New Geometry-Inspired Relaxations and Algorithms for the Metric Steiner Tree Problem*

Deeparnab Chakrabarty[1], Nikhil R. Devanur[2], and Vijay V. Vazirani[1]

[1] College of Computing, Georgia Institute of Technology, Atlanta, GA 30332–0280.
deepc;vazirani@cc.gatech.edu
[2] Toyota Technological Institute nikhil@tti-c.org

Abstract. Determining the integrality gap of the bidirected cut relaxation for the metric Steiner tree problem, and exploiting it algorithmically, is a long-standing open problem. We use geometry to define an LP whose dual is equivalent to this relaxation. This opens up the possibility of using the primal-dual schema in a geometric setting for designing an algorithm for this problem.
Using this approach, we obtain a 4/3 factor algorithm and integrality gap bound for the case of quasi-bipartite graphs; the previous best being 3/2 [RV99]. We also obtain a factor $\sqrt{2}$ strongly polynomial algorithm for this class of graphs.
A key difficulty experienced by researchers in working with the bidirected cut relaxation was that any reasonable dual growth procedure produces extremely unwieldy dual solutions. A new algorithmic idea helps finesse this difficulty – that of reducing the cost of certain edges and constructing the dual in this altered instance – and this idea can be extracted into a new technique for running the primal-dual schema in the setting of approximation algorithms.

1 Introduction

Some of the major open problems left in approximation algorithms are centered around LP-relaxations which researchers believe have not been fully exploited algorithmically, i.e., the best known algorithmic result does not match the best known lower bound on their integrality gaps. One of them is the bidirected cut relaxation for the metric Steiner tree problem [Edm67] and this is the main focus of our paper.

The integrality gap of the bidirected cut relaxation is believed to be very close to 1; the best lower bound known on the gap is 8/7, due to Goemans [Goe96] and Skutella[KPT]. On the other hand, the known upper bound is the same as the weaker undirected cut relaxation which is 2 by a straightforward 2-factor algorithm for it. The only algorithmic results using the bidirected relaxation that we are aware of are: a 6/5 factor algorithm for the class of graphs containing at most 3 required vertices [Goe96] and a factor 3/2 algorithm for the class of

* Work supported by NSF Grant CCF-0728640

quasi-bipartite graphs, i.e., graphs that do not have edges connecting pairs of Steiner vertices [RV99].

In this paper, we use geometry to develop a new way of lower bounding the cost of the optimal Steiner tree. The best such lower bound can be captured via an LP, which we call the *simplex-embedding LP*. A short description of this LP is that it is an l_1-embedding of the given metric on a simplex, maximizing a linear objective function. Interestingly enough, the dual of the simplex-embedding LP is a relaxation of the metric Steiner tree problem having the same integrality gap as the bidirected cut relaxation!

A key difficulty faced by researchers in working with the bidirected cut relaxation is the lack of structure in the dual solutions arising mainly due to the asymmetric nature of the primal. We believe our geometric approach would open up new ways to use the primal-dual schema for the bidirected cut relaxation. In particular, we exhibit one dual growing procedure (the EMBED algorithm in Section 2.1) which helps us prove the following property about the bidirected cut relaxation in quasi-bipartite graphs: If the minimum spanning tree is the optimum Steiner tree, then the LP relaxation is exact (Theorem 5).

A second key feature of our paper is a new algorithmic idea of *reduced costs*. We show that modifying the problem instance by reducing costs of certain edges and then running the primal-dual schema allows us to obtain provably better results. We use this idea first to get a simple and fast algorithm which proves an upper bound of $\sqrt{2}$ on the integrality gap for quasi-bipartite graphs, already improving the previous best of 3/2 [RV99]. Using another way of reducing costs, we improve the upper bound to 4/3. This algorithm (similar to the algorithm in [RV99]) doesn't run in strongly polynomial time, unlike the $\sqrt{2}$ algorithm.

We also give a second geometric LP in which Steiner vertices are not constrained to be on the simplex but are allowed to be embedded "above" the simplex. Again, the dual of this LP is a relaxation of the metric Steiner tree problem. We show that on any instance, the integrality gap of this latter LP is at most that of the bidirected relaxation. It turns out that this LP is in fact exact for Goemans' 8/7 example. However, there are examples for which the gap is 8/7 even for this relaxation. Could it be that this relaxation has a strictly smaller integrality gap than the bidirected cut relaxation?

1.1 Related Work

Historically, the idea of using an extra vertex to get a shorter tree connecting three points on the plane goes back to Torricelli and Fermat in the seventeenth century. The Euclidean Steiner tree problem, in its full generality, was first defined by Gauss in a letter to his student, Schumacher. This problem was made popular by the book of Courant and Robbins [CRS96], who mistakenly attributed it to the nineteenth century geometer, Steiner. The rich combinatorial structure of this problem was explored by many researchers; e.g., see the books by Hwang, Richards and Winter [HRW92] and Ivanov and Tuzhilin [IT94].

The use of the bidirected cut relaxation for the Steiner tree problem goes back to Edmonds[Edm67] who showed the relaxation is exact for the the case of spanning trees. Wong [Won84] gave a dual ascent algorithm for this relaxation, and Chopra and Rao[CR94a,CR94b] studied the properties of facets of the polytope defined by the relaxation. Goemans and Myung[GM93] study various undirected equivalent relaxations to the bidirected cut relaxation.

The bidirected cut relaxation has interesting structural properties which have been exploited in diverse settings. Let α denote the integrality gap of this relaxation. [JV02] use this LP for giving a factor 2 budget-balanced group-strategy-proof cost-sharing method for the Steiner tree game; Agarwal and Charikar [AC04] prove that for the problem of multi-casting in undirected networks, the coding gain achievable using network coding is precisely equal to α. The latter result holds when restricted to quasi-bipartite networks as well. Consequently, for these networks, the previous best bound was 3/2 [RV99], and our result improves it to 4/3.

