

# A GROUP-STRATEGYPROOF COST SHARING MECHANISM FOR THE STEINER FOREST GAME\*

JOCHEN KÖNEMANN<sup>†</sup>, STEFANO LEONARDI<sup>‡</sup>, GUIDO SCHÄFER<sup>§</sup>, AND STEFAN H. M. VAN ZWAM<sup>¶</sup>

**Abstract.** We consider a game-theoretical variant of the Steiner forest problem in which each player  $j$ , out of a set of  $k$  players, strives to connect his terminal pair  $(s_j, t_j)$  of vertices in an undirected, edge-weighted graph  $G$ . In this paper we show that a natural adaptation of the primal-dual Steiner forest algorithm of Agrawal, Klein and Ravi [When trees collide: An approximation algorithm for the generalized Steiner problem in networks, *SIAM Journal on Computing*, 24(3):445–456, 1995] yields a 2-budget balanced and cross-monotonic cost sharing method for this game.

We also present a negative result, arguing that no cross-monotonic cost sharing method can achieve a budget balance factor of less than 2 for the Steiner tree game. This shows that our result is tight.

Our algorithm gives rise to a new linear programming relaxation for the Steiner forest problem which we term the *lifted-cut relaxation*. We show that this new relaxation is stronger than the standard undirected cut relaxation for the Steiner forest problem.

**1. Introduction.** We consider the problem of devising a cost sharing mechanism that is group-strategyproof and satisfies approximate budget balance for a natural game-theoretic variant of the Steiner forest problem. In its most general form, the game-theoretical setting that we consider in this paper can be described as follows.

We are given a service provider and a set  $R$  of potential players (or customers, agents) that are interested in a service offered by the provider. Each player  $j$  in  $R$  has a *utility*  $u_j$  for receiving this service. We assume that  $u_j$  is kept private, i.e., that it is known only to player  $j$ . The service provider now solicits bids  $\{b_j\}_{j \in R}$  from all players and based on these bids (i) determines a set  $Q \subseteq R$  of players that receive the service, (ii) computes a solution to service all players in  $Q$ , and (iii) for each  $j \in Q$  fixes a *price*  $x_j$  that  $j$  has to pay for receiving the service. A *cost sharing mechanism* is simply a strategy that the service provider uses to make these decisions. We assume that the mechanism complies with the following three natural assumptions: (i) a player is not charged more than his bid, (ii) a player is charged only if he receives service, and (iii) a player is guaranteed to receive service if only his bid is large enough.

The total cost that is incurred to establish service for a player set  $Q \subseteq R$  is denoted by  $c(Q)$ . One objective that we wish to achieve is *approximate budget balance*. We say that a cost sharing mechanism is  $\alpha$ -*budget balanced* if

$$\frac{1}{\alpha} \cdot c(Q) \leq \sum_{j \in Q} x_j \leq \text{opt}_Q. \quad (1.1)$$

The first inequality states that at least a fraction  $1/\alpha$  of the total cost  $c(Q)$  of servicing the players in  $Q$  is recovered by the sum of the prices of the players in  $Q$ . The second

---

\*Partially supported by the EU within the 6th Framework Programme under contract 001907 (DELIS). Parts of this work appeared previously in [12] and [13].

<sup>†</sup> Department of Combinatorics and Optimization, University of Waterloo, 200 University Avenue West, Waterloo, ON N2L 3G1, Canada. Email: [jochen@uwaterloo.ca](mailto:jochen@uwaterloo.ca).

<sup>‡</sup> Dipartimento di Informatica e Sistemistica, Università di Roma “La Sapienza”, Via Salaria 113, 00198 Roma, Italy. Email: [leon@dis.uniroma1.it](mailto:leon@dis.uniroma1.it).

<sup>§</sup>Institut für Mathematik, Technische Universität Berlin, Straße des 17. Juni 136, 10623 Berlin, Germany. Email: [schaefer@math.tu-berlin.de](mailto:schaefer@math.tu-berlin.de).

<sup>¶</sup> Department of Mathematics and Computer Science, Eindhoven University of Technology, P.O. Box 513, 5600 MB, Eindhoven, The Netherlands. Email: [svzwam@win.tue.nl](mailto:svzwam@win.tue.nl).

inequality establishes fairness in that the sum of all prices is not allowed to exceed the optimal cost of servicing the players in  $Q$ , denoted  $\text{opt}_Q$ . This second inequality is often referred to as *competitiveness*. A cost sharing mechanism is *budget balanced* if  $\alpha = 1$ . Ideally we obtain cost sharing mechanisms that compute prices in polynomial time and are budget balanced. However, this is clearly impossible if the underlying problem is NP-hard and we therefore resort to cost sharing mechanisms that are approximately budget balanced.

Define the *benefit* of a player  $j$  to be  $u_j - x_j$  if  $j \in Q$ , and zero otherwise. We assume that each player is selfish and may lie about his utility in order to maximize his benefit. The task is to design a cost sharing mechanism that encourages players to bid their true utility, that is, no player or group of players should be able to benefit from lying about their utilities. A cost sharing mechanism is *strategyproof* if the dominant strategy of each player is to bid his utility; it is said to be *group-strategyproof* if this holds even if players are permitted to collude. More precisely, if, for any choice of  $i \in Q' \subseteq Q$ , the utility of a player  $i$  increases as a result of non-truthful behavior of the players in  $Q'$ , then there is at least one other player  $j \in Q'$  whose utility strictly decreases.

In [14], Moulin and Shenker presented a powerful framework that reduces the task of designing a group-strategyproof cost sharing mechanism for a game to that of giving a *cross-monotonic cost sharing method*. In fact, Immorlica et al. [9] showed that all group-strategyproof mechanism that satisfy a certain technical fairness condition can be obtained using Moulin and Shenker's framework. A *cost sharing method*  $\xi$  is an algorithm that, given any subset  $Q \subseteq R$  of players, computes a solution to service  $Q$  and for each  $j \in Q$  determines a non-negative cost share  $\xi_Q(j)$ . Analogously to the definition in (1.1), we say that  $\xi$  is  $\alpha$ -budget balanced if

$$\frac{1}{\alpha} \cdot c(Q) \leq \sum_{j \in Q} \xi_Q(j) \leq \text{opt}_Q.$$

A cost sharing method  $\xi$  is *cross-monotonic* if, for any two sets  $Q$  and  $S$  such that  $Q \subseteq S$ , and any player  $j \in Q$  we have

$$\xi_S(j) \leq \xi_Q(j).$$

In other words, the cost share of any player under the given cost sharing method does not increase if the size of the player set increases.

Given a budget balanced and cross-monotonic cost sharing method  $\xi$  for a game, the following cost sharing mechanism from [14] satisfies budget balance and group-strategyproofness: Initially, let  $Q = R$ . If, for every player  $j \in Q$ , the cost share  $\xi_Q(j)$  is less than or equal to his bid  $b_j$ , stop. Otherwise, remove from  $Q$  all players whose cost shares in  $Q$  are larger than their bids, and repeat. Eventually, let  $x_j = \xi_Q(j)$  be the prices that are charged to players in the final set  $Q$ . Jain and Vazirani [10] later proved that the result of Moulin and Shenker also holds true if one considers cross-monotonic cost sharing methods that are approximately budget balanced.

The underlying optimization problem that we consider in this context is the *Steiner forest problem*. In this problem, we are given an undirected graph  $G = (V, E)$ , a non-negative cost function  $c : E \rightarrow \mathbb{R}^+$  on the edges of  $G$ , and a set of  $k > 0$  terminal pairs  $R = \{(s_1, t_1), \dots, (s_k, t_k)\} \subseteq V \times V$ . Each terminal pair  $(s_j, t_j)$ ,  $1 \leq j \leq k$ , is associated with a player  $j$  that wants to establish a connection between vertices  $s_j$  and  $t_j$ . A feasible solution for terminal set  $R$  is a forest  $F \subseteq E$  such that vertices  $s_j$

and  $t_j$  are in the same tree of  $F$  for all  $1 \leq j \leq k$ . The objective is to find a feasible solution of smallest total cost.

The *Steiner tree* problem is a special case of the Steiner forest problem in which there is a root vertex  $r \in V$  and  $r \in \{s, t\}$  for all terminal pairs  $(s, t) \in R$ . In other words the problem consists of a set of terminals  $R \subseteq V$  and a root vertex  $r \in V$  and the goal is to connect the terminals in  $R$  to  $r$  in the cheapest possible way.

*Previous Work.* Computing minimum-cost Steiner trees and forests is NP-hard [7] and APX-complete [4, 5] and therefore, neither of the two problems admits a polynomial time approximation scheme unless  $P = NP$ . The best known algorithm for the Steiner forest problem, due to Agrawal, Klein and Ravi [2] and generalized by Goemans and Williamson [8], uses the primal-dual schema. The algorithms in [2, 8] achieve an approximation ratio of  $(2 - 1/k)$  and are both based on the classical *undirected cut formulation* for the Steiner forest problem [3]. The integrality gap of this relaxation is known to be  $(2 - 1/k)$  and the results in [2, 8] are therefore tight.

Despite the recent interest in computational game theory, examples of combinatorial optimization problems that possess cross-monotonic cost sharing methods are scarce: Moulin and Shenker [14] gave a cross-monotonic cost sharing method for problems whose optimal cost function is a submodular function of the set  $Q$ . However, this condition does not hold for many important network design problems such as Steiner trees and facility location.

The first cross-monotonic cost sharing method for the minimum-cost spanning tree game is due to Kent and Skorin-Kapov [11]. Jain and Vazirani [10] presented an alternative method that is based on the primal-dual spanning tree algorithm due to Edmonds [6]; the authors then used this result to obtain a 2-budget-balanced, cross-monotonic cost sharing method for the Steiner tree game. Pál and Tardos [15] developed a 3-budget balanced cross-monotonic cost sharing method for the facility location problem and a 15-budget balanced cross-monotonic cost sharing method for the single-source rent-or-buy network design problem.

In most of the methods proposed so far to obtain approximate budget balanced cross-monotonic cost sharing methods, the cost shares are closely related to a feasible dual solution generated by the algorithm and therefore approximate budget balance is an immediate consequence of the approximation guarantee achieved by the algorithm.

Immorlica, Mahdian and Mirrokni [9] showed that NP-hardness of the underlying combinatorial optimization problem is not the only obstruction in achieving budget balance. The authors provide lower bounds on the budget balance factor  $\alpha$  of cross-monotonic cost sharing methods for several problems. Among other results they prove (maybe most surprisingly) lower bounds of  $\Omega(n)$  and  $\Omega(n^{1/3})$  for the budget balance factor of the set cover and the vertex cover problems, respectively. The authors left open the issue of finding a lower bound on the budget balance factor for the Steiner tree problem.

Edmonds [6] proposed the *bidirected cut relaxation* for the Steiner tree problem. It is a well-known fact that an integrality-gap of  $\alpha$  for this formulation implies that no  $(\alpha - \epsilon)$ -budget balanced cross-monotonic cost sharing method can exist for any  $\epsilon > 0$  (see also [9]). The worst example known for the integrality gap of the bidirected-cut relaxation is due to Goemans (c.f. [1]) and shows a gap of  $8/7$ .

