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

Jochen Könemann[1], Stefano Leonardi[2], Guido Schäfer[2], and Stefan van Zwam[3]

[1] Department of Combinatorics and Optimization, University of Waterloo, Canada. jochen@uwaterloo.ca

[2] Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", Italy. {leon,schaefer}@dis.uniroma1.it

[3] Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands. s.h.m.v.zwam@student.tue.nl

Abstract. We consider a game-theoretical variant of the Steiner forest problem, in which each of k users i strives to connect his terminal pair (s_i, t_i) of vertices in an undirected, edge-weighted graph G. In [1] a natural primal-dual algorithm was shown which achieved a 2-approximate budget balanced cross-monotonic cost sharing method for this game.

We derive a new linear programming relaxation from the techniques of [1] which allows for a simpler proof of the budget balancedness of the algorithm from [1]. Furthermore we show that this new relaxation is strictly stronger than the well-known *undirected cut relaxation* for the Steiner forest problem.

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 [1, 2] are essentially tight.

1 Introduction

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 $c : E \to \mathbb{R}^+$ 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 [3] 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 [6] and generalized by

Goemans and Williamson [7], uses the primal-dual schema. The algorithms in [6, 7] achieve an approximation ratio of $(2 - 1/k)$.

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

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

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, denoted $\texttt{opt}_Q$. 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 $\texttt{opt}_Q$ 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 the *cross-monotonic* methods. 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 [8]: any budget balanced cross-monotonic cost sharing method can be turned into a 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 [8] 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 [2] showed a 2-budget balanced and cross-monotonic cost sharing method for Steiner trees. Pál and Tardos [9] 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. Their 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 [2] and Pál and Tardos [9] 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 [1], 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 [6]. 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 [1], that we answer affirmatively in this paper, is this: 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 [6, 7] are based on the classical undirected cut formulation for Steiner forests [11]. The integrality gap of this relaxation is known to be $(2 - 1/k)$ and the results in [6, 7] 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 $(2 - 1/k)$.

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 [1, 2] 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 existed. 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.

2 A Cost Sharing Method for Steiner Forests

We review the cross-monotonic cost sharing method for Steiner forests as given in [1]. The algorithm is similar to the primal-dual algorithm `AKR` due to Agrawal, Klein and Ravi [6]. 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 [1].

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 terminal pairs in R that are separated by S, i.e., $R(S) := \{(s,t) \in R : |\{s,t\} \cap S| = 1\}$. 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.

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

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.

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

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

Suppose now that two active connected components S_1 and S_2 *collide* at time τ 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^τ and continue. S_1 and S_2 are part of the same connected component of $\bar{F}^{\tau'}$ for $\tau' > \tau$.

The following is the main result of [6]:

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

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* $\mathtt{d}(s,t)$ for each terminal pair $(s,t) \in R$ as

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

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

Using the notation introduced above we obtain KLS by modifying the definition of $\mathcal{C}^\tau$. We say that a connected component S of $\bar{F}^\tau$ is *active* at time τ if it contains at least one terminal $v \in S$ with death time at least τ, i.e., S is active iff there exists a $v \in S$ with $\mathtt{d}(v) \geq \tau$. KLS grows all active connected components in $\mathcal{C}^\tau$ uniformly at all times $\tau \geq 0$. Observe that this way KLS also raises dual variables of connected components in $\mathcal{C}^\tau$ 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 $\tau \leq \mathtt{d}(v)$ we let $S^\tau(v)$ be the connected component in $\bar{F}^\tau$ that contains v. Also let $a^\tau(v)$ be the number of terminals in $S^\tau(v)$ whose death time is at least τ. The cost share of terminal vertex $v \in R$ is defined as

$$\xi_R(v) := \int_{\tau=0}^{\mathtt{d}(v)} \frac{1}{a^\tau(v)}\, d\tau \quad . \tag{4}$$

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 [1]. 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, $\mathtt{opt}_R$. 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 $\sum_{S \in \mathcal{U}} y_S \leq \mathtt{opt}_R$. 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 Lifted-Cut LP Relaxation for Steiner Forests

Recall that we let $R = \{(s_1,t_1),\ldots,(s_k,t_k)\}$ be the set of terminal pairs in our instance. Without loss of generality, we assume in the following that $\mathtt{d}(s_1,t_1) \leq \ldots \leq \mathtt{d}(s_k,t_k)$. 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 $s_1 \prec t_1 \prec s_2 \prec t_2 \prec \ldots \prec s_k \prec t_k$. For ease of notation we assume that $v \prec v$ for all $v \in R$.

