SIAM J. COMPUT.
Vol. 39, No. 8, pp. 3616–3632

STRICT COST SHARING SCHEMES FOR STEINER FOREST*

LISA FLEISCHER†, JOCHEN KÖNEMANN‡, STEFANO LEONARDI§, AND GUIDO SCHÄFER¶

Abstract. Gupta et al. [*J. ACM*, 54 (2007), article 11] and Gupta, Kumar, and Roughgarden [in *Proceedings of the ACM Symposium on Theory of Computing*, ACM, New York, 2003, pp. 365–372] recently developed an elegant framework for the development of randomized approximation algorithms for rent-or-buy network design problems. The essential building block of this framework is an approximation algorithm for the underlying network design problem that admits a *strict cost sharing scheme*. Such cost sharing schemes have also proven to be useful in the development of approximation algorithms in the context of two-stage stochastic optimization with recourse. The main contribution of this paper is to show that the Steiner forest problem admits cost shares that are 3-strict and 4-group-strict. As a consequence, we derive surprisingly simple approximation algorithms for the multicommodity rent-or-buy and the multicast rent-or-buy problems with approximation ratios 5 and 6, improving over the previous best approximation ratios of 6.828 and 12.8, respectively. We also show that no approximation ratio better than 4.67 can be achieved using the sample-and-augment framework in combination with the currently best known Steiner forest approximation algorithms. In the context of two-stage stochastic optimization, our result leads to a 6-approximation algorithm for the stochastic Steiner tree problem in the black-box model and a 5-approximation algorithm for the stochastic Steiner forest problem in the independent decision model.

Key words. approximation algorithms, rent-or-buy problems, stochastic optimization, Steiner forests, strict cost shares

AMS subject classifications. 68Q25, 68W25, 68W40

DOI. 10.1137/090767108

1. Introduction.

In the *multicommodity rent-or-buy* (MRoB) *problem* we are given an undirected graph $G = (V, E)$ with nonnegative costs c_e for all edges $e \in E$, a set of k terminal pairs $R = \{(s_1, t_1), \ldots, (s_k, t_k)\} \subseteq V \times V$, a positive demand d_i for every terminal pair $(s_i, t_i) \in R$, and a parameter $M \geq 1$. 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 d_i units of flow 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 that are NP-hard. For $M = 1$ and unit demands, the MRoB problem reduces to the

*Received by the editors August 4, 2009; accepted for publication (in revised form) August 24, 2010; published electronically October 26, 2010. A preliminary version of the results presented in this paper was published in *Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing*, ACM, New York, 2006, pp. 663–670.

http://www.siam.org/journals/sicomp/39-8/76710.html

†T. J. Watson Research Ctr., IBM, Yorktown Heights, NY 10598 (lkf@cs.dartmouth.edu). This author's work was supported in part by NSF grant CCF-0515127.

‡Department of Combinatorics and Optimization, University of Waterloo, 200 University Avenue West, Waterloo, ON N2L 3G1, Canada (jochen@uwaterloo.ca). This author's work was supported in part by NSERC grant 288340.

§Dipartimento di Informatica e Sistemistica, Sapienza University of Rome, Via Ariosto 25, 00185 Rome, Italy (leon@dis.uniroma1.it). Part of this work was done while the author was visiting the School of Computer Science at Carnegie Mellon University.

¶Algorithms, Combinatorics and Optimization Group, CWI Amsterdam, Science Park 123, 1098 XG Amsterdam, The Netherlands (g.schaefer@cwi.nl).

Steiner forest problem which is to compute a minimum-cost forest that contains an s_i, t_i-path for all $1 \le i \le k$. It is well known that this problem is NP-hard [12] 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 [1]; Goemans and Williamson [13] generalize these results to a larger class of network design problems. The *single-sink rent-or-buy* (SSRoB) *problem* refers to the special case of the MRoB problem where all terminal pairs share a common root vertex $r \in V$, i.e., $r \in \{s_i, t_i\}$ for all $1 \le i \le k$.

The *multicast rent-or-buy* (MuRoB) *problem* is a generalization of the MRoB problem. Here one needs to connect terminal subsets of arbitrary size, called *groups*. More precisely, we are given a set of terminal groups $R = \{g_1, \dots, g_k\}$ with $g_i \subseteq V$ for every $1 \le i \le k$, and the goal is to install sufficient capacity on the edges of G such that d_i units of flow can be routed simultaneously between the terminals of every group g_i. The MuRoB problem is equivalent to the MRoB problem if all groups have size two.

Kumar, Gupta, and Roughgarden [23] give the first constant-factor approximation algorithm for the MRoB problem. Gupta, Kumar, and Roughgarden [18] present a randomized framework, called *sample-and-augment*, to derive approximation algorithms for the SSRoB, virtual private network design, and single-sink buy-at-bulk problems. The authors obtain a $(2 + \rho_{ST}) \approx 3.39$-approximation algorithm for the SSRoB problem, where $\rho_{ST} \approx \ln 4$ is the best known performance guarantee for the Steiner tree problem [8]. Based on the ideas in [18], Gupta et al. [16] extend this framework to incorporate the MRoB problem as well. A uniform presentation of the framework and its applications to some other network design problems, including the MuRoB problem, is given in [17]. We briefly review the sample-and-augment algorithm for the MRoB problem. The algorithm proceeds in three steps:

1. *Sampling Step:* Choose a random subset $S \subseteq R$ of terminal pairs by picking every terminal pair $(s_i, t_i) \in R$ independently with probability $p_i := \min\{d_i/M, 1\}$.
2. *Subproblem Step:* Compute an α-approximate Steiner forest F_S on S and buy all edges in F_S.
3. *Augmentation Step:* Augment F_S to a feasible solution for R by renting additional edges to connect all remaining terminal pairs in $R \setminus S$ in the cheapest possible way.

