negative as $c \geq 0$. By repeating this operation, we may clearly ensure property (16).

Let $K \subseteq \Gamma(v)$ be the set $\{u : \alpha^*_u + \beta^*_u = 1\}$ and recall $\bar{u}$ is the unique terminal with $\beta^*_{\bar{u}} = 1$; $\bar{u}$ is clearly a member of K. At $(\alpha^*, \beta^*, \lambda^*)$, we evaluate the objective and collect like terms to get value

$$\begin{aligned}\sum_{U \in \mathcal{U}_v} z_U \rho(U \cap K) + \sum_{u \in K \setminus \bar{u}} (c_{uv} - \sum_{U : u \in U} z_U) + c_{\bar{u}v} &= \sum_{u \in K} c_{uv} + \sum_{U \in \mathcal{U}_v} z_U(\rho(U \cap K) - |(K \setminus \bar{u}) \cap U|) \\ &= \sum_{u \in K} c_{uv} - \sum_{U \in \mathcal{U}_v : U \cap K \neq \varnothing, \bar{u} \notin U} z_U\end{aligned}$$

where the last equality follows by considering cases. Finally, combining the fact that $\sum_{u \in K} c_{uv} \geq C_K$ (since these edges form one possible full component on terminal set K) together with (4) for the pair $(K, \bar{u})$, it follows that the LP's optimal value is non-negative as needed. ∎

Lemma 3.9. *The incidence matrix defined by* (D2') *and* (D3) *is totally unimodular.*

Proof. The incidence matrix has $|\mathcal{U}_v| + 1$ rows ($|\mathcal{U}_v|$ corresponding to (D2') and one last row corresponding to (D3)) and $|\mathcal{U}_v| + 2|\Gamma(v)|$ columns. Furthermore, the columns corresponding to α_u's are same as those corresponding to β_u's, except for the last row, where there are 0's in the α-columns and 1's in the β-columns.

To show that this matrix is totally unimodular we use Ghouila-Houri's characterization of total unimodularity (e.g. see [34, Thm. 19.3]):

Theorem 8 (Ghouila-Houri 1962)**.** *A matrix is totally unimodular iff the following holds for every subset $\mathcal{R}$ of rows: we can assign weights $w_r \in \{-1, +1\}$ to each row $r \in \mathcal{R}$ such that $\sum_{r \in \mathcal{R}} w_r r$ is a $\{0, \pm 1\}$-vector.*

Note that we can safely ignore the columns corresponding to variables λ_U for sets $U \in \mathcal{U}_v$, since each of them contains a single 1 occurring in constraint (D2') for set U.

The row subset $\mathcal{R}$ corresponds to a subset of $\mathcal{U}_v$ — which we will denote $\mathcal{R} \cap \mathcal{U}_v$ — plus possibly the single row corresponding to (D3). Each row in $\mathcal{R} \cap \mathcal{U}_v$ has its values determined by the characteristic vector of $U \cap \Gamma(v)$. So long as any set appears more than once in $\{U \cap \Gamma(v) :$

$U \in \mathcal{R} \cap \mathcal{U}_v\}$ we can assign one copy weight $+1$ and the other copy weight -1; these rows cancel out. Thus, henceforth we assume $\{U \cap \Gamma(v) : U \in \mathcal{R} \cap \mathcal{U}_v\}$ has no duplicate sets.

There is a standard representation of a laminar family as a forest of rooted trees, where there is a node corresponding to each set, with containment in the family corresponding to ancestry in the forest. Given the forest for the laminar family $\{U \cap \Gamma(v) : U \in \mathcal{R} \cap \mathcal{U}_v\}$, the assignment of weights to the rows of the matrix is as follows. Let the root nodes of all trees be at height 0 with height increasing as one goes to children nodes. Give weight -1 to rows corresponding to nodes at even height, and weight $+1$ to rows corresponding to nodes at odd height. If $\mathcal{R}$ contains the row corresponding to (D3), give it weight $+1$.

