

From Primal-Dual to Cost Shares and Back: A Stronger LP Relaxation for the Steiner Forest Problem

Jochen Könemann* Stefano Leonardi† Guido Schäfer† Stefan van Zwam‡

Abstract

In this paper we consider a game theoretical variant of the Steiner forest problem. An instance of this game consists of an undirected graph $G = (V, E)$, non-negative costs $c(e)$ for all edges e in E , and k players. Each player i has an associated pair of terminals s_i and t_i . Consider a forest F in G . We say that player i is serviced if s_i and t_i are connected in F . Player i derives a private utility u_i for receiving service. In a recent paper, Könemann, Leonardi, and Schäfer [12] showed that a natural primal-dual algorithm, KLS, gives rise to a 2-approximate budget balanced and group-strategyproof cost sharing method for the above game.

In this paper we show that the techniques used in [12] yield a new linear programming relaxation for the Steiner forest problem: the *lifted-cut relaxation*. First, we give an alternate proof of the approximate budget-balance result in [12] by showing that the cost shares computed by algorithm KLS are feasible for the dual of this relaxation. Second, we are able to show that this new undirected relaxation for Steiner forests is strictly stronger than the well-studied *undirected cut* relaxation.

We conclude the paper with 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 and Steiner forest games. This shows that the results of [11, 12] are essentially tight.

*Department of Combinatorics and Optimization, University of Waterloo, Canada. Email: jochen@uwaterloo.ca.

†Dipartimento di Informatica e Sistemistica, Università di Roma “La Sapienza”, Italy. Email: {leon,schaefer}@dis.uniroma1.it.

‡Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands. Email: s.h.m.v.zwam@student.tue.nl.

1 Introduction

The Steiner forest problem. In the Steiner forest problem we are given an undirected graph $G = (V, E)$, with vertex set V and edge set E , a non-negative cost function $c : E \rightarrow \mathbb{R}^+$ on the edges of G , and a set of $k > 0$ terminal pairs

$$R = \{(s_1, t_1), \dots, (s_k, t_k)\} \subseteq V \times V.$$

A feasible solution for a Steiner forest instance 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 F of smallest total cost $c(F) := \sum_{e \in F} c(e)$. The *Steiner tree* problem is a special case of the Steiner forest problem that consists of connecting a set of terminals $R \subseteq V$ to a root vertex $r \in V$ in the cheapest possible way.

Computing minimum-cost Steiner trees and forests is NP-hard [7] and APX-complete [3, 4] 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 [1], and Goemans and Williamson [9] use the primal-dual schema. The algorithms in [1, 9] achieve an approximation ratio of $(2 - 1/k)$.

Cost sharing mechanisms. In this paper we will consider the following natural game-theoretic version of the Steiner forest problem: each terminal pair $(s_j, t_j) \in R$ is associated with a player j that wants to establish a connection between s_j and t_j . Player j derives a (privately known) utility value u_j from an existing connection between its terminals.

A *cost sharing method* ξ is an algorithm that, given a subset $Q \subseteq R$ of the players, computes a Steiner forest of cost $c(Q)$ satisfying the connectivity requirements of all players in Q . Moreover, for each player $j \in Q$ it determines a non-negative cost share $\xi_Q(j)$. We say that a cost sharing method is α -*budget balanced* if

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

The first inequality says that at least a $1/\alpha$ fraction of the total cost of servicing the users in Q is recovered by the sum of the cost shares of the users in Q . The second inequality establishes fairness in that the sum of all cost shares is not allowed to exceed the optimum cost of servicing the users in Q . This second inequality is often referred to as *competitiveness*. In this paper we will be interested in cost sharing methods ξ that are computable in polynomial time. Therefore, the cost $c(Q)$ of servicing a set of players Q will necessarily exceed the cost opt_Q of an optimum solution for some instances. This also means that there is no hope to achieve budget balance, i.e. $\alpha = 1$.

An important class of cost sharing methods are those that are *cross-monotonic*. ξ 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 player set increases. The importance of cross-monotonic cost-sharing methods stems from a result by Moulin and Shenker [13]: any (approximately) budget balanced cross-monotonic cost sharing method can be turned into an (approximately) budget balanced group-strategyproof mechanism.

Despite the recent interest in computational game theory, examples for combinatorial optimization problems that possess a cross-monotonic cost sharing method are still few: Moulin and Shenker [13] gave a cross-monotonic cost sharing method for problems whose optimal cost function is a sub-modular function of the set Q . However, this condition does not hold for many important network design problems such as Steiner trees and facility location. Jain and Vazirani [11] showed a 2-budget balanced

and cross-monotonic cost sharing method for Steiner trees. Pál and Tardos [14] later obtained a 3-budget balanced and group-strategyproof cost sharing method for facility location and also provided a 15-budget balanced solution to the single sink rent-or-buy problem.

