Simple Cost Sharing Schemes for Multicommodity Rent-or-Buy and Stochastic Steiner Tree

Lisa Fleischer* Jochen Könemann† Stefano Leonardi‡ Guido Schäfer§

ABSTRACT

In the multi-commodity rent-or-buy network design problem (MRoB) we are given a network together with a set of k terminal pairs $R = \{(s_1,t_1),\ldots,(s_k,t_k)\}$. The goal is to install capacities on the edges of the network so that a prescribed amount of flow f_i can be routed between all terminal pairs s_i and t_i simultaneously. We can either *rent* capacity on an edge at some cost per unit flow or *buy* infinite capacity on an edge at some larger fixed cost. The overall objective is to install capacities at a minimum total cost.

The version of the stochastic Steiner tree problem (SST) considered here is the Steiner tree problem in the model of two-stage stochastic optimization with recourse. In stage one, there is a known probability distribution on subsets of vertices and we can choose to buy a subset of edges at a given cost. In stage two, a subset of vertices T from the prior known distribution is realized, and additional edges can be bought at a possibly higher cost. The objective is to buy a set of edges in stages one and two so that all vertices in T are connected, and the expected cost is minimized.

Gupta et al. (FOCS '03) give a randomized scheme for the MRoB problem that was both used subsequently to improve the approximation ratio for this problem, and extended to yield the best approximation algorithm for SST. One building block of this scheme is a good approximation algorithm for Steiner forests.

We present a surprisingly simple 5-approximation algorithm for MRoB and 6-approximation for SST, improving on the best previous guarantees of 6.828 and 12.6, and show that no approximation ratio better than 4.67 can be achieved using the above mentioned randomized scheme in combination with the currently best known Steiner forest approximation algorithms. A key component of our approach are cost shares that are 3-strict for the *unmodified* primal-dual Steiner forest algorithm.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Non-numerical Algorithms and Problems

General Terms

Algorithms, Design, Theory

Keywords

approximation algorithms, stochastic optimization, cost sharing

1. INTRODUCTION

Multicommodity Rent-or-Buy. In the *multi-commodity rent-or-buy problem* (MRoB) we are given an undirected graph $G = (V,E)$, terminal pairs $R = \{(s_1,t_1),\ldots,(s_k,t_k)\} \subseteq V \times V$, non-negative costs c_e for all edges $e \in E$, and a parameter $M \geq 0$. The goal is to install capacities on the edges of G such that for all $(s_i,t_i) \in R$ we can simultaneously route a given amount of flow f_i from s_i to t_i. We can either *rent* capacity on an edge e at cost $\lambda(e) \cdot c(e)$, where $\lambda(e)$ is the flow traversing edge e, or *buy* infinite capacity on edge e at cost $M \cdot c(e)$. Bought edges have no incremental, flow-dependent cost. The overall objective is to find a feasible solution of smallest total cost.

The MRoB problem generalizes a number of fundamental optimization problems. For $M = \infty$, an optimum solution for an MRoB instance can be found by connecting each pair of terminals by their shortest path.

For $M = 1$, MRoB reduces to the Steiner forest problem. The *Steiner forest problem* is to compute a minimum-cost forest that contains an s_i,t_i-path for all $1 \leq i \leq k$. It is well-known that this problem is NP-hard [9] and even Max-SNP hard [7]. The best known approximation algorithm achieves a performance guarantee of $2 - 1/k$ and is due to Agrawal, Klein and Ravi [3]. Goemans and Williamson [15] generalize these results to a larger class of network design problems.

The MRoB problem is a generalization of the *single-commodity rent-or-buy* problem (SRoB). Here, in addition to the input given in an instance of the MRoB problem, one also has a *root node* $r \in V$.

*Supported in part by NSF grant CCF-0515127. P. O. Box 218, T. J. Watson Research Ctr., IBM, Yorktown Heights, NY 10598, USA. Email: lkf@watson.ibm.com.

†Department of Combinatorics and Optimization, University of Waterloo, 200 University Avenue West, Waterloo, ON N2L 3G1, Canada. Supported by NSERC grant no. 288340-2004 and by an IBM Faculty Award. Email: jochen@uwaterloo.ca.

‡Dipartimento di Informatica e Sistemistica, University of Rome "La Sapienza", Via Salaria 113, 00198 Rome, Italy. Part of this work was done while the author was visiting the School of Computer Science at Carnegie Mellon University. Email: leon@dis.uniroma1.it.

§Supported by the DFG Research Center Matheon "Mathematics for key technologies". Institute for Mathematics, Technical University Berlin, Straße des 17. Juni 136, 10623 Berlin, Germany. Email: schaefer@math.tu-berlin.de.

STOC'06, May21–23, 2006, Seattle, Washington, USA.

The root r is part of every terminal pair, i.e., $r \in \{s_i, t_i\}$ for all $1 \le i \le k$. Gupta et al. [11] gave a randomized 3.55-approximation algorithm for the problem.

Kumar, Gupta and Roughgarden [16] give the first constant-factor approximation algorithm for the MRoB problem. Based on the techniques used by Gupta et al. [11] for the single-commodity rent-or-buy problem, Gupta et al. [10] present a 12-approximation algorithm for the MRoB problem. Becchetti et al. [6] recently obtained the currently best known 6.828-approximation algorithm for this problem.

The MRoB problem is a special case of the *multicommodity buy-at-bulk* (MBaB) problem. The input in this problem is as in the MRoB problem, except for an additional sub-additive monotone function $l : \mathbb{Z}^+ \to \mathbb{R}^+$. A feasible solution consists of a vector $x \in \mathbb{Z}^+_{|E|}$ of edge-capacities that allows for f_i units of flow to be routed between s_i and t_i, for all $(s_i, t_i) \in R$ simultaneously, and feasibly. The cost of installing capacities x is $\sum_{e \in E} l(x_e) c_e$ and the goal is to find a feasible capacity installation x of minimum total cost.

In [2], Awerbuch and Azar present an $O(\alpha)$-approximation for MBaB, assuming that any metric can be probabilistically approximated by a family of tree metrics with an expected distortion at most α. In [4], Bartal shows $\alpha = O(\log^2 n)$ and improves this bound in [5] to $\alpha = O(\log n \log\log n)$. More recently, Fakcharoenphol et al. [8] show that $\alpha = O(\log n)$. Recently, Andrews [1] shows that the results in [2] and [8] are best possible up to constant factors unless $\mathtt{NP} \subseteq \mathtt{ZPTIME}(n^{\mathrm{polylog}(n)})$.

