

Optimal Cost-Sharing Mechanisms for Steiner Forest Problems

Shuchi Chawla¹, Tim Roughgarden², and Mukund Sundararajan²

¹ University of Wisconsin-Madison, Department of Computer Science, 210 W. Dayton Street, Madison, WI 53706, USA

² Stanford University, Department of Computer Science, 353 Serra Mall, Stanford, CA 94305, USA

Abstract. Könemann, Leonardi, and Schäfer [14] gave a 2-budget-balanced and groupstrategyproof mechanism for Steiner forest cost-sharing problems. We prove that this mechanism also achieves an $O(\log^2 k)$ -approximation of the social cost, where k is the number of players. As a consequence, the KLS mechanism has the smallest-possible worst-case efficiency loss, up to constant factors, among all $O(1)$ -budget-balanced Moulin mechanisms for such cost functions. We also extend our results to a more general network design problem.

1 Introduction

We study the design and analysis of cost-sharing mechanisms for fundamental network design problems. A *cost-sharing mechanism* is a protocol that collects bids for a service or good from potential users (players), chooses a subset of players to receive the service and a feasible way of servicing them, and determines prices to charge the chosen players. The mechanism incurs a subset-dependent cost $C(S)$ defined by a known cost function C . In this paper, we are interested in problems where players seek connectivity between a group of vertices, and where the cost $C(S)$ corresponds to the cost of providing such connectivity to the players in the set S .

A cost-sharing mechanism can be viewed as an auction in which any number of players can “win”, but in which the cost incurred by the auctioneer varies with the set of winners. The canonical problem of auctioning off a single good can be viewed as the special case in which the cost $C(S)$ is 0 if $|S| \leq 1$ and is $+\infty$ otherwise.

With more general cost functions, designing a mechanism requires choosing between several desirable but incompatible properties. As is standard, we insist on *incentive-compatibility*, meaning that players are motivated to bid their true private value v_i for receiving the service. We also require *budget-balance*, meaning that the mechanism recovers the incurred cost with the prices charged to the chosen players. Finally, we are interested in the social objective function of *efficiency*. Efficiency states that a set S should be chosen that trades off the cost $C(S)$ incurred and the valuations of the players in S in an optimal way.

Unfortunately, these three properties cannot be simultaneously achieved, even in very simple settings [8, 21].

This impossibility result motivated two distinct approaches to designing cost-sharing mechanisms. The first approach, taken by *VCG mechanisms* (see e.g. [17, 19]), ignores budget-balance. These mechanisms are optimally efficient and incentive compatible. They are typically not approximately budget-balanced for any reasonable approximation factor (see e.g. [5]).

The second approach, adopted in this paper, is to insist on incentive-compatibility and budget-balance, while regarding efficiency as a secondary objective. Moulin [18] introduced a class of mechanisms of this type. Researchers have developed approximately budget-balanced Moulin mechanisms for a number of different combinatorial optimization problems, including fixed-tree multicast [2, 5, 6]; submodular cost-sharing [18, 19]; Steiner tree [11–13]; Steiner forest [14, 15]; facility location [16, 20]; and rent-or-buy network design [9, 20]. The approach of Moulin [18] is the only known general technique for designing budget-balanced mechanisms with non-trivial costs.

Since Moulin mechanisms prioritize budget-balance over efficiency, nearly all previous papers that design Moulin mechanisms do not address the efficiency of the proposed mechanisms. Nevertheless, very recent work [23] shows that it is possible to discuss and compare the efficiency of Moulin mechanisms. Specifically, Roughgarden and Sundararajan [23] measured efficiency via the *social cost*, where the social cost of a set S is defined as the sum of the incurred service cost and the excluded valuations: $C(S) + \sum_{i \notin S} v_i$. This objective function is similar to the “prize-collecting” objectives that are commonly studied in approximation algorithms (see e.g. [7]). We call a mechanism α -*approximate* if it always outputs a solution with social cost at most an α factor times that of an optimal solution.

