Optimal Efficiency Guarantees for Network Design Mechanisms*

Tim Roughgarden**[1] and Mukund Sundararajan***[1]

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

Abstract. A cost-sharing problem is defined by a set of players vying to receive some good or service, and a cost function describing the cost incurred by the auctioneer as a function of the set of winners. A cost-sharing mechanism is a protocol that decides which players win the auction and at what prices. Three desirable but provably mutually incompatible properties of a cost-sharing mechanism are: incentive-compatibility, meaning that players are motivated to bid their true private value for receiving the good; budget-balance, meaning that the mechanism recovers its incurred cost with the prices charged; and efficiency, meaning that the cost incurred and the value to the players served are traded off in an optimal way.

Our work is motivated by the following fundamental question: for which cost-sharing problems are incentive-compatible mechanisms with good approximate budget-balance and efficiency possible? We focus on cost functions defined implicitly by NP-hard combinatorial optimization problems, including the metric uncapacitated facility location problem, the Steiner tree problem, and rent-or-buy network design problems. For facility location and rent-or-buy network design, we establish for the first time that approximate budget-balance and efficiency are simultaneously possible. For the Steiner tree problem, where such a guarantee was previously known, we prove a new, optimal lower bound on the approximate efficiency achievable by the wide and natural class of "Moulin mechanisms". This lower bound exposes a latent approximation hierarchy among different cost-sharing problems.

1 Introduction

Mechanism Design. In the past decade, there has been a proliferation of large systems used and operated by independent agents with competing objectives (most notably the Internet). Motivated by such applications, an increasing amount

---

* Preliminary versions of most of these results appear in a technical report [32].

** Supported in part by ONR grant N00014-04-1-0725, an NSF CAREER Award, and an Alfred P. Sloan Fellowship.

*** Supported in part by OSD/ONR CIP/SW URI "Software Quality and Infrastructure Protection for Diffuse Computing" through ONR Grant N00014-01-1-0795 and by OSD/ONR CIP/SW URI "Trustworthy Infrastructure, Mechanisms, and Experimentation for Diffuse Computing" through ONR Grant N00014-04-1-0725.

of algorithm design research studies optimization problems that involve self-interested entities. Naturally, game theory and economics are important for modeling and solving such problems. *Mechanism design* is a classical area of microeconomics that has been particularly influential. The field of mechanism design studies how to solve optimization problems in which part of the problem data is known only to self-interested players. It has numerous applications to, for example, auction design, pricing problems, and network protocol design [8, 15, 24, 27].

Selling a single good to one of n potential buyers is a paradigmatic problem in mechanism design. Each bidder i has a *valuation* v_i, expressing its maximum willingness to pay for the good. We assume that this value is known only to the bidder, and not to the auctioneer. A *mechanism* (or *auction*) for selling a single good is a protocol that determines the winner and the selling price. Each bidder i is "selfish" in the sense that it wants to maximize its "net gain" $(v_i - p)x_i$ from the auction, where p is the price, and x_i is 1 (0) if the bidder wins (loses).

What optimization problem underlies a single-good auction? One natural goal is *economic efficiency*, which in this context demands that the good is sold to the bidder with the highest valuation. This goal is trivial to accomplish if the valuations are known a priori. Can it be achieved when the valuations are private?

Vickrey [34] provided an elegant solution. First, each player submits a sealed bid b_i to the seller, which is a proxy for its true valuation v_i. Second, the seller awards the good to the highest bidder. This achieves the efficient allocation *if* we can be sure that players bid their true valuations—if $b_i = v_i$ for every i. To encourage players to bid truthfully, we must charge the winner a non-zero price. (Otherwise, all players will bid gargantuan amounts in an effort to be the highest.) On the other hand, if we charge the winning player its bid, it encourages players to underbid. (Bidding your maximum willingness to pay ensures a net gain of zero, win or lose.) Vickrey [34] suggested charging the winner the value of the *second-highest* bid, and proved that this price transforms truthful bidding into an optimal strategy for each bidder, independent of the bids of the other players. In turn, the Vickrey auction is guaranteed to produce an efficient allocation of the good, provided all players bid in the obvious, optimal way.

Cost-Sharing Mechanisms. Economic efficiency is not the only important objective in mechanism design. *Revenue* is a second obvious concern, especially in settings where the mechanism designer incurs a non-trivial cost. This cost can represent production costs, or more generally some revenue target.

A *cost-sharing problem* is defined by a set U of players vying to receive some good or service, and a cost function $C : 2^U \rightarrow \mathcal{R}^+$ describing the cost incurred by the mechanism as a function of the auction outcome—the set S of winners. We assume that $C(S)$ is nonnegative for every set $S \subseteq U$, that $C(\emptyset) = 0$, and that C is nondecreasing ($S \subseteq T$ implies $C(S) \leq C(T)$). Note that there is no explicit limit on the number of auction winners, although a large number of winners might result in extremely large costs. With outcome-dependent costs, the *efficient allocation* is the one that maximizes the *social* *welfare* $W(S) = \sum_{i \in S} v_i - C(S)$—the outcome that trades offs the valuations of the winners and the cost incurred in an optimal way. The problem of selling a single good can be viewed as the special case in which $C(S) = 0$ if $|S| \leq 1$ and $C(S) = +\infty$ otherwise.