A crucial building block of the sample-and-augment algorithm is the Steiner forest algorithm used in the Subproblem Step: Gupta et al. [16, 17] show that if the Steiner forest algorithm has an approximation guarantee of α and additionally admits β-*strict cost shares*, then the sample-and-augment algorithm is an $(\alpha+\beta)$-approximation algorithm for the MRoB problem.

Given a forest F in G, let $G|F$ be 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 $G|F$. A Steiner forest algorithm ALG is said to be β-*strict*, $\beta \ge 1$, if there exist nonnegative cost shares ξ_{st} for all $(s, t) \in R$ such that

(a) $\sum_{(s,t)\in R} \xi_{st} \le \mathsf{opt}_R$, where opt_R is the minimum cost of a Steiner forest for R, and
(b) $c_{G|F_{-st}}(s, t) \le \beta \cdot \xi_{st}$ 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.

The authors originally devised a 6-approximate and 6-strict algorithm for the Steiner forest problem which yields a 12-approximation algorithm for the MRoB problem.

Their analysis can be tightened to achieve an 8-approximation algorithm. Becchetti et al. [6] reduced the approximation ratio to 6.828 by developing a $(2+\sqrt{2})$-approximate and $(2+\sqrt{2})$-strict primal-dual Steiner forest algorithm.

A slight adaptation of the sample-and-augment algorithm yields an $(\alpha+\beta)$-approximation algorithm for the MuRoB problem [17]: Pick every terminal group g_i independently with probability p_i and buy the edges of an α-approximate *group Steiner forest* (also known as *generalized Steiner tree*) F_S on the set S of all selected groups. Then augment F_S by connecting all groups in $R \setminus S$ in the cheapest possible way. The group Steiner forest algorithm needs to connect all terminals that belong to the same group g_i in S. It is easy to see that this can be accomplished by using a Steiner forest algorithm where every group g_i is represented by a set of terminal pairs. However, the major difference is that the Steiner forest algorithm then needs to satisfy a stronger strictness definition, called *group-strictness*.

Given a terminal group $g \in R$, let $c_{G|F}(g)$ denote the minimum cost of connecting all terminals of g in $G|F$. A Steiner forest algorithm ALG is *β-group-strict*, $\beta \geq 1$, if there exist nonnegative cost shares ξ_g for all $g \in R$ such that

(a) $\sum_{g\in R} \xi_g \leq \mathsf{opt}_R$, where opt_R is the minimum cost of a Steiner forest for R, and

(b) $c_{G|F_{-g}}(g) \leq \beta \cdot \xi_g$ for all $g \in R$, where F_{-g} is a Steiner forest for terminal set $R_{-g} = R \setminus \{g\}$ returned by ALG.

Stochastic Steiner tree and forest. The *stochastic Steiner tree* (SST) *problem* that we consider in this paper is the Steiner tree problem in the model of two-stage stochastic optimization with recourse. We are given a undirected graph $G = (V, E)$ with nonnegative costs c_e for all edges $e \in E$, a probability distribution $\pi : 2^V \to [0, 1]$ on the subsets of vertices, and an inflation factor $\sigma \geq 1$. Here $\pi(R)$ is the probability that the subset $R \subseteq V$ is realized as the terminal set. Decisions are made in two stages: In the first stage, we can choose to buy an arbitrary subset F_0 of the edges at cost $c(F_0) := \sum_{e\in F_0} c_e$. In the second stage, a subset R of terminals is realized, and additional edges F_R can be bought at an inflated cost $\sigma \cdot c(F_R)$. The objective is to buy a set of edges in stages one and two so that all vertices in R are connected and the expected total cost $c(F_0) + \sigma\mathbf{E}[c(F_R)]$ is minimized.

We consider two stochastic models in this paper: In the *independent decision model* every vertex $v \in V$ is realized independently with probability $p_v \in [0, 1]$ such that $\sum_{v\in V} p_v = 1$; a set of terminals R is then realized with probability $\Pi_{v\in R} p_v \Pi_{v\notin R}(1 - p_v)$. In the *black-box model*, we make no assumptions about the distribution π except that we have access to it via a sampling oracle: on request, the oracle outputs a subset of vertices R drawn from the distribution. Note that in the latter model the number of scenarios $R \subseteq V$ with positive probability $\pi(R)$ might be exponential in $|V|$.

The *stochastic Steiner forest* (SSF) *problem* is a generalization of the SST problem where π defines a probability distribution over all possible subsets $R \subseteq V \times V$ of terminal pairs and the goal is to buy a subset of edges such that every terminal pair in R is connected. Both models described above naturally extend to the SSF problem.

Extending the strictness notion introduced in [16, 17, 18], Gupta et al. [20] provide a general *boost-and-sample* framework for two-stage stochastic optimization problems in the black-box model. The framework applies to optimization problems that satisfy a certain *subadditivity* condition. We refer the reader to [20] for a general statement of this condition and discuss here only the specific case of the SST problem. In this case, the condition states the following: let R_1 and R_2 be any two samples from the given probability distribution, and let T_1 and T_2 be any two feasible Steiner trees spanning the terminals in R_1 and R_2, respectively. Then (a) $R_1 \cup R_2$ is a potential sample from the given distribution, and (b) $T_1 \cup T_2$ is a feasible Steiner tree for $R_1 \cup R_2$.

It is not difficult to construct examples that show that the SST problem, in general, does not satisfy the above subadditivity condition. However, for the *rooted* version of the problem, where we require that every sample of terminals contain a common root vertex r, the condition is satisfied. In this case, the framework by Gupta et al. yields a 3.39-approximation algorithm. The authors also show that an α-approximate and β-strict algorithm gives rise to an $(\alpha+\beta)$-approximation algorithm for the respective stochastic optimization problem in the independent decision model. Using the result in [16], they obtain an 8-approximation algorithm for the SSF problem in the independent decision model.