Finally, let us argue that these weights have the needed property. Consider first a column corresponding to α_u for any u. The rows of $\mathcal{R}$ with 1 in this column form a path, from the largest set containing u (which is a root node) to the smallest set containing u. The weighted sum in this column is an alternating sum $-1+1-1+1\cdots$, which is either -1 or 0, which is in $\{0, \pm 1\}$ as needed. Second, in a column for some β_u, if $\mathcal{R}$ doesn't contain (resp. contains) the row corresponding to (D3), the weighted sum is the same as for α_u (resp. plus 1); in either case its weighted sum is in $\{0, \pm 1\}$ as needed. ☐

This finishes the proof of Lemma 3.6, and hence also that of Theorem 7. ☐

4 Integrality Gap Upper Bounds

We first show the improved integrality gap upper bound of 73/60 for uniformly quasibipartite graphs. We then consider general graphs, starting with the $(2\sqrt{2}-1) \simeq 1.828$ upper bound that contains the main ideas, then giving the tighter $\sqrt{3} \simeq 1.729$ bound.

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. [24], who gave a 73/60 factor approximation algorithm. In the following, we show that the cost of the returned tree is no more than than $\frac{73}{60}\,\mathrm{OPT}\,(\mathcal{P})$, which upper-bounds the integrality gap by $\frac{73}{60}$. The analysis is related to that of Wolsey [42] for submodular set cover.

We start by describing the algorithm of Gröpl et al. [24] 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 9. *On a uniformly quasibipartite instance,* RatioGreedy *returns a Steiner tree of cost at most* $\frac{73}{60}\,\mathrm{OPT}\,(\mathcal{P})$.

Proof. 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 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; y is not generally feasible, but we will scale it down to make it so. By evaluating a telescoping sum, it is not hard to show that $\sum_i y_{\pi(i)}(r(\pi(i)) - 1) = C(\mathcal{L})$. In the rest of the proof we will show for any $K \in \mathcal{K}$, $\sum_i y_{\pi(i)} \mathtt{rc}_K^{\pi(i)} \leq 73/60 \cdot C_K$ — by scaling, this also proves that $\frac{60}{73}y$ is a feasible dual solution, and hence completes the proof.

Fix any $K \in \mathcal{K}$ and let $|K| = k$. Since the instance in question is uniformly quasibipartite, the full component K is a star with a Steiner centre and edges of a fixed cost c to each terminal in K. For $1 \leq i < k$, let $\tau(i)$ denote the last iteration j in which $\mathtt{rc}_K^{\pi(j)} \geq k - i$. Let K_i denote any subset of K of size $k - i + 1$ such that K_i contains at most one element from each part of $\pi(\tau(i))$; i.e., $|K_i| = k - i + 1$ and $\mathtt{rc}_{K_i}^{\pi(\tau(i))} = k - i$.

Our analysis hinges on the fact that K_i was a valid choice for $L_{\tau(i)+1}$. More specifically, note that $\{L_1, \ldots, L_{\tau(i)}, K_i\}$ is acyclic, hence by the greedy nature of the algorithm, for any $1 \leq i < k$,

$$\theta(\tau(i) + 1) = C_{L_{\tau(i)+1}}/(|L_{\tau(i)+1}| - 1) \leq C_{K_i}/(|K_i| - 1) \leq \frac{c \cdot (k - i + 1)}{k - i}.$$

Moreover, using the definition of τ and telescoping we compute

$$\sum_\pi y_\pi \mathtt{rc}_K^\pi = \sum_{i=0}^{t-1} (\theta(i+1) - \theta(i)) \mathtt{rc}_K^{\pi(i)} = \sum_{i=1}^{k-1} \theta(\tau(i)+1) \leq \sum_{i=1}^{k-1} \frac{c \cdot (k - i + 1)}{k - i} = c \cdot (k - 1 + H(k-1)),$$

where $H(\cdot)$ denotes the harmonic series. Finally, note that $(k - 1 + H(k-1)) \leq \frac{73}{60}k$ for all $k \geq 2$ (achieved at $k = 5$). Therefore, $\frac{60}{73}y$ is a valid solution to ($\mathcal{P}_D$). ☐

4.2 General graphs

We start with a few definitions and notations in order to prove the $2\sqrt{2} - 1$ and $\sqrt{3}$ integrality gap bounds on ($\mathcal{P}$). Both results use similar algorithms, and the latter is a more complex version of the former.