The class of quasi-bipartite graphs is a non-trivial class for the bidirected cut relaxation. In fact, recently Skutella (as reported by Konemann et.al.[KPT]) exhibited a quasi-bipartite graph on 15 vertices for which the integrality gap of the bidirected cut relaxation is 8/7. This ratio matches the erstwhile best known example on general graphs of Goemans[Goe96] where the ratio was met as a limit. Moreover, the best known hardness results for the Steiner tree problem in this class of graphs is quite close to that known in general graphs ($\frac{128}{127}$ versus $\frac{96}{95}$)[CC02].

The best approximation algorithms for the Steiner tree problem is due to Robbins and Zelikovsky [RZ05]. The authors prove a guarantee of 1.55 for general graphs and also show that when restricted to the quasi-bipartite case, the ratio is 1.28. However, it is not clear if these results would imply an upper bound on the integrality gap of the bidirected cut relaxation. Very recently, Konemann et.al.[KPT] introduced another LP relaxation for the minimum Steiner tree problem which is as strong as the bidirected cut relaxation, and showed that the algorithm of Robbins and Zelikovsky can be interpreted as a primal-dual algorithm on their LP! However, even their interpretation does not prove any upper bound on the integrality gap of their relaxation as they also compare with the optimum Steiner tree and not the optimum LP solution. Nevertheless, they show upper bounds for their relaxation for a larger class of graphs called b-quasi-bipartite graphs.[3]

1.2 Preliminaries

Let $G = (V, E)$ be an undirected graph with edge costs $\mathbf{c} : E \to \mathbb{R}$. $R \subset V$ denotes the set of *required* vertices. The vertices in $S = V \setminus R$ are called *Steiner* vertices. The Steiner tree problem is to find the minimum cost tree connecting all the required vertices and some subset of Steiner vertices. We abuse notation and denote both the optimum tree and its solution as OPT. Also, given a set of vertices X, we denote the minimum spanning tree on X and its cost as $MST(X)$.

[3] A graph is b-quasi-bipartite if on deleting all required vertices, the largest size of any component is at most b

The edge costs can be extended to all pairs of vertices such that they satisfy triangle inequality (simply define the cost of (uv) to be the cost of the shortest path from u to v). This version is called the metric Steiner tree problem. The two versions are equivalent.

Let $\mathcal{U} := \{U \subsetneq V : U \cap R \neq \varnothing \text{ and } U^c \cap R \neq \varnothing\}$ denote the subsets of V which contain at least one required vertex but not all. Let $\delta(U)$ denote the edges with exactly one end point in U. The undirected cut relaxation of the Steiner tree problem is:

$$\min\{\sum_{e \in E} \mathbf{c}(e) x_e : \quad x(\delta(U)) \geq 1, \ \forall U \in \mathcal{U}; \ x \geq 0\}$$

The MST on R is guaranteed to be within factor 2 of the fractional optimum of this LP, so this relaxation has an integrality gap of at most 2. The gap can be arbitrarily close to 2, even for instances of the MST problem.

Now replace each undirected edge (uv) with two directed arcs $\boldsymbol{(uv)}$ and $\boldsymbol{(vu)}$, each of cost $\mathbf{c}(uv)$. Call the set of arcs $\boldsymbol{E}$. Fix an arbitrary required vertex r as root. The set of valid sets $\mathcal{U}$ are now those which contain the root but not all the required vertices. If the edges of a Steiner tree are directed to point away from the root, then at least one edge is in the cut set $\delta^+(U)$ of arcs going out of U. This gives the bidirected cut relaxation.

$$\min\{\sum_{e \in \boldsymbol{E}} \mathbf{c}(e) x_{\boldsymbol{e}} : \quad x(\delta^+(U)) \geq 1, \quad \forall U \in \mathcal{U}; \quad x \geq 0\} \tag{1}$$

We denote the optimum of the above LP on a graph G as $BCR(G)$. A graph is called *quasi-bipartite* if there are no edges between two Steiner vertices. The Steiner tree problem is NP-Hard even when restricted to the class of quasi-bipartite graphs [BP89,CC02].

Organization: In Section 2 we show the geometric theorem giving the lower bound, and other results relevant to it. In Sections 3 and 4 we give our $\sqrt{2}$ and $\frac{4}{3}$ factor approximation algorithms respectively.

2 A Geometric Lower Bound and its Consequences

We first present a special case of the geometric theorem, for the sake of ease of presentation. Let Δ_k be the unit simplex in $\mathbb{R}^k$, that is, $\Delta_k := \{x \in \mathbb{R}^k : \sum_{i \in [k]} x(i) = 1\}$, where $x(i)$ is the ith coordinate of x. The corners of Δ_k are the unit vectors in $\mathbb{R}^k$. Let T be any Steiner tree in Δ_k connecting the corners, that is, T is a tree whose vertices are the corners and any number of points in Δ_k. Define the distance between two points to be half the l_1-distance, also called the *variational distance*; for any two points $x, y \in \Delta_k$, $d(x,y) := \frac{1}{2}\sum_{i=1}^k |x(i) - y(i)|$. (The half is so that two corners are at a distance of 1). Let $d(T) := \sum_{e \in T} d(e)$. Then

Theorem 1. $d(T) \geq k - 1$.

Note that if T was a spanning tree, then the relation holds with equality since any two corners are at a distance of 1. The theorem says that Steiner points don't improve upon the MST, w.r.t $d()$. This is somewhat counter-intuitive, since in most geometric spaces, the Steiner points do improve upon the MST. What is special here is the l_1-distance, and the location of the points on the simplex.