An adaptation of their framework was used by Gupta and Pál [19] to derive an approximation algorithm for the SST problem *without* a fixed root in the black-box model. Their algorithm works as follows:

1. *Sampling Step:* Draw $\lfloor \sigma \rfloor$ independent samples $g_1, \dots, g_{\lfloor \sigma \rfloor}$ of terminals from the sampling oracle π, and let $g := \cup_i g_i$.
2. *First Stage Solution:* Compute an α-approximate group Steiner forest F_0 on g.
3. *Second Stage Solution:* When the actual terminal set R realizes, augment F_0 to a feasible solution for R by adding additional edges $F_R \subseteq E$ in the cheapest possible way.

Gupta and Pál [19] show that this boost-and-sample algorithm yields an $(\alpha + \beta)$-approximation algorithm for the SST problem without a root in the black-box model if the Steiner forest algorithm used to build the first stage solution is α-approximate and β-group-strict. The authors then derive a 12.8-approximation for the SST problem by devising a $3 + \sqrt{5}$-approximate and $4 + \frac{8}{\sqrt{5}}$-group-strict Steiner forest algorithm.

Our contributions. The strict Steiner forest algorithms presented in [6, 16, 19] all adapt the primal-dual approximation algorithms for the Steiner forest problem [1, 13]: strictness is achieved by adding extra edges to the approximate Steiner forest produced by these algorithms. This worsens the approximation ratio but reduces the cost of augmenting a feasible forest F_{-g} into a feasible forest for R. In this paper, we show that the primal-dual Steiner forest algorithms [1, 13] are 3-strict and 4-group-strict with appropriate cost sharing rules. We summarize our main result 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 the MRoB problem and a 6-approximation for the MuRoB problem using the framework in [16, 17]. In the context of stochastic optimization, we obtain a 6-approximation for the SST problem (without a fixed root) in the black-box model and a 5-approximation for the SSF problem in the independent decision model using the framework in [19, 20]. We summarize the implications of Theorem 1 in Table 1.

This is the first algorithm to show that the *unmodified* primal-dual Steiner forest algorithms [1, 13] have *constant* strict or group-strict cost shares. Finally, we present an example instance that shows that several natural primal-dual Steiner forest algorithms are not $(\frac{8}{3} - \epsilon)$-strict for any $\epsilon > 0$, therefore showing that the two-stage analysis of Gupta et al. given in [16, 17] is nearly tight for the MRoB problem.

Table 1
Approximation ratios obtained for the different problems considered in this paper. The $\star$ indicates that this refers to the independent decision model only.

Problem	Previous best		This paper
Multicommodity rent-or-buy	6.828	[6]	5
Multicast rent-or-buy	12.8	[19]	6
Stochastic Steiner tree	12.8	[19]	6
Stochastic Steiner forest	$6.828^\star$	[6]	$5^\star$

Related work. The MRoB problem is a special case of the *multicommodity buy-at-bulk* (MBaB) *problem*. An instance of this problem is defined as for the MRoB problem, but additionally we are given a subadditive and 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 us to route d_i units of flow between every terminal pair $(s_i, t_i) \in R$ simultaneously. The cost of installing the capacities given by 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. Awerbuch and Azar [3] present an $O(\alpha)$-approximation for the MBaB problem, assuming that any metric can be probabilistically approximated by a family of tree metrics with an expected distortion at most α. Bartal shows that $\alpha = O(\log^2 n)$ [4] and later improves this bound to $\alpha = O(\log n \log\log n)$ [5]. More recently, Fakcharoenphol, Rao, and Talwar [11] show that $\alpha = O(\log n)$. On the hardness side, Andrews [2] shows that MBaB does not have an $O(\log^{1/4-\epsilon} n)$-approximation algorithm for any $\epsilon > 0$ unless $\mathsf{NP} \subseteq \mathsf{ZTIME}(n^{\mathrm{polylog}(n)})$.

The algorithm by Awerbuch and Azar [3] in combination with Bartal's tree embedding [4, 5, 11] was the best known approximation algorithm for MRoB for several years. The first constant-factor approximation algorithm for the MRoB problem is due to Kumar, Gupta, and Roughgarden [23]. The randomized sample-and-augment algorithm for the MRoB problem was given by Gupta et al. [16, 17]. This framework has been used successfully in recent years to obtain improved approximation algorithms for the MRoB problem [6, 18] and also constitutes the basis for our result.

The sample-and-augment framework was originally introduced by Gupta et al. [17, 18] to derive constant-factor approximation algorithms for the SSRoB, virtual private network design, and single-sink buy-at-bulk problems. The current best approximation algorithm for the SSRoB special case of the MRoB problem is due to Eisenbrand et al. [10], who extend the sample-and-augment framework in [17, 18] to connected facility location problems. They derive a 2.8-approximation algorithm for the SSRoB problem.[1] Moreover, the algorithm can be derandomized leading to a deterministic 3.28-approximation algorithm for SSRoB [10, 27, 28].

The existing literature on two-stage stochastic optimization with recourse is vast. However, only recently have researchers started to attempt to derive algorithms with provable approximation guarantees for stochastic variants of NP-hard optimization problems; see [9, 21, 22, 24] for some examples. General frameworks to derive approximation algorithms for stochastic optimization problems were proposed in [20, 26].