For conciseness we let a "graph" be a triple $G = (V, E, R)$ where $R \subseteq V$ 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.

Our first algorithm contracts full components, and the second one contracts edges. 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. When contractions introduce parallel edges, we delete all but the cheapest edge from each parallel class.

Our first 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 $\mathtt{drop}_T(K; c) := c(T) - \mathtt{mtst}(G/K; c)$. Following [44], 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 $\texttt{gain}_T(K;c) \le 0$. The following useful fact is implicit in [27]; we give a full proof in Section 4.3.

Theorem 10 (Implicit in [27]). *If* $\texttt{mtst}(G;c)$ *is gainless, then* OPT ($\mathcal{P}$) *equals the cost of* $\texttt{mtst}(G;c)$.

We now give the first algorithm and its analysis, which uses a reduced cost trick introduced by Chakrabarty et al. [5].

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 $G_1 := G$, $T_1 := \texttt{mtst}(G_1; c')$, and $i := 1$.
2: The algorithm considers the full components in any order. When we examine a full component K, if $\texttt{gain}_{T_i}(K; c') > 0$, let $K_i := K$, $G_{i+1} := G_i/K_i$, $T_{i+1} := \texttt{mtst}(G_{i+1}; c')$, and $i := i + 1$.
3: Let f be the final value of i. Return the tree $T_{alg} := T_f \cup \bigcup_{i=1}^{f-1} K_i$.

The algorithm functions in an *online* manner in the sense that we irrevocably decide whether to buy a full component or not as soon as it is examined. In-between iterations we maintain information about terminal-terminal edges but not Steiner nodes, so it is also a relative of *streaming* algorithms. However, in both settings all terminal-terminal metric costs need to be known when the algorithm starts.

Theorem 11. $c(T_{alg}) \le (2\sqrt{2} - 1)\,\mathrm{OPT}\,(\mathcal{P})$.

Proof. First we claim that $\texttt{gain}_{T_f}(K; c') \le 0$ for all K. To see this there are two cases. If $K = K_i$ for some i, then we immediately see that $\texttt{drop}_{T_j}(K) = 0$ for all $j > i$ so $\texttt{gain}_{T_f}(K) = -c(K) \le 0$. Otherwise (if for all i, $K \ne K_i$) K had nonpositive gain when examined by the algorithm; and the well-known *contraction lemma* (e.g., see [23, §1.5]) immediately implies that $\texttt{gain}_{T_i}(K)$ is nonincreasing in i, so $\texttt{gain}_{T_f}(K) \le 0$.

When we change the terminal-terminal costs, they are reduced so the optimal LP value does not increase. Moreover, it is easy to see that the LP optimal value does not increase in any iteration of contraction, since a feasible fractional solution on the original graph remains feasible after contraction. Thus the final LP optimal value is at most the original one. Furthermore, by Theorem 10, the final value equals $c'(T_f)$. Using the fact that $c(T_f) = \sqrt{2}c'(T_f)$, we get

$$c(T_f) = \sqrt{2}c'(T_f) \le \sqrt{2}\,\mathrm{OPT}\,(\mathcal{P}) \tag{17}$$

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

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

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

Thus, we have

$$\sum_{i=1}^{f-1} c(K_i) \le \frac{1}{\sqrt{2}} \sum_{i=1}^{f} \texttt{drop}_{T_i}(K_i; c) = \frac{1}{\sqrt{2}}(\texttt{mtst}(G; c) - c(T_f)) \le \frac{1}{\sqrt{2}}(2\,\mathrm{OPT}\,(\mathcal{P}) - c(T_f))$$

where we use the fact that $\texttt{mtst}(G, c)$ is at most twice OPT $(\mathcal{P})$[2]. 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 (17), the proof of Theorem 11 is complete. $\square$