*Our Contribution.* While the performance guarantee of 2 of primal-dual approximation algorithms for the Steiner tree problem is matched by a 2-budget balanced cross-monotonic cost sharing method [10], a similar result for the Steiner forest problem was elusive so far. This contrasts the optimization version of the problem where

primal-dual  $(2 - 1/k)$ -approximation algorithms exist for both problems [2, 8].

In this paper we present a cross-monotonic cost sharing method for the Steiner forest problem that is 2-budget balanced. Our algorithm is a natural adaptation of the primal-dual algorithm for computing Steiner forests due to Agrawal, Klein and Ravi [2], which we will review in Section 2. We then show how a modification of this algorithm turns it into a cross-monotonic cost sharing method in Sections 3 and 4.

An interesting byproduct of the work in this paper is that our Steiner forest algorithm is  $(2 - 1/k)$ -approximate despite the fact that the forest computed by our method is usually costlier than those computed by known primal-dual algorithms in [2, 8]. Although we are able to prove that the cost shares computed by our algorithm are 2-budget balanced they do not correspond to a feasible dual solution for any of the known LP formulations of the Steiner forest problem. Obvious questions that arise are: Is there an alternate Steiner forest LP formulation such that the cost shares computed by our algorithm correspond to a feasible dual solution? If so, how does this new LP relaxation relate to the standard undirected-cut LP relaxation?

We answer these questions by presenting in Section 5 a new LP relaxation for the Steiner forest problem, the *lifted-cut relaxation*. The dual solution computed by our algorithm is feasible for the dual linear program of the lifted-cut relaxation. We prove that our new relaxation is stronger than the well-studied undirected cut relaxation for the Steiner forest problem. There are instances for which the optimal objective function value of our lifted-cut relaxation provides a much better approximation of the optimal cost of a Steiner forest than the undirected cut relaxation.

Unfortunately, there exist instances that show that the LP/IP gap of the lifted-cut relaxation is still nearly 2. For instances in which the lifted-cut relaxation is stronger than the undirected cut relaxation we can, however, show that the additional strength can be used to improve the performance guarantee of the existing primal-dual Steiner forest algorithms in [2, 8]. The details are presented in Section 6.

A natural question is whether there is a cross-monotonic cost sharing method for the Steiner tree and forest games that achieves a budget balance factor smaller than 2. We provide a negative answer to this question by showing (c.f. Section 7) that there is no  $(2 - \epsilon)$ -budget balanced cross-monotonic cost sharing method for Steiner trees for any  $\epsilon > 0$ . This proves that the cross-monotonic cost sharing method for the Steiner tree game [10, 11] as well as our cross-monotonic cost sharing method for the Steiner forest game are tight with respect to the budget balance factor. We remark that our lower bound holds for *any* cross-monotonic cost sharing method for the Steiner tree game, including those taking exponential time.

**2. A Primal-Dual Algorithm for the Steiner Forest Problem.** We review the algorithm of Agrawal, Klein, and Ravi [2]. Subsequently, we use **AKR** to refer to this algorithm. **AKR** is a *primal-dual* algorithm, that is, the algorithm constructs both a feasible and integral primal and a feasible dual solution for a linear programming formulation of the Steiner forest.

A standard integer programming formulation for the Steiner forest problem has a binary variable  $x_e$  for all edges  $e \in E$ :  $x_e$  has value 1 if edge  $e$  is part of the resulting forest and 0 otherwise. A subset  $U \subseteq V$  is a *Steiner cut* if it separates at least one terminal pair in  $R$ . In other words,  $U$  is a Steiner cut iff there is a pair  $(s, t) \in R$  with  $|\{s, t\} \cap U| = 1$ . We use  $\mathcal{S}$  to refer to the set of all Steiner cuts. For a subset  $U \subseteq V$  we define  $\delta(U)$  to be the set of all edges that have exactly one endpoint in  $U$ . Consider a Steiner cut  $U \in \mathcal{S}$ . Any feasible solution  $F$  for a given Steiner forest instance must *cross* this cut at least once, i.e.,  $|\delta(U) \cap F| \geq 1$ . This gives rise to the

following integer programming formulation for the Steiner forest problem, to which we refer as the *undirected cut formulation*:

$$\begin{aligned} \text{opt}_{\text{IP}} = \min \quad & \sum_{e \in E} c(e) \cdot x_e & (\text{IP}) \\ \text{s.t.} \quad & \sum_{e \in \delta(U)} x_e \geq 1 & \forall U \in \mathcal{S} \\ & x_e \in \{0, 1\} & \forall e \in E \end{aligned} \quad (2.1)$$

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

$$\begin{aligned} \text{opt}_{\text{D}} = \max \quad & \sum_{U \in \mathcal{S}} y_U & (\text{D}) \\ \text{s.t.} \quad & \sum_{U \in \mathcal{S}: e \in \delta(U)} y_U \leq c(e) & \forall e \in E \\ & y_U \geq 0 & \forall U \in \mathcal{S} \end{aligned} \quad (2.2)$$

Algorithm **AKR** constructs a primal feasible solution for (IP) and a dual feasible solution for (D). The algorithm starts with an infeasible primal solution and reduces the degree of infeasibility as it progresses. At the same time, it greedily creates a feasible dual packing of subsets of large total value. The algorithm raises dual variables of a laminar family of vertex subsets. The final dual solution is maximal in the sense that no single set can be raised without violating a constraint of type (2.2).

We can think of an execution of **AKR** as a process over time. Let  $x^\tau$  and  $y^\tau$ , respectively, be the primal incidence vector and feasible dual solution at time  $\tau$ . We use  $F^\tau$  to denote the forest corresponding to  $x^\tau$ . Initially,  $x_e^0 = 0$  for all  $e \in E$  and  $y_U^0 = 0$  for all  $U \in \mathcal{S}$ . In the following, we say that an edge  $e \in E$  is *tight* if the corresponding constraint (2.2) holds with equality. Assume that the forest  $F^\tau$  at time  $\tau$  is infeasible. We use  $\bar{F}^\tau$  to denote the subgraph of  $G$  that is induced by the tight edges for dual  $y^\tau$ . In the following, we will also use the term *moat* to refer to a connected component  $U$  of  $\bar{F}^\tau$ . A connected component  $U$  of  $\bar{F}^\tau$  is *active* at time  $\tau$  iff it separates at least one terminal pair, i.e., iff  $U \in \mathcal{S}$ . Let  $\mathcal{A}^\tau$  be the set of all connected components of  $\bar{F}^\tau$  that are active at time  $\tau$ .

**AKR** raises the dual variables for all sets in  $\mathcal{A}^\tau$  uniformly at all times  $\tau \geq 0$ . We say that moats  $U_1$  and  $U_2$  *collide* at time  $\tau$  if

1.  $U_1$  and  $U_2$  are moats at some time  $\tau' < \tau$ , and
2.  $\tau$  is the first time during the execution of the algorithm at which forest  $\bar{F}^\tau$  contains a connected component containing the vertices of both  $U_1$  and  $U_2$ .

If this happens, we add the edges on a shortest  $U_1, U_2$ -path to  $F^\tau$  and continue.

The following is the main result of [2]:

**THEOREM 2.1.** *Suppose that algorithm **AKR** outputs a forest  $F$  and a feasible dual solution  $\{y_U\}_{U \in \mathcal{S}}$ . Then*

$$c(F) \leq \left(2 - \frac{1}{k}\right) \cdot \sum_{U \in \mathcal{S}} y_U \leq \left(2 - \frac{1}{k}\right) \cdot \text{opt}_{\text{R}},$$

where  $\text{opt}_{\text{R}}$  is the minimum-cost of a Steiner forest for the given input instance with terminal set  $R$ .

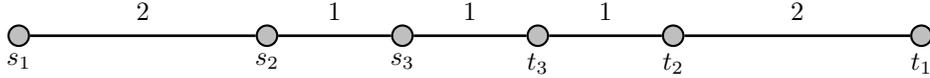


FIG. 3.1. *Distributing the dual growth of each moat  $U$  in AKR uniformly among  $U$ 's active terminals does not lead to a cross-monotonic cost sharing method.*

**3. A Cross-Monotonic Algorithm for the Steiner Forest Game.** In this section we use the ideas presented in the last section to develop a cross-monotonic cost sharing method for the Steiner forest problem.

Consider a subset  $Q$  of players and let  $R$  be the corresponding set of terminal pairs. Running AKR on this instance yields a feasible dual solution for (D) and Theorem 2.1 implies that its value is at least  $\frac{1}{2}\text{opt}_R$  and at most  $\text{opt}_R$ .

Can we distribute the dual computed by AKR as cost shares over the players in  $Q$ ? A natural strategy goes as follows: at any time  $\tau$  during the run of the algorithm, and for any active moat  $U \in \mathcal{A}^\tau$ , distribute the increase in  $y_U$  evenly among the players in  $Q$  whose terminals are separated by  $U$ .

This strategy does not lead to a cross-monotonic cost sharing method as the example instance in Figure 3.1 shows. In the figure, the edges are labeled by their costs. The instance shown has 3 players and the terminal pair of player  $i \in Q = \{1, 2, 3\}$  is  $(s_i, t_i)$ . Distributing the dual growth uniformly as proposed yields a cost share of  $\xi_Q(1) = 3$  for player 1. On the other hand, if player 3 leaves the game and the set of remaining players is  $Q' = \{1, 2\}$ , we have  $\xi_{Q'}(1) = 5/2$ , violating cross-monotonicity.

The example above shows that the *activity time* of a terminal in AKR depends on the presence of other terminals. We now present an adaptation of AKR (subsequently referred to as KLS) that overcomes this problem.

Define the *time of death*  $\mathbf{d}(s, t)$  for each terminal pair  $(s, t) \in R$  as

$$\mathbf{d}(s, t) = \frac{1}{2} \cdot c(s, t), \tag{3.1}$$

where  $c(s, t)$  denotes the cost of the minimum-cost  $s, t$ -path in  $G$ . We assume for ease of presentation that each vertex  $v \in V$  has at most one terminal on it. This assumption is without loss of generality since we can replace each vertex in  $V$  by a sufficient number of copies and link these copies by zero-cost edges. We extend the death time notion to individual terminals and define  $\mathbf{d}(s) = \mathbf{d}(t) = \mathbf{d}(s, t)$  for terminals  $s, t \in R$ .<sup>1</sup>

Recall that AKR raises the dual variables for all sets in  $\mathcal{A}^\tau$ . As a consequence,  $y^\tau$  is a feasible dual solution for (D) at all times  $\tau \geq 0$ . Using the notation introduced in the previous section we obtain KLS by modifying the definition of  $\mathcal{A}^\tau$ . We say that a connected component  $U$  of  $\bar{F}^\tau$  is *active* at time  $\tau$  iff it contains at least one terminal  $v \in U$  with death time at least  $\tau$ , i.e.,  $U$  is active at time  $\tau$  iff there exists  $v \in U$  with  $\mathbf{d}(v) \geq \tau$ . KLS grows all active connected components in  $\mathcal{A}^\tau$  uniformly at all times  $\tau \geq 0$ . Observe that in this way KLS also raises dual variables of connected

<sup>1</sup>Throughout this paper we slightly abuse notation by letting  $R$  refer to both the set of terminal pairs and the set of terminal vertices.

components in  $\mathcal{A}^\tau$  that do *not* correspond to Steiner cuts. In what follows we denote by  $\mathcal{N}$  the set of all *non-Steiner cuts*, i.e.,

$$\mathcal{N} = \{U \subseteq V : U \notin \mathcal{S}, U \cap R \neq \emptyset\}.$$

Furthermore, we let  $\mathcal{U} = \mathcal{S} \cup \mathcal{N}$  be the set of all Steiner and non-Steiner cuts.