Proof. Let $R = \{e_1, e_2, \ldots, e_k\}$ be the unit vectors in $\mathbb{R}^k$. The proof follows by a careful[4] counting argument. First of all, we get rid of the absolute values that occur in the l_1 distance. For every edge (x, y) in T, and $i \in [k]$, if x is on the T-path from y to e_i, then we lower bound $|x(i) - y(i)|$ by $x(i) - y(i)$ (and vice versa). This way the sum $\sum_{e \in T} d(e)$ is lower bounded by a linear combination of the $x(i)$'s, in which the coefficient of $x(i)$ is $\frac{1}{2}(deg_T(x) - 2)$. This is because $x(i)$ occurs with a positive sign for each edge incident at x except one, the first edge on the T-path from x to e_i. The exception is $e_i(i)$ which always occurs with a positive sign, and hence has a coefficient of $\frac{1}{2} deg_T(e_i)$. Therefore,

$$\begin{aligned}\sum_{e \in T} d(e) &\geq \frac{1}{2} \sum_{x \in T, i \in [k]} (deg_T(x) - 2)\, x(i) + \frac{1}{2} \sum_{i \in [k]} 2e_i(i) \\ &= \frac{1}{2} \sum_{x \in T} (deg_T(x) - 2) + \sum_{i \in [k]} 1 \\ &= k - 1.\end{aligned}$$

where the equality in the second line holds because $\sum_{i \in [k]} x(i) = 1$ and the last equality follows from the fact that in a tree $\sum_{x \in T} deg(x) = 2|T| - 2$. □

The general theorem allows two concessions on the location of the points: first, the points need not be on the unit simplex, all point are in the λ-simplex $\Delta_k^{(\lambda)}$, defined as $\{x \in \mathbb{R}^k : \sum_{i \in [k]} x(i) = \lambda\}$, for some parameter $\lambda > 0$. The second is that the required points need not be at the corners of the simplex. In particular, we consider any embedding z that maps every vertex $u \in V = R \cup S$ to a point $z_u \in \Delta_k^{(\lambda)}$, where $k = |R|$. Identify the required vertices with the dimensions, and define $\gamma(z) := \sum_{i \in [k]} z_i(i) - \lambda$. Note that when the required vertices are at the corners of a unit simplex, $\gamma(z) = k - 1$. As before, let T be any Steiner tree connecting all points in R, $d(u, v) = \frac{1}{2} \sum_{i=1}^{k} |z_u(i) - z_v(i)|$ and $d(T) = \sum_{e \in T} d(e)$. The above proof can be easily be modified to give:

Theorem 2. $d(T) \geq \gamma(z)$.

This theorem can be used to get a lower bound on the minimum Steiner tree as follows: Given a graph, a *valid* embedding of the vertices onto any λ-simplex is such that for all edges e, $\mathbf{c}(e) \geq d(e)$. Now given any valid embedding z, for any tree T, $\mathbf{c}(T) \geq d(T) \geq \gamma(z)$. In particular, we have

[4] An easy counting argument shows that $d(T) \geq \frac{1}{2}(k - 1)$.

Theorem 3. *If z is a valid embedding then*

$$OPT \geq \gamma(z).$$

Since the above holds for any embedding, the best lower bound is given by $\max\{\gamma(z) : z \text{ is valid}\}$. Quite interestingly, this maximum value for any graph is equal to $BCR(G)$.

Theorem 4. *Given any graph G,*

$$\max\{\gamma(z) : z \textit{ valid embedding of } G\} = BCR(G).$$

It is not too hard to see that the maximum in the above theorem can be obtained via a polynomial sized linear program, which we call the simplex-embedding LP. In fact, the dual to this program turns out to be a relaxation for the Steiner tree problem and the proof of the above theorem follows by showing its equivalence with the bidirected cut relaxation. In [GM93], Goemans and Myung provide two vertex weighted relaxations which are equivalent to the bidirected cut relaxation. Although our relaxation is different, our proof of equivalence follows on similar lines.

Proof. The simplex-embedding LP is as follows:

$$\begin{aligned}
\text{maximize} \quad & \gamma(z) = \sum_{i\in[k]} z_i(i) - \lambda && (2)\\
\text{subject to} \quad & \sum_{i\in k} z_v(i) = \lambda, \quad \forall v \in V\\
& z_v(i) - z_u(i) \leq d_i(uv), \quad \forall i \in [k], (uv) \in E\\
& z_u(i) - z_v(i) \leq d_i(uv), \quad \forall i \in [k], (uv) \in E\\
& \frac{1}{2}\sum_{i\in[k]} d_i(uv) \leq \mathbf{c}(uv), \quad \forall (uv) \in E\\
& z_v(i), d_i(uv) \geq 0, \quad \forall v \in V, i \in [k], (uv) \in E
\end{aligned}$$

Taking duals, we get the following LP (after a scaling step)

$$\begin{aligned}
\text{minimize} \quad & \sum_{e\in E} \mathbf{c}(e) x_e && (3)\\
\text{subject to} \quad & x_{uv} \geq f_i(uv) + f_i(vu), \quad \forall i \in [k],\ (uv) \in E\\
& \sum_{v:(uv)\ \in E} (f_i(uv) - f_i(vu)) \geq \alpha(v), \quad \forall i \in [k], v \in V - i\\
& \sum_{v:(iv)\ \in E} (f_i(iv) - f_i(vi)) \geq \alpha(i) + 2, \quad \forall i \in [k]\\
& \sum_{v} \alpha(v) = -2\\
& f_i(uv), f_i(vu), x_{uv} \geq 0, \quad \forall (uv)\ \in E, i \in [k]
\end{aligned}$$

To interpret LP3, one can think of x_e as capacity of edge e. There are k flows - f_i for every required vertex i each satisfying the capacity constraint and moreover, the supplies (excess flows) for f_i at every vertex v is $\alpha(v)$ (it could be negative) except at the vertex i, where it is $\alpha(i)+2$. The last equality constraint implies the total supplies sum up to zero.

Theorem 4 follows by showing there exists a feasible solution to LP 1 of value Γ if and only if there exists a feasible solution to LP 3 of value Γ. We only give a sketch for brevity and defer the details to a full version[CDV07].