In this paper, we focus on cost functions that are defined implicitly by an instance of a combinatorial optimization problem. For example, U could represent a set of potential clients, located in an undirected graph with fixed edge costs, that want connectivity to a server r [7, 17]. In this application, $C(S)$ denotes the cost of connecting the terminals in S to r—the cost of the minimum-cost Steiner tree that spans $S \cup \{r\}$.

A *cost-sharing mechanism*, given a set U and a function C, is a protocol that decides which players win the auction and at what prices. Typically, such a mechanism is also (perhaps approximately) *budget-balanced*, meaning that the cost incurred is passed on to the auction's winners. Budget-balanced cost-sharing mechanisms provide control over the revenue generated, relative to the cost incurred by the mechanism designer.

Summarizing, we have identified three natural goals in auction and mechanism design: (1) *incentive-compatibility*, meaning that every player's optimal strategy is to bid its true private value v_i for receiving the service; (2) *budget-balance*, meaning that the mechanism recovers its incurred cost with the prices charged; and (3) *efficiency*, meaning that the cost and valuations are traded off in an optimal way.

Unfortunately, properties (1)–(3) cannot be simultaneously achieved, even in very simple settings [10, 30]. This impossibility result motivates relaxing at least one of the these properties. Until recently, nearly all work in cost-sharing mechanism design completely ignored either budget-balance or efficiency. If the budget balance constraint is discarded, then there is an extremely powerful and flexible mechanism that is incentive-compatible and efficient: the *VCG mechanism* (see e.g. [26]). This mechanism specializes to the Vickrey auction in the case of selling a single good, but is far more general. Since the VCG mechanism is typically not approximately budget-balanced for any reasonable approximation factor (see e.g. [6]), it is not suitable for many applications.

The second approach is to insist on incentive-compatibility and budget-balance, while regarding efficiency as a secondary objective. The only general technique for designing mechanisms of this type is due to Moulin [25]. Over the past five years, researchers have developed approximately budget-balanced Moulin mechanisms for cost-sharing problems arising from numerous different combinatorial optimization problems, including fixed-tree multicast [1, 6, 7]; the more general submodular cost-sharing problem [25, 26]; Steiner tree [17, 18, 20]; Steiner forest [21, 22]; facility location [23, 29]; rent-or-buy network design [14, 29], and various covering problems [5, 16]. Most of these mechanisms are based on novel primal-dual approximation algorithms for the corresponding optimization problem. With one exception discussed below, none of these works provided any guarantees on the efficiency achieved by the proposed mechanisms.

Approximately Efficient Cost-Sharing Mechanisms. Impossibility results are, of course, common in optimization. From conditional impossibility results like Cook's Theorem to information-theoretic lower bounds in restricted models of computation, as with online and streaming algorithms, algorithm designers are accustomed to devising heuristics and proving worst-case guarantees about them using approximation measures. This approach can be applied equally well to cost-sharing mechanism design, and allows us to quantify the inevitable efficiency loss in incentive-compatible, budget-balanced cost-sharing mechanisms. As worst-case approximation measures are rarely used in economics, this research direction has only recently been pursued.

Moulin and Shenker [26] were the first to propose quantifying the efficiency loss in budget-balanced Moulin mechanisms. They studied an additive notion of efficiency loss for submodular cost functions. This notion is useful for ranking different mechanisms according to their worst-case efficiency loss, but does not imply bounds on the quality of a mechanism's outcome relative to that of an optimal outcome. A more recent paper [31] provides an analytical framework for proving approximation guarantees on the efficiency attained by Moulin mechanisms. The present paper builds on this framework. (See [4, 11] for other very recent applications.)

Several definitions of approximate efficiency are possible, and the choice of definition is important for quantifying the inefficiency of Moulin mechanisms. Feigenbaum et al. [6] showed that, even for extremely simple cost functions, budget-balance and social welfare cannot be simultaneously approximated to within any non-trivial factor. This negative approximation result is characteristic of mixed-sign objective functions such as welfare.

An alternative formulation of exact efficiency is to choose a subset minimizing the *social cost*, where the social cost $\pi(S)$ of a set S is the sum of the incurred service cost and the excluded valuations: $C(S) + \sum_{i \notin S} v_i$. Since $\pi(S) = -W(S) + \sum_{i \in U} v_i$ for every set S, where U denotes the set of all players, a subset maximizes the social welfare if and only if it minimizes the social cost. The two functions are not, of course, equivalent from the viewpoint of approximation. Similar transformations have been used for "prize-collecting" problems in combinatorial optimization (see e.g. [3]). We call a cost-sharing mechanism *α-approximate* if it always produces an outcome with social cost at most an α factor times that of an optimal outcome. Also, a mechanism is β-budget-balanced if the sum of prices charged is always at most the cost incurred and at least a $1/\beta$ fraction of this cost.