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:

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

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:

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

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:

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

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 $w \in R$:

$$\begin{aligned}
\texttt{opt}_{\text{LC-P}} := \min \quad & \sum_{e \in E} c(e) \cdot x_e + \sum_{w \in R} \mathtt{d}(w) \cdot x_w & & \text{(LC-P)} \\
\text{s.t.} \quad & \sum_{e \in \delta(S)} x_e + x_w \ge 1 & \forall S \in \mathcal{S}_w, \ \forall w \in R \qquad & (7) \\
& \sum_{e \in \delta(S)} x_e + x_w + x_{\bar{w}} \ge 1 & \forall S \in \mathcal{N}_w, \ \forall w \in R \qquad & (8) \\
& x_e, \ x_w \ge 0 & \forall e \in E, \ \forall w \in R \ . &
\end{aligned}$$

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 $c(F)$ 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 $S \in \mathcal{S}$ 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 $2\mathtt{d}(w, \bar{w})$.

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:*

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

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

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 $T \in F$

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

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 $e \in E(T) \setminus E(P)$. Moreover, we assign $x_w = x_{\bar{w}} := \frac{1}{2}$ and $x_r := 0$ for all terminals $r \in (R \cap V(T)) \setminus \{w, \bar{w}\}$. By definition (3) of death time, $\mathtt{d}(w, \bar{w}) \leq \frac{1}{2} c(P)$. Thus, the objective value for x on T is $c(T) - c(P)/2 + \mathtt{d}(w, \bar{w}) \leq c(T)$.

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 (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 $S \in \mathcal{S}_v$ and $v \prec w$, by assumption, it follows that either $\{w, \bar{w}\} \subseteq S$ or $\{w, \bar{w}\} \cap S = \emptyset$. We also have $\bar{v} \notin S$. As T connects v and $\bar{v}$, it can 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 $\{w, \bar{w}\} \cap S = \emptyset$ 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. □

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

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* $\mathtt{opt}_D \leq \mathtt{opt}_{\mathit{LC\text{-}D}}$.

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., $R(S) = R(\bar{S})$. 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 $w \in R$: $l_w + \sum_{S \in \mathcal{N}_w} y'_S > \mathtt{d}(w)$. Since $y'_S = 0$ for all non-Steiner cuts $S \in \mathcal{N}_w$ we therefore must have $l_w > \mathtt{d}(w) = c(P)/2$, 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

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

and this is a contradiction. □

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

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

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. The proof is omitted.

Lemma 4. *There exist instances for which* $\mathtt{opt}_R / \mathtt{opt}_{LC\text{-}D} = 2 - 2/(k+1)$.

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 by $\mathcal{A}_i$. 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 $1 \le i \le k$.

Lemma 5. *Suppose that there is an α-budget balanced cost sharing method for the Steiner tree game. Then, given an arbitrary player subset $\mathcal{T} \subseteq \mathcal{R}$, there also is an α-budget balanced cost sharing method with $\xi_{\mathcal{T}}(c) = \xi_{\mathcal{T}}(d)$ for all $c, d \in \mathcal{T} \cap \mathcal{A}_i$ and for all $1 \le i \le k$. Moreover, for all $c \in \mathcal{T} \cap \mathcal{A}_i$ and for all $d \in \mathcal{A}_i \setminus \mathcal{T}$, we may assume that $\xi_{\mathcal{T}}(c) = \xi_{(\mathcal{T} \setminus \{c\}) \cup \{d\}}(d)$.*

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 $\mathcal{T} \subseteq \mathcal{R}$. For a player $c \in \mathcal{A}_i \cap \mathcal{T}$ we let $a_c \in A_i$ be the (original) terminal associated with c. Now consider a permutation π_i of the players in class $\mathcal{A}_i$. The terminal of c under permutation π_i is $a_{\pi_i(c)}$.