LP 1 → LP 3: Let $\{y_{(\boldsymbol{uv})}\}_{(\boldsymbol{uv})\in\boldsymbol{E}}$ be a feasible solution of LP 1 of cost Γ with root r. The corresponding solution to LP 3 is as follows: $x_{uv} := y_{(\boldsymbol{uv})} + y_{(\boldsymbol{vu})}$ and $\alpha(v)$ is the *supply* at vertex v which is the difference of the outgoing $y_{(\boldsymbol{vu})}$'s and the incoming $y_{(\boldsymbol{uv})}$'s, except at r where $\alpha(r)$ is the supply -2. What remains is to describe the flows. The flow corresponding to r just mimics the $y_{(\boldsymbol{uv})}$'s. It is easy to see every constraint is till now satisfied. To get the flows corresponding to another required vertex j, we use the fact that the *minimum* $r-j$ *cut* in the digraph with arc set $\boldsymbol{E}$ and capacities $y_{(\boldsymbol{uv})}$'s is at least 1 since the solution is feasible for LP 1. This implies there is a standard flow g_{rj} from r to j in this di-graph. The flow f_j is found by *subtracting* $2g_{rj}$ from f_r. Note that this changes the supplies of no vertex other than r and j, and for these, it changes as it should change. Thus the solution is feasible for LP3 and is of value Γ.

LP 3 → LP 1: Let $(\{x\},\{f_i\},\{\alpha\})$ (respectively over edges, arcs and vertices) be a solution to LP 3. WLOG by adding circulations if necessary, we can assume for all edges (uv), and i we have $x_{uv} = f_i((\boldsymbol{uv})) + f_i((\boldsymbol{vu}))$. For LP 1, let r be the chosen root. Then the solution is: $y_{(\boldsymbol{uv})} := f_r((\boldsymbol{uv}))$. To see feasibility for LP 1, we must show across any cut S separating r and a required vertex j, we have $\sum_{(\boldsymbol{uv})\in\delta^+(S)} f_r((\boldsymbol{uv})) \geq 1$. To see this, consider the flow g_{rj}: the difference between f_r and f_j. To be precise:

$$g_{rj}((\boldsymbol{uv})) = \max[0, f_r((\boldsymbol{uv})) - f_j((\boldsymbol{uv}))] + \max[0, f_j((\boldsymbol{vu})) - f_r((\boldsymbol{vu}))]$$

It is not hard to see for any arc $(\boldsymbol{uv})$, $g_{rj}(uv) \leq 2f_r((\boldsymbol{uv}))$. The proof sketch ends by noting g_{rj} is a standard flow from r to j of value 2, implying across any cut S as above, 2 units (and hence at least 1 unit of f_r) of g_{rj} would go across. □

Interestingly, the above theorems hold even when the Steiner vertices are embedded "above" the simplex. That is, if we have for all Steiner vertices v, $\sum_{i\in[k]} z_v(k) \geq \lambda$ rather than equality. Maximizing over these embeddings give us a better lower bound on OPT and in fact, as we see in Figure 1, sometimes it is strictly better. Thus, we obtain an LP relaxation which could be *tighter* than the bidirected cut relaxation. Although the remainder of the paper does not concern this relaxation, it is an intriguing question if the integrality gap of this revised LP is strictly smaller than that of the bidirected cut relaxation.

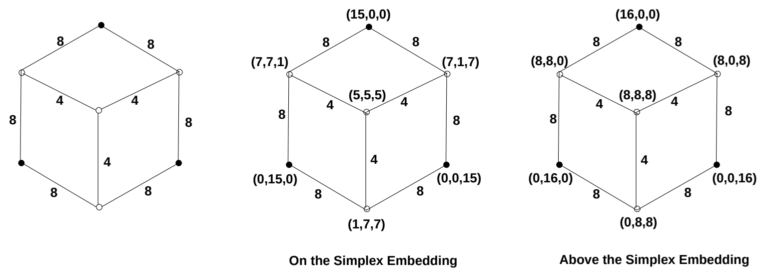

Fig. 1. Integrality gap of the bidirected cut relaxation for the graph is known to be 16/15 (due to Goemans). The middle figure shows an embedding on the simplex attaining a value of 15. The figure to the right shows how we can get a higher value if we allow Steiner vertices to move above the simplex. Note that the Steiner vertex at the center is not on the 16-simplex.

Remark: We should remark that the idea of embedding vertices of a graph onto a simplex is not new. Calinescu et.al.[CKR98] use a similar LP to obtain approximation algorithms for the multi-way cut problem. However, a key difference is that theirs is a primal relaxation while ours is dual. It is not clear if a certain duality between the two LP's can be established.

2.1 An Embedding Algorithm

In this section, we describe a dual growing procedure EMBED which given a quasi-bipartite graph G and a cost function $\mathbf{c}$ does the following.

Case 1: If $MST(R)$ is the optimal Steiner tree, then it returns a feasible embedding z such that $\gamma(z) = MST(R)$. Note, in this case, $MST(R) = OPT = BCR$, since $MST(R) \geq OPT \geq BCR \geq \gamma(z)$.

Case 2: Or, returns a Steiner vertex v whose addition *strictly* helps the MST on R, that is, $MST(R \cup v) < MST(R)$. We say that EMBED *crystallizes* v.

The following theorem is immediate.

Theorem 5. *Given a quasi-bipartite instance G, if the addition of no Steiner vertex reduces the cost of $MST(R)$, then $MST(R) = BCR$. In particular the integrality gap for the instance is* 1.

Note that the theorem implies the following important property about the bidirected cut relaxation for quasi-bipartite graphs: If the minimum spanning tree is optimal, then the relaxation is exact. Only the above property of EMBED is used beyond this section. The rest of the section describes EMBED.

Notation: *Given an embedding $z : V \to \Delta_k^{(\lambda)}$ and the distance $d(\cdot,\cdot)$ induced by it, call an edge (u, v)* tight *if $d(u, v) = \mathbf{c}(uv)$. Call (u, v) under-tight or over-tight if the distance is strictly smaller or larger, respectively.*

The following is a continuous description of the algorithm, which can be easily discretized. The algorithm has a notion of time. It starts at time $t = 0$ and time increases at unit rate. At any time t, all required vertices are on the t-simplex, all Steiner vertices are below the t-simplex, and no edge is over-tight. The algorithm maintains a set of tight terminal-terminal edges T, which form a forest at any time t. Let K denote a connected component of required vertices formed with the edges of T. At time $t = 0$, the algorithm starts with $T = \varnothing$ and the components are singleton required vertices. All vertices start at the origin at $t = 0$.