Previous work [31] demonstrated that $O(\text{polylog}(k))$-approximate, $O(1)$-budget-balanced Moulin mechanisms exist for two important types of cost-sharing problems: submodular cost functions, and Steiner tree cost functions. (Here k denotes the number of players.) This was the first evidence that properties (1)–(3) above can be approximately simultaneously satisfied, and motivates the following fundamental question: *which cost-sharing problems admit incentive-compatible mechanisms that are approximately budget-balanced and approximately efficient?*

Our Results. This paper presents three contributions. We first consider metric uncapacitated facility location (UFL) cost-sharing problems, where the input is a UFL instance, the players U are the demands of this instance, and the cost $C(S)$ is defined as the cost of an optimal solution to the UFL sub-instance induced by S. The only known $O(1)$-budget-balanced Moulin mechanism for this problem is due to Pál and Tardos [29] (the *PT mechanism*). The PT mechanism is 3-budget-balanced [29], and no Moulin mechanism for the problem has better budget balance [16]. We provide the first efficiency guarantee for the PT mechanism by proving that it is $O(\log k)$-approximate, where k is the number of players. Simple examples show that every $O(1)$-budget-balanced Moulin mechanism for UFL is $\Omega(\log k)$-approximate. Thus the PT mechanism simultaneously optimizes both budget balance and efficiency over the class of Moulin mechanisms for UFL.

Second, we design and analyze Moulin mechanisms for rent-or-buy network design cost-sharing problems. For example, the single-sink rent-or-buy (SSRoB) problem is a generalization of the Steiner tree problem in which several source vertices of a network (corresponding to the players U) want to simultaneously send one unit of flow each to a common root vertex. For a subset $S \subseteq U$ of players, the cost $C(S)$ is defined as the minimum-cost way of installing sufficient capacity for the players of S to simultaneously send flow to the root. Capacity on an edge can be rented on a per-unit basis, or an infinite amount of capacity can be bought for M times the per-unit renting cost, where $M \geq 1$ is a parameter. (Steiner tree is the special case where $M = 1$.) Thus the SSRoB problem is a simple model of capacity installation in which costs obey economies of scale. The multicommodity rent-or-buy (MRoB) problem is the generalization of SSRoB in which each player corresponds to a source-sink vertex pair, and different players can have different sink vertices.

Gupta, Srinivasan, and Tardos [14] and Leonardi and Schäfer [23] independently showed how to combine the SSRoB algorithm of [13] with the Jain-Vazirani Steiner tree mechanism [17] to obtain an $O(1)$-budget-balanced SSRoB mechanism. (Earlier, Pál and Tardos [29] designed an $O(1)$-budget-balanced SSRoB mechanism, but it was more complicated and its budget balance factor was larger.) We note that the mechanism design ideas in [14, 23], in conjunction with the recent 2-budget-balanced Steiner forest mechanism due to Könemann, Leonardi, and Schäfer [21], lead to an $O(1)$-budget-balanced MRoB mechanism. Much more importantly, we prove that this SSRoB mechanism and a variant of this MRoB mechanism are $O(\log^2 k)$-approximate, the first efficiency guarantees for any approximately budget-balanced mechanisms for these problems. Our third result below implies that these are the best-achievable efficiency guarantees for $O(1)$-budget-balanced Moulin mechanisms for these problems.

Third, we prove a new lower bound that exposes a non-trivial, latent hierarchy among different cost-sharing problems. Specifically, we prove that every $O(1)$-budget-balanced Moulin mechanism for Steiner tree cost functions is $\Omega(\log^2 k)$-approximate. This lower bound trivially also applies to Steiner forest, SSRoB, and MRoB cost functions.

This lower bound establishes a previously unobservable separation between submodular and facility location cost-sharing problems on the one hand, and the above network design cost-sharing problems on the other. All admit $O(1)$-budget-balanced Moulin mechanisms, but the worst-case efficiency loss of Moulin mechanisms is provably larger in the second class of problems than in the first one.

All previous lower bounds on the efficiency of Moulin mechanisms were derived from either budget-balance lower bounds or, as for the problems considered in this paper, from a trivial example equivalent to a cost-sharing problem in a single-link network [31]. This type of example cannot prove a lower bound larger than the kth Harmonic number $\mathcal{H}_k = \Theta(\log k)$ on the approximate efficiency of a Moulin mechanism. We obtain the stronger bound of $\Omega(\log^2 k)$ by a significantly more intricate construction that exploits the complexity of Steiner tree cost functions.

2 Preliminaries

Cost-Sharing Mechanisms. We consider a cost function C that assigns a cost $C(S)$ to every subset S of a universe U of players. We assume that C is nonnegative and nondecreasing (i.e., $S \subseteq T$ implies $C(S) \leq C(T)$). We sometimes refer to $C(S)$ as the *service cost*, to distinguish it from the social cost (defined below). We also assume that every player $i \in U$ has a private, nonnegative *valuation* v_i.

A *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. In this paper, we focus on cost functions that are defined implicitly as the optimal solution of an instance of a (NP-hard) combinatorial optimization problem. The mechanisms we consider also produce a feasible solution to the optimization problem induced by the served set S, which has cost $C'(S)$ that in general is larger than the optimal cost $C(S)$.