In a recent paper, Immorlica, Mahdian, and Mirrokni [10] show that combinatorial problems that are well-behaved with respect to their approximability may prove hard when looking for approximately budget balanced cross-monotonic cost sharing methods. Among other results they prove 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, and a lower bound of 3 for a facility location game, therefore showing that the result of [14] is tight. Observe that these lower bounds are achieved by using cross-monotonicity only. The authors left open the issue of finding a lower bound on the budget balance factor for the Steiner tree problem.

Our contribution: Both Jain and Vazirani [11], and Pál and Tardos [14] show that the computed cost shares form a feasible solution for the dual of a linear programming relaxation for the problem. Proving competitiveness of the methods can therefore be reduced to an application of weak duality. The budget balance factor corresponds to the performance guarantee of the underlying primal-dual algorithm.

In a recent paper [12], Könemann, Leonardi, and Schäfer depart from this line. The authors present a cost sharing method KLS which is an adaptation of the primal-dual algorithm AKR for Steiner forests due to Agrawal, Klein, and Ravi [1]. The cost shares computed by KLS are proven to be 2-budget balanced. They do not, however, correspond to a feasible dual solution for any of the known Steiner forest duals.

The obvious question left open by [12], that we answer affirmatively in this paper, is: Is there an alternate Steiner forest LP formulation such that the cost shares computed by KLS correspond to a feasible dual solution? If so, how does this new LP relaxation relate to the standard undirected-cut LP relaxation?

Theorem 1. *There is a linear programming relaxation (LC-D) for the Steiner forest problem whose optimum solution value is at most the cost of any feasible Steiner forest for the given instance. (LC-D) is strictly stronger than the well-known undirected-cut relaxation for Steiner forests. The dual solution computed by KLS is feasible for (LC-D).*

The algorithms in [1, 9] are based on the classical undirected cut formulation for Steiner forests [2]. The integrality gap of this relaxation is known to be $(2 - 1/k)$ and the results in [1, 9] are therefore tight. Our lifted-cut dual relaxation is strictly stronger than the classical undirected cut formulation. There are instances in which the dual solution achieved by our relaxation provides a much better approximation of the optimum than the undirected-cut dual relaxation. On the other hand, there are instances in which both relaxations achieve an integrality gap of $(2 - \frac{1}{k})$. Can we characterize the class of instances where the IP/LP-gap of the lifted cut relaxation is close to 2? Can we use the extra strength of the new relaxation to obtain an improved algorithm for the Steiner forest problem?

Secondly, one naturally wonders whether there is a $(2 - \epsilon)$ -budget balanced and cross-monotonic cost sharing method for Steiner trees and forests. The answer to this question is negative, and holds for any cross-monotonic cost sharing method for these games, including those taking exponential time:

Theorem 2. *There is no $(2 - \epsilon)$ -budget balanced, cross-monotonic cost sharing method for Steiner trees for any $\epsilon > 0$.*

This lower bound shows that the results in [11, 12] are essentially best possible. Hence there is no hope to obtain a $(2 - \epsilon)$ -budget-balanced cross-monotonic cost sharing scheme even if a linear

programming relaxation with such integrality gap would exist. Prior to our work, the only upper-bound known for the budget-balance factor of a cross-monotonic cost-sharing method for the Steiner tree and forest games was the IP/LP gap of the bidirected cut relaxation [10] which is at most $8/9$.

Organization of the paper: In Section 2 we recap the primal-dual algorithm for Steiner forests due to Agrawal, Klein and Ravi. We also describe the cross-monotonic adaptation KLS of AKR and state the main results from [12]. Afterwards, in Section 3 we derive the new lifted-cut relaxation for Steiner forests and present a proof of Theorem 1. The lower-bound proof for the budget balance factor of any cross-monotonic cost sharing method for Steiner forests is given in Section 4.

2 Cross-monotonic cost sharing method for Steiner forests

We review the cross-monotonic cost sharing method for Steiner forests as given in [12]. The algorithm is similar to the primal-dual algorithm AKR due to Agrawal, Klein, and Ravi [1]. In this section, we first state the standard LP formulation on which AKR is based, then review AKR, and finally describe the cost sharing algorithm KLS from [12].

Undirected cut LP formulation for Steiner forests. Let $S \subseteq V$. We define $\delta(S)$ to be the set of all edges that have exactly one endpoint in S and we also let $R(S)$ be the set of terminals in S whose mates are in $V \setminus S$, i.e., $R(S) := \{w \in S : (w, \bar{w}) \in R \text{ and } \bar{w} \notin S\}$. We use $r(S)$ for the cardinality of $R(S)$. A subset $S \subseteq V$ is a *Steiner cut* if $r(S) \geq 1$. Let \mathcal{S} be the set of all Steiner cuts.