The rooted stochastic Steiner tree problem has been addressed in [14, 20, 21, 22]. Gupta, Ravi, and Sinha [21] give a constant-factor approximation algorithm for the

[1] The original approximation ratio of 2.92 stated in [10] is based on the Steiner tree approximation ratio $\rho_{ST} = 1.55$ [25]. It reduces to 2.8 by using the currently best Steiner tree approximation algorithm with $\rho_{ST} \approx \ln 4$ [8].

rooted SST problem if the number of scenarios with positive probability is polynomially bounded. The current best approximation algorithm for the rooted SST problem in the black-box model is the randomized 3.39-approximation algorithm based on the boost-and-sample framework given in [20]. This is also the best approximation guarantee currently know for the independent decision model. Van Zuylen [28] derandomized the algorithm and obtained a deterministic 8-approximation algorithm for the rooted SST problem in the independent decision model. Furthermore, Goyal et al. [14] recently gave a deterministic primal-dual 8-approximation algorithm for the rooted SST problem with a polynomial number of scenarios. All known approximation algorithms for the SST problem *without* a root in the black-box model are based on the adaptation of the boost-and-sample framework given by Gupta and Pál [19]. The authors derive a 12.8-approximation algorithm which is also the current best in this setting.

The boost-and-sample framework [20] also provides a means to derive approximation algorithms for the stochastic Steiner forest problem in the black-box model. However, this requires a Steiner forest algorithm that admits a cost sharing scheme that satisfies a very strong notion of strictness (see [20] for more details), and it is still an open question whether such cost shares exist. In the case of the independent decision model, the boost-and-sample framework has been used successfully to obtain constant-factor approximation algorithms for the SSF problem; prior to this work, the best approximation guarantee was 6.828 [6]. Very recently, Gupta and Kumar [15] gave a constant-factor primal-dual approximation algorithm for the SSF problem in the black-box model. Their algorithm does not use the boost-and-sample framework.

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 of Agrawal, Klein, and Ravi [1] and show how the cost of every edge of the computed forest 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 in [16, 17].

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 the 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 $\mathcal{W}_e = \{u, v\}$ to be the *witnesses* of e

and split $\xi(e)$ between the terminals in $\mathcal{W}_e$. 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 $u \in \mathcal{W}_e$ according to the split. The total cost share assigned to terminal u is

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

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 (a) of the definition of group-strictness.

LEMMA 1. *Let F be a Steiner forest computed by an α-approximate algorithm* ALG*, and let $\{\mathcal{W}_e\}_{e \in F}$ 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

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

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

2.1. Symmetric cost share assignment. Crucial to proving the strictness of our cost sharing scheme is to define the witness set $\{\mathcal{W}_e\}_{e \in F}$ 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 $\mathcal{W}_e \cap g = \emptyset$, 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., $\mathcal{W}_e \cap g \neq \emptyset$.

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*: The cost share that each witness $u \in \mathcal{W}_e$ 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 $\{\mathcal{W}_e\}_{e \in F}$ be the witness set. If algorithm* ALG *and $\{\mathcal{W}_e\}_{e \in F}$ 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 $\mathcal{W}_e \cap g \neq \emptyset$, there is some terminal u in g with $\xi_u(e) = \frac{1}{2}\xi(e) = \frac{1}{2\alpha}c(e)$. ∎

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 $\mathcal{W}_e$. We prove that this asymmetric cost sharing scheme yields $\frac{3}{2}\alpha$-strict cost shares, if only algorithm ALG and the witness set $\{\mathcal{W}_e\}_{e\in F}$ 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, then 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 $\mathcal{W}_e \cap g = \emptyset$ 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 works only 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: 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 $\xi_{u\bar{u}}$. $P_{u\bar{u}}$ is the unique path in F connecting u and $\bar{u}$.

We define an *asymmetric cost share assignment* as follows: Consider an edge $e \in F$ and let $\mathcal{W}_e = \{u, v\}$. If u and v belong to the same terminal pair, then we may split $\xi(e)$ arbitrarily between $\xi_u(e)$ and $\xi_v(e)$; e.g., let $\xi_u(e) = \xi_v(e) = \frac{1}{2}\xi(e)$. Without loss of generality, we will now assume that u and v belong to different terminal pairs, and that $(u, \bar{u}) \prec (v, \bar{v})$. We share $\xi(e)$ among the two witnesses u and v as follows:

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

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

LEMMA 3. *Let F be a Steiner forest computed by an α-approximate algorithm ALG and let $\{\mathcal{W}_e\}_{e\in F}$ be the witness set. If algorithm ALG and $\{\mathcal{W}_e\}_{e\in F}$ 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. We prove that we can establish a connection between s and t in the graph $G|F_{-st}$ at cost at most $\frac{3}{2}\alpha$ times the cost share ξ_{st} of terminal pair (s, t). Consider the path P_{st} that connects s and t in F. Let X_{st} be the set of edges of P_{st} that are not part of F_{-st}, i.e., $X_{st} = \{e \in P_{st} : e \notin F_{-st}\}$. From Property 1 we know that each edge $e \in X_{st}$ is witnessed by either s or t. We partition the set of edges in $G|F_{-st}$ that are witnessed by s or t (not necessarily lying on P_{st}) into two sets

$$\mathcal{M}_{st}^{+} = \{e \in G|F_{-st} \,:\, \xi_{st}(e) \ge \tfrac{2}{3}\xi(e)\} \quad \text{and} \quad \mathcal{M}_{st}^{-} = \{e \in G|F_{-st} \,:\, \xi_{st}(e) = \tfrac{1}{3}\xi(e)\}.$$

For an edge set $S \subseteq E$, define $\xi_{st}(S)$ as the total cost share that terminal pair (s, t) receives for the edges in S, i.e., $\xi_{st}(S) = \sum_{e\in S} \xi_{st}(e)$. Note that for each edge