We also impose the following standard restrictions and assumptions. 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 require that all prices are nonnegative ("no positive transfers"). Finally, 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 (set S, prices p, feasible solution with service cost $C'(S)$) of the mechanism. In particular, this requirement implies that the feasible solution produced by the mechanism has cost at most β times that of optimal.

As discussed in the Introduction, 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 instance corresponding to S, plus the sum $\sum_{i \notin S} v_i$ of the excluded valuations. The optimal social cost is $\min_{S \subseteq U}[C(S) + \sum_{i \notin S} v_i]$. A mechanism thus 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 optimization problem induced by S.

Moulin Mechanisms and Cross-Monotonic Cost-Sharing Methods. Next we review *Moulin mechanisms*, the preeminent class of SP, approximately budget-balanced mechanisms. 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 $i \in S$. 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 optimization problem induced by S. A cost-sharing method is β-*budget balanced* for a cost function C and a parameter $\beta \geq 1$ if it always recovers a $1/\beta$ fraction of the cost: $C'(S)/\beta \leq \sum_{i \in S} \chi(i, S) \leq C(S)$, where $C'(S)$ is the cost of the produced feasible solution. A cost-sharing method is *cross-monotonic* if the cost share of a player only increases as other players are removed: for all $S \subseteq T \subseteq U$ and $i \in S$, $\chi(i, S) \geq \chi(i, T)$.

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 optimization problem induced by S and a price $p_i = \chi(i, S)$ for each player $i \in S$. If $p_i \leq b_i$ for all $i \in S$, then halt, output the set S, the corresponding feasible 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 cost-sharing method thus simulates an iterative auction, with the method χ suggesting prices for the remaining players at each iteration. The cross-monotonicity constraint ensures that the simulated auction is ascending, in the sense that the prices that are compared to a player's bid are only increasing with time. 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 [25] proved the following.

Theorem 1 ([25]). *If χ is a cross-monotonic cost-sharing method, then the corresponding Moulin mechanism M_χ is strategyproof.*[1]

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

[1] Moulin mechanisms also satisfy a stronger notion of incentive compatibility called groupstrategyproofness (GSP), which is a form of collusion resistance [26].

Summability and Approximate Efficiency. Roughgarden and Sundararajan [31] 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 and the precise guarantee next.

Definition 1 (Summability [31]). *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 ℓth player of S (with respect to σ), respectively.

Theorem 2 ([31]). *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. Analyzing the summability of a cost-sharing method, while non-trivial, is a tractable problem in many important cases. Because summability is defined as the accrued cost over a worst-case "insertion order" of the players, summability bounds are often reminiscent of performance analyses of online algorithms.

3 An Optimal Facility Location Cost-Sharing Mechanism

In this section we consider the metric uncapacitated facility location (UFL) problem.[2] The input is given by a set U of demands (the players), a set F of facilities, an opening cost f_q for each facility $q \in F$, and a metric c defined on $U \cup F$. The cost $C(S)$ of a subset $S \subseteq U$ of players is then defined as the cost of an optimal solution to the UFL problem induced by S. In other words, $C(S) = \min_{\emptyset \neq F^* \subseteq F} [\sum_{q \in F^*} f_q + \sum_{i \in S} \min_{q \in F^*} c(q,i)]$. We seek an $O(1)$-budget-balanced Moulin mechanism for UFL with the best-possible approximate efficiency. Theorems 1 and 2 reduce this goal to the problem of designing an $O(1)$-budget-balanced cross-monotonic cost-sharing method with the smallest-possible summability.

We begin with a simple lower bound, similar to that given in [31] for submodular cost-sharing problems.

Proposition 1 (Lower Bound on UFL Approximate Efficiency). *For every $k \geq 1$, there is a k-player UFL cost function C with the following property: for every $\beta \geq 1$ and every β-budget-balanced Moulin mechanism M for C, M is no better than $\mathcal{H}_k/\beta$-approximate.*

[2] Due to space constraints, we omit all proofs. Details are in [32].

Pál and Tardos [29] showed that every UFL cost function admits a 3-budget-balanced cross-monotonic cost-sharing method χ_{PT}. We call this the *PT method*, and the induced Moulin mechanism the *PT mechanism*. (See [29] or [32] for details.) Our main result in this section shows that the PT mechanism matches the lower bound in Proposition 1, up to a constant factor.

Theorem 3 (Upper Bound on PT Summabilitity). *Let C be a k-player UFL cost function and χ_{PT} the corresponding PT method. Then χ_{PT} is $\mathcal{H}_k$-summable for C.*

Applying Theorem 2 yields an efficiency guarantee for the PT mechanism.

Corollary 1 (Upper Bound on PT Approximate Efficiency). *Let C be a k-player UFL cost function and M_{PT} the corresponding PT mechanism. Then M_{PT} is $(\mathcal{H}_k + 3)$-approximate.*

Theorem 3 follows from two lemmas. The first states that single-facility instances supply worst-case examples for the summability of the PT method.