Consider a Steiner cut $S \in \mathcal{S}$. Any feasible solution F for a given Steiner forest instance must *cross* this cut at least once, i.e., $|\delta(S) \cap F| \geq 1$. This gives rise to the following integer programming formulation for the Steiner forest problem: We have a variable x_e for each edge $e \in E$ which has value 1 if e is part of the resulting forest and 0 otherwise.

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

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

$$\begin{aligned} \text{opt}_D := \max \quad & \sum_{S \in \mathcal{S}} y_S & \text{(D)} \\ \text{s.t.} \quad & \sum_{S \in \mathcal{S}: e \in \delta(S)} y_S \leq c(e) & \forall e \in E \\ & y_S \geq 0 & \forall S \in \mathcal{S} \end{aligned} \quad (2)$$

Primal-dual algorithm for Steiner forests. AKR is a *primal-dual* algorithm. That is, it constructs both a feasible and integral primal solution for (LP) and a feasible dual solution for (D), respectively. The algorithm starts with an infeasible primal solution and reduces the degree of infeasibility as it progresses. At the same time, it creates a dual feasible packing of sets of largest possible total value. The algorithm raises dual variables of certain subsets of vertices. The final dual solution is maximal in the sense that no single set can be raised without violating a constraint of type (2).

We can think of an execution of AKR as a process over time. Let x^t and y^t be the primal incidence vector and feasible dual solution at time t . We use F^t to denote the forest corresponding to x^t . Initially, $x_e^0 = 0$ for all $e \in E$ and $y_S^0 = 0$ for $S \in \mathcal{S}$. In the following we say that an edge $e \in E$ is *tight* if the corresponding constraint (2) holds with equality.

Assume that the forest F^t at time t is infeasible. We use \bar{F}^t to denote the subgraph of G that is induced by the tight edges for dual y^t . A connected component S of \bar{F}^t is *active* iff S separates at least one terminal pair, i.e., iff $S \in \mathcal{S}$. Let \mathcal{C}^t be the set of all active connected components of \bar{F}^t at time t . AKR raises the dual variables for all sets in \mathcal{C}^t uniformly at all times $t \geq 0$.

Suppose now that two active connected components S_1 and S_2 *collide* at time t in the execution of AKR. In other words, there are terminals $u \in S_1$ and $v \in S_2$ such that a path between u and v becomes tight as a consequence of increasing y_{S_1} and y_{S_2} . If this happens, we add the path to F^t and continue. S_1 and S_2 are part of the same connected component of $\bar{F}^{t'}$ for $t' > t$.

The following is the main result of [1]:

Theorem 3. *Suppose that algorithm AKR outputs a forest F and a feasible dual solution $\{y_S\}_{S \in \mathcal{S}}$. Then $c(F) \leq (2 - 1/k) \cdot \sum_{S \in \mathcal{S}} y_S \leq (2 - 1/k) \cdot \text{opt}_R$, where opt_R is the minimum-cost of a Steiner forest for the given input instance with terminal set R .*

Cross-monotonic cost sharing algorithm. We next describe the modifications that are necessary to turn AKR into a cross-monotonic cost sharing algorithm. We use KLS to refer to this algorithm.

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

$$d(s, t) := \frac{1}{2} \cdot c(s, t), \quad (3)$$

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 $d(s) = d(t) = d(s, t)$ for terminals $s, t \in R$.

Using the notation introduced above we obtain KLS by modifying the definition of \mathcal{C}^t . We say that a connected component S of \bar{F}^t is *active* at time t if it contains at least one terminal $v \in S$ with death time at least t , i.e., S is active iff there exists a $v \in S$ with $d(v) \geq t$. KLS grows all active connected components in \mathcal{C}^t uniformly at all times $t \geq 0$. Observe that this way KLS also raises dual variables of connected components in \mathcal{C}^t that do not correspond to Steiner cuts. In what follows we denote by \mathcal{N} the set of *non-Steiner cuts*: $\mathcal{N} := \{S \subseteq V : S \notin \mathcal{S}, S \cap R \neq \emptyset\}$. Furthermore, we let $\mathcal{U} := \mathcal{S} \cup \mathcal{N}$ be the set of all Steiner and non-Steiner cuts.

The intuition behind KLS is that a terminal pair (s, t) is active for the time it would take s and t to connect in the absence of all other terminals. Therefore its activity time is independent of other terminal pairs and this is crucial to achieve cross-monotonicity.

