A Group-Strategyproof Mechanism for Steiner Forests*

Jochen Könemann† Stefano Leonardi‡ Guido Schäfer‡

Abstract

In this paper we design an approximately budget-balanced and group-strategyproof cost-sharing mechanism for the Steiner forest game. 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 a set $R \subseteq V \times V$ of k terminal pairs. Each terminal pair $(s, t) \in R$ is associated with an agent that wishes to establish a connection between nodes s and t in the underlying network. A feasible solution is a forest F that contains an s, t-path for each connection request $(s, t) \in R$.

Previously, Jain and Vazirani [4] gave a 2-approximate budget-balanced and group-strategyproof cost-sharing mechanism for the Steiner tree game — a special case of the game considered here. Such a result for Steiner forest games has proved to be elusive so far, in stark contrast to the well known primal-dual $(2-1/k)$-approximate algorithms [1, 2] for the problem.

The cost-sharing method presented in this paper is 2-approximate budget-balanced and this is tight with respect to the budget-balance factor.

Our algorithm is an original extension of known primal-dual methods for Steiner forests [1]. 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 fact the dual solution computed by our algorithm is infeasible but we can still prove that its total value is at most the cost of a minimum-cost Steiner forest for the given instance.

1 Introduction

In this paper we consider the problem of designing cost-sharing methods that are approximately budget-balanced and cross-monotonic.

Consider a set R of potential agents (or customers, players) that want to receive a common service, e.g., being connected to a network infrastructure. A *cost-sharing method* ξ is an algorithm that, given any subset $Q \subseteq R$ of agents, computes a solution to service Q and for each $j \in Q$ determines a non-negative cost-share $\xi_Q(j)$. The task is to devise a cost-sharing method ξ that is α*-approximate budget-balanced* and *cross-monotonic*:

α-Approximate Budget-Balance: The sum of the cost-shares recovers at least the total cost $c(Q)$ of the computed solution. Moreover, the sum of the cost-shares is at most $\alpha \geq 1$ times the cost $\texttt{opt}_Q$ of an optimum solution to service Q. That is,

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

Cross-Monotonicity: The cost-share of each individual agent never decreases as the set of agents shrinks, i.e.,

$$\forall Q' \subseteq Q,\ \forall j \in Q', \quad \xi_{Q'}(j) \geq \xi_Q(j).$$

A cost-sharing method is *budget-balanced* if $\alpha = 1$. Obtaining a budget-balanced cost-sharing method is computationally intractable if the underlying problem is NP-hard or if we additionally require cross-monotonicity (see for instance [3]).

Subsequently, we call a cost-sharing method that is α-approximate budget-balanced and cross-monotonic an α*-approximate cross-monotonic cost-sharing method* for short.

In this paper we consider the problem of devising an approximate cross-monotonic cost-sharing method for the *Steiner forest problem*. In this problem, we are given an undirected graph $G = (V, E)$, a non-negative cost function $c : E \to \mathbb{R}^+$ on the edges of G, and a set of $k > 0$ terminal pairs $R = \{(s_1, t_1), \ldots, (s_k, t_k)\} \subseteq V \times V$. Each terminal pair (s_j, t_j), $1 \leq j \leq k$, is associated with an autonomous agent that wants to establish a connection between nodes s_j and t_j. A feasible solution for terminal set R is a forest $F \subseteq E$ such that nodes s_j and t_j are in the same tree of F for all $1 \leq j \leq k$. The

*Partially supported by the EU within the 6th Framework Programme under contract 001907 (DELIS).

†Department of Combinatorics and Optimization, University of Waterloo, 200 University Avenue West, Waterloo, ON N2L 3G1, Canada. Email: jochen@uwaterloo.ca. This work was done while being on leave at the Dipartimento di Informatica e Sistemistica at Università di Roma "La Sapienza", Italy.

‡Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", Via Salaria 113, 00198 Roma, Italy. Email: {leon,schaefer}@dis.uniroma1.it.

objective is to find a feasible solution of smallest total cost.

The reason as to why we are interested in the design of cross-monotonic cost-sharing methods is due to a result of Moulin and Shenker [5]: Any (approximately) budget-balanced cross-monotonic cost-sharing method can be turned into an (approximately) budget-balanced group-strategyproof mechanism, i.e., a mechanism that encourages agents and coalitions of agents to reveal their true utility for receiving the service. (A more detailed description of the mechanism design problem that we consider is given below.)