4.2.1 Improving to $\sqrt{3}$

To get the improved factor of $\sqrt{3}$, we use a more refined iterated contraction approach. The crucial new concept is that of the *loss* of a full component, introduced by Karpinski and Zelikovsky [26]. The intuition is as follows. In each iteration, the $(2\sqrt{2} - 1)$-factor algorithm contracts a full component K, and thus commits to include K in the final solution; the new algorithm makes a smaller commitment, by contracting a *subset* of K's edges, which allows for a possibility of better recovery later.

Given a full component K (viewed as a tree with leaf set K and internal Steiner nodes), $\texttt{loss}(K)$ is defined to be the minimum-cost subset of $E(K)$ such that $(V(K), \texttt{loss}(K))$ has at least one terminal per connected component — i.e. the cheapest way in K to connect each Steiner node to the terminal set. We also use $\texttt{loss}(K)$ to denote the total *cost* of these edges. Note that no two terminals are connected by $\texttt{loss}(K)$. A very useful theorem of Karpinski and Zelikovsky [26] is that for any full component K, $\texttt{loss}(K) \leq c(K)/2$.

Now we have the ingredients to give our new algorithm. In the description below, $\alpha > 1$ is a parameter (which will be set to $\sqrt{3}$). In each iteration, the algorithm contracts the loss of a single full component K.

Procedure Reduced One-Pass Loss-Contracting Heuristic

1: Initially $G_1 := G$, $T_1 := \texttt{mtst}(G; c)$, and $i := 1$.

2: The algorithm considers the full components in any order. When we examine a full component K, if

$$\texttt{gain}_{T_i}(K; c) > (\alpha - 1)\texttt{loss}(K),$$

let $K_i := K$, $G_{i+1} := G_i / \texttt{loss}(K_i)$, $T_{i+1} := \texttt{mtst}(G_{i+1}; c)$, and $i := i + 1$.

3: Let f be the final value of i. Return the tree $T_{alg} := T_f \cup \bigcup_{i=1}^{f-1} \texttt{loss}(K_i)$.

We now analyze the algorithm.

Claim 4.1. $c(T_f) \leq (\frac{1+\alpha}{2})\,\mathrm{OPT}\,(\mathcal{P})$.

Proof. Using the contraction lemma again, $\texttt{gain}_{T_f}(K; c) \leq (\alpha - 1)\texttt{loss}(K)$ for all K, so

$$\texttt{drop}_{T_f}(K; c) \leq c(K) + (\alpha - 1)\texttt{loss}(K) \leq \Big(\frac{1+\alpha}{2}\Big) c(K) \tag{18}$$

since $\texttt{loss}(K) \leq c(K)/2$.

To finish the proof of Claim 4.1, we proceed as in the proof of Equation (17). Define $c'_e := c_e / (\frac{1+\alpha}{2})$ for all edges e which join two vertices of the original terminal set R, and $c'_e = c_e$ for all other edges. Note that (18) implies that T_f is gainless with respect to c'. Thus, by Theorem 10, the value of LP $(\mathcal{P})$ on (G_f, c') equals $c'(T_f)$. Since we only reduce costs (as $\alpha \geq 1$), this optimum is no more than the original OPT $(\mathcal{P})$ giving us $c'(T_f) \leq \mathrm{OPT}\,(\mathcal{P})$. Now using the definition of c', the proof of the claim is complete. $\square$

[2]This follows using standard arguments, and can be seen, for instance, by applying Theorem 10 to the cost-function with all terminal-terminal costs divided by 2, and using short-cutting.

