Quantifying Inefficiency in Cost-Sharing Mechanisms*

Tim Roughgarden† Mukund Sundararajan‡

February 26, 2009

Abstract

In a cost-sharing problem, several participants with unknown preferences vie to receive some good or service, and each possible outcome has a known cost. A cost-sharing mechanism is a protocol that decides which participants are allocated a good and at what prices. Three desirable properties of a cost-sharing mechanism are: incentive-compatibility, meaning that participants 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 economic efficiency, meaning that the cost incurred and the value to the participants are traded off in an optimal way. These three goals have been known to be mutually incompatible for thirty years. Nearly all the work on cost-sharing mechanism design by the economics and computer science communities has focused on achieving two of these goals while completely ignoring the third.

We introduce novel measures for quantifying efficiency loss in cost-sharing mechanisms and prove simultaneous approximate budget-balance and approximate efficiency guarantees for mechanisms for a wide range of cost-sharing problems, including all submodular and Steiner tree problems. Our key technical tool is an exact characterization of worst-case efficiency loss in Moulin mechanisms, the dominant paradigm in cost-sharing mechanism design.

1 Introduction

1.1 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 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 [17, 22, 32, 38].

*Preliminary versions of these results appeared in the Proceedings of the 38th Annual Symposium on Theory of Computing, May 2006, and Proceedings of the 12th Conference on Integer Programming and Combinatorial Optimization, June 2007.

†Department of Computer Science, Stanford University, 462 Gates Building, 353 Serra Mall, Stanford, CA 94305. Supported in part by ONR grant N00014-04-1-0725, DARPA grant W911NF-05-1-0224, an NSF CAREER Award, and an Alfred P. Sloan Fellowship. Email: `tim@cs.stanford.edu`.

‡Department of Computer Science, Stanford University, 470 Gates Building, 353 Serra Mall, Stanford, CA 94305. Supported in part by ONR Grants N00014-01-1-0795 and N00014-04-1-0725. Email: `mukunds@cs.stanford.edu`.

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* for selling a single good is a protocol that determines the winner and the selling price. Each bidder i is "selfish" in 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 if the bidder wins and 0 if the bidder 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 [43] 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 [43] 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.

1.2 Cost-Sharing Mechanisms

The *revenue* obtained by a mechanism can be as or more important than its economic efficiency, especially in settings where the mechanism designer incurs a non-trivial cost, such as production costs. This issue motivates the study of *cost-sharing mechanisms* that guarantee sufficient revenue to cover the incurred costs. Moulin and Shenker [36] describe a range of applications of cost-sharing mechanisms across economics, and Feigenbaum, Papadimitriou, and Shenker [16] motivate the study of such mechanisms from a computer networking perspective.

Formally, a *cost-sharing problem* is defined by a set U of players vying to receive some good or service, and a cost function $C : 2^U \to \mathcal{R}^+$ describing the cost incurred by the mechanism as a function of the auction outcome — the set S of winners. We assume that $C(\emptyset) = 0$ and that C is nondecreasing (i.e., $S \subseteq T$ implies $C(S) \leq C(T)$). We impose no explicit limit on the number of winners, but a large number of winners might result in extremely large costs. 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. A more complex example is a *Steiner tree* cost-sharing problem, where U represents a set of potential clients, located in an undirected graph with fixed edge costs, that want connectivity to a server t [16, 24]. In this application, $C(S)$ denotes the cost of connecting the terminals in S to t — the cost of the minimum-cost Steiner tree that spans $S \cup \{t\}$. For a cost function C and a valuation profile $\{v_i\}_{i \in U}$, the *efficient allocation* is the subset that maximizes the *social welfare*: $W(S) = v(S) - C(S)$, where $v(S)$ denotes $\sum_{i \in S} v_i$.

For a given set U and function C, a *cost-sharing mechanism* is a protocol that decides which players win 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 cost-sharing mechanism design: *incentive-compatibility*, meaning that every player's optimal strategy is to bid its true private value v_i for receiving the service; *budget-balance*, meaning that the mechanism recovers its incurred cost with the prices charged; and *efficiency*, meaning that the cost and valuations are traded off in an optimal way.

Unfortunately, roughly thirty years ago Green, Kohlberg, and Laffont [18] and Roberts [40] ruled out the existence of mechanisms that simultaneously satisfy these three constraints, even in very simple cost-sharing problems. This impossibility result motivates relaxing at least one of these properties. Until recently, nearly all work in cost-sharing mechanism design completely ignored either budget-balance or efficiency. Without the budget-balance constraint, there is an extremely powerful and flexible mechanism that is incentive-compatible and efficient: the *VCG mechanism* (see e.g. [16, 36]). This mechanism specializes to the Vickrey auction when selling a single good, but it is far more general. The VCG mechanism is typically not approximately budget-balanced for any reasonable approximation factor (assuming "individually rational" prices, see e.g. [15] for details).

A second approach is to discard economic efficiency as an objective and insist on incentive-compatibility and budget-balance. Until very recently [33], the only general technique for designing mechanisms of this type was due to Moulin [35]. Researchers have developed numerous approximately budget-balanced Moulin mechanisms for cost-sharing problems arising from different combinatorial optimization problems, including fixed-tree multicast problems [2, 15, 16]; more general submodular problems [35, 36]; scheduling problems [6, 8]; network design problems [19, 20, 24, 25, 27, 29, 39]; facility location problems [30, 39]; and various covering problems [12, 23]. With one exception discussed below, none of these works provided any guarantees on the economic efficiency achieved by the proposed mechanisms.

1.3 Why Quantify Inefficiency?

Impossibility results are, of course, common in optimization. Motivated by conditional impossibility results like Cook's Theorem [10], as well as information-theoretic lower bounds in restricted models of computation like online [7] and streaming algorithms [37], algorithm designers are accustomed to devising heuristics and proving worst-case guarantees about them using approximation measures. This approach can also be applied to cost-sharing mechanism design 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 not been pursued previously.

Quantifying efficiency loss in cost-sharing mechanisms is an important goal for several reasons. First, a quantitative approximation measure is necessary to rigorously compare the economic efficiency of different *mechanisms* for a cost-sharing problem, and to identify a mechanism as "optimally efficient" subject to budget-balance constraints. Second, such a measure allows us to define and compare the intrinsic complexity of cost-sharing *problems*. To give an analogy, recall that the "difficulty" of an NP-hard optimization problem is often identified with the best-possible approximation ratio achievable by a polynomial-time algorithm for it, assuming $P \neq NP$ (see e.g. [3]). For a cost-sharing problem, we can similarly interpret the efficiency guarantee achieved by an optimally efficient mechanism as a measure of the problem's "complexity". Third, even when economic efficiency is not the primary objective, requiring "reasonable" (but not necessarily optimal) efficiency can be useful for constraining the mechanism design space. For example, the intuitively "undesirable" family of mechanisms identified by Immorlica, Mahdian, and Mirrokni [23, Example 4.1], which stubbornly satisfy a long list of standard mechanism design requirements, admit no non-trivial efficiency guarantees.

The sole previous work on quantifying efficiency loss in budget-balanced cost-sharing mechanisms is by Moulin and Shenker [36], who studied submodular cost-sharing problems and an additive notion of efficiency loss. Their results successfully rank different fully budget-balanced mechanisms for an arbitrary but fixed submodular cost-sharing problem according to worst-case efficiency loss (see also Section 4). However, it is not obvious how to use their efficiency loss measure to make comparisons between different cost-sharing problems. Additionally, the approach in [36] has not yet been extended beyond submodular cost-sharing problems, and most of the problems studied in the computer science literature fall outside of this class [6, 8, 19, 20, 23, 24, 25, 27, 29, 30, 39].

1.4 How to Quantify Inefficiency?

The impossibility results in [18, 40] motivate approximate notions of budget-balance and economic efficiency. In this paper, we define a mechanism to be *β-budget-balanced* for a parameter $\beta \geq 1$ if the sum of the prices charged is always at least the cost incurred and is also at most β times this cost. Several previous works instead require that the revenue is no more than and at least a $1/\beta$ fraction of the incurred cost; we obtain similar results for this alternative definition (see Sections 1.7 and 6).

Several definitions of approximate efficiency are possible. Arguably, the most natural requirement is to insist that a mechanism always computes an outcome S that is a ρ-approximation of the social welfare: $W(S) \geq \rho \cdot W(S^*)$, where S^* is the economically efficient solution. Unfortunately, Feigenbaum et al. [15] shattered any hope for such a guarantee, even in very simple cost-sharing problems: for every $\beta \geq 1$ and β-budget-balanced incentive-compatible mechanism, there is a valuation profile such that the efficient solution has strictly positive welfare but the mechanism produces the empty outcome (with zero welfare). Thus every mechanism, no matter how intuitively "good" or "bad", is a 0-approximation algorithm for the social welfare objective. This inapproximability result is characteristic of mixed-sign objective functions such as the social welfare.

We must therefore measure efficiency loss in a different way. Our basic efficiency guarantees have the following form, for a parameter $\rho \geq 0$ and a mechanism for the cost-sharing problem C: for every valuation profile,

$$W(S^*) - W(S) \leq \rho \cdot C(S^*), \tag{1}$$

where S is the output of the mechanism and S^* is an efficient outcome. In this case, we call the mechanism *ρ-approximate*.

We have chosen to present this efficiency guarantee in terms of additive welfare loss, but it is robust and admits several different interpretations. For example, the bound in (1) implies a relative approximation guarantee for a different formulation of economic efficiency. Precisely, define the *social cost* $\pi(S)$ of an outcome S to be the cost incurred by the mechanism plus the sum of the *excluded* valuations (i.e., opportunity cost):

$$\pi(S) = C(S) + v(U \setminus S). \tag{2}$$

Since social cost and social welfare are related by the affine transformation $\pi(S) = -W(S) + v(U)$, minimizing the social cost is ordinally equivalent to maximizing the social welfare. The two objective functions are not, of course, equivalent from an approximation perspective. Indeed, while the impossibility result in Feigenbaum et al. [15] precludes any relative approximation of the social welfare, *every ρ-approximate cost-sharing mechanism also $(\rho + 1)$-approximates the social cost.* Such non-approximation-preserving transformations are common in applications with mixed-sign objective functions, including prize-collecting combinatorial optimization problems (e.g. [5]) and discrete maximum-likelihood problems (e.g. [28]).