What is the intuition behind this? Consider a terminal pair  $(s, t) \in R$  and imagine running the primal-dual Steiner forest algorithm **AKR** on the instance consisting of this terminal pair only. In this case, **AKR** grows two moats corresponding to  $s$  and  $t$ , respectively, at all times  $\tau \leq \mathbf{d}(s, t)$ . At time  $\mathbf{d}(s, t)$  the moats of  $s$  and  $t$  collide and a shortest path connecting the terminals is added. In **KLS** a terminal pair  $(s, t)$  is active for the time it would take  $s$  and  $t$  to connect in the absence of any other terminals. Therefore, the activity time of  $s$  and  $t$  is *independent* of other terminal pairs. This independence is the crucial property leading to cross-monotonicity.

Consider an arbitrary terminal pair  $(s, t) \in R$ . Observe that our choice of the death time  $\mathbf{d}(s, t)$  in (3.1) implies that  $s$  and  $t$  end up in the same connected component of the final forest  $F$ . Therefore **KLS** constructs a feasible solution for the given Steiner forest instance.

For a terminal  $v \in R$  and for  $\tau \leq \mathbf{d}(v)$  we let  $U^\tau(v)$  be the connected component in  $\bar{F}^\tau$  that contains  $v$ . Also let  $a^\tau(v)$  be the number of terminals in  $U^\tau(v)$  whose death time is at least  $\tau$ . We then define the cost share of terminal vertex  $v \in R$  as

$$\xi_R(v) = \int_{\tau=0}^{\mathbf{d}(v)} \frac{1}{a^\tau(v)} d\tau \quad (3.2)$$

and we let  $\xi_R(s, t) = \xi_R(s) + \xi_R(t)$  for all  $(s, t) \in R$ .

The proof of the following theorem is the subject of Section 4.

**THEOREM 3.1.** *The cost shares  $\xi$  computed by **KLS** are cross-monotonic and 2-budget balanced.*

**4. Analysis.** We denote the final forest produced by **KLS**( $R$ ) by  $F$  and use  $\{y_U\}_{U \in \mathcal{U}}$  for the dual computed by our method.

**4.1. Proving Cross-Monotonicity.** In order to prove the cross-monotonicity of **KLS** we consider an arbitrary terminal pair  $(s, t) \in R$  and let  $R_0 = R \setminus \{(s, t)\}$ . In this section we study the effect of the removal of  $(s, t)$  on the cost shares of all other terminal pairs  $(s', t') \in R_0$ .

Let us first introduce some simplifying notation. Assume that **KLS**( $R$ ) terminates at time  $\tau^*$  with forest  $F$ . Similarly, **KLS**( $R_0$ ) finishes at time  $\tau_0^*$  with a forest  $F_0$ . Moreover, for all times  $\tau$  we let  $\mathcal{C}^\tau$  and  $\mathcal{C}_0^\tau$  be the sets of connected components of  $\bar{F}^\tau$  and of  $\bar{F}_0^\tau$ , respectively. The next lemma shows that  $\mathcal{C}_0^\tau$  is a *refinement* of  $\mathcal{C}^\tau$ .

**LEMMA 4.1.** *For all times  $\tau \leq \tau^*$  and for all  $U_0 \in \mathcal{C}_0^\tau$  there must be a set  $U \in \mathcal{C}^\tau$  such that  $U_0 \subseteq U$ .*

*Proof.* The proof is by induction on the time  $\tau$ . It is clear that the claim is true for  $\tau = 0$  since  $\mathcal{C}^0 = \mathcal{C}_0^0 = V$ . Consider a point in time  $0 \leq \tau < \tau^*$  and assume that the claim is true at time  $\tau$ . **KLS**( $R_0$ ) grows active sets in  $\mathcal{C}_0^\tau$  and these are the only sets that can potentially violate the claim at any time  $\tau + \epsilon$  for  $\epsilon > 0$ . Let  $U_0 \in \mathcal{C}_0^\tau$  be an active set at time  $\tau$  in **KLS**( $R_0$ ), i.e., there exists a terminal  $v \in U_0$  with  $\mathbf{d}(v) \geq \tau$ . From the induction hypothesis we know that there is a connected component  $U$  of  $\mathcal{C}^\tau$  that contains  $U_0$ . Then  $U$  must be active in **KLS**( $R$ ) at time  $\tau$  and hence **KLS**( $R$ ) grows  $U$  at time  $\tau$ . The claim follows.  $\square$

Lemma 4.1 immediately implies cross-monotonicity. Let  $\xi(v)$  and  $\xi_0(v)$  be the cost share of terminal  $v \in R_0$  in  $\text{KLS}(R)$  and in  $\text{KLS}(R_0)$ , respectively.

**COROLLARY 4.2.** *Algorithm KLS is cross-monotonic, i.e., for each  $v \in R_0$  we have*

$$\xi_0(v) \geq \xi(v).$$

*Proof.* Let  $U^\tau(v)$  and  $U_0^\tau(v)$  be the moats containing terminal  $v$  at time  $\tau$  in  $\text{KLS}(R)$  and  $\text{KLS}(R_0)$ , respectively. Similarly, let  $a^\tau(v)$  and  $a_0^\tau(v)$  be the number of terminals with death time at least  $\tau$  in  $U^\tau(v)$  and  $U_0^\tau(v)$ . Lemma 4.1 implies that  $U_0^\tau(v) \subseteq U^\tau(v)$  and hence  $a_0^\tau(v) \leq a^\tau(v)$  for all  $\tau \leq \tau^*$  and for all  $v \in R_0$ . Hence we obtain

$$\xi(v) = \int_{\tau=0}^{\text{d}(v)} \frac{1}{a^\tau(v)} d\tau \leq \int_{\tau=0}^{\text{d}(v)} \frac{1}{a_0^\tau(v)} d\tau = \xi_0(v)$$

for all  $v \in R_0$  and the corollary follows.  $\square$

**4.2. Proving Approximate Budget Balance.** We first prove that the cost shares computed by KLS satisfy approximate cost recovery.

**LEMMA 4.3.** *Suppose that algorithm KLS outputs a forest  $F$  and a dual solution  $\{y_U\}_{U \in \mathcal{U}}$ . We then have*

$$c(F) \leq 2 \cdot \sum_{U \in \mathcal{U}} y_U = 2 \cdot \sum_{(s,t) \in R} \xi_R(s,t). \quad (4.1)$$

*Proof.* Using Definition (3.2) it can then be seen that the cost share sum on the right-hand side of (4.1) increases by  $\epsilon$  whenever the total dual value increases by  $\epsilon$  for some  $\epsilon > 0$ . Hence we must have  $\sum_{(s,t) \in R} \xi_R(s,t) = \sum_{U \in \mathcal{U}} y_U$ .

We next prove that  $c(F) \leq 2 \cdot \sum_{U \in \mathcal{U}} y_U$ . We construct a new instance of the Steiner forest problem as follows. For each terminal  $v \in R$ , introduce a new terminal pair  $(\tilde{v}, \tilde{v}')$  and edges  $(v, \tilde{v})$  with  $c(v, \tilde{v}) = 0$  and  $(\tilde{v}, \tilde{v}')$  with  $c(\tilde{v}, \tilde{v}') = 2\text{d}(v)$ . Run the algorithm AKR on the set of terminal pairs  $\tilde{R} \cup R$  where  $\tilde{R} = \{(\tilde{v}, \tilde{v}') : v \in R\}$ . We denote by  $\tilde{\mathcal{S}}$  the set of all Steiner cuts in this new problem, and we use  $\tilde{E}$  for the set of only the new edges. Since the edge  $(v, \tilde{v})$  will go tight at time  $\tau = 0$ , the component containing  $v$  will be active for precisely the same amount of time as in the run of KLS, so we can convert the dual constructed by AKR on the new problem to the dual constructed by KLS, and vice versa. Let  $\{y_U^{\text{AKR}}\}_{U \in \tilde{\mathcal{S}}}$  and  $\{y_U^{\text{KLS}}\}_{U \in \mathcal{U}}$  be the dual solutions computed by AKR and KLS, respectively. Since the new edges do not become tight before the death time of a vertex  $v$ , the solution computed by AKR, when restricted to the original graph, must be equal to the solution computed by KLS. By Theorem 2.1 the solution returned for this new problem is within a factor 2 of the optimal solution for this problem. Using this, we see

$$\begin{aligned} \sum_{e \in E \cup \tilde{E}} c(e)x_e &= \sum_{e \in E} c(e)x_e + \sum_{e \in \tilde{E}} c(e)x_e \\ &\leq 2 \sum_{U \in \tilde{\mathcal{S}}} y_U^{\text{AKR}} = 2 \sum_{U \in \mathcal{S} \cup \mathcal{N}} y_U^{\text{KLS}} + 2 \sum_{v \in R} y_{\{\tilde{v}\}}^{\text{AKR}}. \end{aligned}$$

Furthermore, we know that edge  $(\tilde{v}, \tilde{v}')$  is added exactly at time  $c(\tilde{v}, \tilde{v}')/2$ . Hence

$$\sum_{e \in \tilde{E}} c(e)x_e = 2 \sum_{v \in R} y_{\{\tilde{v}'\}}^{\text{AKR}}.$$

The lemma follows immediately since  $c(F) = \sum_{e \in E} c(e)x_e$ .  
 $\square$

We remark that Lemma 4.3 does *not* imply that the cost  $c(F)$  of forest  $F$  produced by our cost sharing method is at most twice that of an optimal Steiner forest. In fact,  $\{y_U\}_{U \in \mathcal{U}}$  is not a feasible solution for (D) since our algorithm raises duals for active sets that correspond to non-Steiner cuts  $U \in \mathcal{N}$ .<sup>2</sup> Surprisingly however, we can show that the total dual  $\sum_{U \in \mathcal{U}} y_U$  is bounded by the cost  $\text{opt}_R$  of an optimal Steiner forest for the given instance on terminal set  $R$ .

LEMMA 4.4. *Let  $\{y_U\}_{U \in \mathcal{U}}$  be the dual computed by  $\text{KLS}(R)$  and let  $\text{opt}_R$  be the minimum-cost of any feasible Steiner forest for the given instance. We have*

$$\sum_{U \in \mathcal{U}} y_U \leq \text{opt}_R.$$

Lemma 4.3 and 4.4 imply the following corollary on the approximate budget balance of KLS.

COROLLARY 4.5. *Let  $F$  be the Steiner forest computed by  $\text{KLS}(R)$ . We then have*

$$\frac{1}{2} \cdot c(F) \leq \sum_{(s,t) \in R} \xi_R(s,t) \leq \text{opt}_R.$$

It remains to prove Lemma 4.4.