Claim 4.2 (Implicit in [33]). *For any $i \geq 1$, we have $c(T_i) - c(T_{i+1}) \geq \texttt{gain}_{T_i}(K_i; c) + \texttt{loss}(K_i)$.*

Proof. Recall that T_{i+1} is a minimum terminal spanning tree of $G_{i+1} = G_i/\texttt{loss}(K_i)$ under c. Consider the following other terminal spanning tree T of G_{i+1}: take T to be the union of $K_i/\texttt{loss}(K_i)$ with $\texttt{mtst}(G_i/K_i; c)$. Hence $c(T_{i+1}) \leq c(T) = \texttt{mtst}(G_i/K_i; c) + c(K_i) - \texttt{loss}(K_i)$. Rearranging, and using the definition of gain, we obtain:

$$c(T_i) - c(T_{i+1}) \geq c(T_i) - \texttt{mtst}(G_i/K_i; c) - c(K_i) + \texttt{loss}(K_i) = \texttt{gain}_{T_i}(K_i; c) + \texttt{loss}(K_i),$$

and this completes the proof. □

Now we are ready to prove the integrality gap upper bound of $\sqrt{3}$.

Theorem 12. $c(T_{alg}) \leq \sqrt{3}\,\mathrm{OPT}\,(\mathcal{P})$.

Proof. By the algorithm, we have for all i that $\texttt{gain}_{T_i}(K_i) \geq (\alpha-1)\texttt{loss}(K_i)$, and thus $\texttt{gain}_{T_i}(K_i; c) + \texttt{loss}(K_i) \geq \alpha\texttt{loss}(K_i)$. Thus, from Claim 4.2, we get

$$\sum_{i=1}^{f-1} \texttt{loss}(K_i) \leq \frac{1}{\alpha}\sum_{i=1}^{f-1}\Big(c(T_i) - c(T_{i+1})\Big)$$

The right-hand sum telescopes to give us $c(T_1) - c(T_f) = \texttt{mtst}(G; c) - c(T_f)$. Thus,

$$\begin{aligned} c(T_{alg}) = c(T_f) + \sum_{i=1}^{f-1}\texttt{loss}(K_i) &\leq c(T_f) + \frac{1}{\alpha}(\texttt{mtst}(G; c) - c(T_f)) = \frac{1}{\alpha}\texttt{mtst}(G; c) + \frac{\alpha-1}{\alpha}c(T_f) \\ &\leq \Big(\frac{2}{\alpha} + \frac{(\alpha-1)(1+\alpha)}{2\alpha}\Big)\,\mathrm{OPT}\,(\mathcal{P}) = \Big(\frac{\alpha^2+3}{2\alpha}\Big)\,\mathrm{OPT}\,(\mathcal{P}) \end{aligned}$$

which follows from $\texttt{mtst}(G; c) \leq 2\,\mathrm{OPT}\,(\mathcal{P})$ and Claim 4.1. Setting $\alpha = \sqrt{3}$, the proof of the theorem is complete. □

4.3 Gainless MSTs and Hypergraphic Relaxations

Theorem 10 (Implicit in [27]). *If the MST induced by the terminals is gainless, then* $\mathrm{OPT}\,(\mathcal{P})$ *equals the cost of that MST.*

Proof. Let Π be the set of all partitions of the terminal set. As before, we let $r(\pi)$ be the rank of a partition $\pi \in \Pi$, and we use E_π for the set of edges in our graph that *cross* the partition: E_π contains all edges whose endpoints lie in different parts of π. Fulkerson's [19] formulation of the spanning tree polyhedron and its dual are as follows.