Related Work. Though the designing of (approximate) cross-monotonic cost-sharing methods has recently received a growing attention in the computer science literature, such methods are known only for very few combinatorial problems: Moulin and Shenker [5] 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 several network design problems such as Steiner tree and facility location. Jain and Vazirani [4] presented a cross-monotonic cost-sharing method for the minimum spanning tree game and therefore a 2-approximate cross-monotonic cost-sharing method for the Steiner tree game. Pàl and Tardos [6] developed a 3-approximate cross-monotonic cost-sharing method for the facility location problem and a 15-approximate cross-monotonic cost-sharing method for the single-source rent-or-buy network design problem.

In most of the methods that were proposed so far to obtain (approximate) 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.

On the other hand, combinatorial problems that are well-behaved with respect to their approximability may prove hard when looking for approximate cross-monotonic cost-sharing methods. In a recent paper [3], Immorlica, Mahdian, and Mirrokni provide lower bounds on the budget-balance factor α 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. Observe that these lower bounds are achieved by using cross-monotonicity only.

Van Zwam [7] recently proved that for the Steiner tree game there does not exist an α-approximate cross-monotonic cost sharing method with $\alpha < 2$. This implies that the 2-approximate cost-sharing method for Steiner trees of Jain and Vazirani [4] is tight with respect to the budget-balance factor.

Our Contribution. While the 2-approximation achieved by primal-dual algorithms for the Steiner tree problem is matched by a 2-approximate cross-monotonic cost-sharing method [4], a similar result for the Steiner forest problem has proved to be elusive so far. This contrasts the optimization version of the problem where primal-dual $(2 - 1/k)$-approximation algorithms for both problems exist [1, 2].

In this paper we present a cross-monotonic cost-sharing method for the Steiner forest problem that is 2-approximate budget-balanced. The result of van Zwam [7] on Steiner trees shows that our cost-sharing method is tight with respect to the budget-balance factor.

Our algorithm is an original extension of the classical methods for Steiner forests [1, 2]. The Steiner forests produced by our algorithm are generally more expensive than those computed by the algorithms in [1, 2], since a terminal pair (s_j, t_j) will continue to contribute to construct the forest even after s_j and t_j are connected.

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 [1, 2]. In fact the dual solution computed by our algorithm is infeasible but we can still prove that its total value is at most the cost of a minimum-cost Steiner forest for the given instance. This raises the dazzling question of the existence of better primal-dual algorithms for Steiner trees and forests.

Mechanism Design Problem. The mechanism design problem that we consider can be described as follows. Consider a service provider whose set of potential agents (or customers) is R. Each agent j in R has a *utility* u_j which corresponds to the maximum prize j is willing to pay for the service. Moreover, each agent j makes a *bid* b_j for receiving the service. A *cost-sharing mechanism* is an algorithm that, based on the bids $\{b_j\}_{j \in R}$, (i) determines a set $Q \subseteq R$ of agents that receive the service, (ii) computes a solution to service the agents in Q, and (iii) for each $j \in Q$ fixes a *prize* x_j that j has to pay for receiving the service.

We define the *benefit* of an agent j to be $u_j - x_j$ if $j \in Q$, and zero otherwise. We assume that each agent is selfish and therefore may lie about the prize she is willing to pay so as to maximize her benefit. The task is to design a cost-sharing mechanism that encourages agents to bid their true utility: No agent or group of agents should be able to benefit from lying about their utilities. A cost-sharing mechanism is *strategyproof* if the dominant strategy of each agent is to bid her utility;

it is said to be *group-strategyproof* if the same holds even if agents collude.

Moulin and Shenker [5] showed that, given a cross-monotonic cost-sharing method ξ for the underlying problem, the following cost-sharing mechanism is group-strategyproof: Initially, let $Q = R$. If for each agent $j \in Q$ the cost share $\xi_Q(j)$ is less than or equal to her bid b_j, stop. Otherwise, remove from Q all agents whose cost shares are larger than their bids, and repeat. Jain and Vazirani [4] later extended this framework to approximately budget-balanced cross-monotonic cost-sharing methods.