Lemma 1. *For every $k \geq 1$, the summability of PT methods for k-player UFL cost functions is maximized by the cost functions that correspond to single-facility instances.*

Lemma 1 is based on a monotonicity property that we prove for the PT method: increasing the distance between a demand and a facility can only increase cost shares. This monotonicity property allows us to argue that in worst-case UFL instances, players are partitioned into non-interacting groups, each clustered around one facility. We complete the proof of Lemma 1 by arguing that the summability of the PT method for one of these single-facility clusters in at least that in the original facility location instance.

Our second lemma bounds the summability of PT methods in single-facility instances.

Lemma 2. *Let C be a k-player UFL cost function corresponding to a single-facility instance. If χ_{PT} is the corresponding PT method, then χ_{PT} is $\mathcal{H}_k$-summable for C.*

4 Optimal Rent-or-Buy Cost-Sharing Mechanisms

Single-Sink Rent-or-Buy: Next we consider single-sink rent-or-buy (SSRoB) cost-sharing problems. The input is given by a graph $G = (V, E)$ with edge costs that satisfy the Triangle Inequality, a root vertex t, a set U of demands (the players), each of which is located at a vertex of G, and a parameter $M \geq 1$. A feasible solution to the SSRoB problem induced by S is a way of installing sufficient capacity on the edges of G so that every player in S can simultaneously route one unit of flow to t. Installing x units of capacity on an edge e costs $c_e \cdot \min\{x, M\}$; the parameter M can be interpreted as the ratio between the cost of "buying" infinite capacity for a flat fee and the cost of "renting" a single unit of capacity. The cost $C(S)$ of a subset $S \subseteq U$ of players is then defined as the cost of an optimal solution to the SSRoB problem induced by S. We sometimes abuse notation and use $i \in U$ to denote both a player and the vertex of G that hosts the player.

Gupta, Srinivasan, and Tardos [14] and Leonardi and Schäfer [23] independently designed the following $O(1)$-budget-balanced cross-monotonic cost-sharing method for SSRoB, which we call the *GST method*. Given an SSRoB cost function and a set $S \subseteq U$ of players, we use the randomized algorithm of [13] to produce a feasible solution. This algorithm first chooses a random subset $D \subseteq S$ by adding each player $i \in S$ to D independently with probability $1/M$. Second, it computes an approximate Steiner tree spanning $D \cup \{t\}$ using, for example, the 2-approximate MST heuristic [33], and buys infinite capacity on all of the edges of this tree. Third, for each player $i \notin D$, it rents one unit of capacity for exclusive use by i on a shortest path from its vertex to the closest vertex in $D \cup \{t\}$. This defines a feasible solution with probability 1, and the expected cost of this solution at most 4 times that of an optimal solution to the SSRoB instance induced by S [13].

The GST cost share $\chi_{GST}(i, S)$ is defined as the expectation of the following random variable X_i, over the random choice of the set D in the above algorithm: if $i \notin D$, then X_i equals one quarter of the length of the shortest path used to connect i to a vertex in $D \cup \{t\}$; if $i \in D$, then X_i equals $M/2$ times the Jain-Vazirani cost share $\chi_{JV}(i, D)$ of i with respect to the Steiner tree instance defined by G, c, t, and the players D (see [17] for the details of χ_{JV}). These cost shares are 4-budget-balanced with respect to the optimal cost of the SSRoB instance induced by S, as well as the expected cost of the above randomized algorithm that produces a feasible solution to this instance. We prove the following result.

Theorem 4. *For every k-player SSRoB cost function, the corresponding GST mechanism is $O(\log^2 k)$-approximate.*

Theorem 6 below implies that this is the best efficiency guarantee possible for an $O(1)$-budget-balanced SSRoB Moulin mechanism.

With an eye toward extending Theorem 4 to the MRoB problem, we summarize very briefly the main steps in the proof (details are in [32]). First, we decompose each GST cost share $\chi_{GST}(i, S)$ into two terms, a term $\chi_{buy}(i, S)$ for the contributions of samples $D \subseteq S$ in which $i \in D$, and a term $\chi_{rent}(i, S)$ for the contributions of the remaining samples. Proving Theorem 4 reduces to proving that both χ_{buy} and χ_{rent} are $O(\log^2 k)$-summable. Second, we use the $O(\log^2 k)$-summability of χ_{JV} [31] together with a counting argument inspired by [13, 19] to prove that χ_{buy} is $O(\log^2 k)$-summable. Third, we prove that the cost-sharing method χ_{JV} is $O(1)$-*strict* in the sense of [12]. This roughly means that whenever a player i is included in the random sample D, then the cost share $\chi_{JV}(i, D)$ is at least a constant factor times the cost share it would have received had it not been included.[3] We leverage the strictness of χ_{JV} to prove that the summability of χ_{rent} is at most a constant times that of χ_{buy}.