Roughgarden and Sundararajan [23] developed a framework to quantify the extent to which a Moulin mechanism minimizes the social cost. They applied this framework in [22, 23] to well-known mechanisms for submodular [19], facility location [20], Steiner tree [11], and single-sink rent-or-buy [9] cost functions. In particular, all of these mechanisms are both $O(\text{polylog}(k))$ -approximate, where k is the number of players, and are optimal (up to constant factors) among all Moulin mechanisms for the corresponding sets of cost functions.

A consequence of the main result in [23] is that only β -budget-balanced Moulin mechanisms can be β -approximate. The problem of designing Moulin mechanisms with good (say, polylogarithmic) budget-balance is itself highly non-trivial, and is provably impossible for several natural classes of cost functions [10]. Prior work [22, 23] resolved the approximate efficiency of all known Moulin mechanisms with good budget-balance save one: the elegant 2-budget-balanced Moulin mechanism for Steiner forest cost-sharing problems due to Könemann, Leonardi, and Schäfer [14]. We call this the *KLS mechanism*.

There are two reasons that analyzing the approximate efficiency of the KLS mechanism is technically challenging. First, Steiner forest cost functions, defined formally in Section 2, seem more complex than those treated in previous works [22, 23]. Indeed, the problem of designing an $O(1)$ -budget balanced

Moulin mechanism for such functions was a well-known open question for several years prior to the invention of the KLS mechanism. Second, the KLS mechanism itself possesses properties that make the analytic framework in [23] difficult to apply. In fact, prior to the present work, no non-trivial bound on the approximate efficiency of the KLS mechanism was known even when the input is restricted to Steiner tree cost functions. On the other hand, the Steiner tree cost-sharing mechanism due to Jain and Vazirani [11] is known to be $O(\log^2 k)$ -approximate [23]. The analysis of the Jain-Vazirani mechanism does not obviously carry over to the KLS mechanism (specialized to Steiner tree cost functions) since the latter mechanism, intuitively, charges players higher prices and therefore more aggressively discards them from the set S of winners. See Section 3.1 for a more technical discussion of this point.

In this paper, we overcome these difficulties and prove a tight upper bound on the approximate efficiency of the KLS mechanism. Specifically, in Section 3 we prove that the mechanism is $O(\log^2 k)$ -approximate, where k is the number of players. Previous work [22] shows that, even for the special case of Steiner tree cost functions, every $O(1)$ -budget-balanced Moulin mechanism is $\Omega(\log^2 k)$ -approximate. Thus the KLS mechanism has the smallest-possible worst-case efficiency loss, up to constant factors, among all such mechanisms for Steiner forest cost functions. We also extend our results to a more general network design problem (Section 4).

2 Preliminaries

Cost-Sharing Mechanisms. We consider cost functions C that assign a cost $C(S)$ to every subset S of a universe U of players and are defined implicitly via instances of network design problems. We also assume that every player $i \in U$ has a private, nonnegative *valuation* v_i for service. A *generalized Steiner tree (GST)* cost function is defined by a graph $G = (V, E)$, where each edge $e \in E$ possesses a nonnegative cost c_e , and by a set U of players, where each player $i \in U$ is identified with a subset $A_i \subseteq V$ of vertices called *terminals*. For a subset $S \subseteq U$ of players, the cost $C(S)$ is defined as the minimum cost of a subgraph of G that, for each $i \in S$, connects all of the vertices in A_i . A *Steiner forest (SF)* cost function is a special case of a GST function in which every group A_i contains only two terminals, a *source* s_i and a *sink* t_i .

A *cost-sharing mechanism* collects a nonnegative bid b_i from each player $i \in U$, selects a set $S \subseteq U$ of players, and charges every player i a price p_i . The mechanisms we consider also produce a feasible solution to the network design problem induced by the served set S , which has cost $C'(S)$ that in general is larger than the optimal cost $C(S)$. (Of course, evaluating $C(S)$ exactly is NP-hard.) We only allow mechanisms that are “individually rational” in the sense that $p_i = 0$ for players $i \notin S$ and $p_i \leq b_i$ for players $i \in S$. We also require that all prices are nonnegative (“no positive transfers”). As is standard, we assume that players have *quasilinear* utilities, meaning that each player i aims to maximize $u_i(S, p_i) = v_i x_i - p_i$, where $x_i = 1$ if $i \in S$ and $x_i = 0$ if $i \notin S$.