A second interpretation of the bound in (1) is motivated by the examples used in the impossibility result in [15]. These examples are intuitively difficult because the optimal outcome S^* has large cost $C(S^*)$ and value $v(S^*)$ only slightly larger than $C(S^*)$, leaving the mechanism with no "margin for error". Can we obtain a relative approximation of welfare when the value of an optimal outcome is bounded away from its cost? To formalize this question, we say that an outcome S is *η-separated* if $W(S) \geq \eta \cdot C(S)$ or, equivalently, if $v(S) \geq (\eta + 1) \cdot C(S)$. The punchline, proved via a simple calculation, is this: if a mechanism is ρ-approximate, then *ρ is the separation threshold beyond which non-trivial welfare approximation is possible.* Precisely, a ρ-approximate mechanism extracts at least a $(1-\rho/\eta)$ fraction of the optimal welfare when the optimal outcome is η-separated.

1.5 Our Techniques: Moulin Mechanisms and Summability

Our overarching goal is to identify tight upper and lower bounds on the best-possible efficiency guarantees of incentive-compatible and budget-balanced mechanisms for a wide range of cost-sharing problems. Our first contribution is a general analytical framework for proving such bounds (Section 3). The framework applies to *Moulin mechanisms*, the dominant paradigm in budget-balanced cost-sharing mechanism design.

Roughly, a Moulin mechanism simulates an ascending iterative auction. In each iteration, a price $\chi(i, S)$ is offered to each player i of the remaining players S. Players that accept remain in contention; the others are removed. The mechanism halts when all remaining players accept the prices offered to them. To achieve approximate budget-balance, the mechanism offers prices at each iteration that approximately cover the cost that would be incurred if the iteration is the last. To obtain incentive-compatibility, a Moulin mechanism offers each player a non-decreasing sequence of prices. The function χ is called a *cost-sharing method*, and it uniquely defines the corresponding Moulin mechanism. (See Section 2 for formal definitions.) Until very recently, almost all approximately budget-balanced cost-sharing mechanisms were Moulin mechanisms [6, 8, 19, 20, 23, 24, 25, 27, 29, 30, 36, 39], with the mechanisms of Devanur, Mihail, and Vazirani [12] forming a notable exception.

Our first main result is a characterization of the worst-case efficiency loss of a Moulin *mechanism* in terms of a single parameter of its underlying cost-sharing *method*. Given a cost-sharing method χ and a cost function C defined over the same set U of players, this parameter α is easy to describe. We say that the method χ is *α-summable for C* if the following condition holds for every subset $S \subseteq U$ and every ordering of the players of S:

$$\sum_{\ell=1}^{|S|} \chi(i_\ell, S_\ell) \leq \alpha \cdot C(S), \tag{3}$$

where i_ℓ and S_ℓ denote the ℓth player and the set of the first ℓ players in the ordering, respectively. In other words, start with the empty set, add players of S one-by-one according to the given ordering, and let X_ℓ denote the cost share of the ℓth player (according to χ) when the player is first added. The cost-sharing method χ is α-summable for C if the sum $\sum_\ell X_\ell$ only overestimates the cost of $C(S)$ by an α factor (for a worst-case choice of the subset S and the ordering of the players).

For example, in the special case of a symmetric cost function and equal cost shares, summability is a measure of the "amount of concavity" of the cost function. Consider the function $C(S) = |S|^d$ for $d \in [0, 1]$ on the universe $U = \{1, 2, \ldots, n\}$ and the cost-sharing method $\chi(i, S) = C(S)/|S| = |S|^{d-1}$. The summability of χ is then determined by the set $S = U$; the ordering σ is irrelevant. A simple calculation shows that this summability is roughly $1/d$ for fixed $d > 0$ and large n, and grows as $\ln n$ when $d = 0$.

We prove that *summability characterizes approximate efficiency* in the following sense: a Moulin mechanism is $(\alpha-1)$-approximate if and only if its underlying cost-sharing method is α-summable. The key idea behind our proof is to view a Moulin mechanism as a greedy descent algorithm with respect to a type of "potential function". Summability then arises naturally as a measure of proximity between this potential function and the social objective function.

1.6 Our Results: Efficiency Guarantees for Submodular and Steiner Tree Problems

Bounding the summability (3) of a cost-sharing method is a non-trivial but often tractable problem. We demonstrate this by applying our summability framework to obtain matching upper and lower bounds on the best-possible efficiency guarantees of Moulin mechanisms for two widely studied classes of cost-sharing problems, submodular problems (Section 4) and Steiner tree problems (Section 5). Since the conference version of this work [41], many more applications have been found; see Section 7.

A submodular cost-sharing problem is defined by a player set U and a nondecreasing cost function C such that, for every $S_1 \subseteq S_2$ and $i \notin S_2$,

$$C(S_2 \cup \{i\}) - C(S_2) \leq C(S_1 \cup \{i\}) - C(S_1). \tag{4}$$

Submodular cost-sharing problems admit a range of budget-balanced Moulin mechanisms [25, 36]. One is the Shapley mechanism [16, 36], whose underlying cost-sharing method is derived from the Shapley value. As a first application of our framework, we prove that for every submodular cost-sharing problem, the corresponding Shapley cost-sharing method is $\mathcal{H}_k$-summable, where k is the number of players served in an optimal solution, and $\mathcal{H}_k = \sum_{i \leq k} 1/i \approx \ln k$ denotes the kth Harmonic number. Our characterization result then implies that the Shapley mechanism is $(\mathcal{H}_k - 1)$-approximate and also $\mathcal{H}_k$-approximates the social cost for every submodular cost-sharing problem. It also implies that *the Shapley mechanism is an optimal Moulin mechanism* in the following sense: there is a simple submodular cost-sharing problem for which every budget-balanced Moulin mechanism is at least $\mathcal{H}_k$-summable. These results reprove, from a different perspective, earlier results of Moulin and Shenker [36, Proposition 2].

Our most mathematically involved results concern the much more complex class of Steiner tree cost-sharing problems. Such problems are generally not submodular, and no efficiency guarantees of any sort were previously known for approximately budget-balanced mechanisms for such problems. Our main positive result is a proof that the 2-budget-balanced Steiner tree cost-sharing method designed by Jain and Vazirani [24] is $O(\log^2 k)$-summable, where k is again the number of players served in an optimal solution, and thus the corresponding Moulin mechanism (the *JV mechanism*) is $O(\log^2 k)$-approximate. Our proof blends ideas inspired by online algorithms, primal-dual approximation algorithms, and our analysis for submodular cost functions. Techniques from online analysis are useful because summability is defined in terms of a worst-case player ordering; primal-dual arguments arise because the JV mechanism is based on Edmonds's primal-dual branching algorithm [14].

Our efficiency guarantee for the JV mechanism is weaker than that for the Shapley mechanism, and this is no accident: we use our characterization result and a recursive construction to prove that *every $O(1)$-budget-balanced Moulin mechanism for Steiner tree cost-sharing problems is $\Omega(\log^2 k)$-approximate.* Our positive results for submodular problems and this lower bound expose a non-trivial, latent approximation hierarchy among different cost-sharing problems. Of course, this lower bound for Steiner tree problems trivially carries over to the more general network design cost-sharing problems studied in [19, 20, 29, 39].

1.7 Our Results: Budget-Balance vs. Efficiency Trade-Offs

Finally, in Section 6 we extend our summability framework to quantify trade-offs between budget-balance and economic efficiency in cost-sharing mechanisms. In particular, inefficiency can be partially mitigated if the prices charged need not cover the cost incurred. Call a mechanism (β, γ)-*budget-balanced* if the prices charged are always at most a β factor times and at least a $1/\gamma$ fraction of the cost incurred. Permitting $\gamma > 1$ gives rise to a new source of efficiency loss: a mechanism can inadvertently service players with valuations too small to justify service. For example, a mechanism that is (β, γ)-budget-balanced with $\gamma > 1$ might produce an outcome with negative welfare.

We can extend nonetheless our summability characterization of efficiency loss: we prove that every (β, γ)-budget-balanced Moulin mechanism derived from an α-summable cost-sharing method satisfies

$$W(S^*) - W(S) \leq (\alpha + \gamma - 2) \cdot C(S^*) + (\gamma - 1) \cdot v(S \setminus S^*), \tag{5}$$

where S is the output of the mechanism and S^* is an optimal outcome. As a consequence, such a mechanism ρ-approximates the social cost (2), where $\rho = \max\{\gamma, \alpha + \gamma - 1\}$. These guarantees are tight for all values of α and γ.

For example, consider a submodular cost-sharing problem. Dividing the cost shares of the corresponding Shapley mechanism by a $\gamma \geq 1$ factor, we obtain a $(1, \gamma)$-budget-balanced Moulin mechanism induced by an $(\mathcal{H}_n/\gamma)$-summable cost-sharing method, where n is the number of players. Choosing $\gamma = \Theta(\sqrt{\log n})$ yields a $(1, O(\sqrt{\log n}))$-budget-balanced Moulin mechanism that $O(\sqrt{\log n})$-approximates the social cost. Thus budget-balance can be sacrificed to gain efficiency, *but there is also an intrinsic barrier:* our lower bounds imply that no Moulin mechanism $o(\sqrt{\log n})$-approximates the social cost, no matter how poor its budget-balance. Similar trade-offs between approximate budget-balance and efficiency apply to the JV mechanism and Steiner tree cost-sharing problems.

1.8 Organization

Section 2 reviews the basics of cost-sharing mechanism design and Moulin mechanisms, and compares different notions of approximate economic efficiency. Section 3 proves that the worst-case efficiency loss of a Moulin mechanism is characterized by the summability of its cost-sharing method. Sections 4 and 5 prove matching upper and lower bounds on the best efficiency guarantees achievable by Moulin mechanisms for submodular and Steiner tree cost-sharing problems, respectively.

Section 6 extends our characterization result to (β, γ)-budget-balanced Moulin mechanisms and gives quantifiable trade-offs between budget-balance and efficiency in such mechanisms. Section 7 concludes with a discussion of recent related work and open research questions.

2 Preliminaries

After formally defining cost-sharing mechanisms and incentive-compatibility in Section 2.1, we define approximate budget-balance and several notions of approximate efficiency in Section 2.2. Section 2.3 reviews Moulin mechanisms.

2.1 Cost-Sharing Mechanisms

The problem input is a set U of n players and a cost function C that assigns a cost $C(S)$ to every set $S \subseteq U$ of players. We assume that $C(\emptyset) = 0$ and that $C(S) \leq C(T)$ for all $S \subseteq T \subseteq U$. We sometimes refer to $C(S)$ as the *service cost*, to distinguish it from the social cost (2). In addition, every player $i \in U$ possesses a private, nonnegative *valuation* v_i, representing player i's maximum willingness to pay for being included in the chosen set S.