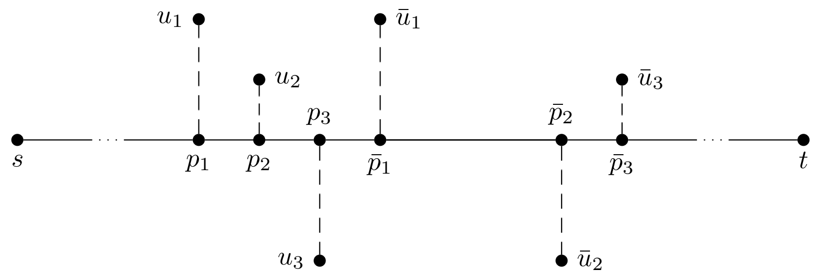

FIG. 1. *The figure shows the path P_{st} and three terminal pairs $(u_i, \bar{u}_i), (u_j, \bar{u}_j) \in I_{st}$ with $1 \leq i < j \leq q$.*

$e \in \mathcal{M}_{st}^+$, $\frac{3}{2}\alpha$ times the cost share $\xi_{st}(e)$ is sufficient to cover the cost of e. We can therefore contract the edges in $\mathcal{M}_{st}^+$ in $G|F_{-st}$ using $\frac{3}{2}\alpha$ times their cost share $\xi_{st}(\mathcal{M}_{st}^+)$. Subsequently, we assume without loss of generality that each edge e in $G|F_{-st}$ that is witnessed by s or t belongs to $\mathcal{M}_{st}^-$. We show that we can connect s and t at a cost at most $\frac{3}{2}\alpha\xi_{st}(\mathcal{M}_{st}^-)$.

Consider an edge $e \in X_{st}$. Then $e \in \mathcal{M}_{st}^-$. By the cost share assignment given in (3), 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 is part of the path $P_{u\bar{u}}$ that connects u and $\bar{u}$ in F. 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 $P_{u\bar{u}}$ provided in F_{-st}.

Let I_{st}^0 be the set of terminal pairs that witness edges in X_{st} together with one of s and t:

$$I_{st}^0 = \{(u, \bar{u}) \in R : \exists e \in X_{st} \text{ such that } \{s, t\} \cap \mathcal{W}_e \neq \emptyset \text{ and } \{u, \bar{u}\} \cap \mathcal{W}_e \neq \emptyset\}.$$

By assumption, s and t obtain cost share $\frac{1}{3}\xi(e)$ for all edges e in X_{st}. Hence, for every $(u, \bar{u}) \in I_{st}^0$ we have $(u, \bar{u}) \prec (s, t)$. It follows from Property 2 that every edge $e \in P_{u\bar{u}} \setminus P_{st}$ that is not witnessed by s or t must be part of F_{-st}. For a terminal pair $(u, \bar{u}) \in I_{st}^0$, we let $\bar{P}_{u\bar{u}} = P_{u\bar{u}} \cap P_{st}$ be the nonempty subpath of $P_{u\bar{u}}$ consisting of edges on P_{st}. For distinct terminal pairs $(u, \bar{u}), (w, \bar{w}) \in I_{st}^0$, we say that $(u, \bar{u})$ *encapsulates* $(w, \bar{w})$ if $\bar{P}_{w\bar{w}} \subseteq \bar{P}_{u\bar{u}}$. As long as I_{st}^0 contains such pairs $(u, \bar{u})$ and $(w, \bar{w})$, remove $(w, \bar{w})$ from I_{st}^0. Call the final set of pairwise nonencapsulating terminal pairs I_{st}.

For ease of notation, we assume that

$$I_{st} = \{(u_1, \bar{u}_1), \dots, (u_q, \bar{u}_q)\},$$

and we use P_i for the unique $u_i, \bar{u}_i$-path in F. Also define p_i and $\bar{p}_i$ to be the endpoints of $\bar{P}_{u_i\bar{u}_i}$ and assign these labels such that p_i is closer to s than $\bar{p}_i$. We will call p_i and $\bar{p}_i$ the *projections* of terminal pair $(u_i, \bar{u}_i) \in I_{st}$. Since the pairs of I_{st} are pairwise nonencapsulating, we may order the indices such that p_i is closer to s than p_j and $\bar{p}_i$ is closer to s than $\bar{p}_j$ for all $1 \leq i < j \leq q$. Refer to Figure 1 for an example.

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

$$\mathcal{P}^i = P_{st}^i \cup P_1 \cup \cdots \cup P_i$$

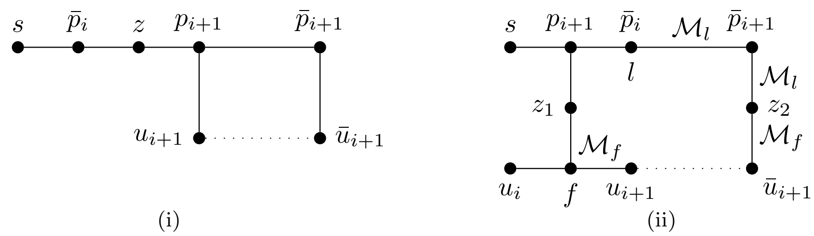

FIG. 2. *The figure illustrates the cases used in the proof of Lemma* 3. *Solid edges represent segments of P_{st}, and dotted edges represent connectivity in F_{-st}.*

as the union of path P^i_{st} and the paths of the first i terminal pairs in I_{st}. It is important for the remainder of this proof to observe that the edges in $\mathcal{P}^i$ form a tree for all i, as F is a forest.

Finally, let

$$\mathcal{M}^i_{st} = \mathcal{M}^-_{st} \cap \mathcal{P}^i$$

be the set of edges in $\mathcal{P}^i$ that are witnessed by s or t. We first prove the following proposition.