Stochastic Steiner Tree. The *stochastic Steiner tree problem (SST)* we consider here is the Steiner tree problem in the model of two-stage stochastic optimization with recourse. In stage one, there is a known probability distribution π on subsets of vertices and we can chose to buy a subset of edges at a given cost. In stage two, a subset of vertices T from the prior known distribution is realized, and additional edges can be bought at a possibly higher cost. The objective is to buy a set of edges in stages one and two so that all vertices in T are connected, and the expected cost is minimized.

We make no assumptions about the distribution π on the subset of vertices, except that we have access to it via a *sampling oracle*: on request, the oracle outputs a subset of vertices T drawn from the distribution. Gupta and Pál give a 12.6-approximation for SST [12]. Prior to their work, there were constant factor guarantees for the problem when all of the possible subsets realized in stage two contain a fixed root terminal [13, 14].

Common Framework. Our work uses a common framework developed in Gupta et al. [10] for MRoB and extended by Gupta and Pál in [12] for SST. This framework first chooses a random subset $S \subseteq R$ of the set of terminal pairs. It then computes an approximate Steiner forest F_S for S using an adaptation of the primal-dual algorithm for Steiner forests and buys its edges. Finally, this forest is augmented to a feasible solution for R by renting additional edges in a cheapest possible way such that all remaining terminals in $R \setminus S$ are connected.

The performance of the above framework depends strongly on a certain *stability* property of the Steiner forest algorithm used to compute F_S. For a forest F in G, let $G|F$ denote the graph resulting from contracting all trees of F. We use $c_{G|F}(u,v)$ to denote the minimum cost of any u,v-path in $G|F$. For a parameter $\beta > 0$, Gupta et al. [10] define the notion of β-*strict* algorithms for the minimum-cost Steiner forest problem:

DEFINITION 1. *An algorithm* ALG *for the Steiner forest problem is* β-strict *if there exist cost shares* ξ_{st} *for all* $(s,t) \in R$ *such that*

1. $\sum_{(s,t) \in R} \xi_{st} \le \mathtt{opt}_R$, *where* $\mathtt{opt}_R$ *is the minimum cost of a Steiner forest for* R, *and*
2. $c_{G|F_{-st}}(s,t) \le \beta \cdot \xi_{st}$ *for all* $(s,t) \in R$, *where* F_{-st} *is a Steiner forest for terminal set* $R_{-st} = R \setminus \{(s,t)\}$ *returned by* ALG.

Gupta et al. [10] then show that using an α-approximate and β-strict Steiner forest algorithm in their framework yields an $(\alpha+\beta)$-approximation algorithm for the MRoB problem. The authors devise a 6-approximate and 6-strict algorithm for Steiner forests which yields a 12-approximate algorithm for MRoB. Their analysis can be tightened to achieve an 8-approximation. Becchetti et al. [6] reduced the approximation ratio to 6.828 by devising a $(2+\sqrt{2})$-approximate and $(2+\sqrt{2})$-strict primal-dual Steiner forest algorithm.

The notion of strictness defined above assumes that R is a set of terminal pairs. To extend this framework to handle SST, Gupta and Pál extend the notion of strictness to a set R of terminal subsets of arbitrary size, called *groups*. For a group $g \in R$, let $c_{G|F}(g)$ denote the minimum cost of connecting all terminals of g in $G|F$.

DEFINITION 2. *An algorithm* ALG *for the Steiner forest problem is* β-group-strict *if there exist cost shares* ξ_g *for all* $g \in R$ *such that*

1. $\sum_{g \in R} \xi_g \le \mathtt{opt}_R$, *where* $\mathtt{opt}_R$ *is the minimum cost of a Steiner forest for* R, *and*
2. $c_{G|F_{-g}}(g) \le \beta \cdot \xi_g$ *for all* $g \in R$, *where* F_{-g} *is a Steiner forest for terminal set* $R_{-g} = R \setminus \{g\}$ *returned by* ALG.

The algorithms in [6], [10], and [12] all adapt the primal-dual Steiner forest algorithm from [3]. In these papers, strictness is achieved by adding extra edges into the Steiner forest produced by the standard primal-dual algorithm. This worsens the approximation ratio but reduces the cost of augmenting a feasible forest F_{-g} into a feasible forest for R.

Our Results. In this paper, we show that the primal-dual algorithms for Steiner forest [3, 15] are 3-strict and 4-group-strict with appropriate cost sharing rules. We summarize our main contribution in the following theorem:

THEOREM 1. *There exists a primal-dual* 2*-approximate algorithm for the Steiner forest problem that is* 3*-strict and* 4*-group-strict.*

This implies a 5-approximation for MRoB and a 6-approximation for SST using the framework in [10, 12]. Moreover, this also implies a 5-approximation algorithm for the 2-stage stochastic Steiner forest problem in the independent decisions model [13].

This is the first algorithm to show that the *unmodified* primal-dual Steiner forest algorithm has *constant* strict or group-strict cost shares. Finally, we present an example instance that shows that the natural primal-dual Steiner forest algorithm is not $(\frac{8}{3} - \varepsilon)$-strict for any $\varepsilon > 0$, therefore showing that the two-stage analysis of Gupta et al. given in [10] is nearly tight for MRoB.

Outline of Paper. In Section 2, we define the structure of our cost shares and give a surprisingly simple property that implies 2α-group-strictness if the cost shares are based on an α-approximate Steiner forest. In Section 2, we also present our improved $\frac{3}{2}\alpha$-strictness result for cost shares that satisfy an additional requirement. In Section 3, we review the Steiner forest algorithm AKR and show how the cost of every edge of the forest that is computed is shared between terminal pairs in order to meet the requirements of our strictness results. Finally, in Section 4, we give examples that demonstrate that our results are nearly tight for the framework proposed by [10].

2. STRICTNESS OF COST SHARING ALGORITHMS FOR STEINER FORESTS

Suppose we are given an α-approximation algorithm ALG that computes a Steiner forest F for the set of terminal groups R. In this section, we define two different cost sharing schemes to distribute a fraction of $\frac{1}{\alpha}$ of the cost of F among the terminals. These schemes crucially rely on a notion of *witnesses* that are associated with each edge $e \in F$. We show that if ALG and the witness definition satisfy certain properties, these cost sharing schemes yield 2α-group-strict and $\frac{3}{2}\alpha$-strict cost shares.

We assume without loss of generality that the terminal sets of two different groups in R are disjoint. If s appears in two groups, g_1 and g_2, we can create two new nodes s_1 and s_2, add edges (s_1, s) and (s_2, s) each of zero cost, and replace s with s_1 in g_1 and with s_2 in g_2.