Example 2.1 (Fixed-Tree Multicast) In a *fixed-tree multicast cost-sharing problem* [16, 36], the cost function is implicitly defined as follows. The input is a tree T with root t and nonnegative edge costs, where each player $i \in U$ is located at some vertex of T. For a subset $S \subseteq U$, the cost $C(S)$ is defined as the sum of the costs of the edges in the (unique) smallest subtree that contains all of the players of S. This cost function is submodular in the sense of (4).

Example 2.2 (Steiner Tree) *Steiner tree cost-sharing problems* [24] generalize fixed-tree multicast problems in that the input is a graph G rather than a tree T. The cost $C(S)$ of a subset of players is defined as that of a minimum-cost subgraph of G that spans all of the players of S as well as the root t. This cost function is not generally submodular.

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. For cost functions that are defined implicitly as the optimal solution of an instance of a combinatorial optimization problem, as in Example 2.2, we also hold the mechanism M responsible for constructing a feasible solution to the optimization problem induced by the served set S. The cost $C_M(S)$ of this feasible solution is in general larger than the cost $C(S)$ of an optimal solution. We insist that all prices are nonnegative ("no positive transfers"), and 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 $i \in S$. As is standard, we assume that every player aims to maximize the quasilinear utility function $u_i(S, p_i) = v_i x_i - p_i$, where $x_i = 1$ if $i \in S$ and $x_i = 0$ if $i \notin S$. Our incentive-compatibility constraint is the well-known strategyproof condition, stating that truthful bidding is a dominant strategy for every player.

Definition 2.3 (Strategyproofness) A mechanism is *strategyproof* 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.

In fact, all of the mechanisms we study meet the more stringent "groupstrategyproof" condition (Remark 2.11).

Remark 2.4 Mechanisms can be defined more generally, but the Revelation Principle [32, P.871] justifies restricting attention to the class of "direct-revelation mechanisms" defined above.

2.2 Approximate Budget-Balance and Economic Efficiency

As discussed in the Introduction, our two cost-sharing mechanism objectives are budget-balance and economic efficiency. A mechanism M for the cost-sharing problem C is (β, γ)-*budget-balanced* if

$$\frac{C_M(S)}{\gamma} \leq \sum_{i \in S} p_i \leq \beta \cdot C(S)$$

for every outcome — set S, prices p, and, if applicable, feasible solution with service cost $C_M(S)$ — of the mechanism. A β-*budget-balanced mechanism* is, by definition, $(\beta, 1)$-budget-balanced. A *no-deficit* mechanism is β-budget-balanced for some $\beta \geq 1$. We focus only on such β-budget-balanced mechanisms except in Section 6.

Remark 2.5 Most previous works on approximately budget-balanced cost-sharing mechanisms define β-budget-balance to mean $(1, \beta)$-budget-balance rather than $(\beta, 1)$-budget-balance. For the class of cost-sharing mechanisms that we study (see Section 2.3), a mechanism meeting one definition can be modified to satisfy the other by scaling its prices accordingly, and thus the two definitions are in some sense equivalent. In this paper, we adopt the definition that is more convenient for stating and proving efficiency guarantees. Analogs for the alternative definition follow from our general results in Section 6.

Our primary definition of approximate efficiency measures additive welfare loss, relative to the service cost of an optimal solution (1). To recap, a mechanism for a cost-sharing problem C is ρ-*approximate* if, assuming truthful bids,

$$W(S^*) - W(S) \leq \rho \cdot C(S^*)$$

for every valuation profile v, where S^* is the optimal outcome for this valuation profile, S is the outcome of the mechanism with this valuation profile, and $W(T) = v(T) - C(T)$ denotes the social welfare of the set $T \subseteq U$. When it is convenient, we sometimes parametrize ρ by the number $n = |U|$ of players or the number $k = |S^*|$ of players served in an optimal outcome. We next establish the robustness of such an approximation bound by demonstrating its consequences for alternative definitions of approximate economic efficiency.

Not all definitions of approximate efficiency provide meaningful information for cost-sharing mechanism design. As noted in Section 1.4, for each $\beta \geq 1$ there are simple cost-sharing problems such that no incentive-compatible, β-budget-balanced mechanism obtains a non-zero fraction of the optimal welfare [15]. Thus, if we insist on adopting a relative approximation measure — by far the most ubiquitous kind across theoretical computer science — we must either change the objective function or restrict the allowable instances. We explore these two approaches in turn.

What is the "smallest perturbation" of the welfare objective that admits non-trivial approximation results? A minimal requirement for a credible reformulation is *ordinal equivalence* — for a fixed cost-sharing function and valuation profile, a subset S should be "better" than a subset T if and only if S has higher welfare than T. This requirement suggests either maximizing $f(W(S))$ for a strictly increasing function f or minimizing $f(W(S))$ for a strictly decreasing function f. Affine functions are in some sense the "least distorting" candidate functions f, and for relative approximation guarantees there is no loss of generality in considering only: (1) minimizing $-W(S) + g(C, v) = C(S) - v(S) + g(C, v)$, where the additive term $g(C, v)$ is positive and independent of S; and (2) maximizing $v(S) - C(S) + h(C, v)$ for a positive additive term $h(C, v)$. Since costs and valuations already occur positively in (1) and (2), respectively, we take g to be independent of C and h to be independent of v. The examples in [15] are strong enough to imply that no non-trivial relative approximation is possible for these objectives unless $g(C, v) \geq v(S^*)$ and $h(C, v) \geq C(S^*)$. To avoid the awkwardness of referencing the optimal solution in the objective function itself, we take $g(C, v) = v(U)$ and $h(C, v) = C(U)$, leading to the objectives of *minimizing social cost*:

$$\min_{S \subseteq U} \pi(S) \equiv -W(S) + v(U) = C(S) + v(U \setminus S); \tag{6}$$

and *maximizing social reward*:

$$\max_{S \subseteq U} R(S) \equiv W(S) + C(U) = v(S) + [C(U) - C(S)]. \tag{7}$$

These answers to our initial question conform to previous approaches to approximating mixed-sign objective functions in other application domains, including prize-collecting combinatorial optimization (e.g. [5]) and maximum-likelihood inference (e.g. [28]).

Simple algebra shows that an efficiency guarantee of the form (1) implies relative approximation guarantees for the social cost and social reward objectives.

Proposition 2.6 (From Additive to Relative Approximation) *If M is a ρ-approximate mechanism for a cost-sharing problem C, then, assuming truthful bids:*

(a) M is a $(\rho + 1)$-approximation algorithm for minimizing social cost; and

(b) M is a $1/(\rho + 1)$-approximation algorithm for maximizing social reward.

The guarantees in Proposition 2.6 hold even if the constants $g(C, v)$ and $h(C, v)$ in the definitions of social cost (6) and social reward (7) are reduced to $v(S^*)$ and $C(S^*)$, respectively.

A second approach to efficiency guarantees is to seek a relative approximation of welfare for the widest class of problems possible. The simple examples in [15] show that restricting only the cost function is insufficient for non-trivial relative welfare guarantees. We instead study "promise problems" in which the value served by an optimal solution is bounded away from its service cost. Recall from the Introduction that an outcome S is *η-separated* for a parameter $\eta \geq 0$ if $W(S) \geq \eta \cdot C(S)$. Call a valuation profile *η-separated* if there is an η-separated efficient outcome. Simple algebra implies the following.

Proposition 2.7 (From Additive Approximation to Promise Problems) *If M is a ρ-approximate mechanism for a cost-sharing problem C, then, assuming truthful bids, M is a $(1 - \frac{\rho}{\eta})$-approximation algorithm for social welfare for η-separated valuation profiles.*

Thus the approximation factor ρ is the separation threshold beyond which the mechanism is guaranteed to approximate the social welfare.

Finally, recall that our critique of the social welfare objective was rooted in the fact that it fails to differentiate between "better" and "worse" cost-sharing mechanisms. Does the approximation framework detailed in this section suffer the same flaw? The answer is "no": the approximation factors (in the sense of (1)) of different mechanisms for a problem can vary widely (Example 2.12 and Proposition 3.12), and the best-achievable approximation factor is different for different types of cost-sharing problems (Section 4 and Theorem 5.10).

2.3 Moulin Mechanisms

Next we review *Moulin mechanisms*, the preeminent class of strategyproof, approximately budget-balanced mechanisms. A Moulin mechanism is driven by a *cost-sharing method*—a function χ that assigns a non-negative *cost share* $\chi(i, S)$ for every subset $S \subseteq U$ of players and every player $i \in S$. For cost functions induced by combinatorial optimization problems (such as Examples 2.1 and 2.2), a cost-sharing method outputs both cost shares and a feasible solution for the optimization problem induced by S. A cost-sharing method is (β, γ)*-budget balanced* for a cost function C and parameters $\beta, \gamma \geq 1$ if

$$\frac{C_\chi(S)}{\gamma} \leq \sum_{i \in S} \chi(i, S) \leq \beta \cdot C(S), \tag{8}$$

where $C_\chi(S)$ is the cost of the feasible solution produced by the method χ. As usual, β-budget-balance is short for $(\beta, 1)$-budget-balance, and such methods are also called *no-deficit*. 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 $i \in S$, $\chi(i, S) \geq \chi(i, T)$.

Example 2.8 (Shapley and Sequential Cost-Sharing) Consider an instance of fixed-tree multicast (Example 2.1) with tree T and player set $U = \{1, 2, \ldots, n\}$. Two 1-budget-balanced cost-sharing methods are as follows. In the *sequential* cost-sharing method χ_{seq}, given a subset $S \subseteq U$, each player $i \in S$ pays the full cost of each edge of its (unique) path to the root of T that is not used by a player of S with lower index. In the *Shapley* method χ_{sh}, each player $i \in S$ pays a "fair share" of each of the edges in its path — c_e/n_e for an edge e of cost c_e, where n_e denotes the number of players of S using edge e to the reach the root of T. Since the amount a player pays for each edge in its path can only increase as other players are removed from S, both of these methods are cross-monotonic.

Given a cost-sharing method χ for a cost function C, we obtain the corresponding Moulin mechanism by simulating an iterative ascending auction, with the method χ suggesting prices for the remaining players at each iteration.