**4.3. A Proof of Lemma 4.4.** Recall the definition of the death time  $\mathbf{d}(s, t)$  of a terminal pair  $(s, t) \in R$ . In the following, let

$$R = \{(s_1, t_1), \dots, (s_k, t_k)\}$$

such that

$$\mathbf{d}(s_1, t_1) \leq \dots \leq \mathbf{d}(s_k, t_k).$$

We define a precedence order  $\prec$  on  $R$  by letting  $(s_i, t_i) \prec (s_j, t_j)$  iff  $i \leq j$ . We extend this order to terminal vertices by letting

$$s_1 \prec t_1 \prec s_2 \prec t_2 \prec \dots \prec s_k \prec t_k.$$

For ease of notation we assume that  $v \prec v$  for all  $v \in R$ .

Let  $U^\tau$  be an active connected component in  $\text{KLS}(R)$  at some time  $\tau \geq 0$ . A terminal vertex  $v \in U^\tau$  is *responsible* for the growth of  $U^\tau$  iff there does not exist a terminal  $u \in U^\tau$  different from  $v$  with  $v \prec u$ . This way, each active moat in KLS has a unique responsible terminal vertex. For a terminal vertex  $v \in R$  and a time  $\tau \geq 0$ , let  $r^\tau(v) = 1$  if  $v$  is responsible at time  $\tau$  and  $r^\tau(v) = 0$  otherwise. We then define the *responsibility time* of a terminal  $v \in R$  as

$$r(v) = \int_{\tau=0}^{\mathbf{d}(v)} r^\tau(v) d\tau. \quad (4.2)$$

<sup>2</sup>Observe, however, that the projection of  $y$  on the set of Steiner cuts is feasible for (D).

As before we let  $U^\tau(v)$  be the connected component of  $\bar{F}^\tau$  containing terminal  $v \in R$ . We can show that a terminal  $v \in R$  is responsible for a unique moat at all times  $0 \leq \tau \leq r(v)$ .

CLAIM 4.6. *Let  $v \in R$  be a terminal and let  $r(v)$  be its responsibility time. Then,  $v$  is responsible for  $U^\tau(v)$  in  $\text{KLS}(R)$  for all  $0 \leq \tau < r(v)$ .*

*Proof.* Assume for the sake of contradiction that there is a point of time  $\tau \in [0, r(v))$  such that  $v$  is not responsible for  $U = U^\tau(v)$ . Since  $U$  is active, we know that there must be a terminal  $u \in U$  that is responsible. We therefore must have  $v \prec u$  and also  $d(v) \leq d(u)$ . Since  $u$  and  $v$  are contained in the same active moat in  $\text{KLS}$  at time  $\tau$ , this means that  $v$  cannot be responsible after time  $\tau$  and hence  $r(v) < \tau$ ; a contradiction.  $\square$

Definition (4.2) also implies that

$$\sum_{U \in \mathcal{U}} y_U = \sum_{v \in R} r(v) \quad (4.3)$$

and hence it suffices to bound the sum on the right-hand side in order to prove Lemma 4.4.

Let  $F^*$  be a minimum-cost Steiner forest for the given instance with terminal set  $R$ . Consider a tree  $T$  in  $F^*$  and suppose that  $T$  connects the terminals  $R(T) = \{v_1, \dots, v_p\}$ . We let  $R^\tau(T)$  be the set of terminal vertices in  $R(T)$  that are responsible at time  $\tau$ , i.e.,

$$R^\tau(T) = \{v \in R(T) : r^\tau(v) = 1\}.$$

The following claim shows that at any time  $\tau$  the moats in

$$U^\tau(T) = \{U^\tau(v) : v \in R^\tau(T)\}$$

are pairwise disjoint.

CLAIM 4.7. *Consider a point of time  $\tau$  and two terminal vertices  $u, v \in R^\tau(T)$ ,  $u \neq v$ . The two moats  $U^\tau(u)$  and  $U^\tau(v)$  must be disjoint.*

*Proof.* Assume for the sake of contradiction that  $U^\tau(u)$  and  $U^\tau(v)$  are not disjoint. Since both  $U^\tau(u)$  and  $U^\tau(v)$  are connected components of  $\bar{F}^\tau$  it must therefore be the case that  $U^\tau(u) = U^\tau(v)$ . Claim 4.6 implies that both  $u$  and  $v$  are responsible for this moat and hence, we must have  $u = v$ . This contradicts our choice of  $u$  and  $v$ .  $\square$

The example in Figure 4.1 shows three terminal pairs  $(s_1, t_1)$ ,  $(s_2, t_2)$  and  $(s_3, t_3)$  that are connected by a tree  $T$  in an optimal solution  $F^*$ . The figure shows a snapshot of algorithm  $\text{KLS}$  at some time  $\tau > 0$ . At this time, five of the terminals are responsible:  $R^\tau(T) = \{s_1, s_2, s_3, t_1, t_3\}$  (assuming that  $t_2 \prec s_1$ ). Consequently, Claim 4.7 implies that the five moats  $U^\tau(s_1), U^\tau(s_2), U^\tau(s_3), U^\tau(t_1)$  and  $U^\tau(t_3)$  are pairwise disjoint. But this means that each of the moats has a non-empty intersection with  $T$  and therefore, we can *charge* their dual growth in the algorithm to the cost  $c(T)$  of tree  $T$ .

Let  $w \in R(T)$  be the terminal vertex with highest responsibility time among all terminals spanned by tree  $T$ . Then, for all terminals  $v_i \in R(T) \setminus \{w\}$  and for all  $0 \leq \tau \leq r(v_i)$ , Claim 4.7 implies that the moats  $U^\tau(w)$  and  $U^\tau(v_i)$  are disjoint. Therefore,

$$\sum_{v_i \in R(T) \setminus \{w\}} r(v_i) \leq c(T).$$

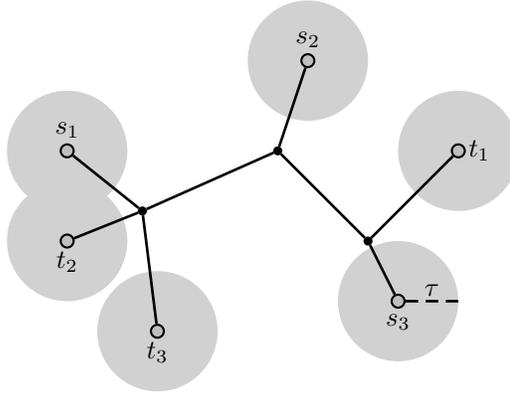


FIG. 4.1. Snapshot of algorithm KLS at some time  $\tau > 0$ .

On the other hand,  $r(w)$  must be at most  $d(w)$  which in turn is at most  $c(T)/2$  and hence, the last inequality implies that

$$\sum_{i=1}^p r(v_i) \leq \frac{3}{2}c(T).$$

In the remainder of this section, we will strengthen the above argument in order to prove Lemma 4.4.

LEMMA 4.8. *If  $\delta(U^\tau(w)) \cap T \neq \emptyset$  for all  $0 \leq \tau < r(w)$  then we must have*

$$\sum_{v \in R(T)} r(v) \leq c(T).$$

*Proof.* Consider any point of time  $\tau \geq 0$  where there are at least two terminals in  $R(T)$  that are responsible, i.e.,  $|R^\tau(T)| > 1$ . By Claim 4.7 we have that the moats in  $U^\tau(T)$  are pairwise disjoint. On the other hand, the vertices in  $R^\tau(T)$  are connected by  $T$  and hence, each of the moats in  $U^\tau(T)$  loads a distinct part of the edges of  $T$ ; see Figure 4.1.

Consider now a time  $\tau$  where  $|R^\tau(T)| = 1$ . It must be the case that  $w$  is the only remaining responsible terminal among the vertices in  $R(T)$ , i.e.,  $R^\tau(T) = \{w\}$ . By assumption,  $U^\tau(w)$  loads at least one edge of  $T$ . This concludes the proof of the lemma.  $\square$

Recall that  $T$  is a tree in an optimal Steiner forest  $F^*$  and that  $T$  spans terminals  $R(T) \subseteq R$ . Furthermore, terminal  $w \in R(T)$  has highest responsibility time among all terminals spanned by  $T$ . In the following, let  $\bar{w}$  be the *mate* of  $w$ , i.e.,  $(w, \bar{w}) \in R$ . From now on we will assume that there is a time  $\tau_0 \in [0, r(w))$  such that  $\delta(U^{\tau_0}(w)) \cap T = \emptyset$  and hence  $T \subseteq E(U^{\tau_0}(w))$ , where  $E(U^{\tau_0}(w))$  denotes the subset of those edges in  $E$  that have both endpoints in  $U^{\tau_0}(w)$ . We also must have  $|R^\tau(T)| = 1$  for all  $\tau \in [\tau_0, r(w))$  since all vertices of  $R(T)$  are in the same connected component of  $\bar{F}^\tau$ . Furthermore, since  $w$  is responsible until time  $r(w)$  we must have  $R^\tau(T) = \{w\}$  for all  $\tau \in [\tau_0, r(w))$  and thus  $u \prec w$  and  $u \prec \bar{w}$  for all  $u \in R(T) \setminus \{w, \bar{w}\}$ .

Let  $P_{w\bar{w}}$  be the unique  $w, \bar{w}$ -path in  $T$ . We define  $I^\tau(T)$  as the set of responsible terminal pairs in  $R^\tau(T) \setminus \{w, \bar{w}\}$  that inflict dual load on path  $P_{w\bar{w}}$  in KLS( $R$ ) at time

$\tau$ , i.e.,

$$I^\tau(T) = \{v \in R^\tau(T) \setminus \{w, \bar{w}\} : \delta(U^\tau(v)) \cap P_{w\bar{w}} \neq \emptyset\}.$$

CLAIM 4.9. *Consider a point in time  $\tau$  and a terminal  $v \in I^\tau(T)$ . Then  $U^\tau(v)$  contains neither  $w$  nor  $\bar{w}$ .*

*Proof.* By definition of  $I^\tau(T)$ , we know that  $v \notin \{w, \bar{w}\}$ . We also know that  $v \prec w$  and  $v \prec \bar{w}$ . The claim follows as  $v$  is responsible for  $U^\tau(v)$  and hence  $\{w, \bar{w}\} \cap U^\tau(v) = \emptyset$ .  $\square$

For a time  $\tau$  and a vertex  $v \in I^\tau(T)$ , let  $p_{w\bar{w}}^\tau(v)$  be the number of intersections of  $P_{w\bar{w}}$  and  $U^\tau(v)$  at time  $\tau$ :

$$p_{w\bar{w}}^\tau(v) = |\delta(U^\tau(v)) \cap P_{w\bar{w}}|. \quad (4.4)$$

We use  $\mathbf{sl}_{w\bar{w}}$  to denote the cost of that part of  $P_{w\bar{w}}$  that does not feel any dual load from any of the terminals in  $R(T)$ . Let  $l_w$  and  $l_{\bar{w}}$  be the total load on  $P_{w\bar{w}}$  coming from terminals  $w$  and  $\bar{w}$ , respectively. We can then express the cost of  $P_{w\bar{w}}$  as

$$c(P_{w\bar{w}}) = l_w + l_{\bar{w}} + \mathbf{sl}_{w\bar{w}} + \int_0^{\tau_0} \sum_{v \in I^\tau(T)} p_{w\bar{w}}^\tau(v) d\tau. \quad (4.5)$$

We obtain the following lemma.