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

$$\xi_R(v) := \int_{t=0}^{d(v)} \frac{1}{a^t(v)} dt. \quad (4)$$

Furthermore, we define $\xi_R(s, t) := \xi_R(s) + \xi_R(t)$ for all $(s, t) \in R$.

Theorem 4. ξ is a cross-monotonic cost sharing method that is 2-budget balanced.

A proof of this theorem was presented in [12]. There, one of the major difficulties was to show that ξ is *competitive*, i.e., that the sum of the cost shares is at most the optimal cost, opt_R . Since we share the entire dual produced during the execution of KLS among the terminal pairs in R , proving competitiveness is equivalent to showing that the dual solution $\{y_S\}_{S \in \mathcal{U}}$ satisfies

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

If y were a feasible solution to (D) this would follow immediately from weak duality. Here, however, we cannot apply this argument, since KLS also raises dual variables of non-Steiner cuts. Subsequently, we present an alternative, LP-based proof for the competitiveness of ξ .

3 A strong non-standard LP relaxation for Steiner forests

Recall that we let $R = \{(s_1, t_1), \dots, (s_k, t_k)\}$ be the set of terminal pairs in our instance. W.l.o.g., we assume in the following that $d(s_1, t_1) \leq \dots \leq d(s_k, t_k)$. We define a precedence order \prec on R by letting $(s_i, t_i) \prec (s_j, t_j)$ iff $i \leq j$ and we extend this order to terminal vertices by letting

$$s_1 \prec t_1 \prec s_2 \prec t_2 \prec s_3 \prec \dots \prec s_k \prec t_k.$$

Consider a terminal w and let \bar{w} be w 's mate in the Steiner forest instance (i.e., $(w, \bar{w}) \in R$). We let $\mathcal{S}_w \subseteq \mathcal{S}$ be the set of Steiner cuts that separate w and \bar{w} and for which (w, \bar{w}) is the highest ranked such terminal pair:

$$\mathcal{S}_w := \{S \in \mathcal{S} : w \in R(S), v \prec w \text{ for all } v \in R(S)\}.$$

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

$$\mathcal{N}_w := \{S \in \mathcal{N} : \{w, \bar{w}\} \subseteq S \cap R, (v, \bar{v}) \prec (w, \bar{w}) \text{ for all } (v, \bar{v}) \in S \cap R\}.$$

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 $w \in R$ is *responsible* for a cut $S \in \mathcal{U}$ if $S \in \mathcal{S}_w \cup \mathcal{N}_w$.

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

$$\begin{aligned} \text{opt}_{\text{LC-D}} &:= \max \sum_{S \in \mathcal{U}} y_S && \text{(LC-D)} \\ \text{s.t.} & \sum_{S \in \mathcal{U} : e \in \delta(S)} y_S \leq c(e) && \forall e \in E && (5) \\ & \sum_{S \in \mathcal{S}_w} y_S + \sum_{S \in \mathcal{N}_w} y_S \leq d(w) && \forall w \in R && (6) \\ & y_S \geq 0 && \forall S \in \mathcal{U} \end{aligned}$$

Notice that a feasible solution to (LC-D) may assign positive values to non-Steiner cuts $S \in \mathcal{N}$. The constraints of type (6) 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_w for every terminal $w \in R$:

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

$$\text{s.t.} \quad \sum_{e \in \delta(S)} x_e + x_w \geq 1 \quad \forall S \in \mathcal{S}_w, \forall w \in R \quad (7)$$

$$\sum_{e \in \delta(S)} x_e + x_w + x_{\bar{w}} \geq 1 \quad \forall S \in \mathcal{N}_w, \forall w \in R \quad (8)$$

$$x_e, x_w \geq 0 \quad \forall e \in E, \forall w \in R$$

Let $\{x_e, x_w\}_{e \in E, w \in R}$ be an integral solution that is feasible for (LC-P). We argue that this solution gives rise to a feasible Steiner forest with cost not exceeding the objective function value. Define $F := \{e \in E : x_e = 1\}$. The total cost of $c(F)$ 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 $S \in \mathcal{S}$ with no crossing edge, i.e., $\delta(S) \cap F = \emptyset$. Let $S \in \mathcal{S}_w$ be such a set and let \bar{w} be the mate of w . Constraint (7) for S and w implies that $x_w = 1$ in this case. Next consider the complement $\bar{S} = V \setminus S$. It can be seen that \bar{w} is responsible for \bar{S} and hence, $\bar{S} \in \mathcal{S}_{\bar{w}}$. As no edge crosses \bar{S} , we must have $x_{\bar{w}} = 1$. Therefore, we can add all edges along the shortest w, \bar{w} -path to F at a cost of $2d(w, \bar{w})$.