$$\min\Big\{\sum_{e\in E} c_e x_e : \quad x \in \mathbf{R}^E_{\geq 0} \qquad (\mathcal{M})$$

$$\sum_{e\in E_\pi} x_e \geq r(\pi) - 1 \quad \forall \pi \in \Pi\Big\} \qquad (19)$$

$$\max\Big\{\sum_{\pi}(r(\pi)-1)\cdot y_\pi : \quad y \in \mathbf{R}^\Pi_{\geq 0} \qquad (\mathcal{M}_D)$$

$$\sum_{\pi : e\in E_\pi} y_\pi \leq c_e, \quad \forall e \in E\Big\} \qquad (20)$$

The high-level overview of the proof is as follows. We use a folklore primal-dual interpretation of Kruskal's minimum-spanning tree algorithm, with respect to Fulkerson's LP (it is similar to

Chopra's analysis [9]). Running Kruskal's algorithm on the terminal set then returns a minimum spanning tree T and a feasible dual y to Equation ($\mathcal{M}_D$) such that

$$c(T) = \sum_{\pi} (r(\pi) - 1) y_\pi.$$

The final step will be to show that, if the returned MST is gainless, then the spanning tree dual y is feasible for ($\mathcal{P}_D$), and its value is $c(T)$ as well. Weak duality and the fact that the optimal value of ($\mathcal{P}$) is at most $c(T)$ imply the theorem.

Kruskal's algorithm can be viewed as a process over time. Let τ be a parameter denoting *time*, initially 0, which increases over the course of the algorithm. The algorithm maintains a forest T^τ, and a feasible dual solution y^τ; initially $T^0 = (V, \varnothing)$ and $y^0 = 0$. Let π^τ be the partition induced by the connected components of T^τ. If T^τ is not a spanning tree, Kruskal's algorithm raises τ and the dual variable y_{π^τ} at the same rate until a constraint (20) for some edge e becomes tight. The algorithm then adds e to the forest and continues. The algorithm stops at time τ^* as soon as T^{τ^*} is a spanning tree.

Let T be the gainless spanning tree returned by Kruskal's algorithm, and let y be the corresponding dual. We claim that y is feasible for ($\mathcal{P}_D$). To see this, consider a full component K. Clearly, the rank contribution $\mathtt{rc}_K^{\pi^0}$ of K to the initial partition π^0 is $|K| - 1$; similarly, the final rank contribution $\mathtt{rc}_K^{\pi^{\tau^*}}$ is 0. Every edge that is added during the algorithm's run either leaves the rank contribution of K unchanged, or it decreases it by 1. Let $e_1, \ldots, e_{|K|-1}$ be the edges of the final tree T whose addition to T decreases K's rank contribution. Also let

$$0 \leq \tau_1 \leq \tau_2 \leq \ldots \leq \tau_{|K|-1} \leq \tau^*$$

be the times where these edges are added. Note that, by definition, we must have $c_{e_i} = \tau_i$ for all i. We therefore have

$$\sum_{i=1}^{|K|-1} c_{e_i} = \sum_{i=1}^{|K|-1} \tau_i. \tag{21}$$

The right-hand side of this equality is easily checked to be equal to

$$\int_0^{\tau^*} \mathtt{rc}_K^{\pi^\tau} d\tau,$$

which in turn is equal to $\sum_\pi \mathtt{rc}_K^\pi y_\pi$, by the definition of Kruskal's algorithm. It is not hard to see that the left-hand side of (21) is the drop $\mathtt{drop}_T(K)$ induced by K. Together with the fact that T is gainless, we obtain

$$c_K \geq \mathtt{drop}_T(K) = \sum_\pi \mathtt{rc}_K^\pi y_\pi.$$

Now observe that the right-hand side of this equation is the left-hand side of (6). It follows that y is feasible for ($\mathcal{P}_D$). $\square$

5 Open Problems

Practical algorithms for the Steiner tree problem are important. Compact LP relaxations independent of r, such as the bidirected cut relaxation, are very practical. For the special case of quasibipartite graphs we showed that the bidirected cut relaxation has the same strength as the hypergraphic relaxations. This implies a 1.28 upper bound on the integrality gap of bidirected cut for quasibipartite instances; but is there a *direct* proof of this fact?