LEMMA 4.10. *If there is a  $\tau_0 \in [0, r(w))$  with  $\delta(U^{\tau_0}(w)) \cap T = \emptyset$  then we must have*

$$\sum_{v \in R(T)} r(v) \leq c(T).$$

*Proof.* Similar to the proof of Lemma 4.8, consider a time  $\tau < r(w)$  where  $R^\tau(T)$  contains more than one terminal. The corresponding moats in  $U^\tau(T)$  are pairwise disjoint by Claim 4.7 and the vertices in  $R^\tau(T)$  are connected by  $T$ . Hence, each of the moats in  $U^\tau(T)$  loads a distinct part of  $T$ . Moreover, using the definition of  $p_{w\bar{w}}^\tau(v)$  in (4.4), for all  $\tau \in [0, \tau_0)$  and  $v \in I^\tau(T)$  moat  $U^\tau(v)$  loads at least  $p_{w\bar{w}}^\tau(v)$  edges of  $T$ .

Recall that  $\mathbf{sl}_{w\bar{w}}$  is the cost of the segments of  $P_{w\bar{w}}$  that do not feel any load from terminals in  $R(T)$ . Furthermore,  $w$  loads edges of  $T$  until time  $\tau_0$  and hence we must have

$$c(T) \geq \tau_0 + \mathbf{sl}_{w\bar{w}} + \int_0^{\tau_0} \sum_{v \in I^\tau(T)} (p_{w\bar{w}}^\tau(v) - 1) d\tau + \sum_{v \in R(T) \setminus \{w\}} r(v). \quad (4.6)$$

Observe that for all  $\tau \in [0, \tau_0)$  and  $v \in I^\tau(T)$ , we account a total contribution of  $p_{w\bar{w}}^\tau(v)$ :  $p_{w\bar{w}}^\tau(v) - 1$  in the first and 1 in the second sum, respectively.

The death time of vertex  $w$  is at most half of the cost of  $P_{w\bar{w}}$ . Using (4.5) we therefore obtain

$$\begin{aligned} r(w) &\leq \frac{l_w + l_{\bar{w}}}{2} + \frac{\mathbf{sl}_{w\bar{w}}}{2} + \frac{1}{2} \cdot \int_0^{\tau_0} \sum_{v \in I^\tau(T)} p_{w\bar{w}}^\tau(v) d\tau \\ &\leq \tau_0 + \mathbf{sl}_{w\bar{w}} + \int_0^{\tau_0} \sum_{v \in I^\tau(T)} (p_{w\bar{w}}^\tau(v) - 1) d\tau, \end{aligned} \quad (4.7)$$

where the second inequality uses the fact that  $\max\{l_w, l_{\bar{w}}\} \leq \tau_0$  and that by Claim 4.9,  $p_{w\bar{w}}^\tau(v) \geq 2$  for all  $v \in I^\tau(T)$ . Combining (4.6) and (4.7) yields the lemma.  $\square$

We can now sum over all trees  $T$  in the forest  $F^*$ . Lemmas 4.8 and 4.10 together with (4.3) imply that

$$\sum_{U \in \mathcal{U}} y_U = \sum_{v \in R} r(v) = \sum_{T \in F^*} \sum_{v \in R(T)} r(v) \leq \sum_{T \in F^*} c(T) = \text{opt}_R.$$

This finishes the proof of Lemma 4.4.

**5. Lifted-Cut LP Relaxation for the Steiner Forest Problem.** Recall that without loss of generality we let

$$R = \{(s_1, t_1), \dots, (s_k, t_k)\}$$

such that

$$d(s_1, t_1) \leq \dots \leq d(s_k, t_k).$$

As before we define a precedence order  $\prec$  on  $R$  by letting  $(s_i, t_i) \prec (s_j, t_j)$  iff  $i \leq j$  and we extend this order to terminal vertices by letting

$$s_1 \prec t_1 \prec s_2 \prec t_2 \prec \dots \prec s_k \prec t_k. \quad (5.1)$$

We assume that  $v \prec v$  for all  $v \in R$ .

Let  $R(U)$  be the set of terminal pairs in  $R$  that are separated by a Steiner cut  $U \in \mathcal{S}$ , i.e.,  $R(U) = \{(s, t) \in R : |\{s, t\} \cap U| = 1\}$ . Consider a terminal  $v$  and let  $\bar{v}$  be  $v$ 's mate in the Steiner forest instance, i.e.,  $(v, \bar{v}) \in R$ . We let  $\mathcal{S}_v \subseteq \mathcal{S}$  be the set of Steiner cuts that separate  $v$  and  $\bar{v}$  and for which  $(v, \bar{v})$  is the highest ranked such terminal pair:

$$\mathcal{S}_v = \{U \in \mathcal{S} : v \in R(U), u \prec v \text{ for all } u \in R(U)\}. \quad (5.2)$$

We also let  $\mathcal{N}_v \subseteq \mathcal{N}$  be the set of all non-Steiner cuts containing  $v$  and  $\bar{v}$  where  $(v, \bar{v})$  is the terminal pair of highest rank:

$$\mathcal{N}_v = \{U \in \mathcal{N} : \{v, \bar{v}\} \subseteq U \cap R, (u, \bar{u}) \prec (v, \bar{v}) \text{ for all } (u, \bar{u}) \in U \cap R\}.$$

Recall that we define  $\mathcal{U} = \mathcal{S} \cup \mathcal{N}$  as the set of all Steiner and non-Steiner cuts. We then say that a terminal  $v \in R$  is *responsible* for a cut  $U \in \mathcal{U}$  if  $U \in \mathcal{S}_v \cup \mathcal{N}_v$ . Observe that for a non-Steiner cut  $U \in \mathcal{N}$  two terminals are responsible. Also note that the responsibility notion introduced here differs from the one that was used in Section 4 in that a terminal can only be responsible for a Steiner cut if the cut separates it from its mate.

The dual of the lifted-cut relaxation for the Steiner forest problem is as follows:

$$\text{opt}_{\text{LC-D}} = \max \sum_{U \in \mathcal{U}} y_U \quad (\text{LC-D})$$

$$\text{s.t.} \quad \sum_{U \in \mathcal{U}: e \in \delta(U)} y_U \leq c(e) \quad \forall e \in E \quad (5.3)$$

$$\sum_{U \in \mathcal{S}_v} y_U + \sum_{U \in \mathcal{N}_v} y_U \leq d(v) \quad \forall v \in R \quad (5.4)$$

$$y_U \geq 0 \quad \forall U \in \mathcal{U}.$$

Notice that a feasible solution to (LC-D) may assign positive values to non-Steiner cuts  $U \in \mathcal{N}$ . The constraints of type (5.4) are necessary as the objective function value of (LC-D) would be unbounded in their absence.

The linear programming dual of (LC-D) has variables  $x_e$  for every edge  $e \in E$  and variables  $x_v$  for every terminal  $v \in R$ :

$$\text{opt}_{\text{LC-P}} = \min \sum_{e \in E} c(e) \cdot x_e + \sum_{v \in R} \mathbf{d}(v) \cdot x_v \quad (\text{LC-P})$$

$$\text{s.t.} \quad \sum_{e \in \delta(U)} x_e + x_v \geq 1 \quad \forall U \in \mathcal{S}_v, \forall v \in R \quad (5.5)$$

$$\sum_{e \in \delta(U)} x_e + x_v + x_{\bar{v}} \geq 1 \quad \forall U \in \mathcal{N}_v, \forall v \in R \quad (5.6)$$

$$x_e, x_v \geq 0 \quad \forall e \in E, \forall v \in R.$$

LEMMA 5.1. *Let  $\{x_e, x_v\}_{e \in E, v \in R}$  be an integral solution that is feasible for (LC-P). Then there is a feasible Steiner forest of cost at most*

$$\sum_{e \in E} c(e) \cdot x_e + \sum_{v \in R} \mathbf{d}(v) \cdot x_v.$$

*Proof.* Given  $\{x_e, x_v\}_{e \in E, v \in R}$ , define  $F = \{e \in E : x_e = 1\}$ . The total cost  $c(F)$  of  $F$  is  $\sum_{e \in E} c(e) \cdot x_e$ .  $F$  is not necessarily a feasible Steiner forest since there might exist a Steiner cut  $U \in \mathcal{S}$  with no crossing edge, i.e.,  $\delta(U) \cap F = \emptyset$ . Let  $U \in \mathcal{S}_v$  be such a set and let  $\bar{v}$  be the mate of  $v$ . Constraint (5.5) for  $U$  and  $v$  implies that  $x_v = 1$  in this case. Next consider the complement  $\bar{U} = V \setminus U$ . It can be seen that  $\bar{v}$  is responsible for  $\bar{U}$  and hence,  $\bar{U} \in \mathcal{S}_{\bar{v}}$ . As no edge crosses  $\bar{U}$ , constraint (5.5) for  $\bar{U}$  and  $\bar{v}$  implies that  $x_{\bar{v}} = 1$ . Therefore, we can add all edges along the shortest  $v, \bar{v}$ -path to  $F$  at a cost of at most  $2\mathbf{d}(v, \bar{v})$ . Observe that this addition is sufficient to satisfy all Steiner cuts in  $\mathcal{S}_v$ , so we only need to add this path once for  $v$  and  $\bar{v}$ . We can therefore repeat this procedure for all remaining terminals  $v \in R$  for which there exists a Steiner cut  $U \in \mathcal{S}_v$  that is not crossed by  $F$ . The total cost in this solution incurred by the additional paths is not more than  $\sum_{v \in R} \mathbf{d}(v) \cdot x_v$ , which completes the proof.  $\square$

In the remainder of this section we prove the following theorem:

THEOREM 5.2. *The objective value of an optimal solution to the lifted-cut relaxation (LC-P) is at most the cost of any feasible Steiner forest for the given instance. The dual linear program (LC-D) is stronger than the well-known undirected-cut relaxation for the Steiner forest problem. Moreover, the dual solution computed by KLS is feasible for (LC-D). There exist instances for which the IP/LP gap is about 2.*

The following lemma relates the cost of any feasible solution for the given Steiner forest instance to the objective function value of an optimal solution for (LC-P).

LEMMA 5.3. *Let  $F$  be a feasible solution for the underlying Steiner forest instance. We can then construct a half-integral solution  $\{x_e, x_v\}_{e \in E, v \in R}$  that is feasible for (LC-P) and satisfies:*

$$\sum_{e \in E} c(e) \cdot x_e + \sum_{v \in R} \mathbf{d}(v) \cdot x_v \leq c(F).$$

*In particular, this implies that  $\text{opt}_{\text{LC-P}} \leq \text{opt}_R$ .*

*Proof.* Let  $T$  be a tree in  $F$ . We use  $E(T)$  and  $V(T)$  to refer to the edges and vertices of  $T$ , respectively. We construct a solution  $\{x_e, x_v\}_{e \in E, v \in R}$  that is feasible for (LC-P) and show that for each tree  $T \in F$

$$\sum_{e \in E(T)} c(e) \cdot x_e + \sum_{v \in R \cap V(T)} d(v) \cdot x_v \leq c(T).$$

The lemma then follows by summing over all trees in  $F$ .