Given that F is produced by an α-approximation algorithm ALG, we define the cost share $\xi(e)$ of an edge $e \in F$ as

$$\xi(e) = \frac{1}{\alpha} c(e). \tag{1}$$

For each edge $e \in F$, we assign two terminals $\mathscr{W}_e = \{u, v\}$ to be the *witnesses* of e and split $\xi(e)$ between the terminals in $\mathscr{W}_e$. In this section, we give two different ways of splitting this cost share, yielding two different strictness results.

Let $\xi_u(e)$ be the share of $\xi(e)$ of terminal $u \in \mathscr{W}_e$ according to the split. The total cost share assigned to terminal u is

$$\xi_u = \sum_{e \in F \,:\, u \in \mathscr{W}_e} \xi_u(e).$$

The cost share of a group of terminals $g \in R$ is $\xi_g = \sum_{u \in g} \xi_u$.

We prove that if cost shares are distributed as described above, the total cost share of all groups of terminals does not exceed the optimum cost. This validates condition (1) of Definition 2.

LEMMA 1. *Let F be a Steiner forest computed by an α-approximate algorithm* ALG *and let $\{\mathscr{W}_e\}_{e \in F}$ be the associated witness set. If the cost shares ξ are computed as described above then*

$$\sum_{g \in R} \xi_g \le \mathsf{opt}_R.$$

PROOF. By our cost sharing rule (1), we have

$$c(F) = \sum_{e \in F} c(e) = \sum_{e \in F} \sum_{u \in \mathscr{W}_e} \alpha \xi_u(e) = \alpha \sum_{g \in R} \xi_g$$

and this implies the lemma as $c(F) \le \alpha \cdot \mathsf{opt}_R$. $\square$

2.1 Symmetric Cost Share Assignment

Crucial to proving the strictness of our cost sharing scheme is to define the witness set $\{\mathscr{W}_e\}_{e \in F}$ to satisfy the following property. For a group of vertices $g \in R$, let T_g denote the unique tree connecting g in F if such a tree exists; otherwise $T_g = \emptyset$. In the following we abuse notation by letting a path P or tree T also stand for the set of edges in it.

PROPERTY 1. *Consider an arbitrary group of terminals $g \in R$ and let e be an edge in tree T_g. If $\mathscr{W}_e \cap g = \emptyset$ then e is part of the forest F_{-g}.*

Remove terminal group g from R and run ALG on the set of terminal groups $R_{-g} = R \setminus \{g\}$. Property 1 implies that if an edge $e \in T_g$ is not part of the forest F_{-g} then e is witnessed by some terminal in g, i.e., $\mathscr{W}_e \cap g \neq \emptyset$.

A natural idea is to split the cost share $\xi(e)$ of e evenly among the two witnesses. This is what we call the *symmetric cost share assignment*:

Symmetric cost share assignment: The cost share that each witness $u \in \mathscr{W}_e$ receives for edge e is

$$\xi_u(e) = \frac{1}{2} \xi(e). \tag{2}$$

It is easy to see that Property 1 together with the symmetric cost share assignment yields cost shares that are 2α-strict.

LEMMA 2. *Let F be a Steiner forest computed by an α-approximate algorithm* ALG *and let $\{\mathscr{W}_e\}_{e \in F}$ be the witness set. If algorithm* ALG *and $\{\mathscr{W}_e\}_{e \in F}$ satisfy Property 1, then the symmetric cost shares ξ are $\beta = 2\alpha$-strict, i.e., for all $g \in R$*

$$c_{G|F_{-g}}(g) \le 2\alpha \cdot \xi_g.$$

PROOF. Property 1 ensures that all edges $e \in T_g$ that are not part of F_{-g} must be witnessed by some terminal in g. The claim of the lemma follows since for each edge e of T_g with $\mathscr{W}_e \cap g \neq \emptyset$, there is some terminal u in g with $\xi_u(e) = \frac{1}{2}\xi(e) = \frac{1}{2\alpha} c(e)$. $\square$

2.2 Asymmetric Cost Share Assignment

We next turn to a refined *asymmetric cost sharing scheme*, where we split the cost share $\xi(e)$ of an edge $e \in F$ unevenly among its witnesses in $\mathscr{W}_e$. We prove that this asymmetric cost sharing scheme yields $\frac{3}{2}\alpha$-strict cost shares, if only algorithm ALG and the witness set $\{\mathscr{W}_e\}_{e \in F}$ satisfy an additional property.

This property is motivated by the following intuition: If terminals in group g witness some edges that are not in T_g, it might be cheaper to connect the terminals of g in F_{-g} by using those edges instead of some edges in $T_g \setminus F_{-g}$. If these alternate edges do not provide a cheaper connection than the corresponding edges in $T_g \setminus F_{-g}$, then they contribute some significant cost share to g that g can then use to pay for edges in $T_g \setminus F_{-g}$.

PROPERTY 2. *There exists an order $\prec$ on the groups of terminals in R such that for any two terminal groups $g, h \in R$, $h \prec g$ implies that all edges e of $T_h \setminus T_g$ with $\mathscr{W}_e \cap g = \emptyset$ are part of the forest F_{-g}.*

While we will show in Section 3.3 that our witness definition satisfies Property 2 for groups of arbitrary size, the asymmetric cost sharing rule below is designed specifically for the case when all groups are pairs. Thus the remainder of this section is devoted to the case when all groups consist of just two terminals. In this case, we specialize our notation as follows: P_{st} is the unique path in F connecting s and t. For terminal u, let $\bar{u}$ be the terminal it is paired with, i.e., $(u, \bar{u}) \in R$. The cost share of a terminal pair $(u, \bar{u}) \in R$ is denoted $\xi_{u\bar{u}}$.

We define an asymmetric cost sharing rule as follows:

Asymmetric cost share assignment: Consider an edge $e \in F$ and let $\mathscr{W}_e = \{u, v\}$. Assume without loss of generality that $(u, \bar{u}) \prec (v, \bar{v})$. We share $\xi(e)$ among the two witnesses u and v as follows:

$$\xi_u(e) = \begin{cases} \frac{1}{3}\xi(e) & \text{if } e \notin P_{u\bar{u}} \\ \frac{2}{3}\xi(e) & \text{if } e \in P_{u\bar{u}} \end{cases} \quad \text{and}$$

$$\xi_v(e) = \begin{cases} \frac{2}{3}\xi(e) & \text{if } e \notin P_{u\bar{u}} \\ \frac{1}{3}\xi(e) & \text{if } e \in P_{u\bar{u}}. \end{cases} \tag{3}$$

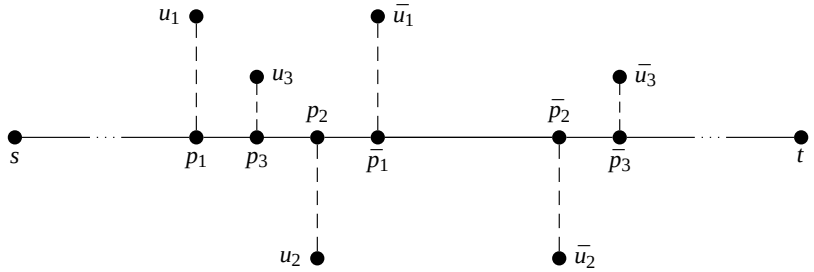

Figure 1: The figure shows the path P_{st} and three terminal pairs $(u_i,\bar{u}_i),(u_j,\bar{u}_j) \in I_{st}$ with $1 \le i < j \le q$.