Organization of the Paper. Our algorithm is based on a primal-dual algorithm for Steiner forests. We introduce the key ideas underlying this algorithm in the following section. Subsequently, we state our cross-monotonic algorithm for the Steiner forest problem in Section 3 and we analyze it in Sections 4 and 5. We comment on the approximation guarantee of our algorithm in Section 6.

2 A primal-dual Steiner forest algorithm

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

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

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

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

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

$$(2.1) \qquad \text{s.t.} \sum_{U \in \mathcal{U}: e \in \delta(U)} y_U \leq c_e \quad \forall e \in E$$

$$y \geq 0$$

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

1. Compute a feasible solution for the given Steiner forest instance. The algorithm reduces the degree of infeasibility as it progresses.
2. Create a dual feasible packing of sets of largest possible total value. The algorithm raises dual variables of certain subsets of nodes at all times. The final dual solution is going to be maximal in the sense that no single set can be raised without violating a constraint of type (2.1).

We think of an execution of algorithm AKR as a process over time and let x^t and y^t be the primal incidence vector and feasible dual solution at time t. We also use F^t to denote the forest corresponding to x^t. Initially, we let $x^0_e = 0$ for all $e \in E$ and $y^0_U = 0$ for $U \in \mathcal{U}$. In the following we say that an edge $e \in E$ is *tight* if the corresponding constraint (2.1) 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 U of $\bar{F}^t$ is *active* iff U separates at least one terminal pair, i.e., iff $U \in \mathcal{U}$. Let $\mathcal{A}^t$ be the set of all active connected components of $\bar{F}^t$ at time t. AKR raises the dual variables for all sets in $\mathcal{A}^t$ uniformly at all times $t \geq 0$.

Suppose now that two active connected components U_1 and U_2 *collide* at time t in the execution of AKR. In other words, there are terminals $u_1 \in U_1$ and $u_2 \in U_2$ such that a path between u_1 and u_2 becomes tight as a consequence of increasing y_{U_1} and y_{U_2}. If this happens, we add the path to F^t and continue. U_1 and U_2 are part of the same connected component of $\bar{F}^{t'}$ for $t' > t$.

Let T be a tree of the final forest F constructed by AKR. We define the age $\texttt{age}(T)$ of T to be the point of time at which T was formed, i.e., the first time t such that $T \subseteq F^t$. The following is the main theorem of [1]:

THEOREM 2.1. *Suppose that algorithm* AKR *outputs a forest F consisting of trees $T_1, \ldots, T_l$ and a feasible dual solution $\{y_U\}_{U \in \mathcal{U}}$. We then have*

$$c(F) \leq 2 \cdot \sum_{U \in \mathcal{U}} y_U - 2 \cdot \sum_{i=1}^{l} \texttt{age}(T_i) \leq \left(2 - \frac{1}{k}\right) \cdot \texttt{opt}_R,$$

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

3 A cross-monotonic algorithm for Steiner forests

In this section we use the ideas presented in the last section to develop a cross-monotonic algorithm for the Steiner forest problem. We refer to this algorithm by CSF throughout the remainder of this paper.

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

$$\mathtt{d}(s,t) = \frac{1}{2} \cdot c(s,t), \tag{3.2}$$

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 0-cost edges. We extend the death time notion to individual nodes and define $\mathtt{d}(r) = \mathtt{d}(s,t)$ for terminals $r,s,t \in R$ iff $r \in \{s,t\}$.

Recall from the last section that AKR raises only node-sets in $\mathcal{U}$ and, as a consequence, y^t is a feasible dual solution for (D) at all times $t \geq 0$. Algorithm CSF on the other hand will also raise subsets of V that do not separate terminal pairs.

Using the notation introduced in Section 2 we obtain CSF by modifying the definition of $\mathcal{A}^t$. We say that a connected component U of $\bar{F}^t$ is *active* at time t if it contains at least one terminal $r \in U$ with death time at least t, i.e., U is active iff there exists $r \in U$ with $\mathtt{d}(r) \geq t$. CSF grows all active connected components in $\mathcal{A}^t$ uniformly at all times $t \geq 0$.