Consider a tree  $T \in F$ . Let  $(w, \bar{w})$  be the terminal pair such that  $w$  and  $\bar{w}$  are responsible for the non-Steiner cut  $V(T)$ . Moreover, let  $P_{w\bar{w}}$  denote the unique  $w, \bar{w}$ -path in  $T$ . We set  $x_e = 1/2$  for each edge  $e \in E(P_{w\bar{w}})$  and  $x_e = 1$  for each edge  $e \in E(T) \setminus E(P_{w\bar{w}})$ . Moreover, we assign  $x_w = x_{\bar{w}} = 1/2$  and  $x_v = 0$  for all terminals  $v \in (R \cap V(T)) \setminus \{w, \bar{w}\}$ . By definition (3.1) of death time,  $d(w, \bar{w}) \leq c(P_{w\bar{w}})/2$ . Thus, the objective value for  $x$  on  $T$  is

$$\sum_{e \in E(T)} c(e) \cdot x_e + \sum_{v \in R \cap V(T)} d(v) \cdot x_v = c(T) - \frac{c(P_{w\bar{w}})}{2} + d(w, \bar{w}) \leq c(T).$$

It remains to be shown that  $x$  is feasible for (LC-P). We show for each tree  $T$  in  $F$  and for all  $v \in R \cap V(T)$  that  $x$  satisfies the cut-requirements of constraints (5.5) and (5.6) for sets  $U \in \mathcal{S}_v \cup \mathcal{N}_v$ .

Consider a cut  $U \in \mathcal{S}_v$  for some  $v \in R \cap V(T)$ . If  $v \in \{w, \bar{w}\}$ , constraint (5.5) holds since  $U$  intersects  $P_{w\bar{w}}$  and  $x_v = 1/2$ . Now let  $v \notin \{w, \bar{w}\}$ . As  $U \in \mathcal{S}_v$  and  $v \prec w$ , by assumption, it follows that either  $\{w, \bar{w}\} \subseteq U$  or  $\{w, \bar{w}\} \cap U = \emptyset$ . We also have  $\bar{v} \notin U$ . As  $T$  connects  $v$  and  $\bar{v}$ , it can be seen that  $U$  either intersects at least one edge  $e$  of  $T$  that is not on  $P_{w\bar{w}}$  (and hence  $x_e = 1$ ) or it intersects at least two edges  $e_1$  and  $e_2$  on  $P_{w\bar{w}}$  (and therefore  $x_{e_1} = x_{e_2} = 1/2$ ). Thus, constraint (5.5) holds in this case as well.

Next consider a non-Steiner cut  $U \in \mathcal{N}_v$  for terminal  $v \in R \cap V(T)$ . If  $v \notin \{w, \bar{w}\}$  then  $\{w, \bar{w}\} \cap U = \emptyset$  and  $U$  crosses at least one edge of  $T$  that is not on  $P_{w\bar{w}}$  or at least two edges of  $P_{w\bar{w}}$ . Hence constraint (5.6) holds. Otherwise,  $U$  may cross no edge of  $T$  but  $x_w + x_{\bar{w}} = 1$  and thus (5.6) is satisfied.  $\square$

Running algorithm KLS on terminal set  $R$  yields a cost share  $\xi_R(s, t)$  for all  $(s, t) \in R$ . It also returns a dual solution  $\{y_U\}_{U \in \mathcal{U}}$  such that  $\sum_{(s,t) \in R} \xi_R(s, t) = \sum_{U \in \mathcal{U}} y_U$ . It is easy to verify that  $y$  is feasible for (LC-D). Lemma 5.3 therefore yields an alternate proof of the competitiveness of KLS:

COROLLARY 5.4.  $\xi$  satisfies competitiveness, i.e.,

$$\sum_{(s,t) \in R} \xi_R(s, t) = \sum_{U \in \mathcal{U}} y_U \leq \text{opt}_{LC-D} = \text{opt}_{LC-P} \leq \text{opt}_R.$$

The next lemma shows that (LC-D) is at least as strong as the standard LP dual (D).

LEMMA 5.5. Let  $\{y_U\}_{U \in \mathcal{S}}$  be a feasible dual solution for (D). Then there is a feasible dual solution  $\{y'_U\}_{U \in \mathcal{U}}$  for (LC-D) with

$$\sum_{U \in \mathcal{S}} y_U \leq \sum_{U \in \mathcal{U}} y'_U.$$

This implies that  $\text{opt}_D \leq \text{opt}_{LC-D}$ .

*Proof.* Let  $y$  be a feasible solution for (D). The sets  $\mathcal{S}_v$  for terminals  $v \in R$  form a partition of  $\mathcal{S}$ :  $\mathcal{S} = \bigcup_{v \in R} \mathcal{S}_v$ . We define a candidate dual solution  $y'$  for (LC-D) as follows: for a set  $U \in \mathcal{S}$ , let  $\bar{U} \in \mathcal{S}$  be its complement and define

$$y'_U = y'_{\bar{U}} = \frac{y_U + y_{\bar{U}}}{2}.$$

Let  $y'_U = 0$  for all non-Steiner cuts  $U \in \mathcal{N}$ .

We claim that  $y'$  satisfies all constraints of type (5.3). To see this, consider an edge  $e \in E$  and observe that

$$\sum_{U \in \mathcal{S}: e \in \delta(U)} y'_U = \sum_{U \in \mathcal{S}: e \in \delta(U)} \frac{y_U + y_{\bar{U}}}{2} = \sum_{U \in \mathcal{S}: e \in \delta(U)} y_U$$

where the last equality uses the fact that  $U$  is a Steiner cut if and only if its complement is. The dual feasibility of  $y$  for (D) shows that  $y'$  satisfies (5.3).

We will now show that  $y'$  also satisfies all constraints of type (5.4). Assume for the sake of contradiction that  $y'$  violates constraint (5.4) for some terminal  $v \in R$ . We then have

$$\sum_{U \in \mathcal{S}_v} y'_U + \sum_{U \in \mathcal{N}_v} y'_U = \sum_{U \in \mathcal{S}_v} y'_U > d(v) = c(P_{v\bar{v}})/2. \quad (5.7)$$

where  $c(P_{v\bar{v}})$  is the cost of a minimum-cost  $v, \bar{v}$ -path in  $G$ .

Consider a Steiner cut  $U \in \mathcal{S}$  and observe that  $U$  and its complement  $\bar{U}$  separate the same set of terminal pairs. Therefore,  $U \in \mathcal{S}_v$  iff  $\bar{U} \in \mathcal{S}_{\bar{v}}$  for a terminal pair  $(v, \bar{v}) \in R$  and thus,

$$\sum_{U \in \mathcal{S}_v} y'_U = \sum_{U \in \mathcal{S}_v} \frac{y_U + y_{\bar{U}}}{2} = \sum_{U \in \mathcal{S}_{\bar{v}}} y'_U. \quad (5.8)$$

Together with (5.7), this implies that

$$\sum_{U \in \mathcal{S}_v} y'_U + \sum_{U \in \mathcal{S}_{\bar{v}}} y'_U > c(P_{v\bar{v}}).$$

On the other hand, adding the constraints of type (2.2) for all edges  $e \in E(P_{v\bar{v}})$  yields

$$\sum_{U \in \mathcal{S}_v} y'_U + \sum_{U \in \mathcal{S}_{\bar{v}}} y'_U \leq \sum_{U \in \mathcal{S}} |\delta(U) \cap P_{v\bar{v}}| \cdot y_U = \sum_{e \in E(P_{v\bar{v}})} \sum_{U \in \mathcal{S}: e \in \delta(U)} y_U \leq c(P_{v\bar{v}}),$$

and this is a contradiction.  $\square$

The dual of the lifted-cut relaxation is stronger than the standard LP dual (D).

LEMMA 5.6. *There exist instances for which  $\text{opt}_D < \text{opt}_{LC-D}$ .*

*Proof.* Consider a cycle of  $2n$  vertices with unit edge costs. Let  $V = \{v_1, \dots, v_{2n}\}$  and define  $R = \{(v_1, v_j)\}_{2 \leq j \leq 2n}$ . The cost of an optimal solution is  $\text{opt}_R = 2n - 1$ .

We define a dual solution as follows:  $y_{\{v\}} = 1/2$  for each  $v \in V$  and  $y_U = 0$  for all other sets  $U \in \mathcal{S}$ . Clearly,  $\{y_U\}_{U \in \mathcal{S}}$  is a feasible solution to (D). It can easily be verified that this is an optimal solution for (D): If we set  $x_e = 1/2$  for each edge  $e$  of the cycle, we obtain a feasible solution for the LP relaxation (LP) having the same objective function value. Thus,  $\text{opt}_D = n$ .

For (LC-D) on the other hand, we can define a dual solution  $y'_{\{v\}} = 1/2$  for each  $v \in V$ ,  $y'_V = n/2 - 1/2$ , and  $y'_U = 0$  for all other sets  $U \in \mathcal{U}$ . It is easy to verify that  $y'$  is a feasible solution for (LC-D). We conclude that

$$\text{opt}_{\text{LC-D}} \geq \sum_{U \in \mathcal{U}} y'_U = \frac{3n}{2} - \frac{1}{2}.$$

The latter term is strictly larger than  $n$  if  $n > 1$ .  $\square$

Unfortunately, as with the undirected cut formulation for the Steiner forest problem, the IP/LP gap of the lifted-cut relaxation is close to 2 for certain instances.

**LEMMA 5.7.** *There exist instances for which  $\text{opt}_R / \text{opt}_{\text{LC-P}} = 2 - 2/(k + 1)$ , where  $k$  is the number of terminal pairs.*

*Proof.* Consider a clique  $K_n$  with vertices  $V = \{v_1, v_2, \dots, v_n\}$  and unit edge costs. Define  $R = \{(v_1, v_j)\}_{2 \leq j \leq n}$ . Without loss of generality, let  $(w, \bar{w}) = (v_1, v_2)$  be the highest ranked terminal pair among all terminal pairs in  $R$ .

Consider path  $P = (v_2, v_3, \dots, v_n, v_1)$  spanning all vertices of  $K_n$ . The following is a feasible solution for (LC-P): set  $x_w = x_{\bar{w}} = 1/2$  and  $x_v = 0$  for all  $v \in V \setminus \{w, \bar{w}\}$ , set  $x_e = 1/2$  for all edges  $e \in E(P)$  and  $x_e = 0$  for all edges  $e \notin E(P)$ . This solution satisfies constraints (5.5) and (5.6). The objective function value for  $x$  is  $n/2$ . Next consider the following dual solution. Let  $y_{\{v\}} = 1/2$  for all  $v \in V$  and  $y_U = 0$  for all other  $U \in \mathcal{U}$ . Then  $y$  satisfies constraints (5.3) and (5.4). The objective value of  $y$  is  $n/2$  and thus  $x$  and  $y$  are optimal solutions to (LC-P) and (LC-D), respectively.