Multicommodity Rent-or-Buy: We next extend Theorem 4 to the MRoB problem, where each player i corresponds to a vertex pair (s_i, t_i). (All other aspects of the problem are the same.) The high-level approach is similar, but the technical challenges are much more formidable. In the proof of Theorem 4, the Jain-Vazirani cost-sharing method χ_{JV} played a heroic role: it is cross-monotonic, which is necessary for the GST cost-sharing method to be cross-monotonic; it is $O(\log^2 k)$-summable, which is necessary for χ_{buy} to be $O(\log^2 k)$-summable; and it is $O(1)$-strict in the sense of [12] with respect to the MST heuristic for Steiner tree, which is necessary for χ_{rent} to be $O(\log^2 k)$-summable. Is there a comparably all-purpose cost-sharing method for the *Steiner Forest* problem—the problem of finding the min-cost subgraph of a given graph that includes a path between every given vertex pair (s_i, t_i)? The only known cross-monotonic cost-sharing method χ_{KLS} for Steiner Forest cost-sharing problems was recently given by Könemann, Leonardi, and Schäfer [21]. This method is defined by a primal-dual algorithm; the cost shares are a natural byproduct of a dual growth process, and the primal is a 2-approximate feasible solution to the given Steiner Forest instance. Using the ideas in [9, 12, 14, 23], these facts suffice to define an $O(1)$-budget-balanced Moulin mechanism for MRoB cost-sharing problems. Moreover, the KLS method was very recently shown to be $O(\log^2 k)$-summable [4]; thus, the corresponding cost-sharing method χ_{buy} is $O(\log^2 k)$-summable. Unfortunately, the KLS cost-sharing method is $\Omega(k)$-strict with respect to the corresponding primal solution [12], which precludes bounding the summability of χ_{rent} in terms of χ_{buy}. While several strict cost-sharing methods are known for different Steiner Forest approximation algorithms [2, 9, 12, 28], none of these are cross-monotonic methods.

Our high-level approach is to modify the above composition of the KLS method with the mechanism design techniques of [14, 23] in a way that achieves $O(1)$-strictness while sacrificing only a small constant factor in the budget balance. Similar ideas have been used previously to obtain strictness guarantees for other Steiner forest algorithms [2, 12, 28].

Theorem 5. *Every k-player MRoB cost function admits an $O(1)$-budget-balanced, $O(\log^2 k)$-approximate Moulin mechanism.*

5 An $\Omega(\log^2 k)$ Lower Bound for Steiner Tree Problems

An instance of the *Steiner tree cost-sharing problem* [17] is given by an undirected graph $G = (V, E)$ with a root vertex t and nonnegative edge costs, with each player of U located at some vertex of G. For a subset $S \subseteq U$, the cost $C(S)$ is defined as that of a minimum-cost subgraph of G that spans all of the players of

[3] Formally, strictness of a cost-sharing method is defined with respect to some primal algorithm; see [12] for a precise definition.

S as well as the root t. There are $O(1)$-budget-balanced, $O(\log^2 k)$-approximate Moulin mechanisms for such problems [4, 17, 21, 31]. The main result of this section is a matching lower bound on the approximate efficiency of every $O(1)$-budget-balanced Moulin mechanism.

Theorem 6. *There is a constant $c > 0$ such that for every constant $\beta \geq 1$, every β-budget-balanced Moulin mechanism for Steiner tree cost-sharing problems is at least $(\beta^{-1} c \log^2 k)$-approximate, where k is the number of players served in an optimal outcome.*

Theorem 6 implies that Steiner tree cost-sharing problems and their generalizations are fundamentally more difficult for Moulin mechanisms than facility location (Theorem 3) and submodular cost-sharing problems (see [31]).

We now outline the proof of Theorem 6. Fix values for the parameters $k \geq 2$ and $\beta \geq 1$. We construct a sequence of networks, culminating in G. The network G_0 consists of a set V_0 of two nodes connected by an edge of cost 1. One of these is the root t. The player set U_0 is $\sqrt{k}$ players that are co-located at the non-root node. (Assume for simplicity that k is a power of 4.) For $j > 0$, we obtain the network G_j from G_{j-1} by replacing each edge (v, w) of G_{j-1} with m internally disjoint two-hop paths between v and w, where m is a sufficiently large function of k of β. (We will choose $m \geq 8\beta\sqrt{k} \cdot (2\beta)^{\sqrt{k}}$.) See Figure 1. The cost of each of these $2m$ edges is half of the cost of the edge (v, w). Thus every edge in G_j has cost 2^{-j}.

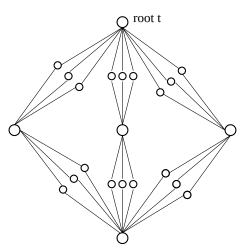

Fig. 1. Network G_2 in the proof of Theorem 6, with $m = 3$. All edges have length 1/4.

Let V_j denote the vertices of G_j that are not also present in G_{j-1}. We augment the universe by placing $\sqrt{k}$ new co-located players at each vertex of V_j; denote these new players by U_j. The final network G is then G_p, where $p = (\log k)/2$. Let $V = V_0 \cup \cdots \cup V_p$ and $U = U_0 \cup \cdots \cup U_p$ denote the corresponding vertex and player sets. Let C denote the corresponding Steiner tree cost function.

Now fix $\beta \geq 1$ and an arbitrary cross-monotonic, β-budget balanced Steiner tree cost-sharing method χ. By Theorem 2, we can prove Theorem 6 by exhibiting a subset $S \subseteq U$ of size k and an ordering σ of the players of S such that $\sum_{\ell=1}^{k} \chi(i_\ell, S_\ell) \geq (c \log^2 k/\beta) \cdot C(S)$, where i_ℓ and S_ℓ denote the ℓth player and the first ℓ players with respect to σ.