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 1. *Let F be a feasible solution for the underlying Steiner forest instance. We can then construct a solution x that is feasible for (LC-P) and satisfies:*

$$\sum_{e \in E} c(e) \cdot x_e + \sum_{w \in R} d(w) \cdot x_w \leq c(F).$$

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

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 that is feasible for (LC-P) and show that for each tree $T \in F$

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

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 that is responsible for the non-Steiner cut $V(T)$. Moreover, let P denote the unique w, \bar{w} -path in T . We set $x_e := \frac{1}{2}$ for each edge $e \in E(P)$ and $x_e := 1$ for each edge $e \in E(T \setminus P)$. Moreover, we assign $x_w = x_{\bar{w}} := \frac{1}{2}$ and $x_r := 0$ for all terminals $r \in (R \cap V(T)) \setminus \{w, \bar{w}\}$.

The objective value for x on T is

$$c(T) - \frac{1}{2}c(P) + \frac{1}{2}d(w) + \frac{1}{2}d(\bar{w}) = c(T) - \frac{1}{2}c(P) + d(w, \bar{w}) \leq c(T),$$

where the last inequality holds since $d(w, \bar{w}) \leq \frac{1}{2}c(P)$, by the definition (3) of death time.

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 (7) and (8) for sets $S \in \mathcal{S}_v \cup \mathcal{N}_v$.

Consider a cut $S \in \mathcal{S}_v$ for some $v \in R \cap V(T)$. If $v \in \{w, \bar{w}\}$, constraint (7) holds since S intersects P and $x_v = \frac{1}{2}$. Now let $v \notin \{w, \bar{w}\}$. As (w, \bar{w}) is responsible for $V(T)$ we must have $(v, \bar{v}) \prec (w, \bar{w})$ and hence $\{w, \bar{w}\} \cap S = \emptyset$. It can now be seen that S either intersects at least one edge e of T that is not on P (and hence $x_e = 1$) or it intersects at least two edges e_1 and e_2 on P (and therefore $x_{e_1} = x_{e_2} = \frac{1}{2}$). Thus, constraint (7) holds in this case as well.

Next consider a non-Steiner cut $S \in \mathcal{N}_v$ for terminal $v \in R \cap V(T)$. If $v \notin \{w, \bar{w}\}$ then $\{w, \bar{w}\} \cap S = \emptyset$ as before and S crosses at least one edge of T that is not on P or at least two edges of P . Hence constraint (8) holds. Otherwise, S may cross no edge of T but $x_w + x_{\bar{w}} = 1$ and thus (8) is satisfied. \square

Running algorithm KLS on terminal set R yields a cost-share $\xi_R(s, t)$ for all $(s, t) \in R$. The algorithm also returns a dual solution y such that

$$\sum_{(s,t) \in R} \xi_R(s, t) = \sum_{S \in \mathcal{U}} y_S.$$

It is easy to verify that y is feasible for (LC-D). Lemma 1 therefore yields an alternate proof of the competitiveness of KLS:

Corollary 1. ξ satisfies competitiveness: $\sum_{(s,t) \in R} \xi_R(s, t) = \sum_{S \in \mathcal{U}} y_S \leq \text{opt}_{LC-D} \leq \text{opt}_R$.

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

Lemma 2. Let $\{y_S\}_{S \in \mathcal{S}}$ be a feasible dual solution for (D). Then there is a feasible dual solution $\{y'_S\}_{S \in \mathcal{U}}$ for (LC-D) with

$$\sum_{S \in \mathcal{S}} y_S \leq \sum_{S \in \mathcal{U}} y'_S.$$

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

Proof. Let y be a feasible solution for (D). The sets \mathcal{S}_w for terminals $w \in R$ form a partition of \mathcal{S} : $\mathcal{S} = \bigcup_{w \in R} \mathcal{S}_w$. First, we argue that we can assume that y is *symmetric* in the following sense. y is symmetric if for all $(s, t) \in R$:

$$l_s := \sum_{S \in \mathcal{S}_s} y_S = \sum_{S \in \mathcal{S}_t} y_S =: l_t.$$

Suppose that this equality does not hold for some $(s, t) \in R$ and, without loss of generality, assume that $l_s > l_t$. Then, let $S \in \mathcal{S}_s$ be a set with $y_S > 0$.

Consider the set $\bar{S} = V \setminus S$ and observe that this set is a Steiner cut as well. Moreover, S and \bar{S} trivially separate the same terminal pairs in R , i.e., if $w \in R(S)$ for some terminal pair $(w, \bar{w}) \in R$ then $\bar{w} \in R(\bar{S})$. It therefore follows that t must be the responsible terminal for \bar{S} and hence $\bar{S} \in \mathcal{S}_t$. Finally, notice that $\delta(S) = \delta(\bar{S})$ and hence we can increase $y_{\bar{S}}$ and decrease y_S at the same rate without violating any of the constraints of type (2). Continuing this procedure will lead to a symmetric dual y that is feasible for (D).