Observe that with this cost sharing rule, the total cost share that the witnesses in $\mathscr{W}_e$ receive for $e \in F$ is at most $\xi(e)$.

LEMMA 3. *Let F be a Steiner forest computed by an α-approximate algorithm* ALG *and let $\{\mathscr{W}_e\}_{e\in F}$ be the witness set. If algorithm* ALG *and $\{\mathscr{W}_e\}_{e\in F}$ satisfy Properties 1 and 2, then the asymmetric cost shares ξ are $\beta = \frac{3}{2}\alpha$-strict, i.e., for all $(s,t) \in R$*

$$c_{G|F_{-st}}(s,t) \le \frac{3}{2}\alpha \cdot \xi_{st}.$$

PROOF. Consider path P_{st} in F. From Property 1 we know that each edge $e \in P_{st}$ that is missing in F_{-st} is witnessed by at least one of s and t. If the cost share of terminal pair (s,t) for such an edge e is at least $\frac{2}{3}\xi(e)$, then $\frac{3}{2}\alpha \cdot \xi_{st}(e)$ suffices to reconstruct e. We can therefore reconstruct all such edges using $\frac{3}{2}\alpha$ times their total cost share. By contracting all such edges in F_{-st}, we can assume without loss of generality that if s or t witness an edge $e \notin F_{-st}$, then $\xi_{st}(e) = \frac{1}{3}\xi(e)$. Let $\mathscr{M}_{st}$ be the set of all such edges.

$$\mathscr{M}_{st} = \{e \in G|F_{-st} : \xi_{st}(e) = \frac{1}{3}\xi(e)\}$$

We show that we can establish connection between s and t in the contracted graph at a cost at most $\frac{3}{2}\alpha\sum_{e\in\mathscr{M}_{st}}\xi_{st}(e)$. By then uncontracting the edges for which $\xi_{st}(e) \ge \frac{2}{3}\xi(e)$, we have established a connection between s and t in $G|F_{-st}$ at cost at most $\frac{3}{2}\alpha\xi_{st}$.

By Property 2 and the cost share assignment given in (3), if $e \in P_{st} \cap \mathscr{M}_{st}$ then the following must hold: (i) there is a terminal $u \notin \{s,t\}$ that together with s or t witnesses e, (ii) $(u,\bar{u}) \prec (s,t)$ and (iii) $e \in P_{u\bar{u}}$. For these edges we need to collect additional cost share from edges in $P_{u\bar{u}} \setminus P_{st}$ witnessed by (s,t) and possibly exploit the connectivity provided by $P_{u\bar{u}}$ in F_{-st}.

Let I_{st} be the set of terminal pairs that witness edges on P_{st} together with one of s and t:

$$I_{st} = \{(u,\bar{u}) \in R : \exists e \in P_{st} \text{ such that } \{s,t\} \cap \mathscr{W}_e \neq \emptyset \text{ and } \{u,\bar{u}\} \cap \mathscr{W}_e \neq \emptyset\}.$$

Assume that $I_{st} = \{(u_1,\bar{u}_1),\ldots,(u_q,\bar{u}_q)\}$ for some $q \ge 0$. We use P_i, $1 \le i \le q$, to refer to the $u_i,\bar{u}_i$-path in F. For $1 \le i \le q$, let p_i and $\bar{p}_i$ be the first and last vertices on P_i that are on P_{st}. We call p_i and $\bar{p}_i$ the *projections* of u_i and $\bar{u}_i$, respectively, on path P_{st}. We choose the label p_i so that it is closer to s than $\bar{p}_i$ for all $1 \le i \le q$. We order the indices of pairs in I_{st} so that $\bar{p}_i$ is closer to s than $\bar{p}_j$ for all $1 \le i < j \le q$. Refer to Figure 1 for an example.

In the following, we let P^i_{st} be the $s,\bar{p}_i$-segment of P_{st} for all $0 \le i \le q$ (where we define $\bar{p}_0 = s$). We then define

$$\mathscr{P}^i = P^i_{st} \cup P_1 \cup \ldots \cup P_i$$

as the union of path P^i_{st} and the paths of the first i terminal pairs in I_{st}. Finally, let $\mathscr{M}^i_{st} = \mathscr{M}_{st} \cap \mathscr{P}^i$ be the set of edges in $\mathscr{P}^i$ that are witnessed by s or t.

For an edge set $S \subseteq \mathscr{M}_{st}$, let $\xi_{st}(S)$ be a short-hand for $\sum_{e\in S}\xi_{st}(e)$. Using induction over $0 \le i \le q$, we will now prove that we can reconstruct an s,z-path in F_{-st} for all vertices $z \in \mathscr{P}^i$ at a cost at most $\frac{3}{2}\alpha \cdot \xi_{st}(\mathscr{M}^i_{st})$.

This is clearly true for $i = 0$. Now assume that the claim holds for some $0 \le i < q$. Consider first the case where $\bar{p}_i$ precedes p_{i+1} on P_{st} (see also Figure 2.(i)). Let z be a node on the $\bar{p}_i, p_{i+1}$-segment P of P_{st}. Observe that none of the edges $e \in P$ can be contained on P_j for any $1 \le j \le q$. Hence, if s or t witnessed this edge, they would receive at least a cost share of $\frac{2}{3}\xi(e)$. We excluded those edges initially and hence, using the inductive hypothesis, we can reconstruct an s,z-path in F_{-st} at a cost at most $\frac{3}{2}\alpha \cdot \xi_{st}(\mathscr{M}^i_{st})$.

For the above argument, we will now assume that p_{i+1} precedes $\bar{p}_i$ on P_{st}. Let f and l be the first and last vertices on P_{i+1}, respectively, that are incident to edges of $\mathscr{P}^i$ (refer to Figure 2.(ii)). Clearly, the unique f,l-path P_{fl} in F must be contained in $\mathscr{P}^i$. Let z_1 be a node on P_{fl}. We can then use the induction hypothesis in order to reconstruct an s,z_1-path in F_{-st} at a cost at most $\frac{3}{2}\alpha \cdot \xi_{st}(\mathscr{M}^i_{st})$.