We construct the set S iteratively. For $j = 0, 1, \dots, p$, we will identify a subset $S_j \subseteq U_j$ of players; the set S will then be $S_0 \cup \dots \cup S_p$. Recall that U_j consists of groups of $\sqrt{k}$ players, each co-located at a vertex of V_j, with m such groups for each edge of G_{j-1}. The set S_j will consist of zero or one such group of $\sqrt{k}$ players for each edge of G_{j-1}.

The set S_0 is defined to be U_0. For $j > 0$, suppose that we have already defined $S_0, \dots, S_{j-1}$. Call a vertex $v \in V_0 \cup \dots \cup V_{j-1}$ *active* if v is the root t or if the $\sqrt{k}$ players co-located at v were included in the set $S_0 \cup \dots \cup S_{j-1}$. Call an edge (v, w) of G_{j-1} *active* if both of its endpoints are active and *inactive* otherwise.

To define S_j, we consider each edge (v, w) of G_{j-1} in an arbitrary order. Each such edge gives rise to m groups of $\sqrt{k}$ co-located players in G_j. If (v, w) is inactive in G_{j-1}, then none of these $m\sqrt{k}$ players are included in S_j. If (v, w) is active in G_{j-1}, then we will choose precisely one of the m groups of players, and will include these $\sqrt{k}$ co-located players in S_j. We first state two lemmas that hold independently of how this choice is made; we then elaborate on our criteria for choosing groups of players.

Lemma 3. *For every $j \in 1, 2, \dots, p$, $|S_j| = 2^{j-1}\sqrt{k}$. Also, $|S_0| = \sqrt{k}$.*

Lemma 3 implies that $|S| = \sqrt{k}(1 + \sum_{j=0}^{p-1} 2^j) = k$. The next lemma states that our construction maintains the invariant that the players selected in the first j iterations lie "on a straight line" in G.

Lemma 4. *For every $j \in 0, 1, \dots, p$, $C(S_0 \cup \dots \cup S_j) = 1$.*

Lemmas 3 and 4 both follow from straightforward inductions on j.

We now explain how to choose one out of the m groups of co-located players that arise from an active edge. Fix an iteration $j > 0$ and let $\hat{S}$ denote the set of players selected in previous iterations $(S_0, \dots, S_{j-1})$ and previously in the current iteration. Let (v, w) be the active edge of G_{j-1} under consideration and $A_1, \dots, A_m \subseteq U_j$ the corresponding groups of co-located players. We call the group A_r *good* if the $\sqrt{k}$ players of A_r can be ordered $i_1, i_2, \dots, i_{\sqrt{k}}$ so that

$$\chi(i_\ell, \hat{S} \cup \{i_1, \dots, i_\ell\}) \geq \frac{1}{4\beta} \cdot \frac{2^{-j}}{\ell} \tag{1}$$

for every $\ell \in \{1, 2, \dots, \sqrt{k}\}$. We then include an arbitrary good group A_r in the set S_j. See [32] for a proof of the following lemma.

Lemma 5. *Provided m is a sufficiently large function of k and β, for every $j \in \{1, \dots, p\}$, every ordering of the active edges of G_{j-1}, and every edge (v, w) in this ordering, at least one of the m groups of players of U_j that corresponds to (v, w) is good. Also, the group S_0 is good.*

We conclude by using the lemma to finish the proof of Theorem 6.

We have already defined the subset $S \subseteq U$ of players. We define the ordering σ of the players in S as follows. First, for all $j \in \{1, \dots, p\}$, all players of S_{j-1}

precede all players of S_j in σ. Second, for each $j \in \{1, \ldots, p\}$, the players of S_j are ordered according to groups, with the $\sqrt{k}$ players of a group appearing consecutively in σ. The ordering of the different groups of players of S_j is the same as the corresponding ordering of the active edges of G_{j-1} that was used to define these groups. Third, each (good) group of $\sqrt{k}$ co-located players is ordered so that (1) holds.

Now consider the sum $\sum_{\ell=1}^{k} \chi(i_\ell, S_\ell)$, where i_ℓ and S_ℓ denote the ℓth player and the first ℓ players of S with respect to σ, respectively. Since (1) holds for every group of players, for every $j \in \{0, 1, \ldots, p\}$, every group of players in S_j contributes at least

$$\sum_{\ell=1}^{\sqrt{k}} \frac{1}{4\beta} \cdot \frac{2^{-j}}{\ell} = \frac{2^{-j}\mathcal{H}_{\sqrt{k}}}{4\beta}$$

to this sum. By Lemma 3, for each $j \in \{1, \ldots, p\}$, there are 2^{j-1} such groups. There is also the group S_0. Thus the sum $\sum_{\ell=1}^{k} \chi(i_\ell, S_\ell)$ is at least

$$\frac{\mathcal{H}_{\sqrt{k}}}{4\beta} \left(1 + \sum_{j=1}^{(\log k)/2} 2^{j-1} \cdot 2^{-j} \right) \geq \frac{c}{\beta} \log^2 k = \left(\frac{c}{\beta} \log^2 k \right) \cdot C(S)$$

for some constant $c > 0$ that is independent of k and β. This completes the proof of Theorem 6.