Our incentive-compatibility constraint is the well-known strategyproofness condition, which intuitively requires that a player cannot gain from misreporting its bid. Formally, a mechanism is *strategyproof (SP)* if for every player i , every bid vector b with $b_i = v_i$, and every bid vector b' with $b_j = b'_j$ for all $j \neq i$, $u_i(S, p_i) \geq u_i(S', p'_i)$, where (S, p) and (S', p') denote the outputs of the mechanism for the bid vectors b and b' , respectively.

For a parameter $\beta \geq 1$, a mechanism is β -*budget balanced* if $C(S)/\beta \leq \sum_{i \in S} p_i \leq C(S)$ for every outcome (S, p) of the mechanism. For a mechanism that outputs a feasible solution with cost $C'(S) \geq C(S)$, we require the stronger condition that $C'(S)/\beta \leq \sum_{i \in S} p_i \leq C(S)$. In particular, this requirement implies that the feasible solution produced by the mechanism has cost at most a β factor times that of optimal.

As discussed in the Introduction, we measure efficiency using the objective of *social cost minimization*. A cost-sharing mechanism is α -*approximate* if, assuming truthful bids, it always produces a solution with social cost at most an α factor times that of an optimal solution. Here, the social cost incurred by the mechanism is defined as the service cost $C'(S)$ of the feasible solution it produces for the network design instance corresponding to S , plus the sum $\sum_{i \notin S} v_i$ of the excluded valuations. Such a mechanism has two sources of inefficiency: first, it might choose a suboptimal set S of players to serve; second, it might produce a suboptimal solution to the network design instance induced by S .

Moulin Mechanisms and Cross-Monotonic Cost-Sharing Methods. Next we review *Moulin mechanisms*, a class of cost-sharing mechanisms that, for many cost functions, are SP, approximately budget-balanced, and approximately efficient. Such mechanisms are based on *cost sharing methods*, defined next.

A cost-sharing method χ is a function that assigns a non-negative *cost share* $\chi(i, S)$ for every subset $S \subseteq U$ of players and every player $i \in S$. A cost-sharing method is β -*budget balanced* for a cost function C and a parameter $\beta \geq 1$ if it always recovers β fraction of the cost: $C(S)/\beta \leq \sum_{i \in S} \chi(i, S) \leq C(S)$. We consider cost-sharing methods that, given a set S , produce both the cost shares $\chi(i, S)$ for all $i \in S$ and also a feasible solution for the network design problem induced by S . As above, we use the stronger condition $C'(S)/\beta \leq \sum_{i \in S} \chi(i, S) \leq C(S)$ for such methods, where $C'(S)$ is the cost of the produced feasible solution. A cost-sharing method is *cross-monotonic* if adding players to a set S only decreases the cost shares of players: for all $S \subseteq X \subseteq U$ and $i \in S$, $\chi(i, S) \geq \chi(i, X)$.

A cost-sharing method χ for C defines the following *Moulin mechanism* M_χ for C . First, collect a bid b_i for each player i . Initialize the set S to all of U and invoke the cost-sharing method χ to define a feasible solution to the network design problem induced by S and a price $p_i = \chi(i, S)$ for each player i . If $p_i \leq b_i$ for all $i \in S$, then halt, output the set S , the corresponding network design solution, and charge prices p . If $p_i > b_i$ for some player $i \in S$, then remove an arbitrary such player from the set S and iterate. A Moulin mechanism based on a cross-monotonic cost-sharing method thus simulates an iterative ascending auction, with the method χ suggesting prices for the remaining players at each

iteration. Note that if χ produces a feasible solution in polynomial time, then so does M_χ . Also, M_χ clearly inherits the budget-balance factor of χ . Finally, Moulin [18] proved the following.

Theorem 1 ([18]). *If χ is a cross-monotonic cost-sharing method, then the corresponding Moulin mechanism M_χ is strategyproof.*

Theorem 1 reduces the problem of designing an SP³, β -budget-balanced mechanism to that of designing a cross-monotonic, β -budget-balanced cost-sharing method.

Summability and Approximate Efficiency. Roughgarden and Sundararajan [23] showed that the approximate efficiency of a Moulin mechanism is completely controlled by its budget-balance and one additional parameter of its underlying cost-sharing method. We define this parameter next.