Definition 2.9 (Moulin Mechanisms) Let U be a universe of players and χ a cost-sharing method defined on U. The *Moulin mechanism* $M(\chi)$ *induced by* χ is the following.

1. Collect a bid b_i from each player $i \in U$.
2. Initialize $S := U$.
3. If $b_i \geq \chi(i, S)$ for every $i \in S$, then halt. Output the set S, the feasible solution constructed by χ, and charge each player $i \in S$ the price $p_i = \chi(i, S)$.
4. Let $i^* \in S$ be a player with $b_{i^*} < \chi(i^*, S)$.
5. Set $S := S \setminus \{i^*\}$ and return to Step 3.

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. This implies that the outcome of a Moulin mechanism is uniquely defined, independent of the choices made in Step 4. Also, the Moulin mechanism $M(\chi)$ clearly inherits the budget-balance factors of the cost-sharing method χ. Finally, Moulin [35] proved the following.

Theorem 2.10 (Strategyproofness of Moulin Mechanisms [35]) *If χ is a cross-monotonic cost-sharing method, then the corresponding Moulin mechanism $M(\chi)$ is strategyproof.*

Theorem 2.10 reduces the problem of designing an strategyproof, (β, γ)-budget-balanced cost-sharing mechanism to that of designing a cross-monotonic, (β, γ)-budget-balanced cost-sharing method. As noted in the Introduction, until recently almost all known approximately budget-balanced cost-sharing mechanisms were Moulin mechanisms.

Remark 2.11 Moulin mechanisms also satisfy a stronger notion of incentive compatibility called *groupstrategyproofness* [35, 36], which states that every coordinated set of false bids by a coalition should decrease the utility of some player in the coalition (or should have no effect).

By Theorem 2.10, the sequential and Shapley cost-sharing methods of Example 2.8 induce strategyproof and fully budget-balanced mechanisms for fixed-tree multicast cost-sharing problems. The classical impossibility results [18, 40] imply that neither mechanism can be fully efficient. We conclude the section with concrete examples demonstrating this.

Example 2.12 (Excludable public good) Consider an instance of fixed-tree multicast consisting of one link with cost $1+\epsilon$ and a set of n players co-located opposite the root. Such a cost function is often called an *excludable public good* in the economic cost-sharing literature (e.g. [11, 31]). For a valuation profile v, the efficient outcome is U if $v(U) > 1 + \epsilon$ and $\emptyset$ otherwise. The idea is to determine "worst-case valuations" for the Moulin mechanisms $M(\chi_{seq})$ and $M(\chi_{sh})$ induced by the sequential and Shapley cost-sharing methods, respectively. We do this by setting the valuations of players to be as large as possible, subject to the constraint that the mechanism terminates with the empty outcome.

If all players have valuation 1 and bid truthfully, then $M(\chi_{seq})$ outputs the empty outcome. If player i has valuation $1/i$ for $i \in \{1, 2, \ldots, n\}$ and players bid truthfully, then $M(\chi_{sh})$ outputs the empty outcome. These examples show that the first mechanism is no better than $\approx (n-1)$-approximate, while the second is no better than $\approx (\mathcal{H}_n - 1)$-approximate, where $\mathcal{H}_n = \sum_{i=1}^{n} 1/i$ denotes the nth Harmonic number.

3 Summability Characterizes Approximate Efficiency

This section proves that the summability of a cost-sharing method characterizes the approximate efficiency of the corresponding Moulin mechanism. After Section 3.1 defines summability, Section 3.2 proves that it upper bounds approximate efficiency and Section 3.3 explores the senses in which this bound is tight.

3.1 Summability

Intuitively, summability quantifies the efficiency loss from the overly aggressive removal of players by a Moulin mechanism. We motivate the formal definition via a generalization of Example 2.12, which strongly suggests that summability lower bounds the approximate efficiency of a Moulin mechanism.

Example 3.1 (Generic Lower Bound on Efficiency Loss) Let χ be a cross-monotonic cost-sharing method for the cost function C, defined on the universe U. Assume for simplicity that the method only assigns positive cost shares: $\chi(i,S) > 0$ for all $S \subseteq U$ and $i \in S$. Pick an ordering σ of the players of U and a subset S. Let i_ℓ denote the ℓth player and S_ℓ the first ℓ players of S with respect to σ and define the parameter $\alpha_{S,\sigma}$ by

$$\alpha_{S,\sigma} = \frac{1}{C(S)} \sum_{\ell=1}^{|S|} \chi(i_\ell, S_\ell). \tag{9}$$

In other words, we start with the empty set, add players of S one-by-one according to σ, and consider the cost share of the ℓth player when it is initially added. The parameter $\alpha_{S,\sigma}$ is the factor by which the sum of these cost shares overestimates the cost $C(S)$ of serving all of the players.

We claim that the Moulin mechanism $M(\chi)$ is no better than $(\alpha_{S,\sigma} - 1)$-approximate for C. To see this, define the valuation v_ℓ of the ℓth player of S (according to σ) to be $\chi(i_\ell, S_\ell) - \epsilon$, where $\epsilon > 0$ is arbitrarily small. Give players of $U \setminus S$ zero valuations. The Moulin mechanism $M(\chi)$ will output the empty set. The optimal welfare is bounded below by $v(S) - C(S) \approx \alpha_{S,\sigma} \cdot C(S) - C(S) = (\alpha_{S,\sigma} - 1) \cdot C(S)$. Since valuations outside S are zero, there is an efficient outcome $S^* \subseteq S$, and hence the welfare loss of $M(\chi)$ on this valuation profile is at least $(\alpha_{S,\sigma} - 1) \cdot C(S^*)$.

The *summability* of a cost-sharing method is then defined as the worst-case ratio of the form (9) over choices of sets S and orderings σ.

Definition 3.2 (Summability) Let C and χ be a cost function and a cost-sharing method, respectively, defined on a common universe U of n players. The method χ is *α-summable for C* for a function $\alpha : \{0, 1, 2, \ldots, n\} \to \mathcal{R}^+$ if

$$\sum_{\ell=1}^{|S|} \chi(i_\ell, S_\ell) \leq \alpha(|S|) \cdot C(S) \tag{10}$$

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.

Remark 3.3 We define summability as a function rather than a scalar in order to parametrize our efficiency guarantees by the number k of players served in an efficient outcome (which can be much smaller than the universe size). For example, in Sections 4 and 5 we establish summability bounds of the form $\alpha(|S|) \leq \mathcal{H}_{|S|}$ and $\alpha(|S|) = O(\log^2 |S|)$ for all $S \subseteq U$, which will lead to Moulin mechanisms that are $\mathcal{H}_k$- and $O(\log^2 k)$-approximate, respectively.

3.2 Efficiency Guarantees

The central result of this section is the following efficiency guarantee for Moulin mechanisms derived from cost-sharing methods with small summability.

Theorem 3.4 (Summability Upper Bounds Approximate Efficiency) *Let C be a cost function defined on a universe U and χ a cross-monotonic, no-deficit, α-summable cost-sharing method for C. Then $M(\chi)$ is an $(\alpha(k) - 1)$-approximate mechanism, where k is the size of an efficient outcome.*

Propositions 2.6 and 2.7 immediately give the following corollaries.

Corollary 3.5 *Let C be a cost function defined on a universe U and χ a cross-monotonic, no-deficit, α-summable cost-sharing method for C. Then $M(\chi)$ is:*

(a) an $\alpha(k)$-approximation algorithm for minimizing the social cost;

(b) a $1/\alpha(k)$-approximation algorithm for maximizing the social reward;

(c) a $[1-(\alpha(k)-1)/\eta]$-approximation algorithm for maximizing welfare for η-separated valuation profiles.

We emphasize that Theorem 3.4 is completely problem-independent. Together with Definition 3.2, it distills the problem-specific aspect of simultaneously achieving good budget-balance and efficiency in Moulin mechanisms: designing a cross-monotonic and approximately budget-balanced cost-sharing method with small summability. We illustrate the generality of Theorem 3.4 in Sections 4–6 by showing matching upper and lower bounds of $\Theta(\log k)$ and $\Theta(\log^2 k)$ on the approximate efficiency of Moulin mechanisms for submodular and Steiner tree cost-sharing problems, respectively, and to quantifiable trade-offs between budget-balance and economic efficiency.

We now build up to a proof of Theorem 3.4. Fix a cost function C defined on a universe U, a valuation profile v, and an α-summable and a no-deficit cross-monotonic cost-sharing method for C. Let σ denote the reversal of the order in which the mechanism $M(\chi)$ deletes players (in some fixed trajectory), with players in the final output set S^M ordered arbitrarily among the first $|S^M|$ positions.

A crucial tool in our proof is the following *potential function* Φ_σ, which we define for each subset $S \subseteq U$ as

$$\Phi_\sigma(S) = v(U \setminus S) + \sum_{i_\ell \in S} \chi(i_\ell, S_\ell), \tag{11}$$

where for every $\ell \in \{1, 2, \ldots, |S|\}$, S_ℓ denotes the first ℓ players of S and i_ℓ the ℓth player of S according to σ.

The ordering σ and the potential function Φ_σ are defined to ensure that the potential function value decreases with each iteration in our fixed trajectory of $M(\chi)$. We use this fact in the next lemma.

Lemma 3.6 *If S^M is the final output of $M(\chi)$ and S^* is an efficient outcome for a valuation profile v, then*

$$\Phi_\sigma(S^M \cap S^*) \leq \Phi_\sigma(S^*).$$

Proof: The idea is to delete players from S^* in the same order as $M(\chi)$ to obtain the set $S^M \cap S^*$. More precisely, order the players $i_1, i_2, \ldots, i_m$ of $S^* \setminus S^M$ according to their deletion by $M(\chi)$, with player i_1 deleted first. This ordering is consistent with σ. For a player $i_j \in S^* \setminus S^M$, let S_j denote the set of players from which it was removed by $M(\chi)$, and let S_j^* denote $S^* \setminus \{i_1, \ldots, i_{j-1}\}$. Note that $S_j \supseteq S_j^*$ for every j. By the definition of $M(\chi)$, the valuation v_j of player i_j is less than $\chi(i_j, S_j)$. Cross-monotonicity of χ then implies that $v_j < \chi(i_j, S_j^*)$ for every player $i_j \in S^* \setminus S^M$. Using the definition of Φ_σ, we have

$$\Phi_\sigma(S^*) = \Phi_\sigma(S_1^*) > \Phi_\sigma(S_2^*) > \cdots > \Phi_\sigma(S_{m+1}^*) = \Phi_\sigma(S^M \cap S^*).$$

■

Also, by definition, summability (10) bounds the distance between the potential function (11) and the social cost (2) in the following sense.