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 $t \leq \mathtt{d}(s,t)$. At time $\mathtt{d}(s,t)$ the moats of s and t meet and a path connecting the terminals is added. In CSF 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. The death time of s and t is *independent* of other terminal pairs that are present. 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 $\mathtt{d}(s,t)$ in (3.2) implies that s and t end up in the same connected component of the final forest F and thus CSF constructs a feasible solution for the given Steiner forest instance.

We now detail the cost-share computation. For a terminal $r \in R$ and for $t \leq \mathtt{d}(r)$ we let $U^t(r)$ be the connected component in $\bar{F}^t$ that contains r. We also call $U^t(r)$ the *moat* around r at time t. Let $a^t(r)$ be the number of terminals in $U^t(r)$ whose death time is at least t. We then define the cost-share of terminal node $r \in R$ as

$$\xi_R(r) = 2 \cdot \int_{t=0}^{\mathtt{d}(r)} \frac{1}{a^t(r)}\, dt \tag{3.3}$$

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 Sections 4 and 5.

Theorem 3.1. *Algorithm* CSF *is a cross-monotonic cost-sharing method for the Steiner forest game that is* 2*-approximate budget-balanced.*

We let the final forest produced by CSF(R) be denoted by F and we use $\{y_U\}_{U \subseteq V}$ for the dual computed by our method.

4 Analysis: Proving cross-monotonicity

In order to prove cross-monotonicity of CSF 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 CSF(R) terminates at time t^* with forest F. Similarly, CSF(R_0) finishes at time t_0^* with a forest F_0. Moreover, for all times t we let $\mathcal{C}^t$ and $\mathcal{C}_0^t$ be the sets of connected components of $\bar{F}^t$ and of $\bar{F}_0^t$, respectively. The next lemma shows that $\mathcal{C}_0^t$ is a refinement of $\mathcal{C}^t$.

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

Proof. The proof is by induction on the time t. It is clear that the claim is true for $t = 0$ since $\mathcal{C}^0 = \mathcal{C}_0^0 = V$.

Consider a point in time $0 \leq t < t^*$ and assume the claim is true at time t. CSF(R_0) grows active sets in $\mathcal{C}_0^t$ and these are the only sets that can potentially violate the claim at any time $t + \epsilon$ for $\epsilon > 0$. Let $U_0 \in \mathcal{C}_0^t$ be an active set at time t in CSF(R_0), i.e., there exists a terminal $r \in U_0$ with $\mathtt{d}(r) \geq t$. From the induction hypothesis we know that there is a connected component U of F^t that contains U_0. Then U must be active in CSF(R) at time t and hence CSF(R) grows U at time t. The claim follows.

This claim immediately implies cross-monotonicity. Let $\xi(r)$ and $\xi_0(r)$ be the cost-share of terminal $r \in R_0$ in CSF(R) and in CSF(R_0), respectively.

COROLLARY 4.1. *Algorithm* CSF *is cross-monotonic, i.e., for each* $r \in R_0$ *we have*

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

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

$$\xi(r) = 2 \cdot \int_{t=0}^{\mathtt{d}(r)} \frac{1}{a^t(r)}\, dt \leq 2 \cdot \int_{t=0}^{\mathtt{d}(r)} \frac{1}{a_0^t(r)}\, dt = \xi_0(r)$$

for all $r \in R_0$ and the corollary follows.

5 Analysis: Competitiveness and cost-recovery

Recall that we let $\{y_U\}_{U\subseteq V}$ denote the dual solution computed by $\mathtt{CSF}(R)$ and we let F be the corresponding forest.

LEMMA 5.1. *Suppose that algorithm* CSF *outputs a forest* F *and a (possibly infeasible) dual solution* $\{y_U\}_{U\subseteq V}$*. We then have*

$$c(F) \leq 2 \cdot \sum_{U\subseteq V} y_U = \sum_{(s,t)\in R} \xi_R(s,t). \tag{5.4}$$

Proof. The proof of Theorem 2.1 implies that $c(F) \leq 2 \cdot \sum_{U\subseteq V} y_U$. Using Definition (3.3) it can then be seen that the cost-share sum on the right-hand side of (5.4) increases by 2ϵ whenever the total dual value increases by ϵ for some $\epsilon > 0$. Hence we must have $\sum_{(s,t)\in R} \xi_R(s,t) = 2 \cdot \sum_{U\subseteq V} y_U$.