Clearly, the optimal solution  $\text{opt}_R$  has cost  $n - 1$ . The ratio between  $\text{opt}_R$  and  $\text{opt}_{\text{LC-D}}$  is  $2 - 2/n$ . Since  $k = n - 1$  the lemma follows.  $\square$

**6. Algorithmic Consequences of the Lifted-Cut Relaxation.** In this section we show that, for some instances of the Steiner forest problem, we can use the additional strength of the lifted-cut relaxation in order to prove that algorithm AKR returns a Steiner forest of cost strictly less than  $(2 - 1/k)\text{opt}_R$ .

Consider an instance of the Steiner forest problem with terminal set  $R$ . Assume that algorithm AKR, when executed on this instance, finishes at time  $\tau^* \geq 0$  with forest  $F$  and feasible dual solution  $\{y_U\}_{U \in \mathcal{S}}$ . Let  $U_1, \dots, U_p$  be the connected components of  $\bar{F}^{\tau^*}$  and define  $R_i \subseteq R$  to be the set of terminal pairs contained in  $U_i$ , for all  $1 \leq i \leq p$ . Further let  $(s_i, t_i)$  be the terminal pair in  $R_i$  of highest rank according to the precedence order  $\prec$  defined in (5.1), i.e.,

$$(s, t) \prec (s_i, t_i)$$

for all  $(s, t) \in R_i$  and for all  $1 \leq i \leq p$ . For  $1 \leq i \leq p$ , we now define the combined *slack*  $\text{sl}_i$  of the constraints (5.4) for the terminal vertices  $s_i$  and  $t_i$  with respect to dual solution  $y$ :

$$\text{sl}_i = 2\text{d}(s_i, t_i) - \sum_{U \in \mathcal{S}_{s_i} \cup \mathcal{S}_{t_i}} y_U.$$

Let  $\text{sl}_R = \max_{1 \leq i \leq p} \text{sl}_i$  be the slack of the given instance of the Steiner forest problem.

**THEOREM 6.1.** *The forest  $F$  returned by AKR for an instance of the Steiner forest problem with terminal pairs  $R$  has cost at most*

$$\left(2 - \frac{1}{k}\right) \left(\frac{Y}{Y + \text{sl}_R/2}\right) \text{opt}_R$$

where  $Y$  is the objective function value of the dual computed by AKR.

*Proof.* From the proof of Lemma 5.5 (see (5.8)) we know that we may assume w.l.o.g. that  $y$  is symmetric, i.e., we may assume that

$$\sum_{U \in \mathcal{S}_s} y_U = \sum_{U \in \mathcal{S}_t} y_U$$

for all  $(s, t) \in R$ .

We observe that the proof of Lemma 5.5 works for any fixed precedence order  $\prec$  on  $R$ ; in particular, at no point in the proof of this lemma do we use the fact that  $(s, t) \prec (s', t')$  implies  $d(s, t) \leq d(s', t')$ .

Choose  $1 \leq q \leq p$  such that  $\mathbf{sl}_q = \max_{1 \leq i \leq p} \mathbf{sl}_i$ . We will now define an alternative order  $\prec'$  on  $R$  in which the terminal pairs in  $R_q$  have highest rank. The order on terminal pairs in  $R \setminus R_q$  and the order within  $R_q$  is that induced by  $\prec$ . Formally, consider two terminal pairs  $(s, t), (s', t') \in R$ . We let  $(s, t) \prec' (s', t')$  iff

- $(s, t) \prec (s', t')$  and either  $\{(s, t), (s', t')\} \subseteq R \setminus R_q$  or  $\{(s, t), (s', t')\} \subseteq R_q$ , or
- $(s, t) \in R \setminus R_q$  and  $(s', t') \in R_q$ .

Similar to the definition of  $\mathcal{S}_v$  in (5.2) we let  $\mathcal{S}'_v$  be the set of Steiner cuts that separate  $v$  and its mate  $\bar{v}$  and for which  $(v, \bar{v})$  has highest  $\prec'$ -rank among all such terminal pairs. The definition of  $\prec'$  implies that  $(s, t) \prec (s', t')$  if and only if  $(s, t) \prec' (s', t')$  for all  $\{(s, t), (s', t')\} \subseteq R_i$  for all  $1 \leq i \leq p$ . Therefore, we also must have

$$\sum_{U \in \mathcal{S}_v} y_U = \sum_{U \in \mathcal{S}'_v} y_U$$

for all terminals  $v \in R$ . Specifically, this and the symmetry of  $y$  imply that

$$\begin{aligned} \sum_{U \in \mathcal{S}'_{s_q}} y_U + \sum_{U \in \mathcal{N}_{s_q}} y_U &\leq d(s_q, t_q) - \frac{\mathbf{sl}_q}{2} \\ \sum_{U \in \mathcal{S}'_{t_q}} y_U + \sum_{U \in \mathcal{N}_{t_q}} y_U &\leq d(s_q, t_q) - \frac{\mathbf{sl}_q}{2} \end{aligned}$$

where  $\mathcal{N}_{s_q} = \mathcal{N}_{t_q}$  and  $y_U = 0$  for all  $U \in \mathcal{N}_{s_q}$ . Finally notice that  $V \in \mathcal{N}_{s_q}$  as  $(s_q, t_q)$  is the highest ranked terminal pair in  $R$  under  $\prec'$ . We now let  $y'_U = y_U$  for all Steiner cuts  $U \in \mathcal{S}$  and we define  $y'_V = \mathbf{sl}_q/2$ . It is not hard to see that  $y'$  is feasible for the lifted-cut dual (LC-D) for the given instance of the Steiner forest problem.

In the following, we use  $Y$  as a short for  $\sum_{U \in \mathcal{S}} y_U$ . We then have

$$\left(1 + \frac{\mathbf{sl}_R}{2Y}\right) \cdot Y = y'_V + \sum_{U \in \mathcal{S}} y'_U \leq \mathbf{opt}_R$$

and this together with Theorem 2.1 implies

$$c(F) \leq \left(2 - \frac{1}{k}\right) \cdot Y \leq \left(2 - \frac{1}{k}\right) \left(\frac{Y}{Y + \mathbf{sl}_R/2}\right) \mathbf{opt}_R.$$

□

Suppose now that we are given an instance of the Steiner tree problem with terminal set  $R$  and root vertex  $r$ . Let  $\Delta_R$  be the maximum distance among any two terminals in  $R \cup \{r\}$ . We call  $\Delta_R$  the *diameter* of the given instance. Let  $r'$  be an

arbitrary terminal in  $R \cup \{r\}$  such that there exists a terminal  $u \in R \cup \{r\}$  with  $c(r', u) = \Delta_R$ . The Steiner forest instance with terminal pairs

$$R' = \{(u, r') : u \in R \cup \{r\}\}$$

is easily seen to be equivalent to the given instance of the Steiner tree problem. Suppose again, that AKR finishes at time  $\tau^*$ , when run on this instance. It is not hard to convince oneself that the slack  $\text{sl}_{R'}$  of this instance is

$$\text{sl}_{R'} = \Delta_R - \tau^*.$$

We therefore obtain the following corollary of Theorem 6.1:

**COROLLARY 6.2.** *Given an instance of the Steiner tree problem with terminal set  $R$ , AKR returns a tree  $T$  of cost at most*

$$\left(2 - \frac{1}{|R|}\right) \left(\frac{Y}{Y + (\Delta_R - \tau^*)/2}\right) \text{opt}_R$$

where  $Y$  is the objective function value of the dual computed by AKR.

**7. A Lower Bound for the Steiner Tree Game.** We next prove that no cross-monotonic cost sharing method for the Steiner tree game can achieve a budget balance factor better than 2.

**THEOREM 7.1.** *There is no  $(2 - \epsilon)$ -budget balanced, cross-monotonic cost sharing method for the Steiner tree game for any  $\epsilon > 0$ .*

The tools used in this section are adaptations of those used in [9]. In particular we consider any given cross-monotonic cost sharing method  $\xi$  for the Steiner tree game and show that there is an instance of the game where the sum of the cost shares of all players is considerably smaller than the cost of an optimal solution. Instead of using a probabilistic argument similar to the one described in [9], we use a more direct (but ultimately equivalent) proof based on convex combinations.

The family of instances used in our proof resembles the one used for the facility location lower bound in [9]. We construct an undirected graph  $G = (V, E)$ . First we describe the vertex set. There are  $k$  pairwise disjoint sets  $A_i$ ,  $i = 1, \dots, k$ , each of which contains  $m$  vertices. Every one of these vertices corresponds to a player who wants to connect this vertex with a root vertex (which is different from the vertices in the  $A_i$ ). The set of all players that have a vertex associated with them in  $A_i$  is denoted by  $\mathcal{A}_i$ . The set of all players is  $\mathcal{R} = \bigcup_{i=1}^k \mathcal{A}_i$ .

Let  $\mathcal{B}$  be the collection of all sets with exactly one element from each of the  $A_i$ , i.e.,

$$\mathcal{B} = \{\{a_1, \dots, a_k\} : a_i \in A_i, i = 1, \dots, k\}.$$

For each set  $B \in \mathcal{B}$ , we introduce a unique vertex  $f_B$  and edges  $(b, f_B)$  of cost 1 for all vertices  $b \in B$ . The distance to the vertices not in  $B$  is, by the triangle inequality, equal to 3. Finally there is, for each  $B$ , an edge  $(f_B, r)$  of cost 3. See Figure 7.1.

The following lemma argues that we may assume that  $\xi$  is *symmetric*, i.e. that it does not differentiate between players from the same set  $\mathcal{A}_i$ .

**LEMMA 7.2.** *Suppose that there is an  $\alpha$ -budget balanced cost sharing method for the Steiner tree game. Then there is also an  $\alpha$ -budget balanced cost sharing method that satisfies, for every subset  $\mathcal{Q} \subseteq \mathcal{R}$  of players,*

$$\xi_{\mathcal{Q}}(c) = \xi_{\mathcal{Q}}(d)$$

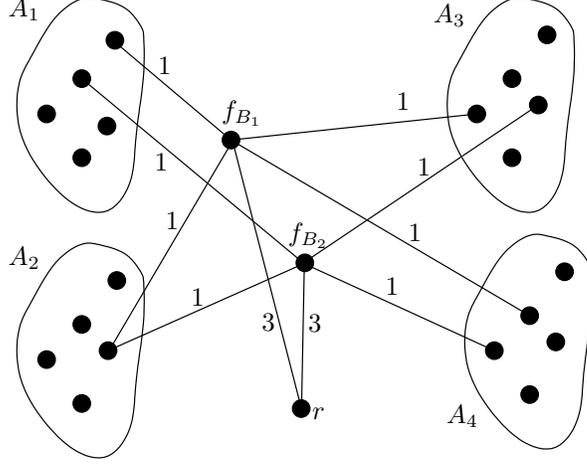


FIG. 7.1. Example of  $G$  in which  $k = 4$ ,  $m = 5$ , and only 2 of the  $f_B$  are drawn.

for all  $c, d \in \mathcal{Q} \cap \mathcal{A}_i$  and for all  $1 \leq i \leq k$ . Moreover, for all  $c \in \mathcal{Q} \cap \mathcal{A}_i$  and for all  $d \in \mathcal{A}_i \setminus \mathcal{Q}$ ,

$$\xi_{\mathcal{Q}}(c) = \xi_{(\mathcal{Q} \setminus \{c\}) \cup \{d\}}(d).$$

*Proof.* Let  $\tilde{\xi}$  be an  $\alpha$ -budget balanced cost sharing method for the Steiner tree game. Let  $\Pi$  be the set of permutations of  $\mathcal{R}$  that leave the  $\mathcal{A}_i$  invariant, i.e. if  $\pi \in \Pi$  and  $c \in \mathcal{A}_i$ , then  $\pi(c) \in \mathcal{A}_i$ . Then  $|\Pi| = (m!)^k$ . Write  $\pi(\mathcal{Q}) := \{\pi(c) : c \in \mathcal{Q}\}$ . Define, for  $c \in \mathcal{R}$ ,

$$\xi_{\mathcal{Q}}(c) := \sum_{\pi \in \Pi} \frac{1}{(m!)^k} \tilde{\xi}_{\pi(\mathcal{Q})}(\pi(c)).$$

Notice that, for a player  $c \notin \mathcal{Q}$ , the value  $\xi_{\mathcal{Q}}(c)$  is 0 as  $\pi(c) \notin \pi(\mathcal{Q})$  for all  $\pi \in \Pi$ . Since we average over all player permutations, for all  $1 \leq i \leq k$  and for any two players  $c, d \in \mathcal{A}_i \cap \mathcal{Q}$ , we have  $\xi_{\mathcal{Q}}(c) = \xi_{\mathcal{Q}}(d)$ . It remains to show that  $\xi$  is cross-monotonic and  $\alpha$ -budget balanced.