References

1. 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.
2. L. Becchetti, J. Könemann, S. Leonardi, and M. Pál. Sharing the cost more efficiently: Improved approximation for multicommodity rent-or-buy. In *SODA '05*, pages 375–384.
3. D. Bienstock, M. X. Goemans, D. Simchi-Levi, and D. P. Williamson. A note on the prize-collecting traveling salesman problem. *Mathematical Programming*, 59(3):413–420, 1993.
4. S. Chawla, T. Roughgarden, and M. Sundararajan. Optimal cost-sharing mechanisms for network design. In *WINE '06*.
5. N. R. Devanur, M. Mihail, and V. V. Vazirani. Strategyproof cost-sharing mechanisms for set cover and facility location games. In *EC '03*, pages 108–114.
6. J. Feigenbaum, A. Krishnamurthy, R. Sami, and S. Shenker. Hardness results for multicast cost sharing. *Theoretical Computer Science*, 304:215–236, 2003.
7. J. Feigenbaum, C. Papadimitriou, and S. Shenker. Sharing the cost of multicast transmissions. *Journal of Computer and System Sciences*, 63(1):21–41, 2001.
8. J. Feigenbaum and S. J. Shenker. Distributed algorithmic mechanism design: Recent results and future directions. In *DIAL M '02*, pages 1–13.
9. L. Fleischer, J. Könemann, S. Leonardi, and G. Schäfer. Simple cost-sharing schemes for multicommodity rent-or-buy and stochastic Steiner tree. In *STOC '06*, pages 663–670.

10. J. Green, E. Kohlberg, and J. J. Laffont. Partial equilibrium approach to the free rider problem. *Journal of Public Economics*, 6:375–394, 1976.
11. A. Gupta, J. Könemann, S. Leonardi, R. Ravi, and G. Schäfer. An efficient cost-sharing mechanism for the prize-collecting Steiner forest problem. In *SODA '07*.
12. A. Gupta, A. Kumar, M. Pál, and T. Roughgarden. Approximation via cost-sharing: A simple approximation algorithm for the multicommodity rent-or-buy problem. In *FOCS '03*, pages 606–615.
13. A. Gupta, A. Kumar, and T. Roughgarden. Simpler and better approximation algorithms for network design. In *STOC '03*, pages 365–372.
14. A. Gupta, A. Srinivasan, and É. Tardos. Cost-sharing mechanisms for network design. In *APPROX '04*, pages 139–150.
15. J. D. Hartline. *Optimization in the Private Value Model: Competitive Analysis Applied to Auction Design*. PhD thesis, University of Washington, 2003.
16. N. Immorlica, M. Mahdian, and V. S. Mirrokni. Limitations of cross-monotonic cost-sharing schemes. In *SODA '05*, pages 602–611.
17. K. Jain and V. Vazirani. Applications of approximation algorithms to cooperative games. In *STOC '01*, pages 364–372.
18. K. Jain and V. Vazirani. Equitable cost allocations via primal-dual-type algorithms. In *STOC '02*, pages 313–321.
19. D. R. Karger and M. Minkoff. Building Steiner trees with incomplete global knowledge. In *FOCS '00*, pages 613–623.
20. K. Kent and D. Skorin-Kapov. Population monotonic cost allocation on MST's. In *Operational Research Proceedings KOI*, pages 43–48, 1996.
21. J. Könemann, S. Leonardi, and G. Schäfer. A group-strategyproof mechanism for Steiner forests. In *SODA '05*, pages 612–619.
22. 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 *ICALP '05*, pages 1051–1063.
23. S. Leonardi and G. Schäfer. Cross-monotonic cost-sharing methods for connected facility location. In *EC '04*, pages 242–243.
24. A. Mas-Colell, M. D. Whinston, and J. R. Green. *Microeconomic Theory*. Oxford University Press, 1995.
25. H. Moulin. Incremental cost sharing: Characterization by coalition strategy-proofness. *Social Choice and Welfare*, 16:279–320, 1999.
26. H. Moulin and S. Shenker. Strategyproof sharing of submodular costs: Budget balance versus efficiency. *Economic Theory*, 18:511–533, 2001.
27. M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, 1994.
28. M. Pál. *Cost Sharing and Approximation*. PhD thesis, Cornell University, 2005.
29. M. Pál and É. Tardos. Group strategyproof mechanisms via primal-dual algorithms. In *FOCS '03*, pages 584–593.
30. K. Roberts. The characterization of implementable choice rules. In J. J. Laffont, editor, *Aggregation and Revelation of Preferences*. North-Holland, 1979.
31. T. Roughgarden and M. Sundararajan. New trade-offs in cost-sharing mechanisms. In *STOC '06*, pages 79–88.
32. T. Roughgarden and M. Sundararajan. Approximately efficient cost-sharing mechanisms. Technical Report cs.GT/0606127, arXiv, 2006.
33. V. V. Vazirani. *Approximation Algorithms*. Springer, 2001.
34. W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16(1):8–37, 1961.