This does *not* mean that the cost $c(F)$ of the forest F produced by our cost-sharing method is at most twice that of an optimum Steiner forest. In fact, $\{y_U\}_{U\subseteq V}$ is not a feasible solution for (D) since our algorithm raises duals for active sets that are not in $\mathcal{U}$. Surprisingly, we can however show that the total dual $\sum_{U\subseteq V} y_U$ is bounded by the cost $\mathtt{opt}_R$ of an optimum Steiner forest for the given instance on terminal set R.

LEMMA 5.2. *Let* y *be the (infeasible) dual computed by* $\mathtt{CSF}(R)$ *and let* $\mathtt{opt}_R$ *be the minimum-cost of any feasible Steiner forest for the given instance. We have*

$$\sum_{U\subseteq V} y_U \leq \mathtt{opt}_R.$$

Lemma 5.1 and 5.2 imply the following corollary on the approximate budget-balance of CSF.

COROLLARY 5.1. *Let* F *be the Steiner forest computed by* $\mathtt{CSF}(R)$*. We then have*

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

We now prove Lemma 5.2.

5.1 A proof of Lemma 5.2

Recall the definition of the death time $\mathtt{d}(s,t)$ of a terminal pair $(s,t) \in R$. In the following, let

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

such that

$$\mathtt{d}(s_1,t_1) \leq \ldots \leq \mathtt{d}(s_k,t_k).$$

We define a total order on the set of terminal nodes as follows. Let $u \in \{s_i,t_i\}$ and $v \in \{s_j,t_j\}$ for $i,j \in \{1,\ldots,k\}$ such that $u \neq v$. We then define $u \prec v$ if $i < j$ or if $i = j$ and $u = s_i$.

Let U^t be an active connected component in $\mathtt{CSF}(R)$ at some time $t \geq 0$. A terminal node $v \in U^t$ is *responsible* for the growth of U^t iff there does not exist a terminal $u \in U^t$ different from v with $v \prec u$. This way, each active moat in CSF has a unique responsible terminal node.

For a terminal node $v \in R$ and a time $t \geq 0$, let $r^t(v) = 1$ if v is responsible at time t, and 0 otherwise. We then define the *total responsibility time* of a terminal $v \in R$ as

$$r(v) = \int_{t=0}^{\mathtt{d}(v)} r^t(v)\, dt. \tag{5.5}$$

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

CLAIM 5.1. *Let* $v \in R$ *be a terminal and let* $r(v)$ *be its total responsibility time. Then, for any point of time* $0 \leq t < r(v)$*,* v *is responsible for* $U^t(v)$ *in* $\mathtt{CSF}(R)$*.*

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

Definition (5.5) also implies that

$$\sum_{U\subseteq V} y_U = \sum_{u\in R} r(u)$$

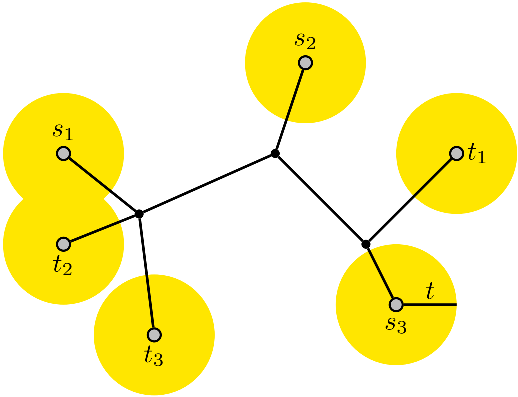

Figure 1: A tree T spanning terminals $R(T) = \{s_i, t_i\}_{1 \le i \le 3}$. The set of responsible terminal nodes at time t is $R^t(T) = \{s_1, s_2, s_3, t_1, t_3\}$ (where we assume that $\mathtt{d}(s1) > \mathtt{d}(t2)$). The corresponding moats in $U^t(T) = \{U^t(v)\}_{v \in R^t(T)}$ are pairwise disjoint and each moat loads at least one edge of T.

and hence it suffices to bound the sum on the right-hand side in order to prove Lemma 5.2.

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 $R(T) = \{v_1, \ldots, v_p\}$; see Figure 1. The idea is to bound the total growth of the moats of terminals in $R(T)$ by the cost $c(T)$ of the tree T spanning $R(T)$.