Lemma 3.7 *For every subset $S \subseteq U$,*

$$\Phi_\sigma(S) \le v(U \setminus S) + \alpha(|S|) \cdot C(S).$$

We are now prepared to prove Theorem 3.4.

Proof of Theorem 3.4: Fix a universe U, a cost function C, and a set v of truthful bids. Let S^* be an efficient outcome. Let χ be an α-summable, no-deficit, cross-monotonic cost-sharing method for C and S^M the output of the corresponding Moulin mechanism $M(\chi)$ for the profile v. Define the player ordering σ and the potential function Φ_σ as in (11). We can then derive

$$\begin{aligned} v(U \setminus S^M) + C(S^M) &\le v(U \setminus S^M) + \sum_{i \in S^M} \chi(i, S^M) \\ &\le v(U \setminus S^M) + v(S^M \setminus S^*) + \sum_{i \in S^M \cap S^*} \chi(i, S^M) \\ &\le \Phi_\sigma(S^M \cap S^*) \\ &\le \Phi_\sigma(S^*) \\ &\le v(U \setminus S^*) + \alpha(|S^*|) \cdot C(S^*), \end{aligned}$$

where the first inequality follows from the no-deficit condition (8), the second from the fact that $\chi(i, S^M) \le v_i$ for every $i \in S^M$, the third from the cross-monotonicity of χ, the fourth from Lemma 3.6, and the fifth from Lemma 3.7. Rearranging terms then proves the theorem. ■

Remark 3.8 When the method χ is the Shapley cost-sharing method (see Section 4), our definition (11) of the potential function Φ_σ essentially coincides with that of Hart and Mas-Colell [21] for cooperative games.

Remark 3.9 The results of this section can be interpreted as efficiency guarantees for the noncooperative *participation games* studied by Monderer and Shapley [34] and Moulin [35]. For example, Corollary 3.5(a) implies that for the social cost objective (6), the "strong price of anarchy" [1] in such a game is at most the summability of the underlying cost-sharing method.

3.3 Matching Lower Bounds

We now discuss the senses in which the bound in Theorem 3.4 is tight. The argument in Example 3.1 implies the following lower bound for strictly positive cost-sharing methods.

Proposition 3.10 (Summability Lower Bounds Approximate Efficiency I) *Let χ be a cross-monotonic cost-sharing method for a cost-sharing problem C with universe U that is everywhere positive and at least α-summable. Then $M(\chi)$ is no better than $(\alpha(k) - 1)$-approximate, where k is the size of an efficient outcome.*

The assumption that all cost shares are positive is similar to the "strong consumer sovereignty" assumption in Moulin [35].

For technical reasons, summability need not lower bound the approximate efficiency of cost-sharing methods that can employ zero cost shares. To informally illustrate the issue, consider a cost-sharing problem with universe $U = \{1, 2, \ldots, n\}$ and two cost-sharing methods χ_1, χ_2 defined for the restriction of this problem to $U \setminus \{1\}$, where the summability of χ_2 is much larger than that of χ_1. Define χ on U by setting cost shares equals to those of χ_1 for sets that include the first player and equal to those of χ_2 for sets that do not; the first player always receives a zero cost share. The summability of χ is as large as that of χ_2, but the Moulin mechanism $M(\chi)$ will never delete the first player and will therefore only assign cost shares according to the method χ_1 that has small summability. Thus the summability of χ is strictly larger than the approximate efficiency of the induced Moulin mechanism.

There is nevertheless a variant of Proposition 3.10 for non-positive cost-sharing methods. To state it, note that a Moulin mechanism $M(\chi)$ for a cost-sharing problem naturally induces a Moulin mechanism for each induced subproblem (via the restriction of χ to the subproblem). We say that a Moulin mechanism $M(\chi)$ is *strongly ρ-approximate* if every induced mechanism is ρ-approximate for the corresponding induced cost-sharing problem. The proof of Theorem 3.4 extends directly to this notion of strong approximation.

Corollary 3.11 *Let C be a cost function defined on a universe U and χ a cross-monotonic, no-deficit, α-summable cost-sharing method for C. Then $M(\chi)$ is a strongly $(\alpha(k)-1)$-approximate mechanism, where k is the size of an efficient outcome.*

Summability is a valid lower bound for strong approximate efficiency, even for cost-sharing methods that use zero cost shares.

Proposition 3.12 (Summability Lower Bounds Approximate Efficiency II) *Let χ be a cross-monotonic cost-sharing method for a cost-sharing problem C with universe U that is at least α-summable. Then $M(\chi)$ is no better than strongly $(\alpha(k)-1)$-approximate, where k the size of an efficient outcome.*

Proof Sketch: Choose k, a set S with $|S| = k$, and an ordering of the players of S so that $\sum_{\ell=1}^{k} \chi(i_\ell, S_\ell) \geq \alpha(k) \cdot C(S)$, where S_ℓ and i_ℓ are defined in the usual way. Obtain R from S by discarding players with $\chi(i_\ell, S_\ell) = 0$. Since χ is cross-monotonic and C is nondecreasing, the induced ordering on R satisfies $\sum_{\ell=1}^{|R|} \chi(i_\ell, R_\ell) \geq \alpha(k) \cdot C(R)$ with all cost shares positive. Mimicking Example 3.1 in the problem induced by R, the welfare loss of the induced Moulin mechanism is at least $(\alpha(k)-1) \cdot C(R^*)$, where R^* denotes an optimal outcome to this induced problem. ■

The construction in Example 3.1 also demonstrates the tightness of the alternative guarantees in Corollary 3.5.

Proposition 3.13 *Let χ be a cross-monotonic cost-sharing method for a cost-sharing problem C with universe U that is everywhere positive and at least α-summable. Then:*

(a) $M(\chi)$ is no better than an $\alpha(k)$-approximation algorithm for minimizing social cost;

(b) $M(\chi)$ is no better than a $1/\alpha(k)$-approximation algorithm for maximizing social reward;

(c) there are $(\alpha(k)-1)$-separated valuation profiles for which $M(\chi)$ obtains zero welfare.

Similar results apply for non-positive cost-sharing methods and "strong" versions of these three types of efficiency guarantees.

4 Submodular Cost-Sharing Problems

This section illustrates our approximation framework using submodular cost-sharing problems. We show how existing results of Moulin and Shenker [36] imply approximation bounds in this special case, and also derive identical bounds using the summability approach of Section 3.

We first recall a well-known mechanism based on a generalization of the Shapley method χ_{sh} described in Example 2.8. Let C be a submodular cost function (recall (4)) defined on a player set U. The *Shapley cost share* $\chi_{sh}(i,S)$ of player i in the set S is defined as follows. For a permutation σ of the players of S, let $\Delta_\sigma(i)$ denote the increase $C(A \cup \{i\}) - C(A)$ in cost due to i's arrival, where $A \subseteq S$ is the set of players that precede i in σ. The Shapley cost share $\chi_{sh}(i,S)$ is then the expected value of $\Delta_\sigma(i)$, where the expectation is over the (uniform at random) choice of σ. As is well known and easily checked, Shapley cost shares are 1-budget-balanced, and are cross-monotonic when the function C is submodular. The corresponding Moulin mechanism $M(\chi_{sh})$ is called the *Shapley mechanism for C* [36].

Moulin and Shenker [36, Proposition 2] proved that, for every submodular cost function C defined on a universe U of n players, the corresponding Shapley mechanism minimizes the worst-case (over valuation profiles) additive welfare loss, over all 1-budget-balanced Moulin mechanisms. Precisely, they showed that this worst-case welfare loss, compared to an efficient solution, is at least

$$\sum_{S \subseteq U} \frac{(|S|-1)!(n-|S|)!}{n!} C(S) - C(U) \tag{12}$$

for every 1-budget-balanced Moulin mechanism, with equality holding for the Shapley mechanism. Since $C(S) \le C(U)$ for every $S \subseteq U$, the worst-case welfare loss for the Shapley mechanism is at most

$$C(U) \cdot \left(\sum_{|S|=1}^{n} \binom{n}{|S|} \frac{(|S|-1)!(n-|S|)!}{n!} \right) - C(U) = C(U) \cdot \left(\sum_{|S|=1}^{n} \frac{1}{|S|} \right) - C(U) = (\mathcal{H}_n - 1) \cdot C(U),$$

and thus this mechanism is at most $(\mathcal{H}_n - 1)$-approximate for every submodular cost-sharing problem. Since $C(S) = C(U)$ for every non-empty set $S \subseteq U$ in the excludable public good problem (Example 2.12), it provides a matching lower bound: there is submodular cost-sharing problem for which every 1-budget-balanced Moulin mechanism is no better than $(\mathcal{H}_n - 1)$-approximate.

These bounds can also be derived from summability arguments, and in the process extended to all no-deficit (not necessarily 1-budget-balanced) Moulin mechanisms. The lower bound is again for the special case of an excludable public good with n players. For every Moulin mechanism $M(\chi)$ induced by a cross-monotonic, no-deficit cost-sharing method χ, we can inductively order the players $1, 2, \ldots, n$ such that $\chi(i, \{i, i+1, \ldots, n\}) \ge 1/(n-i+1)$ for every i. Defining valuations as in Example 3.1 then shows that $M(\chi)$ is no better than $(\mathcal{H}_n - 1)$-approximate.

To obtain an upper bound of $(\mathcal{H}_k - 1)$ for the approximation factor of the Shapley mechanism, where k is the number of players served in an optimal solution, fix a submodular cost function C with players U, with χ_{sh} the corresponding Shapley cost-sharing method. By Definition 3.2 and Theorem 3.4, we only need to show that

$$\sum_{\ell=1}^{|S|} \chi_{sh}(i_\ell, S_\ell) \le \mathcal{H}_{|S|} \cdot C(S) \tag{13}$$

for every $S \subseteq U$ and ordering σ of U, where S_ℓ and i_ℓ are defined in the usual way. A remarkable result of Hart and Mas-Colell [21, Footnote 7], a variant of which is also used in [36] to establish (12), implies that the left-hand side of (13) is *independent of the ordering* induced by σ on the players of S. (This can also be established directly by a counting argument.) Choosing an ordering of the players of S uniformly at random, the facts that C is nondecreasing and χ_{sh} is 1-budget-balanced imply that $E[\chi_{sh}(i_\ell, S_\ell)] = E[C(S_\ell)]/\ell \leq C(S)/\ell$ for each ℓ. Summing over all ℓ and using the linearity of expectation shows that the expected value under a random ordering (and hence the value under every ordering) of the left-hand side of (13) is at most $\mathcal{H}_{|S|} \cdot C(S)$, completing the argument.