Definition 1 ([23]). *Let C and χ be a cost function and a cost-sharing method, respectively, defined on a common universe U of players. The method χ is α -summable for C if*

$$\sum_{\ell=1}^{|S|} \chi(i_\ell, S_\ell) \leq \alpha \cdot C(S)$$

for every ordering σ of U and every set $S \subseteq U$, where S_ℓ and i_ℓ denote the set of the first ℓ players of S and the ℓ th player of S (with respect to σ), respectively.

We next summarize the main result in [23].

Theorem 2 ([23]). *Let U be a universe of players and C a nondecreasing cost function on U with $C(\emptyset) = 0$. Let M be a Moulin mechanism for C with underlying cost-sharing method χ . Let $\alpha \geq 0$ and $\beta \geq 1$ be the smallest numbers such that χ is α -summable and β -budget-balanced. Then the mechanism M is $(\alpha + \beta)$ -approximate and no better than $\max\{\alpha, \beta\}$ -approximate.*

In particular, an $O(1)$ -budget-balanced Moulin mechanism is $\Theta(\alpha)$ -approximate if and only if the underlying cost-sharing method is $\Theta(\alpha)$ -summable.

The KLS cost-sharing method. Könemann, Leonardi and Schäfer [14] devised cross-monotonic, 2-budget-balanced cost-sharing methods for all Steiner forest cost functions. The cost-sharing method is based on a variant of the primal-dual method. By Theorem 1, this yields 2-budget-balanced and GSP mechanisms for all such functions. Due to space constraints, we refer the reader to [14] for a description of the KLS cost-sharing method; its details are important primarily for Sections 3.3 and 4.

³ Moulin mechanisms satisfy a stronger notion of incentive compatibility called group-strategyproofness (GSP), which is a form of collusion resistance. Almost all known GSP cost-sharing mechanisms are Moulin mechanisms (see [10, 18, 19]).

3 The Efficiency of the KLS Mechanism

We now analyze the efficiency of the KLS mechanism. Our main result is the following.

Theorem 3. *For every Steiner forest cost function with k players, the KLS cost-sharing method is $O(\log^2 k)$ -summable.*

Since the KLS cost-sharing method is 2-budget-balanced, Theorems 2 and 3 immediately give a guarantee on the approximate efficiency of the KLS mechanism.

Corollary 1. *For every Steiner forest cost function with k players, the KLS mechanism is $O(\log^2 k)$ -approximate.*

Since every $O(1)$ -budget-balanced Moulin mechanism for Steiner tree cost functions is $\Omega(\log^2 k)$ -approximate [22], the KLS mechanism is an optimal mechanism of this type (up to constant factors).

3.1 Overview of the Proof of Theorem 3

This section provides an overview of our analysis. By the definition of summability (Definition 1), proving Theorem 3 requires analyzing the following procedure. Given an arbitrary Steiner forest instance and an arbitrary ordering of the players (source-sink pairs), we consider adding the players to the instance one-by-one, according to the given ordering. Each time we add a new player, we recompute the KLS cost shares using the KLS primal-dual algorithm and consider the cost share of the most recently added player. The key question is: by how much can the sum of these successive cost shares exceed the cost of servicing all of the players?

Our analysis proceeds in two steps. The first step is motivated by the difficulty in directly bounding the above successive cost shares in a general network. The idea of this step is to replace the given network by a forest with cost at most an $O(\log k)$ times that of an optimal Steiner forest. In addition, to facilitate our charging argument in the second step, we require that each tree of this forest be an ultrametric—i.e. all root-leaf paths have equal length. While this goal is reminiscent of probabilistic tree embeddings (see e.g. [3, 4]), we cannot apply such an embedding as a black box. The reason is that our charging argument requires structure beyond the low distortion guarantee—it also needs the distances in the ultrametric to be tightly coupled with the dual growth process used to define the KLS cost-shares.

In the second step, we demonstrate how to charge the k successive KLS cost shares to the ultrametries constructed in the first step. Loosely speaking, we show how subtrees in each ultrametric correspond to active components during the execution of the primal-dual algorithm that defines the KLS cost shares. Our charging scheme charges each point of each ultrametric $O(\log k)$ times, proving an $O(\log^2 k)$ bound on the summability of the KLS cost-sharing method.