Consider adding a player  $d$  to set  $\mathcal{Q}$ . We have to argue that the cost share of an individual player cannot increase. For a player  $c \in \mathcal{Q}$  we see that

$$\xi_{\mathcal{Q} \cup \{d\}}(c) = \sum_{\pi \in \Pi} \frac{1}{(m!)^k} \tilde{\xi}_{\pi(\mathcal{Q} \cup \{d\})}(\pi(c)) \leq \sum_{\pi \in \Pi} \frac{1}{(m!)^k} \tilde{\xi}_{\pi(\mathcal{Q})}(\pi(c)) = \xi_{\mathcal{Q}}(c).$$

This follows since  $\pi(\mathcal{Q} \cup \{d\}) = \pi(\mathcal{Q}) \cup \{\pi(d)\}$  and hence the cross-monotonicity of  $\tilde{\xi}$  can be applied to each term.

Now we show  $\alpha$ -budget balance. To this end we must specify which solution is returned by the algorithm. If we denote with  $S^\pi$  the solution returned by cost sharing method  $\tilde{\xi}$  when run on set  $\pi(\mathcal{Q})$ , we return the solution  $S \in \{S^\pi : \pi \in \Pi\}$  with cost  $c(S) = \min_{\pi \in \Pi} c(S^\pi)$ .

Of course this solution is not necessarily feasible for the original player set, but because of the symmetry of the instance there is a graph isomorphism that maps the solution back to a feasible one without changing the cost.

Now we can write

$$\begin{aligned} \sum_{c \in \mathcal{Q}} \xi_{\mathcal{Q}}(c) &= \sum_{c \in \mathcal{Q}} \sum_{\pi \in \Pi} \frac{1}{(m!)^k} \tilde{\xi}_{\pi(\mathcal{Q})}(\pi(c)) = \sum_{\pi \in \Pi} \frac{1}{(m!)^k} \sum_{c \in \mathcal{Q}} \tilde{\xi}_{\pi(\mathcal{Q})}(\pi(c)) \\ &\geq \sum_{\pi \in \Pi} \frac{1}{(m!)^k} \frac{1}{\alpha} \cdot c(S^\pi) \geq \sum_{\pi \in \Pi} \frac{1}{(m!)^k} \frac{1}{\alpha} \cdot c(S) = \frac{1}{\alpha} \cdot c(S). \end{aligned}$$

Competitiveness can be proved using a similar line of reasoning: the cost of the optimal solution must be the same in any permutation. With that, the proof is complete.  $\square$

Now suppose we are given a symmetric cost sharing method  $\xi$ . From this point on we will identify players and vertices to avoid complication of notation. Ask the algorithm for cost shares for a subset of players  $\{a_1, \dots, a_k\}$  where  $a_i \in A_i$ . By construction of the graph, all these terminals can connect to vertex  $f_{\{a_1, \dots, a_k\}}$  at cost 1, at which point they are only 3 units away from the root. Hence there is a solution of cost  $k + 3$  for this subset. Competitiveness states that

$$\sum_{j=1}^k \xi_{\{a_1, \dots, a_k\}}(a_j) \leq \text{opt}_{\{a_1, \dots, a_k\}} \leq k + 3.$$

Therefore there must be at least one index  $i$  such that  $\xi_{\{a_1, \dots, a_k\}}(a_i) \leq (k + 3)/k$  and Lemma 7.2 implies that

$$\xi_{\{a_1, \dots, a_{i-1}, c, a_{i+1}, \dots, a_k\}}(c) \leq (k + 3)/k \quad (7.1)$$

for all  $c \in A_i$ .

For this index  $i$  we consider the instance with subset  $Q = \{a_1, \dots, a_k\} \cup A_i$ . We bound the sum of the cost shares for this set as follows:

$$\begin{aligned} \sum_{c \in Q} \xi_Q(c) &= \sum_{c \in A_i} \xi_Q(c) + \sum_{j \neq i} \xi_Q(a_j) \\ &\leq \sum_{c \in A_i} \xi_{\{a_1, \dots, a_{i-1}, c, a_{i+1}, \dots, a_k\}}(c) + \sum_{j \neq i} \xi_{\{a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_k\}}(a_j) \quad (7.2) \end{aligned}$$

$$\leq m \cdot \frac{k + 3}{k} + k + 2. \quad (7.3)$$

The first inequality is an application of cross-monotonicity; the second follows from (7.1) and the fact that there is a solution of cost  $k + 2$  for a set

$$\{a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_k\}$$

of players where  $a_j \in A_j$ .

Due to the large amount of symmetry in the instance, we can in fact describe the optimal solution.

**LEMMA 7.3.** *The optimal solution for connecting the players in a set  $Q$ , as defined above, to the root has cost  $2m + k + 1$ .*

*Proof.* We observed above that connecting all terminals  $\{a_1, \dots, a_k\}$  via  $f_{\{a_1, \dots, a_k\}}$  to the root has cost  $k + 3$ . Fix a terminal  $a_j \in Q$  with  $a_j \notin A_i$ . Each of the remaining  $m - 1$  terminals in  $A_i \setminus \{a_i\}$  can connect to  $a_j$  at cost 2. Thus,  $\text{opt}_Q \leq k + 3 + 2(m - 1) = 2m + k + 1$ .

We next show that  $2m + k + 1$  is a lower bound on the optimal cost. Suppose  $F$  is the set of vertices  $f_B$ ,  $B \in \mathcal{B}$ , that are used to connect all terminals in  $Q$  to the root  $r$  and define  $f = |F|$ . Clearly,  $1 \leq f \leq m$ . The cost of connecting all vertices in  $F$  to the root is  $3f$ . Moreover, connecting all  $k - 1$  terminals in  $Q \setminus A_i$  to  $F$  has cost at least  $k - 1$ . At most  $f$  terminals in  $A_i$  are adjacent to a vertex in  $F$  and the total cost of connecting these terminals to  $F$  is  $f$ . The remaining  $m - f$  terminals in  $A_i$  are not adjacent to any of the  $f$  vertices in  $F$  and therefore the cost of connecting these terminals to  $F$  is at least  $2(m - f)$ . Hence, the cost of connecting all terminals in  $Q$  via vertices in  $F$  is at least

$$3f + k - 1 + f + 2(m - f) = 2m + k + 2f - 1 \geq 2m + k + 1.$$

□

Combining Lemma 7.3 with Inequality (7.3), we can now prove Theorem 7.1.

*Proof.* [Theorem 7.1] The ratio between the cost shares of players in the subset  $Q$  as defined above and the cost of the network they use can be bounded as follows:

$$\frac{\sum_{c \in Q} \xi_Q(c)}{c(Q)} \leq \frac{\sum_{c \in Q} \xi_Q(c)}{\text{opt}_Q} \leq \frac{m \frac{k+3}{k} + k + 2}{2m + k + 1} = \frac{k^2 + 4k + 2}{2k^2 + k + 1},$$

where the last equality holds if we choose  $m = k^2$ . This ratio tends to  $1/2$  as  $k \rightarrow \infty$ , which completes the proof. □

#### REFERENCES

- [1] A. AGARWAL AND M. CHARIKAR, *On the advantage of network coding for improving network throughput*, in Proceedings, IEEE Information Theory Workshop, 2004.
- [2] 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.
- [3] Y. P. ANEJA, *An integer linear programming approach to the Steiner problem in graphs*, Networks, 10 (1980), pp. 167–178.
- [4] S. ARORA, C. LUND, R. MOTWANI, M. SUDAN, AND M. SZEGEDY, *Proof verification and the hardness of approximation problems*, J. ACM, 45 (1998), pp. 501–555.
- [5] M. BERN AND P. PLASSMANN, *The Steiner problem with edge lengths 1 and 2*, Inform. Process. Lett., 32 (1989), pp. 171–176.
- [6] J. EDMONDS, *Optimum branchings*, J. Res. Nat. Bur. Standards Sect. B, 71B (1967), pp. 233–240.
- [7] M. R. GAREY AND D. S. JOHNSON, *Computers and intractability*, W. H. Freeman and Co., San Francisco, Calif., 1979. A guide to the theory of NP-completeness.
- [8] M. X. GOEMANS AND D. P. WILLIAMSON, *A general approximation technique for constrained forest problems*, SIAM J. Comput., 24 (1995), pp. 296–317.
- [9] N. IMMORLICA, M. MAHDIAN, AND V. S. MIRROKNI, *Limitations of cross-monotonic cost sharing schemes*, in Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, ACM Press, 2005, pp. 602–611.
- [10] K. JAIN AND V. V. VAZIRANI, *Applications of approximation algorithms to cooperative games*, in Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing, 2001, pp. 364–372.
- [11] K. KENT AND D. SKORIN-KAPOV, *Population monotonic cost allocation on mst’s*, in Operational Research Proceedings KOI, 1996, pp. 43–48.
- [12] J. KÖNEMANN, S. LEONARDI, AND G. SCHÄFER, *A group-strategyproof mechanism for Steiner forests*, in Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, ACM Press, 2005, pp. 612 – 619.
- [13] J. KÖNEMANN, S. LEONARDI, G. SCHÄFER, AND S. VAN ZWAM, *From primal-dual to cost shares and back: A stronger LP relaxation for the Steiner forest problem*, in Proceedings of the 32nd International Colloquium on Automata, Languages and Programming, vol. 3580 of Lecture Notes in Computer Science, Springer Berlin/Heidelberg, 2005, pp. 930–942.

- [14] H. MOULIN AND S. SHENKER, *Strategyproof sharing of submodular costs: budget balance versus efficiency*, *Econom. Theory*, 18 (2001), pp. 511–533.
- [15] M. PÁL AND É. TARDOS, *Group strategyproof mechanisms via primal-dual algorithms*, in *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, 2003, pp. 584–593.