Now consider a node $z_2 \in P_{i+1} \setminus P_{fl}$. Assume that z_2 is contained on the $l,\bar{u}_{i+1}$-segment of P_{i+1} (the case where z_2 is contained on the u_{i+1},f-segment of P_{i+1} works analogously). We consider two ways to reconstruct an s,z_2-path in F_{-st}:

1. Use P_f for the union of the u_{i+1},f- and $z_2,\bar{u}_{i+1}$-segments of P_{i+1}, and let $\mathscr{M}_f$ be the set of edges on P_f that are witnessed by s or t. We can then inductively reconstruct an s,f-path in F_{-st} at a cost at most $\frac{3}{2}\alpha \cdot \xi(\mathscr{M}^i_{st})$. Using the fact that u_{i+1} and $\bar{u}_{i+1}$ are connected in F_{-st}, we obtain an s,z_2-path by reconstructing P_f. This costs at most $3\alpha \cdot \xi_{st}(\mathscr{M}_f)$.

2. Use P_l for the l,z_2-segment of P_{i+1} and let $\mathscr{M}_l$ be a short-hand for $\mathscr{M}_{st} \cap P_l$. We can then inductively reconstruct an s,l-path in F_{-st} at a cost at most $\frac{3}{2}\alpha \cdot \xi(\mathscr{M}^i_{st})$. We obtain an s,z_2-path by reconstructing P_l at a cost at most $3\alpha \cdot \xi_{st}(\mathscr{M}_l)$.

In summary, reconstructing an s,z_2-path in F_{-st} costs at most

$$\frac{3}{2}\alpha \cdot \xi_{st}(\mathscr{M}^i_{st}) + 3\alpha \cdot \min\{\xi_{st}(\mathscr{M}_f), \xi_{st}(\mathscr{M}_l)\}$$
$$\le \frac{3}{2}\alpha \cdot \left(\xi_{st}(\mathscr{M}^i_{st}) + \xi_{st}(\mathscr{M}_f) + \xi_{st}(\mathscr{M}_l)\right).$$

Observing that $\mathscr{M}_f \cup \mathscr{M}_l \cup \mathscr{M}^i_{st}$ is a partition of $\mathscr{M}^{i+1}_{st}$ finishes the proof of the lemma. □

3. A PRIMAL-DUAL BASED STRICT ALGORITHM FOR STEINER FORESTS

In this section we review a $(2-1/k)$-approximate primal-dual algorithm for Steiner forests. The algorithms for Steiner forest presented in [3] and [15] differ only slightly. In this paper, we focus on

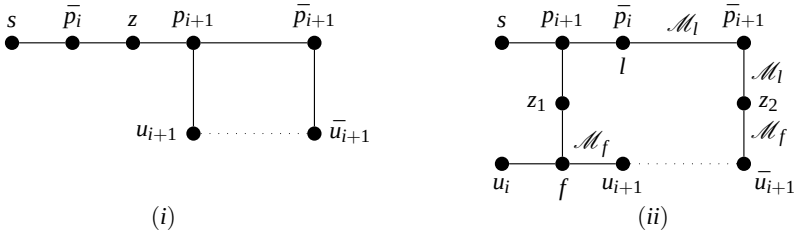

Figure 2: The figure illustrates the cases used in the proof of Lemma 3. Solid edges represent segments of P_{st}, dotted edges represent connectivity in F_{-st}.

the viewpoint taken in [3]. We use AKR to refer to this algorithm. We then show that AKR together with an appropriate witness definition satisfies Properties 1 and 2.

While the Steiner forest problem is traditionally defined on pairs of nodes, it is easy to extend the definition to groups of nodes. However, in the case of the Steiner forest problem, the group problem can be modeled and solved as the problem defined on pairs, by creating a pair for each pair of nodes in a group.

3.1 Primal-Dual Algorithms for Steiner Forests

The primal-dual algorithm AKR constructs both a feasible primal and a feasible dual solution for a linear programming formulation of the Steiner forest problem and its dual, respectively. A standard integer programming formulation for the Steiner forest problem has a binary variable x_e for all edges $e \in E$. Variable x_e has value 1 if edge e is part of the resulting forest. We let $\mathscr{U}$ contain exactly those subsets U of V that *separate* at least one terminal pair in R. In other words, $U \in \mathscr{U}$ iff there is $(s,t) \in R$ with $|\{s,t\} \cap U| = 1$.

For a subset U of the nodes we also let $\delta(U)$ denote the set of those edges that have exactly one endpoint in U. We then obtain the following integer linear programming formulation for the Steiner forest problem:

$$\begin{aligned} \min \quad & \sum_{e \in E} c_e \cdot x_e & \text{(IP)} \\ \text{s.t.} \quad & \sum_{e \in \delta(U)} x_e \geq 1 \quad \forall U \in \mathscr{U} \\ & x \text{ integer} \end{aligned}$$

The linear programming dual of the LP-relaxation (LP) of (IP) has a variable y_U for all node sets $U \in \mathscr{U}$. There is a constraint for each edge $e \in E$ that limits the total dual assigned to sets $U \in \mathscr{U}$ that contain exactly one endpoint of e to be at most c_e.

$$\begin{aligned} \max \quad & \sum_{U \in \mathscr{U}} y_U & \text{(D)} \\ \text{s.t.} \quad & \sum_{U \in \mathscr{U} : e \in \delta(U)} y_U \leq c_e \quad \forall e \in E & (4) \\ & y \geq 0 \end{aligned}$$

Algorithm AKR constructs a primal solution for (LP) and a dual solution for (D). The algorithm has two goals:

1. Compute a feasible solution for the given Steiner forest instance. The algorithm reduces the degree of infeasibility as it progresses.

2. Create a dual feasible packing of sets of largest possible total value. The algorithm raises dual variables of certain subsets of nodes at all times. The final dual solution is maximal in the sense that no single set can be raised without violating a constraint of type (4).

Consider the execution of algorithm AKR as a process over time and let x^τ and y^τ be the primal incidence vector and feasible dual solution at time τ. Note that in any optimal solution to (IP), $x_e \in \{0,1\}$. Let F^τ denote the forest corresponding to the set of edges with $x_e^\tau = 1$. Initially, let $x_e^0 = 0$ for all $e \in E$ and $y_U^0 = 0$ for all $U \in \mathscr{U}$. The algorithm maintains the invariant $x_e^\tau \leq x_e^{\tau'}$ and $y_U^\tau \leq y_U^{\tau'}$ for all $\tau < \tau'$.