Remark 4.1 While the approximation bound of $\mathcal{H}_k - 1$ is tight for an excludable public good, both of the derivations above can obviously be sharpened for particular cost functions. For example, for the cost function $C(S) = |S|^d$ with $d \in (0,1]$ and n large, the Shapley mechanism remains optimal and is roughly $(\frac{1}{d} - 1)$-approximate. See Brenner and Schäfer [8] for a related discussion.

Remark 4.2 Computational complexity is not a focus of this paper, but we note in passing that Shapley cost shares are generally hard to compute, in myriad senses, even for monotone and submodular cost functions [4]. The following randomized variant of the Shapley cost-sharing method is polynomial-time computable, cross-monotonic with probability 1, and arbitrarily close to $\mathcal{H}_k$-summable with high probability: choose in advance a sufficiently large polynomial number of player permutations uniformly at random, and estimate every expectation of the form $E[\Delta_\sigma(i)]$ by the average value of $\Delta_\sigma(i)$ over the randomly chosen permutations.

5 Steiner Tree Cost-Sharing Problems

This section uses the summability framework of Section 3 to prove matching upper and lower bounds on the best-possible approximate efficiency of no-deficit Moulin mechanisms for Steiner tree cost-sharing problems. Both the upper and lower bounds are much more intricate than for submodular cost-sharing problems. Section 5.1 reviews a mechanism of Jain and Vazirani [24], and Section 5.2 proves that this mechanism is $O(\log^2 k)$-approximate for all Steiner tree problems. Section 5.3 proves that this mechanism is optimally approximately efficient (up to constant factors).

5.1 The JV Steiner Tree Mechanism

Recall that a Steiner tree cost-sharing problem (Example 2.2) is defined via an undirected graph $G = (V, E)$ with nonnegative edge costs, a root vertex t, and a set U of players that inhabit the vertices of G. The cost $C(S)$ of a subset $S \subseteq U$ is defined as the cost of an optimal Steiner tree of G that spans $S \cup \{t\}$. Such cost functions are not generally submodular, and the corresponding Shapley cost-sharing methods are not generally cross-monotonic. Several researchers have designed 2-budget-balanced and cross-monotonic Steiner tree cost-sharing methods [24, 25, 29], and no cross-monotonic method can have better budget-balance [23, 29]. We work with the first of these, designed by Jain and Vazirani [24].

Put succinctly, the *JV cost-sharing method* χ_{JV} for a Steiner tree problem is defined by equally sharing the dual growth that occurs in Edmonds's primal-dual branching algorithm [14]. In more detail, this method works as follows.

First, given a subset $S \subseteq U$, form a complete directed graph $H = (V_H, A_H)$. The vertices V_H are t and the vertices of G that contain at least one player of S. The cost c_{uw} of an arc (u, w) of H equals the length of a minimum-cost u-w path in G. (Since G is undirected, arcs (u, w) and (w, u) of H have equal cost.) We then define both a feasible Steiner tree and cost shares using *Edmonds's algorithm*, as follows. Initialize a timer to time $\tau = 0$ and increase time at a uniform rate. Initialize a subset $F \subseteq A_H$ to $\emptyset$. At every moment in time, the algorithm increases at unit rate a variable y_A for every weakly connected component A of (V_H, F) other than the one containing the root t. When an inequality of the form $\sum_{A \subseteq V_H \,:\, u \in A, w \notin A} y_A \le c_{uw}$ first holds with equality, the corresponding arc (u, w) is added to F and the algorithm continues. (When this occurs for several inequalities simultaneously, all of the corresponding arcs are added.) When the algorithm terminates, the graph (V_H, F) contains a directed path from every vertex to the root t. To obtain a subgraph of G that spans t and the players of S, select an arbitrary branching B (a spanning tree directed toward t) of (V_H, F) and output the union of the minimum-cost paths of G that correspond to the arcs of B. To obtain cost shares, let u_i denote the vertex of V_H at which player i resides and set

$$\chi_{JV}(i, S) = \sum_{A \subseteq V_H \,:\, u_i \in A} \frac{y_A}{\kappa(A)},$$

where $\kappa(A)$ is the population of S in A. Equivalently, cost shares can be defined in tandem with the above algorithm: whenever a variable y_A is increased, this increase is distributed equally among the cost shares of the players of S contained in A.

Jain and Vazirani [24] proved that the method χ_{JV} is cross-monotonic and 2-budget-balanced in the sense of the inequalities (8). The next proposition summarizes the additional properties of the JV cost-sharing method that are important for bounding its summability. To state them, we say that a player $i \in S$ is *active at time* τ in Edmonds's algorithm if it is not in the same weakly connected component as the root t at time τ. The *activity time* of a player is the latest moment in time at which it is active. The notation $d_G(i, j)$ refers to the minimum cost of an i-j path in the graph G.

Proposition 5.1 *Let $G = (V, E)$ be a Steiner tree instance with root t and player set S.*

(a) While player i is active in Edmonds's algorithm and belongs to a component with $m - 1$ other (active) players, it accumulates an instantaneous cost share of $\frac{dt}{m}$. The final JV cost share for player i equals the integral of its instantaneous cost share up to its activity time.

(b) The activity time of a player $i \in S$ in Edmonds's algorithm is at most the length of a shortest i-t path in G.

(c) For every pair $i, j \in S$, by the time $d_G(i, j)$ in Edmonds's algorithm, players i and j are in the same weakly connected component.

Proposition 5.1 follows easily from the definition of Edmonds's algorithm and the JV cost shares.

5.2 The JV Mechanism is $O(\log^2 k)$-Approximate

Our main result in this section is that, for every Steiner tree cost-sharing problem, the Moulin mechanism induced by the corresponding JV method is $O(\log^2 k)$-approximate.

Theorem 5.2 *There are constants $a, b > 0$ such that the following statement holds: for every Steiner tree cost-sharing problem, the Moulin mechanism induced by the corresponding JV method is $(a \log^2 k + b)$-approximate, where k is the size of an efficient outcome.*

Next we discuss our high-level proof approach. By Theorem 3.4, we need to show that

$$\sum_{\ell=1}^{|S|} \chi_{JV}(i_\ell, S_\ell) = O(\log^2 |S|) \cdot C(S)$$

for every Steiner tree problem C with JV method χ_{JV}, every subset S of players, and every ordering of the players (where i_ℓ and S_ℓ are defined in the usual way). The challenge in proving this stems from the adversarial ordering of the players (cf., Example 5.9 below). Our proof of Theorem 5.2 resolves this difficulty with the following three-step approach. First, we build a tree T on the player set, with the same root as the given Steiner tree problem, that intuitively "inverts" an arbitrary ordering so that players closer to the root in T appear earlier in the ordering than their descendants. We pay a price for this inversion: the sum of the edge costs of T is $O(\log |S|)$ times the cost of an optimal Steiner tree.

In the second step we define "artificial cost shares" for the players. These cost shares will approximate the JV cost shares of players in G, but it will also be straightforward to upper bound their sum. More precisely, we define the artificial cost share of the ith player (according to the given adversarial ordering) as its Shapley cost share in the tree T, assuming that precisely the first i players are present. By inequality 13, the sum of these artificial cost shares is at most $\mathcal{H}_{|S|}$ times the sum of the edge costs of T, which in turn is $O(\log^2 |S|)$ times the cost of an optimal Steiner tree in G.

In the third step, we prove that Shapley cost shares in T approximate JV cost shares in G: for every player, the former is at least a constant fraction of the latter. We feel that this final step is by far the most surprising, as it relates two sets of cost shares that are defined by different methods as well as in different graphs. This final step uses properties of both the JV dual growth process and the edge cost structure in the tree T.

We now supply the details. Fix a Steiner tree cost-sharing problem with universe U, graph $G = (V, E)$ with edge costs c, and root vertex $t \in V$. We begin with the construction of the tree T, given a subset $S \subseteq U$ of players and an ordering σ of the players. The tree T will contain a root vertex t_0 that corresponds to t, and will contain one additional vertex for each player in S. We sometimes refer to a non-root node of T and to the corresponding player of S in G interchangeably.

Each vertex $i \neq t_0$ of T will be associated with a *radius* r_i that serves distinct purposes in the tree T and the original graph G. First, the edge from i to its parent in T will have cost r_i. Second, r_i will denote the radius of a ball B_i in the graph G centered at the player i. These balls will be used to determine ancestor-descendant relationships in T.

We initialize the tree T to contain only the root vertex t_0. We give t_0 a radius of $+\infty$, and the ball B_{t_0} of t_0 is defined as the entire player set S. We then add players of S to the tree T one-by-one, in the order prescribed by σ. When adding a new player i, we consider all of the balls of previously added players that contain i. If nothing else, the ball B_{t_0} contains i. Among all such balls, let B_j be one of minimum radius r_j. First, we add the node i to the tree T by making i a child of j. Second, we define the radius r_i as follows. If $j = t_0$, then r_i is half the shortest-path distance between the root t and the player i in the graph G. If $j \neq t_0$, then we define $r_i = r_j/2$.

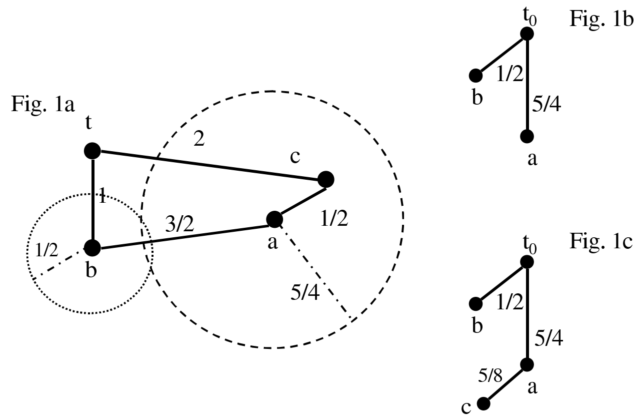

Figure 1: Proof of Theorem 5.2: the construction of the tree T (Figs. 1b and 1c) from the graph G (Fig. 1a) and ordering $\sigma = a, b, c$ of the players. Fig. 1b depicts T after players a and b have been considered, and Fig. 1a shows the balls corresponding to these players.