While portions of this argument are similar to that used in [23] to upper bound the summability of the Jain-Vazirani Steiner tree cost-sharing method [11],

the refined ultrametric structure and the charging argument in this paper are new. One reason we require the ultrametric structure is that the primal-dual algorithm underlying the KLS mechanism determines cost shares using fixed “death times”, rather than via the component structure in the dual growth process. While crucial for cross-monotonicity, this property can cause a terminal to accumulate a cost share beyond the point at which it is connected to its mate, and it is not obvious how to bound this additional accumulation. In fact, we can exhibit an example with k players for which the summability of the KLS method is an $\Omega(\log k)$ factor times larger than that of the Jain-Vazirani method. Nonetheless, we prove in this section that the KLS method is always $O(\log^2 k)$ -summable, matching the (tight) worst-case bound for the Jain-Vazirani method.

3.2 Building the Forest

In the first step of our proof of Theorem 3, we define a procedure with the following properties. The procedure takes as input a Steiner forest instance $G = (V, E)$ with edge costs and an (adversarial) ordering σ of the source-sink pairs $(s_1, t_1), \dots, (s_k, t_k)$. It constructs a forest F , defined on the terminals, that has cost $O(\log k)$ times that of a minimum-cost Steiner forest, as well as other desirable structure. While the following description will be algorithmic, we emphasize that this construction is purely for the purposes of analyzing the summability of the KLS cost-sharing method.

Consider an optimal solution to the given Steiner forest instance. Our forest F will have one tree for each connected component of this optimal solution. We will construct these trees independently of each other, so we can restrict our description to a single component T^* of the optimal Steiner forest. Let A^* denote the terminals spanned by T^* . The vertex set of the tree T that we construct will contain all the terminals in A^* as well as some auxiliary vertices.

We now describe the construction of T . The ordering $\sigma = (s_1, t_1), \dots, (s_k, t_k)$ on source-sink pairs induces an ordering $s_1, t_1, s_2, t_2, \dots, s_k, t_k$ on the terminals and also an ordering of A^* . We construct T by adding terminals in A^* to it in this order. When a terminal is considered, we attach it to the existing tree and endow it with a *radius*. The *ball* of a terminal x with radius r is defined as the terminals of A^* at distance at most r from x in the given graph G . We begin with the first terminal (say x_1) of A^* , which is given an infinite radius. For technical reasons, we introduce an auxiliary root x_0 and create an edge e_0 between x_0 and x_1 of length D_{max} , where D_{max} is half the largest distance in G between two terminals of A^* . We call this edge e_0 the *backbone edge*.

Now consider some subsequent terminal x . Among all of the previously added terminals whose ball contains x , we define the terminal y with the minimum radius to be the *parent* of x and write $p(x) = y$. If y has finite radius—i.e., is not the first terminal of A^* with respect to σ —then we define x ’s radius r_x to be half of its parent’s radius. Otherwise, we define the radius r_x to be half of the shortest-path distance between x and y in G . To attach x to the tree T , consider the path from y to x_0 in T . We connect x to the point along this path at a distance r_x from y , possibly creating a new internal node. The backback edge

and the definition of D_{max} ensure that this is always possible. Call this point $v(x)$. The length of the edge between $v(x)$ and x is defined to be r_x . We assign this new node $v(x)$ a label with value y ; this label plays a role in our subsequent proofs.

We next prove several facts about this construction. We begin with an easy lemma.

Lemma 1. *The backbone edge e_0 has length at most $c(T^*)$, where T^* is the component of the optimal solution that spans A^* .*

Using this lemma and arguments similar to those in [23] for a related tree construction, we can bound the sum of edge costs in T^* .

Lemma 2. *The sum of the costs of the edges in T is $O(\log k) \cdot c(T^*)$.*

We next study distances between terminals in the tree T . We begin by noting that our construction does indeed produce an ultrametric.