An edge $e \in E$ is *tight* if the corresponding constraint (4) holds with equality; and a path is *tight* if every edge in the path is tight. Assume that the forest F^τ at time τ is infeasible. A terminal node $v \in R$ is *active* at time τ if v and its *mate* $\bar{v}$, i.e., $(v,\bar{v}) \in R$, are in different trees in the forest F^τ; v is *inactive* otherwise.[1] Let $\bar{F}^\tau$ denote the subgraph of G that is induced by the tight edges for dual y^τ. To avoid confusion between connected components in F^τ and those in $\bar{F}^\tau$, the term *moat* refers to a connected component in $\bar{F}^\tau$. The algorithm maintains that if $C \in F^\tau$ then $C \subseteq U \in \bar{F}^\tau$. A moat U of $\bar{F}^\tau$ is active at time τ if U contains an active terminal; U is inactive otherwise. Let A^τ be the set of all active moats in $\bar{F}^\tau$ at time τ. AKR raises the dual variables for all sets in A^τ uniformly at all times $\tau \geq 0$, so that if U is active from time τ' until time τ'', then $y_U = \tau'' - \tau'$.

Two disjoint moats *collide* at time τ in the execution of AKR if there is a path in G from one moat to the other that becomes tight at time τ. In order for this to happen, at least one of the two moats must be active. Suppose that a path P connecting two *active* terminals u and u' becomes tight at time τ in the execution of AKR. Then u is contained in some active moat U and u' is in a disjoint active moat U'. When this happens, AKR adds the edges of P not already in F^τ to F^τ: that is, for all $e \in P$, the algorithm sets $x_e^\tau = 1$. For $\tau' > \tau$, sets U and U' are part of the same moat of $\bar{F}^{\tau'}$.

Subsequently, we use $U^\tau(v)$ to refer to the moat in $\bar{F}^\tau$ that contains node $v \in V$ at time τ. Similarly, we let $U^\tau(C)$ denote the moat in $\bar{F}^\tau$ that contains the connected component $C \in F^\tau$ at time τ. Let F be the final forest.

The following is the main theorem of [3]:

THEOREM 2. *Let F be the forest computed by* AKR *on terminal set R. We then have $c(F) \leq \left(2 - \frac{1}{k}\right) \cdot \mathsf{opt}_R$, where opt_R is the minimum cost of a Steiner forest for the given input instance with terminal set R.*

3.2 Witness Definition

We define a set $\{\mathscr{W}_e\}_{e \in F}$ of witnesses that are used to distribute the cost shares as described in Section 2. In Section 3.3 we show

[1] Note that for the problem defined on groups, each terminal in the group will become inactive at exactly the same time, since if the group is not connected, then each terminal is not connected to some other terminal in the group.

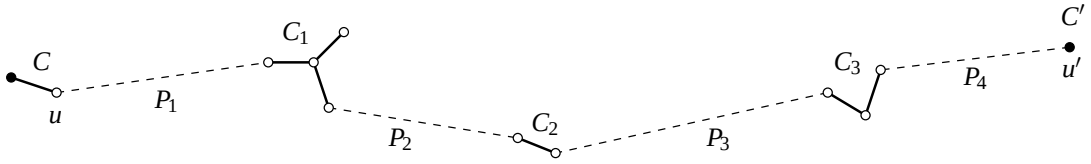

Figure 3: A path P that becomes tight at time τ_P in $\mathsf{AKR}(R)$.

that AKR together with this witness definition satisfies Properties 1 and 2.

Let $\mathsf{AKR}(S)$ refer to the execution of AKR on terminal set $S \subseteq R$. Let F be the forest computed by $\mathsf{AKR}(R)$ for terminal set R. The witnesses $\mathscr{W}_e$ for each edge $e \in F$ are defined by the execution of $\mathsf{AKR}(R)$.

Consider a path P that becomes tight at time τ_P in $\mathsf{AKR}(R)$, as depicted in Figure 3. Path P starts from a node u in a connected component C of F^{τ_P}, passes through a (possibly empty) sequence $C_1, \ldots, C_l$ of connected components of F^{τ_P}, and ends in a node u' of a connected component C' of F^{τ_P}. Let $P_1, \ldots, P_{l+1}$ be the sequence of paths of $P \setminus F^{\tau_P}$ and let $\mathscr{P}_P$ be the set of edges in $P \setminus F^{\tau_P}$. When P becomes tight, the set $\mathscr{P}_P$ is added to F and we determine for each edge $e \in \mathscr{P}_P$ the corresponding witnesses $\mathscr{W}_e$ as follows.

Each edge $e \in \mathscr{P}_P$ will have the same witness set $\mathscr{W}_e = \{w, w'\}$. We will also say that P is witnessed by w and w'. Since the moats $U^{\tau_P}(C)$ and $U^{\tau_P}(C')$ are active at time τ_P, both C and C' must contain at least one active terminal. We will choose one witness among the active terminals in each of C and C'. Intuitively, the witness chosen in C is the active terminal whose moat intersects P_1 earliest among all active terminals in C. Similarly, the moat of the witness chosen in C' is the first to intersect P_{l+1} among all active terminals in C'. To make this precise, let A_C be the set of terminals in C that are active at time τ_P. By definition of C, all terminals in A_C are connected to u in F^{τ_P}. If $u \in A_C$, then $w = u$. Otherwise, let τ_u be the first time that moat $U^{\tau}(u)$ collides with a moat U_u containing a terminal in A_C. (Let $\tau_u = 0$ if $u \in A_C$.)

LEMMA 4. *There is a terminal w in $U_u \cap A_C$ whose moat collides with u's moat at time τ_u even if all terminals in $A_C \setminus \{w\}$ are not part of the terminal set R.*

PROOF. We will prove the lemma by showing a stronger claim: For all terminals v in C that become inactive before time τ_P, let τ_v be the first time that moat $U^{\tau}(v)$ collides with a moat U_v containing at least one terminal from A_C. Then there is a terminal $w \in U_v \cap A_C$ whose moat collides with v's moat at time τ_v even if all terminals in $A_C \setminus \{w\}$ are not part of the terminal set R. This clearly implies the lemma.

Fix a terminal v in C that becomes inactive at some time before τ_P. Observe that by the definition of τ_v, $U^{\tau}(v)$ does not intersect A_C before time τ_v, and therefore the growth of v's moat until time τ_v is not affected by the removal of A_C.

The proof is by induction on $|U_v|$. If $|U_v| = 1$ then the set consists of only a terminal $w \in A_C$, and the growth of w's moat is not affected by the removal of other terminals in A_C.