PROPOSITION 1. *For every $0 \le i \le q$ and for every vertex $z \in \mathcal{P}^i$, we can reconstruct an s, z-path in $G|F_{-st}$ at a cost at most $\frac{3}{2}\alpha\xi_{st}(\mathcal{M}^i_{st})$.*

Proof. The proof is by induction over $0 \le i \le q$. The claim 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} and $\bar{p}_i \ne p_{i+1}$ (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 $1 \le j \le q$ since the projections of the pairs in I_{st} are nonnested. If s or t witnessed some edge $e \in P$, they would receive at least a cost share of $\frac{2}{3}\xi(e)$, and thus $e \in \mathcal{M}^+_{st}$. We excluded those edges initially, and hence, using the inductive hypothesis, we can reconstruct an s, z-path in $G|F_{-st}$ at a cost at most $\frac{3}{2}\alpha\xi_{st}(\mathcal{M}^i_{st})$.

For the above argument, we will now assume that p_{i+1} precedes $\bar{p}_i$ on P_{st} (possibly $\bar{p}_i = p_{i+1}$). Let f and l be the first and last vertices on P_{i+1}, respectively, that are incident to edges of $\mathcal{P}^i$ (refer to Figure 2(ii)). Clearly, the unique f, l-path P_{fl} in F must be contained in $\mathcal{P}^i$. 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 $G|F_{-st}$ at a cost at most $\frac{3}{2}\alpha\xi_{st}(\mathcal{M}^i_{st})$.

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 $G|F_{-st}$:

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 $\mathcal{M}_f$ 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 $G|F_{-st}$ at a cost at most $\frac{3}{2}\alpha\xi(\mathcal{M}^i_{st})$. 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. Recall that the edges of P_f that are missing in F_{-st} must be witnessed by s or t (because $(u_{i+1}, \bar{u}_{i+1}) \prec (s, t)$ and by Property 2). Thus this costs at most $3\alpha\xi_{st}(\mathcal{M}_f)$.
2. Use P_l for the l, z_2-segment of P_{i+1}, and let $\mathcal{M}_l$ be the set of edges on P_l that are witnessed by s or t. We can then inductively reconstruct an s, l-path in $G|F_{-st}$ at a cost at most $\frac{3}{2}\alpha\xi(\mathcal{M}^i_{st})$. We obtain an s, z_2-path by

reconstructing P_l. Note that the only edges in P_l that are missing in F_{-st} must be witnessed by s or t. Thus, reconstructing P_l costs at most $3\alpha\xi_{st}(\mathcal{M}_l)$.

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

$$\frac{3}{2}\alpha\xi_{st}(\mathcal{M}_{st}^i) + 3\alpha\min\{\xi_{st}(\mathcal{M}_f), \xi_{st}(\mathcal{M}_l)\} \leq \frac{3}{2}\alpha\left(\xi_{st}(\mathcal{M}_{st}^i) + \xi_{st}(\mathcal{M}_f) + \xi_{st}(\mathcal{M}_l)\right).$$

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

By the above proposition, we can establish a connection between s and $\bar{p}_q$ in $G|F_{-st}$ at a cost at most

$$\frac{3}{2}\alpha\xi_{st}(\mathcal{M}_{st}^q) \leq \frac{3}{2}\alpha\xi_{st}(\mathcal{M}_{st}^-).$$

Finally, observe that every edge $e \in X_{st}$ on the $\bar{p}_q, t$-segment of P_{st} must belong to $\mathcal{M}_{st}^+$. We excluded these edges from $G|F_{-st}$ initially. Thus we can reconstruct an s, t-path in $G|F_{-st}$ at a total cost at most $\frac{3}{2}\alpha\xi_{st}(\mathcal{M}_{st}^-)$, which concludes the proof. □

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 forests presented in [1] and [13] differ only slightly. In this paper, we focus on the viewpoint taken in [1]. 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 $\mathcal{U}$ contain exactly those subsets U of V that *separate* at least one terminal pair in R. In other words, $U \in \mathcal{U}$ 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:

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

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

c_e.

$$\text{(D)} \qquad \max \sum_{U \in \mathcal{U}} y_U$$

$$\text{(4)} \qquad \text{s.t.} \sum_{U \in \mathcal{U}: e \in \delta(U)} y_U \leq c_e \quad \forall e \in E,$$

$$y \geq 0.$$

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 the 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 $U \in \mathcal{U}$. 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.[2] 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 $C \subseteq U$ for some moat $U \in \bar{F}^\tau$. A moat U of $\bar{F}^\tau$ is active at time τ if U contains an active terminal; U is inactive otherwise. Let $\mathcal{A}^\tau$ be the set of all active moats in $\bar{F}^\tau$ at time τ. AKR raises the dual variables for all sets in $\mathcal{A}^\tau$ 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 [1].

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*

[2]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.

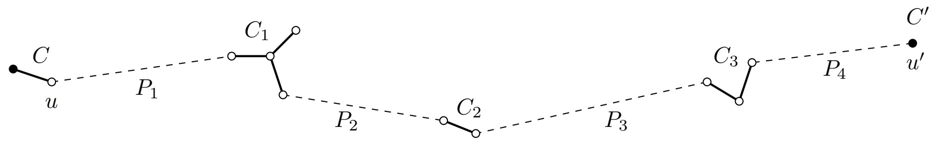

FIG. 3. *A path P that becomes tight at time τ_P in* $\mathsf{AKR}(R)$.

the given input instance with terminal set R.

3.2. Witness definition. We define a set $\{\mathcal{W}_e\}_{e\in F}$ of witnesses that are used to distribute the cost shares as described in section 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 $\mathcal{W}_e$ 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 $C_1, \dots, C_l$ of connected components of F^{τ_P}, and ends in a node u' of a connected component C' of F^{τ_P}. Let $P_1, \dots, P_{l+1}$ be the sequence of paths of $P \setminus F^{\tau_P}$, and let $\mathcal{P}_P$ be the set of edges in $P \setminus F^{\tau_P}$. When P becomes tight, the set $\mathcal{P}_P$ is added to F, and we determine for each edge $e \in \mathcal{P}_P$ the corresponding witnesses $\mathcal{W}_e$ as follows.