Third, we set the cost of the edge (i, j) in T to be this radius r_i. Finally, we define the ball B_i of player i to be the players of S that lie within distance r_i of i in the graph G. See also Figure 1.

To begin, we record some simple relations between shortest-path distances in T and in G; we omit the simple proofs.

Lemma 5.3 *Let i, j be a pair of vertices in T and P_{ij} the unique i-j path in T.*

(a) The cost of P_{ij} is at most four times the cost of its most expensive edge.

(b) The cost of P_{ij} is at least $d_G(i, j)/2$.

Now let OPT denote the cost of a minimum-cost Steiner tree in G that spans $S \cup \{t\}$. We next give a series of three lemmas, culminating in a proof that the sum of the costs of the edges of T exceeds OPT by an $O(\log |S|)$ factor. The first lemma states that two edges of the tree T that have roughly equal cost correspond to well-separated players in the graph G; it follows easily from the way we construct T.

Lemma 5.4 *Suppose (i_1, j_1) and (i_2, j_2) are edges of T, directed toward the root t_0, with costs c_1 and c_2, respectively. If $c_1 \leq c_2 < 2c_1$, then $d_G(i_1, i_2) \geq c_1$.*

We next show how to use Lemma 5.4 to upper bound the number of edges of T with cost in a given range.

Lemma 5.5 *For every $\nu \geq 1$, the number of edges of T that have cost in the interval $[OPT/\nu, 2OPT/\nu)$ is at most 2ν.*

Proof: Fix $\nu \geq 1$ and suppose that q edges of T have cost at least OPT/ν and less than $2OPT/\nu$. Lemma 5.4 implies that there is a set $A \subseteq S$ of q players that are mutually far apart in G: $d_G(i, i') \geq OPT/\nu$ for every pair i, i' of distinct players of A.

Consider an optimal Steiner tree T^* in G that spans $S \cup \{t\}$ (with cost OPT). Order the players of $A = \{i_1, \ldots, i_q\}$ according to a pre-order traversal of T^* (starting from the root, say).

As is well known, we can double every edge of T^* and decompose the resulting multigraph into a collection of paths that connect pairs of adjacent players (including i_1 and i_q). This proves that $\sum_{j=1}^{q} d_G(i_j, i_{j+1}) \leq 2OPT$, where i_{q+1} refers to player i_1. Thus $d_G(i_j, i_{j+1}) \leq 2OPT/q$ for some $j \in \{1, 2, \ldots, q\}$. Since $d_G(i, i') \geq OPT/\nu$ for every $i, i' \in A$, $q \leq 2\nu$. ■

We now combine Lemma 5.5 with a grouping argument to upper bound the sum of the edge costs in the tree T.

Lemma 5.6 *The sum of the costs of the edges in T is at most $(4 \log_2 |S| + 5) \cdot OPT$.*

Proof: First, note that every edge cost in T is bounded above by the distance $d_G(i, t)$ in G between the root t and some player i of S. Since every such distance is a lower bound on OPT, every edge of T has cost at most OPT.

Next, let $k = |S|$ and consider the edges with cost in the interval $[2^i OPT/k, 2^{i+1} OPT/k)$ for some $i \in \{0, 1, \ldots, \lfloor \log_2 k \rfloor\}$. By Lemma 5.5, there are at most $k/2^{i-1}$ edges in this group. The sum of the edge costs in each of the $\lceil \log_2 k \rceil$ groups is therefore at most $4OPT$. Since T has $k + 1$ vertices, it has k edges, and thus the total cost of the edges not in any of these groups — each of which has cost less than OPT/k — is at most OPT. Summing over all of the edges proves the lemma. ■

Next, let $\chi^T_{sh}(i_\ell, S_\ell)$ denote the Shapley cost share of the ℓth player (in the given ordering σ) in the fixed-tree multicast instance corresponding to the tree T and the set S_ℓ of the first ℓ players according to σ. Since fixed-tree multicast cost-sharing problems are submodular (Example 2.1), inequality (13) and Lemma 5.6 immediately give the following upper bound on the sum of these Shapley cost shares.

Lemma 5.7 *Let i_ℓ denote the ℓth player and S_ℓ the first ℓ players of S according to σ, respectively. Then*

$$\sum_{\ell=1}^{|S|} \chi^T_{sh}(i_\ell, S_\ell) \leq (\ln |S| + 1) \cdot (4 \log_2 |S| + 5) \cdot OPT.$$

Finally, we show that the JV cost share of a player in G is at most a constant factor times its Shapley cost share in T. This is the step of the proof of Theorem 5.2 where we use specific properties of the JV cost-sharing method (Proposition 5.1).

Lemma 5.8 *Let i_ℓ denote the ℓth player and S_ℓ the first ℓ players of S according to σ, respectively. For every $\ell \in \{1, 2, \ldots, |S|\}$,*

$$\chi_{JV}(i_\ell, S_\ell) \leq 8 \cdot \chi^T_{sh}(i_\ell, S_\ell).$$

Proof: Fix $\ell \in \{1, 2, \ldots, |S|\}$ and let $e_1, e_2, \ldots, e_p$ denote the sequence of edges in the i_ℓ-t_0 path in T. Let c_j denote the cost of edge e_j. Let $A_j \subseteq S_\ell$ denote the players of S_ℓ whose path to t_0 in T contains the edge e_j. Let m_j denote the number $|A_j|$ of such players.

Our tree construction ensures that children of i_ℓ correspond only to players subsequent to i_ℓ in the ordering σ, and no such players are in S_ℓ. Thus $A_1 = \{i_\ell\}$, and of course $A_1 \subseteq \cdots \subseteq A_p \subseteq S_\ell$. First, observe that

$$\chi^T_{sh}(i_\ell, S_\ell) = \sum_{j=1}^{p} \frac{c_j}{m_j}. \tag{14}$$

Next, fix $j \in \{2, 3, \ldots, p\}$ and consider a player $i \in A_j$ distinct from i_ℓ. Since the edge e_j separates players i and i_ℓ from t_0 in T, the most expensive edge on the i_ℓ-i path P in T has cost at most c_{j-1}. By Lemma 5.3(a), the path P has cost at most $4c_{j-1}$. By Lemma 5.3(b), the distance $d_G(i_\ell, i)$ between the players in G is at most $8c_{j-1}$. By Proposition 5.1(c), the players i_ℓ and i are in a common connected component by the time $8c_{j-1}$ in the execution of Edmonds's algorithm that defines the JV cost share $\chi_{JV}(i_\ell, S_\ell)$. Crucially, it follows that if player i_ℓ is active at a time subsequent to $8c_{j-1}$ in this execution, then its weakly connected component at this time does not contain the root t and contains at least the m_j (active) players of A_j. Similarly, Lemma 5.3 and Proposition 5.1(b) imply that player i_ℓ is inactive by the time $8c_p$.

Combining these observations with Proposition 5.1(a), we obtain

$$\chi_{JV}(i_\ell, S_\ell) \le \sum_{j=1}^{p} \int_{8c_{j-1}}^{8c_j} \frac{dt}{m_j} \le 8 \sum_{j=1}^{p} \frac{c_j}{m_j}, \tag{15}$$

where we are interpreting c_0 as 0. Comparing equality (14) and inequality (15) proves the lemma. ■

Theorem 5.2 now follows immediately from Lemma 5.7, Lemma 5.8, and Theorem 3.4.

5.3 Every Moulin Mechanism is $\Omega(\log^2 k)$-Approximate

This section proves that the JV mechanism is an *optimal* Moulin mechanism for Steiner tree cost-sharing problems, in the sense that every no-deficit mechanism for such problems is $\Omega(\log^2 k)$-approximate, where k is the size of an efficient outcome. To motivate our proof of this result, we begin with an example showing that our analysis of the JV mechanism is tight up to constant factors.

Example 5.9 We construct a Steiner tree instance in rounds by iteratively bisecting an edge of cost 1 as follows. Initially we place the root t at one end of the edge and $\sqrt{n}$ players at the other end of the edge. (Think of n as a large power of 4.) In the second round, we bisect the edge with a new vertex in the middle and add $\sqrt{n}$ further players co-located at this vertex. In round j, we bisect the existing 2^{j-1} edge segments and, for each new node, we add $\sqrt{n}$ new co-located players. The construction concludes when there are n players, after $\Theta(\log n)$ rounds.

Order the players in the same order in which they were added during the construction; break ties among players added in the same round arbitrarily. This defines n successive Steiner tree instances. Consider the cost share of the most recently added player of one of these instances. The JV cost-sharing method satisfies the following property: if a player is co-located with $i - 1$ other players (all added earlier) and is distance c away from the nearest non-co-located player that was added in an earlier round, then its cost share in this instance is $\Omega(c/i)$. Because of this, the sum of the cost shares of players added in the jth round of the above construction is $\Omega(\log n)$. Since there are $\Omega(\log n)$ rounds, the sum of all of these successive cost shares is $\Omega(\log^2 n)$. Since the minimum-cost Steiner tree of the full instance has cost 1 and the JV cost-sharing method is positive in this instance, Proposition 3.10 implies that the induced Moulin mechanism is $\Omega(\log^2 n)$-approximate.

The main result of this section is a comparable lower bound for *every* $O(1)$-budget-balanced Moulin mechanism.

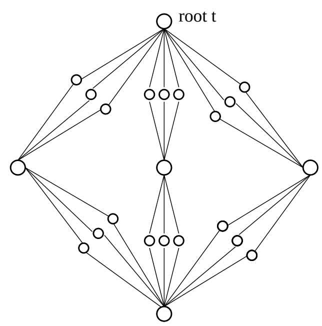

Figure 2: Network G_2 in the proof of Theorem 5.10, with $m = 3$. All edges have length 1/4.

Theorem 5.10 *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 no better than strongly $(c \log^2 k)$-approximate, where k is the number of players served in an optimal outcome.*

Theorem 5.10 implies that Steiner tree cost-sharing problems are fundamentally more difficult for Moulin mechanisms than submodular cost-sharing problems (cf., Section 4).

We now outline the proof of Theorem 5.10. At the highest level, our goal is to exhibit a (large) network G such that every $O(1)$-budget-balanced Steiner tree Moulin mechanism behaves like the JV mechanism in Example 5.9 on some subnetwork of G.