Required vertices: For each component K and required vertex $i \in K$, the algorithm increases the jth coordinate of i at rate $1/|K|$, for each $j \in K$. Clearly, this will keep required vertex i on the t-simplex. When an edge (ij) goes tight, the algorithm merges the components containing i and j and adds (ij) to T. It is instructive to note that when restricted to only required vertices, this actually mimics Kruskal's MST algorithm.

Steiner vertices: A Steiner vertex v remains at the origin until it *links* to a required vertex. It links to required vertex i at time $t = \mathbf{c}(iv)$, if it is not already linked to another required vertex in the same component as i. The edge (iv) is called a *link*. We say that v is linked to a component K if it is linked to any required vertex in K. For each component K that v is linked to, the coordinates of v corresponding to K increase at rate $1/|K|$.

The algorithm terminates if the number of components becomes 1 (Case 1) or a Steiner vertex v hits the simplex (Case 2). The example in Figure 2 illustrates the algorithm on a graph with three required vertices.

We now show the above procedure satisfies the conditions. In Case 1, the algorithm returns the embedding obtained after running the following projection step. If Case 1 happens at time $t = \lambda$, then the algorithm projects each Steiner vertex onto the λ-simplex. For every Steiner vertex v and coordinate j, $z_v(j) \leftarrow z_v(j)\frac{\lambda}{||z_v||_1}$. The coordinates of the required vertices are kept the same. It is easy to show that z is feasible, that is, no edge is over-tight. We need to show that tree T has cost $\gamma(z)$. In fact we prove something stronger. Given any connected component K, denote the restriction of T to K as $T[K]$.

Lemma 1. *At any instant of time t, for any connected component K,* $\mathbf{c}(T[K]) = \sum_{i \in K} z_i(i) - t$.

Proof. At time $t = 0$, the lemma holds vacuously. Since the quantity $\sum_{i \in K} z_i(i)$ increases at the same rate as time, we need to prove the lemma only in the time instants when two components merge. Suppose K, K' merge at time instant t due to edge (ij) which comes in the tree, with $i \in K, j \in K'$. Note $d(i,j) = \mathbf{c}(ij) = t$. So for the new connected component $K \cup K'$, $\sum_{i \in K \cup K'} z_i(i) - t = \sum_{i \in K} z_i(i) - t + \sum_{i \in K'} z_i(i) - t + t = \mathbf{c}(T[K]) + \mathbf{c}(T[K']) + \mathbf{c}(ij) = \mathbf{c}(T[K \cup K'])$. $\square$

In Case 2, when v hits the simplex, the algorithm returns v as the Steiner vertex helping the minimum spanning tree. In fact, we show that if v is linked

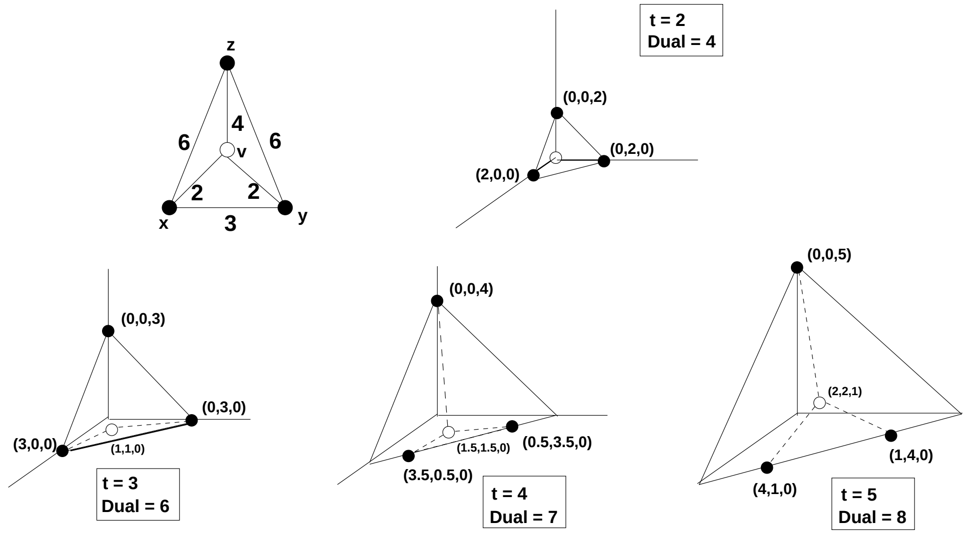

Fig. 2. Snapshots of the running of EMBED on the graph above at times $t = 2, 3, 4, 5$. At time $t = 2$, the Steiner vertex v links to the required vertices x and y, and increases its x and y coordinates at rate 1. At time $t = 3$, x, y merge. The edge (x, y) goes into $Remove(v)$. At time $t = 4$, v links to z, and moves in the zth coordinate as well. At $t = 5$, it hits the 5-simplex, terminating the algorithm. The tree shown with dotted lines pays exactly for the dual and is cheaper than the MST.

to $K_1, \cdots, K_r$ when it hits the simplex, then v helps the MST of the required vertices in $P = \bigcup_l K_l$. This suffices since $\bigcup_l T[K_l]$ can be extended to an MST of R.

With each Steiner vertex v, we associate a subset of edges $Remove(v)$ of T. Suppose v is linked to K and K' and these merge at time t, due to edge (ij), $i \in K$ and $j \in K'$. At this point, (ij) is added to the set $Remove(v)$. Thus, a Steiner vertex may have more than one link into the same component, but for each extra link, there is an edge in $Remove(v)$. Let T_v be the tree formed by adding all the links incident at v to $\bigcup_l T[K_l]$ and deleting $Remove(v)$. The proof of the following lemma is very similar to that of Lemma 1.