Define the map π by letting $\pi(c) = \pi_i(c)$ 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. The cost sharing method ξ is then given by

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

for all $c \in \mathcal{R}$.

The important observation here is that the cost sharing method $\tilde{\xi}$ works on vertices. So if we swap the terminals associated with two players in $\mathcal{T}$, 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 set of terminals, but the players associated with these terminals may have changed.

Notice that, for a player $c \notin \mathcal{T}$, the value $\xi_{\mathcal{T}}(c)$ is 0 as $\pi(c) \notin \pi(\mathcal{T})$ for all $\pi \in \Pi$. It is now not difficult to see that, for all $1 \le i \le k$ and for any two players $c, d \in \mathcal{A}_i \cap \mathcal{T}$, we have $\xi_{\mathcal{T}}(c) = \xi_{\mathcal{T}}(d)$ as we average over all player permutations. It remains to show that ξ is cross-monotonic and α-budget balanced.

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

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

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. □

Now suppose we are given a symmetric cost sharing method ξ. 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 $\{a_1, \ldots, a_k\}$ where $a_i \in A_i$. By construction of the graph, all these terminals can connect to vertex $f_{\{a_1,\ldots,a_k\}}$ 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

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

Therefore there must be at least one index i such that $\xi_{\{a_1,\ldots,a_k\}}(a_i) \leq (k+3)/k$. By Lemma 5, this holds for every set $\{a_1, \ldots, a_{i-1}, c, a_{i+1}, \ldots, a_k\}$ 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 $Q := \{a_1, \ldots, a_k\} \cup A_i$. We bound the sum of the cost shares for this set as follows:

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

The inequality in (9) is due to cross-monotonicity: the cost share of a player over a subset of Q cannot 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 $\{a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_k\}$, 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 $\xi_{\{a_1,\ldots,a_{i-1},c,a_{i+1},\ldots,a_k\}}(c) \leq (k+3)/k$.

Due to the large amount of symmetry in the instance, we can in fact describe the optimal solution. The proof of the following lemma is omitted.

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

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

Proof (Theorem 2). 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:

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

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

References

1. Könemann, J., Leonardi, S., Schäfer, G.: A group-strategyproof mechanism for Steiner forests. In: Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms. (2005) 612–619
2. Jain, K., Vazirani, V.V.: Applications of approximation algorithms to cooperative games. In: Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing. (2001) 364–372
3. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-completeness. Freeman, San Francisco (1979)
4. Arora, S., Lund, C., Motwani, R., Sudan, M., Szegedy, M.: Proof verification and intractability of approximation problems. In: Proceedings of the Thirty-Third IEEE Symposium on Foundations of Computer Science. (1992) 210–214
5. Bern, M., Plassmann, P.: The Steiner problems with edge lengths 1 and 2. Information Processing Letters **32** (1989) 171–176
6. Agrawal, A., Klein, P., Ravi, R.: When trees collide: An approximation algorithm for the generalized Steiner problem in networks. SIAM Journal on Computing **24** (1995) 445–456
7. Goemans, M.X., Williamson, D.P.: A general approximation technique for constrained forest problems. SIAM Journal on Computing **24** (1995) 296–317
8. Moulin, H., Shenker, S.: Strategyproof sharing of submodular costs: budget balance versus efficiency. Economic Theory **18** (2001) 511–533
9. Pál, M., Tardos, É.: Group strategyproof mechanisms via primal-dual algorithms. In: Proceedings of the Forty-Fourth Annual IEEE Symposium on Foundations of Computer Science. (2003) 584–593
10. Immorlica, N., Mahdian, M., Mirrokni, V.S.: Limitations of cross-monotonic cost sharing schemes. In: Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms. (2005) 602–611
11. Aneja, Y.P.: An integer linear programming approach to the Steiner problem in graphs. Networks **10** (1980) 167–178