Lemma 3. *The tree T is an ultrametric, with all root-leaf paths having length D_{max} . Moreover, the leaves of T are in bijective correspondence with the terminals A^* .*

Lemma 3 follows from an easy induction. In particular, when a new terminal x is added to the tree T , the distance from $v(x)$ to x equals the distance from $v(x)$ to $p(x)$.

For every two terminals x, y in A^* , let $d_T(x, y)$ and $d_G(x, y)$ denote the distances between x and y in the tree T and in the graph G , respectively. The next lemma follows immediately from the construction.

Lemma 4. *For every terminal $x \in A^*$ with parent $p(x)$, $d_T(x, p(x)) = 2r_x = r_{p(x)} \geq d_G(x, p(x))$.*

We next extend Lemma 4 to every pair $x, y \in A^*$ of terminals x, y . The idea is to consider a walk W_{xy} between x and y in T and relate the length of this walk to both $d_T(x, y)$ and $d_G(x, y)$.

Precisely, fix $x, y \in A^*$ and consider the (unique) path P_{xy} between x and y in the tree T . The length of this path is $d_T(x, y)$. To construct the walk W_{xy} , consider the sequence S_{xy} of vertices that the path P_{xy} visits; apart from x and y , all of these are internal nodes of T . Obtain a sequence S'_{xy} of terminals from S_{xy} by replacing the internal nodes of S_{xy} by their label values (terminals) and then removing duplicates. Obtain the walk W_{xy} by visiting the terminal nodes in S'_{xy} in order, along the unique paths in T that connect nodes. The walk W_{xy} contains P_{xy} as a subgraph, and may be decomposed into P_{xy} and a set of circuits, each of which starts and ends at an internal node of P_{xy} , visiting the terminal node corresponding to the label of the internal node along the way.

Let ℓ_{xy} denote the length of this walk. We prove the following three lemmas, with the third an immediate consequence of the first two.

Lemma 5. *For every pair $x, y \in A^*$ of terminals in T , $\ell_{x,y} \geq d_G(x, y)$.*

Lemma 6. *For every pair $x, y \in A^*$ of terminals in T , $d_T(x, y) \geq \ell_{x, y}/5$.*

Lemma 7. *For every pair $x, y \in A^*$ of terminals in T , $d_T(x, y) \geq d_G(x, y)/5$.*

Lemma 5 follows from Lemma 4 and the fact that consecutive nodes in S'_{xy} share a parent-child relationship. The idea of the proof of Lemma 6 is as follows. Consider one circuit of W_{xy} rooted at the internal point $v(z)$ and visiting the terminal $p(z)$. The length of the circuit is at most $2r_z$, while the length of the segment of P_{xy} immediately preceding $v(z)$ is at least $r_z/2$ (r_z minus the radius of any of z 's children). Therefore, we can charge the length of every circuit to 4 times the section of P_{xy} that immediately precedes it in the walk W_{xy} .

For technical reasons, we multiply all of the edge costs of T by 10, yielding the tree T' . The following is just a restatement of Lemmas 2, 3, and 7.

Lemma 8. *T' satisfies the following properties:*

- (a) *The cost of T' is $O(\log k) \cdot c(T^*)$.*
- (b) *T' is an ultrametric, with the terminals of A^* appearing only as leaves.*
- (c) *For every pair $x, y \in A^*$ in T' , $d_G(x, y) \leq \frac{1}{2}d_{T'}(x, y)$.*

3.3 The Charging Argument

We are now ready to bound the summability of the KLS cost-share. Our charging argument will proceed independently for each ultrametric constructed in Sections 3.2; for most of this section, we will fix one such ultrametric T , spanning a set A^* of terminals.

Let x_ℓ and A_ℓ denote the ℓ th terminal and the first ℓ terminals of A^* , respectively, with respect to the ordering induced by σ .

We aim to charge the KLS cost share $\chi^{KLS}(x_\ell, A_\ell)$ of a terminal $x_\ell \in A^*$ to points of the tree T . (A technical detail: since matched pairs of terminals appear consecutively in the ordering induced by σ , the set A_ℓ contains only matched pairs of terminals, plus possibly an orphaned source s_i . In either case, $\chi^{KLS}(x_\ell, A_\ell)$ denotes the KLS cost share assigned to the terminal x_ℓ in the Steiner forest instance induced by all of the players with at least one terminal in the set A_ℓ .)