Lemma 2. *At any instant of time,* $c(T_v) = \sum_{i \in P} z_i(i) - \sum_{i \in P} z_v(i)$.

Hence when v hits the simplex, $\sum_{i \in P} z_v(i) = t$, and so $c(T_v) = \sum_{i \in P} z_i(i) - t < MST(P)$.

Remark: Note that the above algorithm and analysis do not use the fact the cost satisfies triangle inequality. We would need this for our algorithms to work.

3 The $\sqrt{2}$ Factor Approximation Algorithm

Notation 1 $MST(U; \mathbf{c})$ *denotes the minimum cost spanning tree on vertices U given the costs c. Based on the context, it also denotes the cost of this tree.*

We start by giving a high level idea of our algorithm. The algorithm will return a cost $\mathbf{c}_2$ and a subset of Steiner vertices $X \subseteq S$ such that

1. The optimal Steiner tree w.r.t. $\mathbf{c}_2$ is the MST. Equivalently, EMBED when run on $G, \mathbf{c}_2$ terminates with a feasible embedding z with $\gamma(z) = MST(R; \mathbf{c}_2)$.
2. $MST(X \cup R; \mathbf{c}) \leq \sqrt{2} \cdot MST(R; \mathbf{c}_2)$

The costs $\mathbf{c}_2$ will be only smaller than $\mathbf{c}$; therefore, z is also feasible for $\mathbf{c}$. Hence, the two conditions imply that we get a factor $\sqrt{2}$ approximation.

Initially, $X = \emptyset$ and we obtain $\mathbf{c}_2$ by reducing the costs of the required-required edges by a factor of $\sqrt{2}$ and leaving the costs of required-Steiner edges unchanged. We denote the reduced cost at this point as $\mathbf{c}_1$ which we use later. Clearly Condition 2 is satisfied now, and will remain an *invariant* of the algorithm.

Suppose that condition 1 is not satisfied, that is, when EMBED is run on $G, \mathbf{c}_2$, a Steiner vertex $v \in S$ is crystallized. At this point, the algorithm adds v to X, and will modify $\mathbf{c}_2$ by reducing the costs of certain required-required edges further, as detailed below. This has the effect that if EMBED is run with these new costs, v does not crystallize, while still maintaining the invariant. Hence in each iteration, a new Steiner vertex is added to X, implying termination in at most $|S|$ rounds.

We now give the intuition behind modifying the costs so that the invariant is maintained. The first step of scaling all the required-required edges acts as a "global filter" which filters out Steiner vertices that only help a little. If a Steiner vertex v does crystallize, then adding it to X reduces the cost of $MST(R \cup X; \mathbf{c})$ so much that decreasing the cost of required-required edges "local" to it to $\frac{1}{2}$ of the original costs still maintains the invariant. This requires an involved argument (Theorem 6) that amortizes the improvements due to all the vertices previously added to X. This has the additional required effect that v itself is filtered out.

Now the formal description of the algorithm follows.

Definition 1. *Applying the* global filter *with parameter $\rho > 1$ gives a cost $\mathbf{c}_1$ defined as $\mathbf{c}_1(ij) = \frac{\mathbf{c}(ij)}{\rho}$ for all $i, j \in R$, and $\mathbf{c}_1(iv) = \mathbf{c}(iv)$ for all $i \in R$ and $v \in S$.*

Definition 2. *Applying a* local *filter w.r.t $X \subseteq S$ gives a cost $\mathbf{c}_2$. Let $T_1 = MST(R \cup X; \mathbf{c}_1)$, and for each $u \in X$, $Clos(u)$ denote the closest required vertex to u. The cost $\mathbf{c}_2$ after applying the local filter w.r.t X is defined as $\mathbf{c}_2(Clos(u), j) = \frac{1}{2}\mathbf{c}(Clos(u), j)$ (half the original cost), for every $u \in X$ and $j \in R$ ($j \neq Clos(u)$) that is adjacent to u in T_1. $\mathbf{c}_2(e) = \mathbf{c}_1(e)$ (the global filter is retained) otherwise.*

Algorithm Primal-Dual

1. Apply global filter with parameter $\rho = \sqrt{2}$ to get $\mathbf{c}_1$.
   Initialize $X \leftarrow \varnothing$; $\mathbf{c}_2 \leftarrow \mathbf{c}_1$.
2. **Repeat** till Embed returns z
   Run Embed on $G, \mathbf{c}_2$.
   If Embed returns v then
   $X = X \cup v$; Apply local filter w.r.t X to get $\mathbf{c}_2$.
3. Return $T_1 = MST(R \cup X; \mathbf{c}_1), z$.

Theorem 6. *The algorithm* Primal-Dual *terminates in at most* $|S|$ *rounds, returning a Steiner tree* T_1 *and a feasible embedding* z *of* $G, \mathbf{c}$ *such that* $\mathbf{c}(T_1) \leq \sqrt{2} \cdot \gamma(z) \leq \sqrt{2} \cdot OPT$.

Proof. (Sketch) Let $T_1 = E_0 \cup E_1$, where E_0 denotes the required-Steiner edges and E_1 denotes the required-required edges of T_1. We bound the costs of these two sets separately. Let E_2 be the set of edges modified by the local filter, that is, e such that $\mathbf{c}_2(e) = \frac{1}{2}\mathbf{c}(e)$. Define T_2 to be $E_2 \cup E_1$. It can be shown that T_2 is an MST with costs $\mathbf{c}_2$, and hence $\mathbf{c}_2(T_2) = \gamma(z)$. We have $\mathbf{c}(T_1) = \mathbf{c}(E_0) + \mathbf{c}(E_1)$, $\mathbf{c}_2(T_2) = \mathbf{c}_2(E_2) + \mathbf{c}_2(E_1)$ and it is enough to prove that