Each edge $e \in \mathcal{P}_P$ will have the same witness set $\mathcal{W}_e = \{w, w'\}$. 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 $\mathcal{A}_C$ be the set of terminals in C that are active at time τ_P. By definition of C, all terminals in $\mathcal{A}_C$ are connected to u in F^{τ_P}.

LEMMA 4. *Let τ_u be the first time that moat $U^\tau(u)$ collides with a moat U_u containing a terminal in $\mathcal{A}_C$. There is a terminal w in $U_u \cap \mathcal{A}_C$ whose moat collides with u's moat at time τ_u even if all terminals in $\mathcal{A}_C \setminus \{w\}$ are not part of the terminal set R.*

Proof. If $u \in \mathcal{A}_C$, then $\tau_u = 0$ and $w = u$. Otherwise, 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 $\mathcal{A}_C$. Then there is a terminal $w \in U_v \cap \mathcal{A}_C$ whose moat collides with v's moat at time τ_v even if all terminals in $\mathcal{A}_C \setminus \{w\}$ 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 $\mathcal{A}_C$ before time τ_v, and therefore the growth of v's moat until time τ_v is not affected by the removal of $\mathcal{A}_C$.

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

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 $\mathcal{A}_C$, 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 $\mathcal{A}_C \setminus \{w\}$ are not part of the terminal set.

Assume that z is not in $\mathcal{A}_C$. 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 $\mathcal{A}_C$. 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 $\mathcal{A}_C \setminus \{w\}$ are not part of the terminal set R. Since w is in $\mathcal{A}_C$, it causes the moat containing z to grow after time τ_z regardless of other terminals in $\mathcal{A}_C$. Thus, w's moat collides with that of v at time τ_v, and this finishes the proof of 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 $\{\mathcal{W}_e\}_{e\in F}$ be the set of witnesses assigned by AKR. Let $\mathcal{G}_{-g}$ (where $\mathcal{G} = U, F$ or $\bar{F}$) refer to set $\mathcal{G}$ 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 $\tau < \tau_g$. We will show that it remains true at time $\tau + \epsilon$ for any small $\epsilon > 0$.

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 $U^{\tau+\epsilon}_{-g}(v) = U^{\tau+\epsilon}(v)$ if $U^{\tau+\epsilon}(v) \cap g = \emptyset$; and $U^{\tau+\epsilon}_{-g}(v) \subseteq U^{\tau+\epsilon}(v)$ otherwise. Now assume $U^\tau(v) \cap g \neq \emptyset$ and thus $U^\tau_{-g}(v) \subseteq U^\tau(v)$. Clearly, $U^{\tau+\epsilon}(v) \cap g \neq \emptyset$. Since $\tau < \tau_g$, all terminals in g are active at time τ and thus $U^\tau(v)$ is active at time τ. It follows that $U^{\tau+\epsilon}_{-g}(v) \subseteq U^{\tau+\epsilon}(v)$. ∎

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 components C and C' that becomes tight at time $\tau_P \le \tau_g$ in the execution of $\mathsf{AKR}(R)$. Recall that the moats $U^{\tau_P}(C)$ and $U^{\tau_P}(C')$ are active at time τ_P. 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 that P is witnessed by w and w'.

LEMMA 6. *Assume that neither of the two witnesses w, w' of P is in g, i.e., $\mathcal{W}_e \cap g = \emptyset$ for all edges $e \in \mathcal{P}_P$. 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, the edges in $\mathcal{P}_P$ are 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 $1 \le i \le l$, let $\tau_i = \tau$ be the first time at which the moat $U^\tau(C_i)$ of component C_i becomes inactive. Then $\tau_i < \tau_P \le \tau_g$ and thus $C_i \cap g = \emptyset$. Then, by Lemma 5, $U^\tau_{-g}(v) = U^\tau(v)$ for all $\tau \le \tau_i$, for all $v \in C_i$, and 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 $\mathcal{A}_C$. By the definition of $\hat{\tau}$, $U^\tau(u) \cap g = \emptyset$ for all $\tau \in [0, \hat{\tau})$. Thus, by Lemma 5, $U^\tau_{-g}(u) = U^\tau(u)$ for all $\tau \in [0, \hat{\tau})$; and the contribution to (4) from

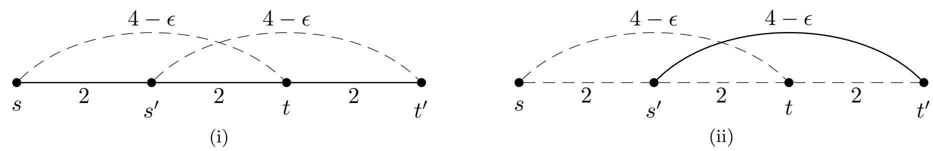

FIG. 4. *Instance used in the lower bound argument.*

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 $\mathsf{AKR}(R_{-g})$ before time τ_P. Hence the edges in $\mathcal{P}_P$ are added at time τ_P in $\mathsf{AKR}(R_{-g})$, and the lemma follows. $\square$

Note that the above lemma implies Property 1 for our definition of witnesses. To see this, consider an arbitrary edge e in the tree T_g of $\mathsf{AKR}(R)$ and assume $\mathcal{W}_e \cap g = \emptyset$. Let P be the path that becomes tight at time τ_P in the run $\mathsf{AKR}(R)$ with $e \in \mathcal{P}_P$. Then $\tau_P \leq \tau_g$. By Lemma 6, the edges in $\mathcal{P}_P$ are added at time τ_P in the run $\mathsf{AKR}(R_{-g})$, and thus $e \in F_{-g}$.