The charging proceeds as follows. Let P_ℓ be the unique path in T from x_ℓ to x_0 , and consider the primal-dual algorithm that assigns the KLS cost share $\chi^{KLS}(x_\ell, A_\ell)$. At each moment in time τ up to the death time of x_ℓ , the terminal's cost share increases at a positive rate, equal to the inverse of the number of active terminals in x_ℓ 's component at time τ . For each such time τ , we charge this (marginal) increment in x_ℓ 's cost-share to the point $g_\ell(\tau)$ which is at distance τ from x_ℓ along the path P_ℓ .

Since every leaf-root path of T has length at least D_{max} (Lemma 8(b))—half of the largest distance between two terminals of A^* —and since D_{max} is at least the death time of every terminal of A^* , this procedure fully charges the sum $\sum_\ell \chi^{KLS}(x_\ell, A_\ell)$ of the KLS cost shares to T .

We now claim that for every point g of the tree T , the sum of the (marginal) charges to g by the terminals of A^* is only $O(\log k)$. Fix a point g of T . Only terminals in the subtree of T rooted at g charge part of their cost share to g . By the ultrametric property (Lemma 8(b)), all of these terminals are equidistant from the point g in T ; let this common distance be τ_g . Such a terminal charges part of its cost share to g if and only if its death time is at least τ_g ; let B denote these terminals.

Using Lemma 8(c) we now show that, for a terminal $x_\ell \in B$, at time τ_g in the run of the primal-dual algorithm that defines the KLS cost share $\chi^{KLS}(x_\ell, A_\ell)$, the component containing x_ℓ also contains all of the terminals of $B \cap A_\ell$.

Lemma 9. *If $x, x' \in B$, then $d_G(x, x') \leq \tau_g$.*

Lemma 10. *Suppose $x_\ell \in B$ and $x \in A_\ell \cap B$. Then at time τ_g in the run of the primal-dual algorithm that defines the KLS cost share $\chi^{KLS}(x_\ell, A_\ell)$, the terminal x is active and lies in the same component as x_ℓ .*

Since the KLS cost-sharing method splits the increase in value of an active dual variable equally among the active terminals contained in the corresponding component, Lemma 10 implies that the marginal charge to the point g by the terminal $x_\ell \in B$ is at most $1/|B \cap A_\ell|$. Summing over the contributions of the terminals in B , we obtain the following.

Lemma 11. *For every point g of T , the total marginal charge to g is at most $\mathcal{H}_{|B|}$, where $\mathcal{H}_j = \sum_{i \leq j} 1/i$ denotes the j th Harmonic number.*

Theorem 3 now follows easily from Lemmas 8(a) and 11.

4 A Generalized Steiner Tree Mechanism

We now briefly consider an extension of the Steiner forest problem, deferring a detailed discussion to the full version. We consider a problem in which each player i controls a group A_i of terminals, and is interested in connecting all of these terminals together. This problem is also called the *generalized Steiner tree (GST)* problem [1]. Let k and n denote the number of terminal groups (players) and terminals, respectively. The Steiner forest problem is the special case where each group contains only two terminals.

Consider the following naive reduction to the Steiner forest problem. For each group of terminals corresponding to a player, nominate one of these terminals as a leader. Create terminal pairs by pairing each terminal in the group with the leader. Invoke the KLS cost-sharing method on these terminal pairs, and define the cost share of a player to be the sum of the cost shares assigned to its corresponding terminal pairs. Cross-monotonicity and 2-budget-balance are straightforward to establish, and Theorem 3 then implies that there is an $O(\log^2 n)$ -approximate Moulin mechanism for all GST cost functions.

We can improve the above approximation factor by changing the way that the KLS primal-dual algorithm splits the increase in active dual variables between the active terminals. Specifically, if we modify the algorithm to split each

such increase equally between the *players* that have at least one active terminal in the corresponding dual variable, rather than equally among the terminals themselves, then we obtain the following theorem.

Theorem 4. *Every GST cost function with k players and n terminals admits a 2-budget-balanced, $O(\log n \log k)$ -approximate Moulin mechanism.*

Using techniques from [22], we can show that the bound in Theorem 4 is the best possible for a $O(1)$ -budget-balanced Moulin mechanism for GST cost functions.