– $\mathbf{c}(E_0) \leq \sqrt{2}\mathbf{c}_2(E_2)$.
  This is essentially a consequence of the observation that $\mathbf{c}_1(T_1) \leq MST(R; \mathbf{c}_1)$. Since $T_2 = E_1 \cup E_2$ is also a spanning tree of R, we get $\mathbf{c}_1(T_1) \leq \mathbf{c}_1(T_2)$. Expanding the costs, we get

$$\mathbf{c}_1(E_0) + \mathbf{c}_1(E_1) \leq \mathbf{c}_1(E_2) + \mathbf{c}_1(E_1).$$

  Since E_0 are required vertex-Steiner edges, $\mathbf{c}_1(E_0) = \mathbf{c}(E_0)$. $\mathbf{c}_1(E_2) = \mathbf{c}(E_2)/\sqrt{2} = \sqrt{2}\mathbf{c}_2(E_2)$ by definition, giving us $\mathbf{c}(E_0) \leq \sqrt{2}\mathbf{c}_2(E_2)$.
– $\mathbf{c}(E_1) \leq \sqrt{2}\mathbf{c}_2(E_1)$.
  Since E_1 costs are not modified by the local filter, $\mathbf{c}_2(E_1) = \mathbf{c}_1(E_1)$ and in fact the relation holds with equality. $\square$

In fact, the above algorithm has a faster implementation. Although the algorithm constructs the set X in a certain order, it turns out that the order does not matter. Hence it is enough to simply apply the global filter and go through the Steiner vertices (in any order) once, picking the ones that help.

Algorithm Reduced One-Pass Heuristic

1. Apply global filter with parameter $\rho = \sqrt{2}$ to get $\mathbf{c}_1$.
   Initialize $X \leftarrow \varnothing$;
2. For all $v \in S$,
   If $MST(R \cup X \cup v; \mathbf{c}_1) < MST(R \cup X; \mathbf{c}_1)$, then
   $X = X \cup v$;
3. Return $T_1 = MST(R \cup X; \mathbf{c}_1)$.

Theorem 7. *There exists a feasible embedding z of $G, \mathbf{c}$ such that for T_1 returned by Algorithm* Reduced One-Pass Heuristic, $\mathbf{c}(T_1) \leq \sqrt{2} \cdot \gamma(z)$.

The proof of Theorem 7 is similar to Theorem 6. Note that the above algorithm makes at most $|S|$ minimum spanning tree computations and is hence is very efficient. In particular, it runs in strongly polynomial time.

4 The $\frac{4}{3}$ Factor Approximation Algorithm

The primal-dual $\frac{4}{3}$ approximation algorithm is along the lines of the one in the previous section, with the major difference being that it drops Steiner vertices from X when beneficial. The other differences are that it applies the global filter with $\rho = 4/3$, and the definition of a local filter is somewhat different. And like the earlier algorithm, the order of vertices picked/dropped does not matter. As a result it can be implemented as a local search algorithm with an extra global filtering step, which is what we present here.

Algorithm Reduced-Local-Search

1. Apply global filter with parameter $\rho = 4/3$ to get $\mathbf{c}_1$.
   Initialize $X \leftarrow \varnothing$, $T_1 = MST(R; \mathbf{c}_1)$;
2. Repeat
   If $\exists v$ such that $MST(R \cup X \cup v; \mathbf{c}_1) < \mathbf{c}_1(T_1)$, $X = X \cup v$.
   If $\exists v$ such that $MST(R \cup X \setminus v; \mathbf{c}_1) < \mathbf{c}_1(T_1)$, $X = X \setminus v$.
   $T_1 = MST(R \cup X; \mathbf{c}_1)$.
   Until No such v exists.
3. Return T_1.

The plain local search algorithm (without the global filtering step) was studied [RV99] who showed that this algorithm gives a 3/2 factor approximation for quasi-bipartite graphs. This factor is tight. So the simple modification of applying a global filter provably improves the performance of this algorithm. It was shown in [Riz03] that this algorithm can be implemented efficiently.

We show that T_1 returned by the algorithm is within 4/3 of the optimal by exhibiting an embedding z of value greater than 3/4 times the cost of T_1. As in Section 3, the analysis proceeds by defining cost $\mathbf{c}_2$ and constructing tree T_2. The factor 4/3 comes from the parameter ρ used in the global filter and the following property of T_1.

Lemma 3. *The degree of every Steiner vertex in T_1 is at least 4.*

Proof. It is easy to see that T_1 doesn't have vertices of degree 1 or 2. Suppose there existed a Steiner vertex $v \in T_1$ with $deg(v) = 3$. Let a, b, c be the required vertices connected to v and assume $\mathbf{c}_1(va) \leq \mathbf{c}_1(vb) \leq \mathbf{c}_1(vc)$ without loss of generality. Now by triangle inequality property of $\mathbf{c}$, we know $\mathbf{c}(va) + \mathbf{c}(vb) \geq \mathbf{c}(ab)$. Since $\mathbf{c}(va) = \mathbf{c}_1(va)$ and $\mathbf{c}(vb) = \mathbf{c}_1(vb)$, we get $\frac{3}{4}(\mathbf{c}_1(va) + \mathbf{c}_1(vb)) \geq \frac{3}{4}\mathbf{c}(ab) = \mathbf{c}_1(ab)$. Similarly $\frac{3}{4}(\mathbf{c}_1(va) + \mathbf{c}_1(vc)) \geq \mathbf{c}_1(ac)$. Thus $\mathbf{c}_1(ab) + \mathbf{c}_1(ac) \leq \frac{3}{4}(2\mathbf{c}_1(va) + \mathbf{c}_1(vb) + \mathbf{c}_1(vc)) \leq \mathbf{c}_1(va) + \mathbf{c}_1(vb) + \mathbf{c}_1(vc)$. Thus $MST(R \cup X)$ would choose (ab) and (ac), rather than choosing $(va), (vb), (vc)$. □

Theorem 8. *For the tree T_1 returned by* Reduced-Local-Search*, there exists a feasible embedding z such that $\mathbf{c}(T_1) \leq \frac{4}{3} \cdot \gamma(z)$.*