Now define $y'_S = y_S$ if S is a Steiner cut and let $y'_S = 0$ otherwise. y' clearly satisfies all constraints of type (5). We will now show that y' also satisfies all constraints of type (6) and this will finish the proof of the lemma.

Assume for the sake of contradiction that y' violates constraint (6) for some terminal $w \in R$:

$$l_w + \sum_{S \in \mathcal{N}_w} y'_S > d(w).$$

Since $y'_S = 0$ for all non-Steiner cuts $S \in \mathcal{N}_w$ we therefore must have $l_w > d(w) = c(P)/2$, where $c(P)$ is the cost of a minimum-cost w, \bar{w} -path in G .

Using the symmetry of y we know that $l_w = l_{\bar{w}}$ and hence we must have $l_w + l_{\bar{w}} > c(P)$. On the other hand, adding the constraints of type (2) for all edges $e \in P$ yields

$$l_w + l_{\bar{w}} \leq \sum_{S \in \mathcal{S}} |\delta(S) \cap P| \cdot y_S = \sum_{e \in P} \sum_{S \in \mathcal{S}: e \in \delta(S)} y_S \leq c(P)$$

and this is a contradiction. \square

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

Lemma 3. *There exist instances for which $\text{opt}_D < \text{opt}_{LC-D}$.*

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

The solution $y_{\{v\}} := \frac{1}{2}$ for each $v \in V$ and $y_S := 0$ otherwise is a feasible solution to (D). Observe that this is an optimal solution for (D) as there is a half-integral solution for the LP relaxation (LP) having the same cost: set $x_e := \frac{1}{2}$ for each edge e of the cycle. Thus, $\text{opt}_D = n/2$.

On the other hand, for (LC-D) we can define a dual solution $y'_{\{v\}} := \frac{1}{2}$ for each $v \in V$, $y'_V := n/4 - 1/2$, and $y'_S := 0$ otherwise. It is easy to verify that y' is a feasible solution for (LC-D). We therefore have

$$\text{opt}_{LC-D} \geq \sum_{S \in \mathcal{U}} y'_S = \frac{1}{2} \cdot n + \frac{1}{4} \cdot n - \frac{1}{2} = \frac{3n}{4} - \frac{1}{2}.$$

The latter term is strictly larger than $n/2$ if $n > 2$. \square

Lemmas 1, 2, and 3 together with Corollary 1 finish the proof of Theorem 1. Unfortunately, as with the undirected cut formulation for Steiner forests, the IP/LP gap of the lifted-cut relaxation is about 2 for certain instances.

Lemma 4. *There exist instances for which $\text{opt}_R / \text{opt}_{LC-D} = 2 - 1/n$, where n is the number of nodes.*

Proof. We consider the following minimum-spanning tree instance. Let K_n be a clique with vertices $V := \{v_1, v_2, \dots, v_n\}$ and unit edge costs. Let $R := \{(v_1, v_j)\}_{2 \leq j \leq n}$. Clearly, the optimal solution opt_R has cost $n - 1$. 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 $x_w = x_{\bar{w}} := \frac{1}{2}$ and $x_v := 0$ for all $v \in V \setminus \{w, \bar{w}\}$, and $x_e := \frac{1}{2}$ for all edges $e \in P$ and $x_e := 0$ for all edges $e \in E \setminus P$. This solution satisfies constraints (7) and (8). The objective function value for x is $(n - 1) \cdot \frac{1}{2} + \frac{1}{2}d(w) + \frac{1}{2}d(\bar{w}) = n/2$.

Next consider the following dual solution. Let $y_{\{v\}} := \frac{1}{2}$ for all $v \in V$ and $y_S := 0$ otherwise. Then y satisfies constraints (5) and (6). The objective value of y is $n/2$ and thus x and y are optimal solutions to (LC-P) and (LC-D), respectively. The ratio between opt_R and opt_{LC-D} is $2 - 1/n$. \square

4 A lower bound for the Steiner tree game

The tools used in this section are adaptations of those used in [10]. In particular we consider any given cross-monotonic cost sharing method ξ 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 optimum solution. Instead of using a probabilistic argument similar to the one described in [10], 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 [10]. We construct an undirected graph $G = (V, E)$. In this graph, there are k pairwise disjoint classes 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 the root. The set of all players that have a vertex associated with them in A_i is denoted with \mathcal{A}_i . The set of all players is $\mathcal{R} := \bigcup_{i=1}^k \mathcal{A}_i$.