We let $R^t(T)$ be the set of terminal nodes in $R(T)$ that are responsible at time t, i.e.,

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

The following claim shows that at any time t the moats in

$$U^t(T) = \{U^t(v)\}_{v \in R^t(T)}$$

are pairwise disjoint.

Claim 5.2. *Consider a point of time t and two terminal nodes $v, u \in R^t(T)$, $v \neq u$. The two moats $U^t(v)$ and $U^t(u)$ must be disjoint.*

Proof. Assume for the sake of contradiction that $U^t(v)$ and $U^t(u)$ are not disjoint. Since both $U^t(v)$ and $U^t(u)$ are connected components of $\bar{F}^t$, it must therefore be the case that $U^t(v) = U^t(u)$. Claim 5.1 implies that both v and u are responsible for this moat and hence we must have $v = u$. This contradicts our choice of u and v.

Let $w \in R(T)$ be the terminal node with highest responsibility time. It is tempting to believe that w is the node with largest death time among nodes in $R(T)$ and that at time $t = r(w)$ all nodes in $R(T)$ are in the same connected component $U^t(w)$ of $\bar{F}^t$. However, this need not necessarily be true; see Figure 2.

Lemma 5.3. *If $\delta(U^t(w)) \cap T \neq \emptyset$ for all $0 \le t < r(w)$ then we must have $\sum_{v \in R(T)} r(v) \le c(T)$.*

Proof. Consider any point of time $t \ge 0$ where there are at least two terminals in $R(T)$ that are responsible, i.e., $|R^t(T)| > 1$. By Claim 5.2 we have that the moats in $U^t(T)$ are pairwise disjoint. On the other hand, the nodes in $R^t(T)$ are connected by T and hence, each of the moats in $U^t(T)$ loads a distinct part of the edges of T; see Figure 1.

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

In the following, let $\bar{w}$ be the *mate* of w, i.e., $(w, \bar{w}) \in R(T)$ and $\mathtt{d}(w) = \mathtt{d}(w, \bar{w})$. From now on we will assume that there is a time $t_0 \in [0, r(w))$ such that $\delta(U^{t_0}(w)) \cap T = \emptyset$ and hence $T \subseteq E(U^{t_0}(w))$, where $E(U^{t_0}(w))$ denotes the subset of those edges in E that have both endpoints in $U^{t_0}(w)$.

We also must have $|R^t(T)| = 1$ for all $t \in [t_0, r(w))$ since all nodes of $R(T)$ are in the same connected component of $\bar{F}^t$. Furthermore, since w is responsible until time $r(w)$ we must have $R^t(T) = \{w\}$ for all $t \in [t_0, r(w))$ and thus $u \prec w$ and $u \prec \bar{w}$ for all $u \in R(T) \setminus \{w, \bar{w}\}$.

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

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

Claim 5.3. *Consider a point in time t and a terminal $v \in I^t(T)$. Then $U^t(v)$ does not contain either w or $\bar{w}$.*

Proof. By definition of $I^t(T)$, 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 the growth of $U^t(v)$ and hence $\{w, \bar{w}\} \cap U^t(v) = \emptyset$.

For a time t and a node $v \in I^t(T)$, let $p^t_w(v)$ be the number of intersections of P_w and $U^t(v)$ at time t:

$$p^t_w(v) = |\delta(U^t(v)) \cap P_w|. \tag{5.6}$$

We use $\mathtt{sl}_w$ to denote the cost of that part of P_w that does not feel any dual load from any of the terminals

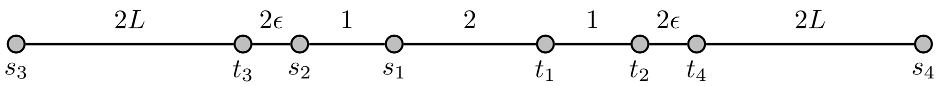

Figure 2: Let L be sufficiently large such that t_3 and t_4 are responsible terminals during the entire execution of CSF. The optimal solution contains the tree T spanning terminals $\{s_1, s_2, t_1, t_2\}$. We have $r(s_2) = r(t_2) = \epsilon$ and $r(s_1) = r(t_1) = \frac{1}{2}$. Moreover, $\mathtt{d}(s_2) = \mathtt{d}(t_2) = 2$ and $\mathtt{d}(s_1) = \mathtt{d}(t1) = 1$.

in $R(T)$. Let l_w and $l_{\bar{w}}$ be the total load on P_w coming from terminals w and $\bar{w}$, respectively. We can then express the cost of P_w as

$$c(P_w) = l_w + l_{\bar{w}} + \mathtt{sl}_w + \int_0^{t_0} \sum_{v \in I^t(T)} p_w^t(v)\, dt. \tag{5.7}$$

We obtain the following lemma.