Now assume that $|U_v| > 1$. Let $z \in U_v$ be the endpoint of the path P_v that becomes tight when U_v collides with $U^{\tau}(v)$. If z is in A_C we are done: We define $w = z$ and observe that w's moat intersects P_v at all times $0 \le \tau \le \tau_v$ even if the terminals in $A_C \setminus \{w\}$ are not part of the terminal set.

Assume that z is not in A_C. In this case, let τ_z be the first time that $U^{\tau}(z)$ collides with a moat U_z that contains a terminal from A_C. We have $|U_z| < |U_v|$ and can therefore apply the induction hypothesis to z and U_z. That is, there is a terminal $w \in U_z$ whose moat collides with z's moat at time τ_z even if all terminals in $A_C \setminus \{w\}$ are not part of the terminal set R. Since w is in A_C, it causes the moat containing z to grow after time τ_z regardless of other terminals in A_C. Thus, w's moat collides with that of v at time τ_v and this finishes the lemma. □

The witness w is a terminal described by Lemma 4. The witness w' with respect to C' is defined analogously.

3.3 Properties of AKR

We show that Properties 1 and 2 hold for AKR and the witness definition given above. Let $\{\mathscr{W}_e\}_{e \in F}$ be the set of witnesses assigned by AKR. Let $\mathscr{G}_{-g}$ (where $\mathscr{G} = U, F$ or $\bar{F}$) refer to set $\mathscr{G}$ in run $\mathsf{AKR}(R_{-g})$. For example, $U^{\tau}_{-g}(u)$ refers to the moat of u at time τ in $\mathsf{AKR}(R_{-g})$. Let τ_g denote the time at which all terminals in group g become inactive in $\mathsf{AKR}(R)$. Subsequently, we abuse notation by letting R also refer to the set of all terminals that are contained in the groups of R.

LEMMA 5. *For all $\tau \le \tau_g$ and for all terminals $v \in R_{-g}$: $U^{\tau}_{-g}(v) \subseteq U^{\tau}(v)$. Moreover, if $U^{\tau}(v) \cap g = \emptyset$, then $U^{\tau}_{-g}(v) = U^{\tau}(v)$.*

PROOF. We prove the lemma by induction over time τ. At time $\tau = 0$ we have $U^{\tau}_{-g}(v) = U^{\tau}(v)$ for all $v \in R_{-g}$ and thus the induction hypothesis clearly holds. Assume the induction hypothesis holds at time τ. We will show that it remains true at time $\tau + \varepsilon$ for any small $\varepsilon > 0$.

Consider the case $U^{\tau}(v) \cap g = \emptyset$ and thus $U^{\tau}_{-g}(v) = U^{\tau}(v)$. That is, $U^{\tau}_{-g}(v)$ is active at time τ iff $U^{\tau}(v)$ is active at that time. Then $U^{\tau+\varepsilon}_{-g}(v) = U^{\tau+\varepsilon}(v)$ if $U^{\tau+\varepsilon}(v) \cap g = \emptyset$; and $U^{\tau+\varepsilon}_{-g}(v) \subseteq U^{\tau+\varepsilon}(v)$ otherwise. Now assume $U^{\tau}(v) \cap g \neq \emptyset$ and thus $U^{\tau}_{-g}(v) \subseteq U^{\tau}(v)$. Clearly, $U^{\tau+\varepsilon}(v) \cap g \neq \emptyset$. Since $\tau \le \tau_g$, all terminals in g are active at time τ and thus $U^{\tau}(v)$ is active at time τ. It follows $U^{\tau+\varepsilon}_{-g}(v) \subseteq U^{\tau+\varepsilon}(v)$. □

COROLLARY 1. *Consider a terminal $v \in R_{-g}$. If v is active at time $\tau \le \tau_g$ in $\mathsf{AKR}(R)$ then v must be active until time at least τ in $\mathsf{AKR}(R_{-g})$.*

As in Figure 3, let P be a path connecting two active components C and C' that becomes tight at time $\tau_P \le \tau_g$ in the execution of $\mathsf{AKR}(R)$. As before, let u and u' be the two endpoints in C and C', respectively, and let $C_1, \ldots, C_l$ be the connected components of F^{τ_P} that lie on P. Moreover, assume P is witnessed by w and w'. The following lemma implies Property 1 for our definition of witnesses.

LEMMA 6. *Assume that none of the two witnesses w, w' of P is in g, i.e., $\mathscr{W}_e \cap g = \emptyset$ for all edges $e \in \mathscr{P}_P$. Then for each edge $e \in P$, the contribution to (4) before time τ_P is the same in $\mathsf{AKR}(R)$ as it is in $\mathsf{AKR}(R_{-g})$. In particular, P is added at time τ_P in both runs.*

PROOF. First we show that the contribution to (4) from variables corresponding to moats not containing u and u' is the same in both runs. For all $1 \le i \le l$, C_i is inactive at time $\tau_P < \tau_g$, so that $C_i \cap g =$

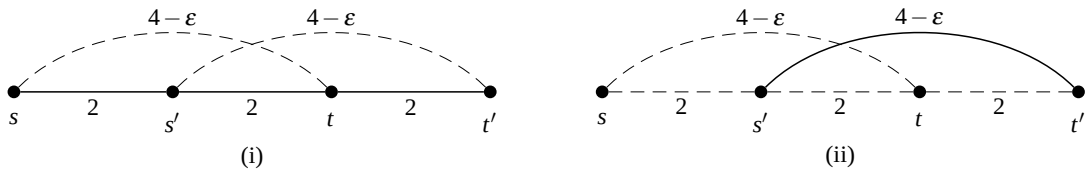

Figure 4: Instance used in the lower bound argument.

$\emptyset$. Let $\tau_i < \tau_P$ be the time at which component C_i becomes inactive. Then, by Lemma 5, $U_g^\tau(v) = U^\tau(v)$ for all $\tau \le \tau_i$, for all $v \in C_i$, for all $1 \le i \le l$. Thus, the dual variable values for all sets restricted to subsets of C_i are the same in both $\mathsf{AKR}(R)$ and $\mathsf{AKR}(R_{-g})$.