Let \mathcal{B} be a set containing 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 with distance 1 to all vertices in B . The distance to the vertices not in B is, by triangle inequality, equal to 3. Finally, every vertex f_B is connected to the root r , with edges of length 3.

The following lemma argues that we may assume that ξ is *symmetric* in the following sense: Consider a subset $\mathcal{T} \subseteq \mathcal{R}$. We then may assume that $\xi_{\mathcal{T}}(c) = \xi_{\mathcal{T}}(d)$ for any two players $c, d \in \mathcal{A}_i \cap \mathcal{T}$ and for any $1 \leq i \leq k$.

Lemma 5. *If there exists an α -budget balanced cost sharing method for the Steiner tree game, there exists also an α -budget balanced cost sharing method satisfying that for all i and for all sets of players $\mathcal{T} \subseteq \mathcal{R}$, $\xi_{\mathcal{T}}(c) = \xi_{\mathcal{T}}(d)$ for all $c, d \in \mathcal{T} \cap \mathcal{A}_i$. Moreover, for all $c \in \mathcal{T} \cap \mathcal{A}_i$, $d \in \mathcal{A}_i \setminus \mathcal{T}$, $\xi_{\mathcal{T}}(c) = \xi_{(\mathcal{T} \setminus \{c\}) \cup \{d\}}(d)$.*

Proof. Let $\tilde{\xi}$ be an α -budget balanced cost sharing method for the Steiner tree game. Pick an index i and a set of players $\mathcal{T} \subseteq \mathcal{R}$. Let π_i be a permutation of the elements of a class \mathcal{A}_i . This means that, for $c \in \mathcal{A}_i$, if c had as terminal vertex a_{ci} , it will now have as terminal vertex $a_{\pi_i(c)i}$. Define the map π by $\pi(c) = \pi_i(c)$ iff $c \in \mathcal{A}_i$. For a set of players \mathcal{T} , we define

$$\pi(\mathcal{T}) := \{d \in \mathcal{R} : \exists c \in \mathcal{T} : d = \pi(c)\}.$$

Let Π be the set of all $(m!)^k$ possible maps that arise in this way. Define the cost sharing method ξ by

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

The important observation here is that the cost sharing method $\tilde{\xi}$ works on vertices. So if we swap two players, the cost shares for these players will be swapped as well. In other words, for any two permutations the algorithm will be presented with the same number of terminals, but these terminals may have switched location in the graph - though they do stay within the same class.

For a player $c \notin \mathcal{T}$ the value $\xi_{\mathcal{T}}(c)$ will be 0 because $\pi(c) \notin \pi(\mathcal{T})$. For all players $c \in \mathcal{T} \cap \mathcal{A}_i$, the sum will result in the same answer. Since we average over all permutations, the location of a demand is no longer important. So if $\tilde{\xi}$ is indeed cross-monotonic and if it is α -budget balanced, the proof is completed.

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

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

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

Now we show α -budget balance. To this end we must specify which solution is returned by the algorithm. If we denote with S^π the solution returned by cost sharing method $\tilde{\xi}$ when run on set $\pi(\mathcal{T})$, 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

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

Competitiveness can be proven 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 cost sharing method ξ as in the lemma. 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 $\xi_{\{a_1, \dots, a_k\}}(a_i) \leq \frac{k+3}{k}$. By Lemma 5, this holds for every set $\{a_1, \dots, a_{i-1}, c, a_{i+1}, \dots, a_k\}$ where $c \in A_i$ and the other a_j are the same as before.

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:

$$\begin{aligned} \sum_{c \in Q} \xi_Q(c) &= \sum_{c \in A_i} \xi_Q(c) + \sum_{j \neq i} \xi_Q(a_j) \leq \sum_{c \in A_i} \xi_{\{a_1, \dots, a_{i-1}, c, a_{i+1}, \dots, a_k\}}(c) + \sum_{j \neq i} \xi_{\{a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_k\}}(a_j) \quad (9) \\ &\leq m \cdot \frac{k+3}{k} + k+2 \quad (10) \end{aligned}$$

The inequality in (9) is due to cross-monotonicity: the cost share of a player over a subset of Q can not be smaller than the cost share for that player over Q .

We know that a set of players never pays more than the cost of the optimal tree connecting these players. For set $\{a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_k\}$, which has one player from $k-1$ of the sets A_j , there is a solution of $3+k-1 = k+2$. This provides an upper bound on the rightmost term of (9). For the leftmost term, we argued above that $\xi_{\{a_1, \dots, a_{i-1}, c, a_{i+1}, \dots, a_k\}}(c) \leq \frac{k+3}{k}$.