Lemma 5.4. *If there is a $t_0 \in [0, r(w))$ with $\delta(U^{t_0}(w)) \cap T = \emptyset$ then we must have $\sum_{v \in R(T)} r(v) \leq c(T)$.*

Proof. Similar to the proof of Lemma 5.3, consider a time $t < r(w)$ where $R^t(T)$ contains more than one terminal. The corresponding moats in $U^t(T)$ are pairwise disjoint by Claim 5.2 and the nodes in $R^t(T)$ are connected by T. Hence, each of the moats in $U^t(T)$ loads a distinct part of T.

Moreover, using the definition of $p_w^t(v)$ in (5.6), for all $t \in [0, t_0)$ and $v \in I^t(T)$ moat $U^t(v)$ loads at least $p_w^t(v)$ edges of T.

Recall that $\mathtt{sl}_w$ is the cost of the segments of P_w that do not feel any load from terminals in $R(T)$. Furthermore, w loads edges of T until time t_0 and hence we must have

$$\begin{aligned} c(T) \geq \mathtt{sl}_w + \int_0^{t_0} \sum_{v \in I^t(T)} (p_w^t(v) - 1)\, dt \\ + t_0 + \sum_{v \in R(T) \setminus \{w\}} r(v) \end{aligned} \tag{5.8}$$

The death time of node w is at most half of the cost of P_w. Using (5.7) we therefore obtain

$$r(w) \leq \frac{l_w + l_{\bar{w}}}{2} + \frac{\mathtt{sl}_w}{2} + \frac{1}{2} \cdot \int_0^{t_0} \sum_{v \in I^t(T)} p_w^t(v)\, dt$$

$$\leq t_0 + \mathtt{sl}_w + \int_0^{t_0} \sum_{v \in I^t(T)} (p_w^t(v) - 1)\, dt, \tag{5.9}$$

where the second inequality uses the fact that $\max\{l_w, l_{\bar{w}}\} \leq t_0$ and that by Claim 5.3, $p_w^t(v) \geq 2$ for all $v \in I^t(T)$. Combining (5.8) and (5.9) yields the lemma.

We can now sum over all trees T in the forest F^*. Lemmas 5.3 and 5.4 imply that

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

This finishes the proof of Lemma 5.2.

6 Algorithmic consequences

In the previous section we have shown that the dual solution $\{y_U\}_{U \subseteq V}$ computed by our algorithm CSF, although being possibly infeasible, yields a lower bound on the optimum cost $\mathtt{opt}_R$:

$$\mathtt{opt}_R \geq \sum_{U \subseteq V} y_U.$$

Following the proof of Theorem 2.1 of Agrawal, Klein, and Ravi [1], we can use this fact and prove that our algorithm achieves the same approximation guarantee as the known primal-dual algorithms [1, 2]. This is surprising, since the forest constructed by our algorithm is usually costlier than those computed by the algorithms in [1, 2].

Let T be a tree in the final forest F constructed by CSF. We define the age $\mathtt{age}(T)$ of T to be the point of time at which the final moat that contains T stops growing, i.e., $\mathtt{age}(T) = \max\{r(v) : v \in R(T)\}$.

Theorem 6.1. *Suppose that algorithm CSF outputs a forest F consisting of trees $T_1, \ldots, T_l$ and a (possibly infeasible) dual solution $\{y_U\}_{U \subseteq V}$. We then have*

$$c(F) \leq 2 \cdot \sum_{U \subseteq V} y_U - 2 \cdot \sum_{i=1}^{l} \mathtt{age}(T_i) \leq \left(2 - \frac{1}{k}\right) \cdot \mathtt{opt}_R,$$

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

Since CSF also raises dual variables for node-sets that do not separate any terminal pair, one could hope that CSF always constructs a better lower bound than those obtained from the feasible dual solution of the algorithms in [1, 2].