We show that the following precedence order $\prec$ together with the witness definition described above implies Property 2 for AKR. Consider the run $\mathsf{AKR}(R)$, and fix an order on the terminal groups in $R = \{g_i\}_{1 \leq i \leq k}$ such that

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

We define $g_i \prec g_j$ if $i \leq j$ 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 of tree T_h in F. If $\mathcal{W}_e \cap g = \emptyset$, 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 $\tau \leq \tau_h \leq \tau_g$. By Lemma 6 and since $\mathcal{W}_e \cap g = \emptyset$, 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 $\xi_{st} \leq \frac{3}{2}$. Running AKR on terminal set $R_{-st} = \{(s',t')\}$ yields the forest in Figure 4(ii). As $c_{G|F_{-st}}(s,t) = 4-\epsilon$, this example shows that the strictness of AKR is at least $(4-\epsilon)/\frac{3}{2} \approx \frac{8}{3}$ 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 [16] 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 [16] 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.

REFERENCES

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

[2] M. Andrews, *Hardness of buy-at-bulk network design*, in Proceedings of the IEEE Symposium on Foundations of Computer Science, IEEE, Washington, DC, 2004, pp. 115–124.

[3] B. Awerbuch and Y. Azar, *Buy-at-bulk network design*, in Proceedings of the IEEE Symposium on Foundations of Computer Science, IEEE, Washington, DC, 1997, pp. 542–547.

[4] Y. Bartal, *Probabilistic approximation of metric spaces and its algorithmic applications*, in Proceedings of the IEEE Symposium on Foundations of Computer Science, IEEE, Washington, DC, 1996, pp. 184–193.

[5] Y. Bartal, *On approximating arbitrary metrics by tree metrics*, in Proceedings of the ACM Symposium on Theory of Computing, ACM, New York, 1998, pp. 161–168.

[6] L. Becchetti, J. Könemann, S. Leonardi, and M. Pál, *Sharing the cost more efficiently: Improved approximation for multicommodity rent-or-buy*, ACM Trans. Algorithms, 3 (2007), article 23.

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

[8] J. Byrka, F. Grandoni, T. Rothvoss, and L. Sanità, *An improved LP-based approximation for Steiner tree*, in Proceedings of the ACM Symposium on Theory of Computing, ACM, New York, 2010, pp. 583–592.

[9] S. Dye, L. Stougie, and A. Tomasgar, *The stochastic single resource service-provision problem*, Naval Res. Logist., 50 (2003), pp. 869–887.

[10] F. Eisenbrand, F. Grandoni, T. Rothvoss, and G. Schäfer, *Approximating connected facility location problems via random facility sampling and core detouring*, in Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, SIAM, Philadelphia, 2008, pp. 1174–1183.

[11] J. Fakcharoenphol, S. Rao, and K. Talwar, *A tight bound on approximating arbitrary metrics by tree metrics*, in Proceedings of the ACM Symposium on Theory of Computing, ACM, New York, 2003, pp. 448–455.

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

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

[14] V. Goyal, A. Gupta, S. Leonardi, and R. Ravi, *Pricing tree access networks with connected backbones*, in Proceedings of the European Symposium on Algorithms, Lecture Notes in Comput. Sci. 4698, Springer, New York, 2007, pp. 498–509.

[15] A. Gupta and A. Kumar, *A constant-factor approximation for stochastic Steiner forest*, in Proceedings of the ACM Symposium on Theory of Computing, ACM, New York, 2009, pp. 659–668.

[16] 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 of the IEEE Symposium on Foundations of Computer Science, IEEE, Washington, DC, 2003, pp. 606–617.

[17] A. Gupta, A. Kumar, M. Pál, and T. Roughgarden, *Approximation via cost sharing: Simpler and better approximation algorithms for network design*, J. ACM, 54 (2007), article 11.

[18] A. Gupta, A. Kumar, and T. Roughgarden, *Simpler and better approximation algorithms for network design*, in Proceedings of the ACM Symposium on Theory of Computing, ACM, New York, 2003, pp. 365–372.

[19] A. Gupta and M. Pál, *Stochastic Steiner trees without a root*, in Proceedings of the International Colloquium on Automata, Languages and Programming, Lisbon, Portugal, 2005, pp. 1051–1063.

[20] A. Gupta, M. Pál, R. Ravi, and A. Sinha, *Boosted sampling: Approximation algorithms for stochastic optimization*, in Proceedings of the ACM Symposium on Theory of Computing, ACM, New York, 2004, pp. 417–426.

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

[22] N. Immorlica, D. Karger, M. Minkoff, and V. S. Mirrokni, *On the costs and benefits of procrastination: Approximation algorithms for stochastic combinatorial optimization problems*, in Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, SIAM, Philadelphia, 2004, pp. 691–700.

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

[24] R. Ravi and A. Sinha, *Hedging uncertainty: Approximation algorithms for stochastic optimization problems*, Math. Program., 108 (2006), pp. 97–114.

[25] G. Robins and A. Zelikovsky, *Improved Steiner tree approximation in graphs*, in Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, SIAM, Philadelphia, 2000, pp. 770–779.

[26] D. B. Shmoys and C. Swamy, *An approximation scheme for stochastic linear programming and its application to stochastic integer programs*, J. ACM, 53 (2006), pp. 978–1012.

[27] D. Williamson and A. van Zuylen, *A simpler and better derandomization of an approximation algorithm for single source rent-or-buy*, Oper. Res. Lett., 35 (2007), pp. 707–712.

[28] A. van Zuylen, *Deterministic sampling algorithms for network design*, in Proceedings of the European Symposium on Algorithms, Lecture Notes in Comput. Sci. 5193, Springer, New York, 2008, pp. 830–841.