Now consider the contribution to (4) from variables corresponding to moats containing u. Let $\hat{\tau}$ be the first time at which moat $U^{\hat{\tau}}(u)$ collides with a moat $\hat{U}$ containing a terminal in A_C. By the definition of $\hat{\tau}$, $U^\tau(u) \cap g = \emptyset$ for all $\tau \in [0, \hat{\tau})$. Thus by Lemma 5, $U_{-g}^\tau(u) = U^\tau(u)$ for all $\tau \in [0, \hat{\tau})$; and the contribution to (4) from variables corresponding to moats containing u before time $\hat{\tau}$ is the same in $\mathsf{AKR}(R)$ and $\mathsf{AKR}(R_{-g})$. From time $\hat{\tau}$, w and u are in the same moat in $\mathsf{AKR}(R)$, and by Lemma 4, they are also in the same moat at this time in $\mathsf{AKR}(R_{-g})$. By Lemma 5, w is still active at time τ_P in both $\mathsf{AKR}(R)$ and $\mathsf{AKR}(R_{-g})$. Thus the contribution to (4) of variables corresponding to moats containing u from time $\hat{\tau}$ until time τ_P is also the same in both runs. A symmetric argument for variables corresponding to moats containing u' shows that path P is tight at time τ_P in $\mathsf{AKR}(R_{-st})$.

Finally, note that Lemma 5 also implies that w and w' are contained in disjoint moats in $\mathsf{AKR}(R_{-st})$ before time τ_P. Hence path P is added at time τ_P in $\mathsf{AKR}(R_{-st})$ and the lemma follows. $\square$

We show that the following precedence order $\prec$ together with the witness definition described above imply Property 2 for AKR.

Consider a run $\mathsf{AKR}(R)$. For each group of terminals $g \in R$, let τ_g be the time at which the terminals in g become inactive. Fix an order on the terminal groups in $R = \{g_i\}_{1 \le i \le k}$ such that

$$\tau_{g_1} \le \tau_{g_2} \le \cdots \le \tau_{g_k}.$$

We define $g_i \prec g_j$ if $i \le j$ in this order.

The following lemma implies Property 2.

LEMMA 7. *Let g and h be two groups of terminals in R such that $h \prec g$ and let e be an edge on tree T_h in F. If $\mathscr{W}_e \cap g = \emptyset$ then $e \in F_{-g}$.*

PROOF. The proof is by contradiction. Assume that edge e is not part of F_{-g}. Edge $e \in T_h$ is added to F at time $\tau \le \tau_h \le \tau_g$. By Lemma 6 and since $\mathscr{W}_e \cap g = \emptyset$, e is picked at time τ in $\mathsf{AKR}(R_{-g})$. This is a contradiction. $\square$

4. A LOWER BOUND ON THE STRICTNESS FACTOR

Figure 4 shows a simple Steiner forest instance with two terminal pairs $R = \{(s,t),(s',t')\}$. The solid lines in Figure 4.(i) correspond to edges of forest F returned by algorithm AKR when run on this instance. The total cost share of all edges in F is 3 and therefore, there must be a terminal pair in R whose cost share is at most $\frac{3}{2}$. Without loss of generality, assume that $\xi_{st} \le \frac{3}{2}$. Running AKR on terminal set $R_{-st} = \{(s',t')\}$ yields the forest in Figure 4.(ii). As $c_{G|F_{-st}}(s,t) = 4-\varepsilon$, this example shows that the strictness of AKR is at least $(4-\varepsilon)/\frac{3}{2} \approx \frac{8}{3}$ whenever the sum of the cost shares of all terminal pairs is at most half of the cost of the computed forest.

We remark that the previously known algorithms for the MRoB problem in [10] and [6] essentially distribute half of the cost of a forest computed by AKR as cost shares among the terminal pairs. Given a terminal pair $(s,t) \in R$, both of these algorithms use an adaptation of the standard primal-dual Steiner forest algorithm (so called *timed* or *boosted* primal-dual algorithms) to compute a forest F_{-st}. In a nutshell, the idea behind these adaptations is to produce a forest whose connectivity is higher than that of a forest produced by standard primal-dual algorithms. For the example above, however, both algorithms in [10] and [6] return the forest in Figure 4.(ii). Thus, the above example provides an $\frac{8}{3}$ lower bound for the strictness of these algorithms as well.

5. REFERENCES

[1] M. Andrews. Hardness of buy-at-bulk network design. In *Proceedings, IEEE Symposium on Foundations of Computer Science*, pages 115–124, 2004.

[2] B. Awerbuch and Y. Azar. Buy-at-bulk network design. In *Proceedings, IEEE Symposium on Foundations of Computer Science*, pages 542–547, 1997.

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

[4] Y. Bartal. Probabilistic Approximation of Metric Spaces and its Algorithmic Applications. In *Proceedings, IEEE Symposium on Foundations of Computer Science*, pages 184–193, 1996.

[5] Y. Bartal. On approximating arbitrary metrics by tree metrics. In *Proceedings, ACM Symposium on Theory of Computing*, pages 161–168, 1998.

[6] L. Becchetti, J. Könemann, S. Leonardi, and M. Pál. Sharing the cost more efficiently: Improved approximation for multicommodity rent-or-buy. In *Proceedings, ACM-SIAM Symposium on Discrete Algorithms*, pages 375–384, 2005.

[7] M. Bern and P. Plassmann. The Steiner problem with edge lengths 1 and 2. *Inform. Process. Lett.*, 32(4):171–176, 1989.

[8] J. Fakcharoenphol, S. Rao and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *Proceedings, ACM Symposium on Theory of Computing*, pages 448–455, 2003.

[9] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. Freeman, San Francisco, 1979.

[10] A. Gupta, A. Kumar, M. Pál, and T. Roughgarden. Approximation via cost-sharing: A simple approximation algorithm for the multicommodity rent-or-buy problem. In *Proceedings, IEEE Symposium on Foundations of Computer Science*, pages 606–617, 2003.

[11] A. Gupta, A. Kumar, and T. Roughgarden. Simpler and better approximation algorithms for network design. In *Proceedings, ACM Symposium on Theory of Computing*, pages 365–372, 2003.

[12] A. Gupta and M. Pál. Stochastic Steiner trees without a root. In *Proceedings, International Colloquium on Automata, Languages and Programming*, pages 1051–1063, 2005.

[13] A. Gupta, M. Pál, R. Ravi, and A. Sinha. Boosted sampling: Approximation algorithms for stochastic optimization. In *Proceedings, ACM Symposium on Theory of Computing*, pages 417–426, 2004.

[14] A. Gupta, R. Ravi, and A. Sinha. An edge in time saves nine: LP rounding approximation algorithms. In *Proceedings, IEEE Symposium on Foundations of Computer Science*, pages 218–227, 2004.

[15] M. X. Goemans and D. P. Williamson. A general approximation technique for constrained forest problems. *SIAM J. Comput.*, 24:296–317, 1995.

[16] A. Kumar, A. Gupta, and T. Roughgarden. A constant factor approximation algorithm for the multicommodity rent-or-buy problem. In *Proceedings, IEEE Symposium on Foundations of Computer Science*, pages 333–344, 2002.