In fact, depending on the underlying instance, CSF may yield a significantly stronger lower bound than the

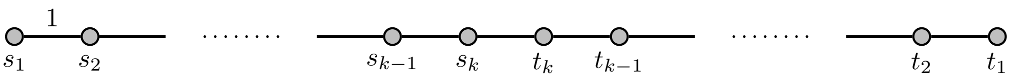

Figure 3: Chain with k terminal pairs $R = \{(s_i, t_i)\}_{1 \leq i \leq k}$ and unit edge costs.

one obtained from AKR. As an example, consider the instance given in Figure 3. The optimal cost $\texttt{opt}_R$ to connect all terminal pairs in R is $2k-1$. The total dual raised during the execution of AKR equals $2k \cdot \frac{1}{2} = k$, while the total dual of the solution constructed by CSF is $(2k-1) \cdot \frac{1}{2} + \frac{1}{2} c(s_1, t_1) = 2k - 1 = \texttt{opt}_R$. That is, for this particular instance, the lower bound of CSF proves optimality of the computed solution. Observe that this example also shows that the bound stated in Lemma 5.2 is tight.

On the other hand, it is an easy exercise to construct example instances on which the lower bounds of CSF and AKR are equally close to the optimum, or on which the lower bound of AKR is better than to the one of CSF.

An interesting observation is that CSF may even produce a solution whose total dual is strictly larger than the objective value of an *optimal* solution to the standard LP-relaxation (LP) on which the algorithms in [1, 2] are based. To see this, consider an even-length cycle $C = (v_0, \ldots, v_{n-1})$ on n nodes with unit edge costs and define $n-1$ terminal pairs $R = \{(v_0, v_i)\}_{1 \leq i \leq n-1}$. The total dual constructed by AKR is $\sum_{U \in \mathcal{U}} y_U = n/2$. Note that this is an optimal solution for the dual (D) of the standard LP-relaxation, since there exists a half-integral solution for (LP) having the same cost: set $x_e = \frac{1}{2}$ for each edge e of the cycle. The total dual constructed by CSF is

$$\sum_{U \subseteq V} y_U = \frac{1}{2} \cdot (n-1) + \frac{1}{2} \cdot \frac{n}{2} = \frac{3n}{4} - \frac{1}{2}.$$

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

A special case: Rooted Steiner tree games. The rooted Steiner tree game is a special case of the Steiner forest game. In the Steiner tree game, we are given a subset $R' \subseteq V$ of terminal nodes that want to be connected to a designated root node r; that is, agents correspond to nodes and the root node in particular is not part of the agents-set. A feasible solution is a tree that spans $R' \cup \{r\}$.

Jain and Vazirani [4] gave a 2-approximate cross-monotonic cost-sharing method for the Steiner tree game. Their method is based on a budget balanced cross-monotonic cost-sharing method for the minimum spanning tree game with a pre-specified root.

We can use algorithm CSF to obtain a 2-approximate budget-balanced cost-sharing mechanism ξ^{ST} for the Steiner tree game: Define the set of terminal pairs as $R = \{(r, v)\}_{v \in R'}$ and let algorithm CSF run on this instance. Recall that the root node r is not part of the agent-set in the Steiner tree game. We therefore define the cost-share of a terminal node $v \in R'$ as $\xi^{ST}_{R'}(v) = \xi_R(r, v)$. By Corollary 4.1 and Corollary 5.1, ξ^{ST} is a 2-approximate budget-balanced cross-monotonic cost-sharing method for the Steiner tree game.

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] M. X. Goemans and D. P. Williamson. A general approximation technique for constrained forest problems. *SIAM Journal on Computing*, 24:296–317, 1995.

[3] 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.

[4] 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.

[5] H. Moulin and S. Shenker. Strategyproof sharing of submodular costs: budget balance versus efficiency. *http://www.aciri.org/shenker/cost.ps*, 1997.

[6] 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.

[7] S. van Zwam. A lower bound on the cost recovery of the Steiner tree game with cross-monotonic cost shares. Technical Report 18-04, Dipartimento di Informatica e Sistemistica, Università di Roma “La Sapienza”, 2004.