References

1. A. Agrawal, P. Klein, and R. Ravi. When trees collide: an approximation algorithm for the generalized Steiner problem on networks. *SIAM Journal on Computing*, 24(3):440–456, 1995.
2. A. Archer, J. Feigenbaum, A. Krishnamurthy, R. Sami, and S. Shenker. Approximation and collusion in multicast cost sharing. *Games and Economic Behavior*, 47(1):36–71, 2004.
3. Y. Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 184–193, 1996.
4. J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *Proceedings of the 35th Annual ACM Symposium on the Theory of Computing (STOC)*, 2003.
5. J. Feigenbaum, A. Krishnamurthy, R. Sami, and S. Shenker. Hardness results for multicast cost sharing. *Theoretical Computer Science*, 304:215–236, 2003.
6. J. Feigenbaum, C. Papadimitriou, and S. Shenker. Sharing the cost of multicast transmissions. *Journal of Computer and System Sciences*, 63(1):21–41, 2001.
7. M. X. Goemans and D. P. Williamson. A general approximation technique for constrained forest problems. *SIAM Journal on Computing*, 24(2):296–317, 1995.
8. J. Green, E. Kohlberg, and J. J. Laffont. Partial equilibrium approach to the free rider problem. *Journal of Public Economics*, 6:375–394, 1976.
9. A. Gupta, A. Srinivasan, and É. Tardos. Cost-sharing mechanisms for network design. In *Proceedings of the 7th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, volume 3122 of *Lecture Notes in Computer Science*, pages 139–150, 2004.
10. N. Immorlica, M. Mahdian, and V. S. Mirrokni. Limitations of cross-monotonic cost-sharing schemes. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 602–611, 2005.
11. K. Jain and V. Vazirani. Applications of approximation algorithms to cooperative games. In *Proceedings of the 33rd Annual ACM Symposium on the Theory of Computing (STOC)*, pages 364–372, 2001.
12. K. Jain and V. Vazirani. Equitable cost allocations via primal-dual-type algorithms. In *Proceedings of the 34th Annual ACM Symposium on the Theory of Computing (STOC)*, pages 313–321, 2002.
13. K. Kent and D. Skorin-Kapov. Population monotonic cost allocation on mst’s. In *Operational Research Proceedings KOI*, pages 43–48, 1996.

14. J. Könemann, S. Leonardi, and G. Schäfer. A group-strategyproof mechanism for Steiner forests. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 612–619, 2005.
15. J. Könemann, S. Leonardi, G. Schäfer, and S. van Zwam. From primal-dual to cost shares and back: A stronger LP relaxation for the steiner forest problem. In *Proceedings of the 32nd Annual International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 3580 of *Lecture Notes in Computer Science*, pages 1051–1063, 2005.
16. S. Leonardi and G. Schäfer. Cross-monotonic cost-sharing methods for connected facility location. In *Proceedings of the Fifth ACM Conference on Electronic Commerce (EC)*, pages 242–243, 2004.
17. A. Mas-Colell, M. D. Whinston, and J. R. Green. *Microeconomic Theory*. Oxford University Press, 1995.
18. H. Moulin. Incremental cost sharing: Characterization by coalition strategy-proofness. *Social Choice and Welfare*, 16:279–320, 1999.
19. H. Moulin and S. Shenker. Strategyproof sharing of submodular costs: Budget balance versus efficiency. *Economic Theory*, 18:511–533, 2001.
20. M. Pál and É. Tardos. Group strategyproof mechanisms via primal-dual algorithms. In *Proceedings of the 44th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 584–593, 2003.
21. K. Roberts. The characterization of implementable choice rules. In J. J. Laffont, editor, *Aggregation and Revelation of Preferences*. North-Holland, 1979.
22. T. Roughgarden and M. Sundararajan. Approximately efficient cost-sharing mechanisms. Submitted, 2006.
23. T. Roughgarden and M. Sundararajan. New trade-offs in cost-sharing mechanisms. In *Proceedings of the 38th Annual ACM Symposium on the Theory of Computing (STOC)*, pages 79–88, 2006.