Proof. (Sketch) As in the proof of Theorem 6, denote the edges of T_1 as $E_1 \cup E_0$. Define $\mathbf{c}_2$ as: For every Steiner vertex $v \in T_1$ and for every $j \neq Clos(v)$ connected to v in T_1, let $\mathbf{c}_2(Clos(v), j) = \mathbf{c}_1(vj)$. Note that $\mathbf{c}_1(vj) \leq \mathbf{c}_1(Clos(v), j)$, for otherwise T_1 would have picked $(Clos(v), j)$ instead of (vj). Call these required vertex-required vertex edges *diminished*. For every other edge, $\mathbf{c}_2(e) := \mathbf{c}_1(e)$. Let E_2 be the set of diminished edges and let $T_2 := E_1 \cup E_2$, be a required vertex spanning tree. By the conditions of the algorithm, since T_1 is an MST of $R \cup X$ with costs $\mathbf{c}_1$ and no Steiner vertices help X, T_2 is an MST of R with costs $\mathbf{c}_2$ and no Steiner vertex helps T_2. Thus, by Theorem 5 running Embed on $G, \mathbf{c}_2$ returns a feasible embedding z of value $\mathbf{c}_2(T_2)$. We now bound the cost of T_1.

We have $\mathbf{c}(T_1) = \mathbf{c}(E_1) + \mathbf{c}(E_0) = \mathbf{c}(E_1) + \mathbf{c}_1(E_0)$. Note that $\mathbf{c}_2(T_2) = \mathbf{c}_1(E_1) + \mathbf{c}_2(E_2)$ since E_1 is not diminished. As in the proof of Theorem 6, we argue term by term. By definition we have $\mathbf{c}(E_1) = \frac{3}{4}\mathbf{c}_1(E_1)$.

Every Steiner vertex $v \in T_1$ contributes $deg(v) - 1$ edges to E_2 and $deg(v)$ edges in E_0, where $deg(v)$ is the degree of v in T_1. By definition the $deg(v) - 1$ edges have cost exactly the cost of the largest $deg(v) - 1$ edges of the $deg(v)$ edges it contributes to E_0. By lemma 3, $deg(v) \geq 4$ and thus we get $\mathbf{c}_1(E_0) \leq \frac{3}{4}\mathbf{c}_2(E_2)$. Adding, we get $\mathbf{c}(T_1) \leq \frac{4}{3}\mathbf{c}_2(T_2) = \frac{4}{3}\gamma(z)$. □

5 Discussion

Clearly the most important question to address is whether the geometric approached to the bidirected cut relaxation describe here can be extended to general graphs. In fact, there is a natural generalization of the Embed procedure described above to the case where there are Steiner-Steiner edges; however, it has not yielded any results for the general case. As noted above, one crucial property possessed by quasi-bipartite graphs is Theorem 5: if the spanning tree is optimal, then the relaxation is exact. However, this property is not satisfied by general graphs. An interesting question would be upper bounding the gap in such instances, and then perhaps our techniques of reducing costs may be useful.

References

[AC04] A. Agarwal and M. Charikar. On the advantage of network coding for improving network throughput. In *Proceedings of the IEEE Information Theory Workshop*, 2004.

[BP89] M. Bern and P. Plassman. The Steiner problem with edge lengths 1 and 2. *Inform. Process. Lett.*, 32:171–176, 1989.

[CC02] M. Chlebik and J. Chlebikova. Approximation hardness of the steiner tree problem on graphs. *Proceedings of Scandinavian Workshop on Algorithm Theory*, 2002.

[CDV07] D. Chakrabarty, N. R. Devanur, and V. V. Vazirani. New geometry-inspired relaxations and algorithms for the Metric Steiner Tree Problem. *Available at* `http://www.cc.gatech.edu/d̃eepc`, 2007.

[CKR98] G. Calinescu, H. Karloff, and Y. Rabani. An improved approximation algorithm for multiway cut. *STOC*, 1998.

[CR94a] S. Chopra and M. R. Rao. The Steiner tree problem I: Formulations, compositions and extension of facets. *Math. Programming*, 64:209–229, 1994.

[CR94b] S. Chopra and M. R. Rao. The Steiner tree problem II: Properties and classes of facets. *Math. Programming*, 64:231–246, 1994.

[CRS96] R. Courant, H. Robbins, and I. Stewart. **What Is Mathematics?: An Elementary Approach to Ideas and Methods**. Oxford Papebacks, 1996.

[Edm67] J. Edmonds. Optimum branchings. *Journal of Research of the National Bureau of Standards. Section B*, 71:233–240, 1967.

[GM93] M. Goemans and Y. Myung. A catalog of Steiner tree formulations. *NETWORKS*, 23:19–23, 1993.

[Goe96] M. Goemans. Personal communication. 1996.

[HRW92] F. K. Hwang, D. S. Richards, and P. Winter. **The Steiner Tree Problem**, volume 53 of *Annals of Discrete Mathematics*. North-Holland, Amsterdam, Netherlands, 1992.

[IT94] A. O. Ivanov and A. A. Tuzhilin. **The Steiner problem and its generalizations**. CRC Press, BocaRaton, Ann Arbor, London, Tokyo, 1994.

[JV02] K. Jain and V. V. Vazirani. Equitable cost allocations via primal-dual-type algorithms. In *Proceedings of 33rd ACM Symposium on Theory of Computing*, 2002.

[KPT] J. Konemann, D. Pritchard, and K. Tan. A partition based relaxation for Steiner trees. *Manuscript*.

[Riz03] R. Rizzi. On Rajagopalan and Vazirani's 3/2-approximation bound for the iterated 1-Steiner heuristic. *Information Processing Letters*, 86:335–338, 2003.

[RV99] S. Rajagopalan and V. Vazirani. On the bidirected cut relaxation for the metric Steiner tree problem. In *Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*, 1999.

[RZ05] G. Robins and A. Zelikovsky. Tighter bounds for graph Steiner tree approximation. *SIAM Journal of Discrete Mathematics*, 19:122–134, 2005.

[Won84] R. T. Wong. A dual ascent approach for Steiner trees on a directed graph. *Mathematical Programming*, 28:271–287, 1984.