An intruiging open problem is to *exactly* optimize over the hypergraphic relaxations when *all* full components are present (e.g. when $r = |R|$). Before now only approximations were known, by taking r suitably large. Our methods have partially solved a special case: in *quasibipartite instances*, the lifting-based proof of Theorem 7 shows that one can exactly compute an dual optimal solution.

Recently the integrality gap lower bound for the bidirected cut relaxation was raised from $\frac{8}{7}$ to $\frac{36}{31}$ [4]. For the hypergraphic relaxation, the $\frac{8}{7}$ attained by the elegant Skutella graph remains the best known, and it would be nice to improve upon it.

References

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

[2] G. Baudis, S. H. C. Gröpl, T. Nierhoff, and H. Prömel. Approximating minimum spanning sets in hypergraphs and polymatroids. Technical report, Humboldt-Universität zu Berlin, Institut für Informatik, 2000. ICALP 2000, accepted but withdrawn, http://publications.mi.fu-berlin.de/329.

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

[4] J. Byrka, F. Grandoni, T. Rothvoß, and L. Sanità. Steiner tree approximation via iterative randomized rounding. *J. ACM*, To appear. Preliminary version appeared in *Proc. 42nd STOC*, pages 583–592, 2010.

[5] 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.

[6] D. Chakrabarty, J. Könemann, and D. Pritchard. Hypergraphic LP relaxations for Steiner trees. In *Proc. 14th IPCO*, pages 383–396, 2010.

[7] D. Chakrabarty, J. Könemann, and D. Pritchard. Integrality gap of the hypergraphic relaxation of Steiner trees: A short proof of a 1.55 upper bound. *Operations Research Letters*, 38(6):567–570, 2010.

[8] 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.

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

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

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

[12] G. Cornuéjols, D. Naddef, and J. Fonlupt. The traveling salesman problem on a graph and some related integer polyhedra. *Math. Programming*, 33:1–27, 1985.

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

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

[15] J. Edmonds. Optimum branchings. *Journal of Research of the National Bureau of Standards B*, 71B:233–240, 1967.

[16] J. Edmonds. Matroids and the greedy algorithm. *Math. Programming*, 1:127–136, 1971.

[17] J. Edmonds and R. Giles. A min-max relation for submodular functions on graphs. *Annals of Discrete Mathematics*, 1:185–204, 1977.

[18] A. Frank, T. Király, and Z. Király. On the orientation of graphs and hypergraphs. *Discrete Appl. Math.*, 131(2):385–400, 2003.

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

[20] D. Gijswijt and G. Pap. An algorithm for weighted fractional matroid matching. arXiv:0806.1818, 2008.

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

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

[23] 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.

[24] 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.

[25] 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.

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

[27] J. Könemann, D. Pritchard, and K. Tan. A partition-based relaxation for Steiner trees. *Mathematical Programming*, 127:345–370, 2011. 10.1007/s10107-009-0289-2.

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

[29] 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.

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

[31] D. Pritchard. *Linear Programming Tools & Approximation Algorithms for Combinatorial Optimization.* PhD thesis, University of Waterloo, 2009.

[32] 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.

[33] 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.

[34] A. Schrijver. *Theory of Linear and Integer Programming.* Wiley, Chichester, 1986.

[35] A. Schrijver. *Combinatorial optimization.* Springer, New York, 2003.

[36] 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.

[37] R. P. Stanley. *Enumerative Combinatorics*, volume 1. Wadsworth & Brooks/Cole, 1986.

[38] L. Trevisan. Non-approximability results for optimization problems on bounded degree instances. In *Proc. 33rd STOC*, pages 453–461, 2001.

[39] J. H. V. Vate. Fractional matroid matchings. *J. Comb. Theory, Ser. B*, 55(1):133–145, 1992.

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

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

[42] L. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4):385–393, 1982.

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

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