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

Lemma 6. *The optimal solution for connecting the players in a set Q , as defined above, to the root has cost $2m+k+1$.*

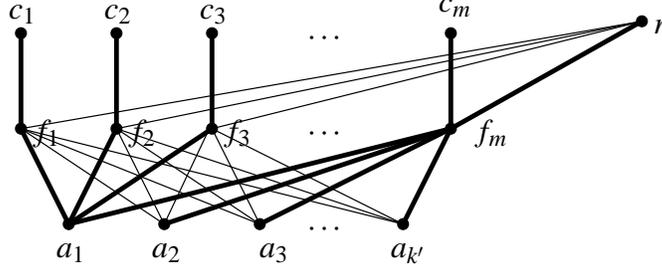


Figure 1: Optimal solution for a set of players $Q := \{a_1, \dots, a_k\} \cup A_i$. In this figure, $A_i = \{c_1, \dots, c_m\}$ and $Q \setminus A_i = \{a_1, \dots, a_k\}$; $Q \setminus A_i$ contains one vertex from each of the $k - 1$ classes A_j with $j \neq i$.

Proof. See Figure 1, in which vertices in classes A_j that have no demand, as well as irrelevant vertices f_B have been omitted. All players have to connect to a vertex f_B for a certain set B as defined above. The vertices $f_{\{a_1, \dots, a_{i-1}, c_j, a_{i+1}, \dots, a_k\}}$, for $c_j \in A_i$ ($j = 1, \dots, m$), have been labeled f_1, \dots, f_m . It is obvious that the other vertices f_B play no role here, since they can connect fewer of the selected players to the root simultaneously. From the vertex f_j players have the choice of connecting directly to the root, at cost 3, or finding another way. Clearly it is efficient to have as many as possible players connect to the same vertex f_B . We can connect at most k players this way. For the remaining players, all in A_i , it is now possible to connect to the partially constructed tree at cost 2. This results in a total cost of $k + 3 + 2(m - 1) = 2m + k + 1$.

Using more than one (f_B, r) edge does not improve the solution. At most one player from A_i can benefit from this edge - the other players in A_i can reach the vertex f_B at cost 3, which gives no improvement. Moreover, even for this one player the cost would increase with 1. This concludes the proof. \square

Combining Lemma 6 with Inequality (10), we can now prove Theorem 2 which we restate here.

Theorem 2. *There is no $(2 - \varepsilon)$ -budget balanced, cross-monotonic cost sharing method for Steiner trees for any $\varepsilon > 0$.*

Proof. 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 $\frac{1}{2}$ as $k \rightarrow \infty$, which completes the proof. \square

References

- [1] A. Agrawal, P. Klein, and R. Ravi. When trees collide: An approximation algorithm for the generalized Steiner problem in networks. *SIAM Journal on Computing*, 24(3):445–456, 1995.
- [2] Y.P. Aneja. An integer linear programming approach to the Steiner problem in graphs. *Networks*, 10:167–178, 1980.
- [3] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and intractability of approximation problems. In *Proc. 33rd IEEE Symp. on Foundations of Computer Science*, 1992.
- [4] M. Bern and P. Plassmann. The Steiner problems with edge lengths 1 and 2. *Information Processing Letters*, 32:171–176, 1989.
- [5] S. Chopra and M. R. Rao. The Steiner tree problem 1: Formulations, compositions, and extension of facets. *Mathematical Programming*, 64:209–229, 1994.
- [6] S. Chopra and M. R. Rao. The Steiner tree problem 2: Properties and classes of facets. *Mathematical Programming*, 64:231–246, 1994.
- [7] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. Freeman, San Francisco, 1979.
- [8] M. X. Goemans and Y. S. Myung. A catalog of Steiner tree formulations. *Networks*, 23:19–28, 1993.
- [9] M. X. Goemans and D. P. Williamson. A general approximation technique for constrained forest problems. *SIAM Journal on Computing*, 24:296–317, 1995.
- [10] 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. to appear.
- [11] 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*, pages 364–372, 2001.
- [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. to appear.
- [13] H. Moulin and S. Shenker. Strategyproof sharing of submodular costs: budget balance versus efficiency. <http://www.aciri.org/shenker/cost.ps>, 1997.
- [14] M. Pál and É. Tardos. Group strategyproof mechanisms via primal-dual algorithms. In *Proceedings of the Forty-Fourth Annual IEEE Symposium on Foundations of Computer Science*, pages 584–593, 2003.
- [15] S. Rajagopalan and V. Vazirani. On the bidirected cut relaxation for the metric Steiner tree problem. In *Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 742–751, 1999.
- [16] G. Robins and A. Zelikovsky. Improved steiner tree approximation in graphs. In *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 770–779, 2000.
- [17] R. T. Wong. A dual ascent approach for steiner tree problems on a directed graph. *Mathematical Programming*, 28:271–287, 1984.