Fix values for the parameters k and β, where k is a power of 4. Let m be an integer with $m \geq 8\beta\sqrt{k} \cdot (2\beta)^{\sqrt{k}}$. 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. For $j > 0$, we obtain the network G_j from G_{j-1} by replacing each edge (u, w) of G_{j-1} with m internally disjoint two-hop paths between u and w. See Figure 2. The cost of each of these $2m$ edges is half of the cost of the edge (u, w). Thus every edge in G_j has cost 2^{-j}.

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; call each of these groups a *j-group* and denote the union of them by U_j. The final network G is then G_p, where $p = (\log_2 k)/2$. 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.

A *line* in G_j is a subgraph defined inductively as follows. The only line in G_0 is all of G_0. Each line L_{j-1} of G_{j-1} gives rise to a set of m^{2^j} lines in G_j, each obtained by replacing each edge of L_{j-1} by one of the m two-hop paths to which it corresponds in G_j. Every line in the network G_j has 2^j vertices other than the root, 2^j edges, and unit total cost. In G_p, $\sqrt{k}$ players inhabit each of the $2^p = \sqrt{k}$ non-root vertices on a line.

Now fix an arbitrary cross-monotonic, β-budget balanced Steiner tree cost-sharing method χ. Our plan is to identify a line of G_p and an ordering of the players on this line such that χ behaves like the JV cost-sharing method in Example 5.9. We construct this line iteratively via the following key technical lemma.

Lemma 5.11 *Let $S \subseteq U$ be a subset of players that lies on a line in G_p, includes at least one player of U_0, and includes at least one player each from a pair u, w of vertices that are adjacent in*

G_{j-1}. Let $A_1, \ldots, A_m$ denote the j-groups that correspond to the edge (u, w). Then for some group A_q, its players can be ordered $i_1, \ldots, i_{\sqrt{k}}$ so that

$$\chi(i_\ell, S \cup \{i_1, \ldots, i_\ell\}) \geq \frac{2^{-j}}{4\ell} \tag{16}$$

for every $\ell \in \{1, 2, \ldots, \sqrt{k}\}$.

Before proving Lemma 5.11, we use it to prove Theorem 5.10 by inductively constructing player sets $S_0, \ldots, S_p$ and orderings $\sigma_0, \ldots, \sigma_p$ with the following properties.

(1) For every $j \in \{0, 1, 2, \ldots, p\}$, S_j corresponds to the $\sqrt{k} \cdot 2^j$ players occupying some line L_j of G_j.

(2) σ_j is an ordering of S_j that orders the $\sqrt{k}$ players of each of its j-groups A consecutively and in a way that (16) holds with S equal to the predecessors of A in σ_j.

For the base case, set $S_0 = U_0$. Since χ is no-deficit, the players of S_0 can be ordered $i_1, \ldots, i_{\sqrt{k}}$ so that $\chi(i_\ell, \{i_1, \ldots, i_\ell\}) \geq C(\{i_1, \ldots, i_\ell\})/\ell = 1/\ell$ for every ℓ. Let σ_0 denote this ordering of S_0.

For the inductive step, let L_{j-1} be the line of G_{j-1} corresponding to S_{j-1}, and consider the edges of L_{j-1} in an arbitrary order. Each such edge gives rise to m j-groups; applying Lemma 5.11 with S equal to the players already chosen (in this and previous steps), one of these j-groups can be ordered so that (16) holds. Add an arbitrary such group to the player set, ordered after all previously chosen players and so that (16) holds. After all of the edges of L_{j-1} have been processed, we obtain a player set S_j and ordering σ_j of them that satisfy the inductive invariants (1) and (2).

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_p with respect to σ_p, respectively. For $j > 0$, the 2^{j-1} j-groups of S_p each contribute at least

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

to this sum; the 0-group S_0 also contributes at least this amount. Thus the sum $\sum_{\ell=1}^{k} \chi(i_\ell, S_\ell)$ is at least

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

for a constant $c > 0$ that is independent of k. This, combined with Proposition 3.12, completes the proof of Theorem 5.10.

To conclude, we provide a proof of Lemma 5.11.

Proof of Lemma 5.11: Let $A_1^1, \ldots, A_m^1$ denote the j-groups corresponding to the edge (u, w) of G_{j-1} and set $X^1 = \cup_{r=1}^{m} A_r^1$. The proof plan is to inductively identify subcollections of these j-groups such that inequality (16) holds for an increasing number of the players in the remaining j-groups. Toward this end, call a set A_r^1 *1-eligible* if

$$\sum_{i \in A_r^1} \chi(i, S \cup X^1) \geq \frac{2^{-j}}{4}. \tag{17}$$

Every 1-eligible group contains a player i for which $\chi(i, S \cup X^1) \geq 2^{-j}/4\sqrt{k}$.

Our key claim is that at least $m/2\beta$ groups are 1-eligible. We prove this claim via an averaging argument that relies on the β-budget-balance and cross-monotonicity of χ. Precisely, reindex the 1-eligible groups $A_1^1, \ldots, A_q^1$ and let Y^1 denote their union. An optimal Steiner tree spanning $S \cup Y^1$ consists of a line through S and one group of Y^1, plus $q-1$ "spokes" attaching the rest of the groups to either u or w. Thus $C(S \cup Y^1) = 1 + (q-1)2^{-j}$. Since χ is cross-monotonic and β-budget-balanced, we have

$$\sum_{i \in S \cup Y^1} \chi(i, S \cup X^1) \leq \sum_{i \in S \cup Y^1} \chi(i, S \cup Y^1) \leq \beta(1 + (q-1)2^{-j}).$$

Since (17) fails for ineligible groups, and there at most m such groups,

$$\sum_{i \in X^1 \setminus Y^1} \chi(i, S \cup X^1) \leq \frac{m2^{-j}}{4}.$$

On the other hand, since $C(S \cup X^1) = 1 + (m-1)2^{-j}$ and χ is no-deficit,

$$\sum_{i \in S \cup X^1} \chi(i, S \cup X^1) \geq 1 + (m-1)2^{-j}.$$

Combining these three inequalities and rearranging gives the constraint

$$q \geq \frac{3m}{4\beta} - 2^j - \frac{1}{\beta} \geq \frac{m}{2\beta},$$

where the second inequality holds because m is sufficiently large.

Now we iterate the process. In more detail, obtain A_r^2 from each 1-eligible group A_r^1 by removing a player i for which $\chi(i, S \cup X^1) \geq 2^{-j}/4\sqrt{k}$. (Such a player must exist by 1-eligibility.) Let X^2 denote the union of these sets. Such a set A_r^2 is *2-eligible* if

$$\sum_{i \in A_r^2} \chi(i, S \cup X^2) \geq \frac{2^{-j}}{4}.$$

Every 2-eligible j-group contains a player i for which $\chi(i, S \cup X^2) \geq 2^{-j}/4(\sqrt{k}-1)$. Arguing as above, at least a $1/2\beta$ fraction of the sets A_r^2 are 2-eligible.

Iterating this procedure and reindexing the eligible groups after each iteration, we inductively obtain a collection of disjoint sets $A_1^h, \ldots, A_{q_h}^h$ for each $h \in \{1, 2, \ldots, \sqrt{k}\}$ with the following properties:

(1) $q_h \geq m/(2\beta)^h$;

(2) for each $r \in \{1, \ldots, q_h\}$, A_r^h contains a player i_r^h such that $\chi(i_r^h, S \cup X_h) \geq 2^{-j}/4(\sqrt{k} - h + 1)$, where $X_h = \cup_r A_r^h$;

(3) for each $r \in \{1, \ldots, q_h\}$ and $h > 1$, $A_r^h = A_r^{h-1} \setminus \{i_r^{h-1}\}$.

Since m is sufficiently large, $q_{\sqrt{k}} \geq 1$. By properties (2) and (3) and cross-monotonicity of χ, the group A_1^1 that corresponds to $A_1^{\sqrt{k}}$ satisfies the requirements of the lemma. ■

6 Budget-Balance vs. Economic Efficiency Trade-Offs

No-deficit Moulin mechanisms are inefficient because of their overzealous removal of players that cannot pay their cost share (cf., Examples 2.12 and 3.1). This suggests a possible trade-off between budget-balance and economic efficiency: if we relax the requirement that the prices charged cover the cost incurred, then a Moulin mechanism can employ smaller cost shares and reduce the worst-case efficiency loss from regrettable player deletions. This section extends the efficiency guarantees of Section 3 to mechanisms that need not cover the incurred cost, and uses these extensions to quantify the trade-off between budget-balance and economic efficiency in Moulin mechanisms for submodular and Steiner tree cost-sharing problems. In particular, we show that relaxing budget-balance permits mechanisms with strictly better efficiency guarantees than those possible for no-deficit Moulin mechanisms.

Recall that a Moulin mechanism is (β, γ)-*budget-balanced* if the sum of the prices charged is at least $1/\gamma$ and at most β times the incurred service cost. When $\gamma > 1$, Moulin mechanisms can suffer efficiency loss from the unjustified service of players with low valuations. (See Example 6.5 below.) For this reason, an efficiency guarantee for a (β, γ)-budget-balanced Moulin mechanism must reference both the parameter γ and the summability of its underlying cost-sharing method. We provide such a guarantee next.

Theorem 6.1 *Let C be a cost function defined on a universe U and χ a cross-monotonic, (β, γ)-budget-balanced, α-summable cost-sharing method for C. Let S^M and S^* denote the outcome chosen by $M(\chi)$ and an optimal outcome, respectively, for a valuation profile v. Then,*

$$W(S^*) - W(S^M) \leq (\alpha(|S^*|) - 1 + \beta(\gamma - 1)) \cdot C(S^*) + (\gamma - 1) \cdot v(S^M \setminus S^*).$$

Proof: Define an ordering σ on U and a potential function Φ_σ as in the proof of Theorem 3.4. By following the steps in that proof and using the (β, γ)-budget-balance of χ, we obtain

$$\begin{aligned}
v(U \setminus S^M) + C(S^M) &\leq v(U \setminus S^M) + \gamma \sum_{i \in S^M} \chi(i, S^M) \\
&\leq v(U \setminus S^M) + \gamma \cdot v(S^M \setminus S^*) + \gamma \sum_{i \in S^M \cap S^*} \chi(i, S^M) \\
&\leq \Phi_\sigma(S^M \cap S^*) + (\gamma - 1) \cdot v(S^M \setminus S^*) + (\gamma - 1) \sum_{i \in S^M \cap S^*} \chi(i, S^M \cap S^*) \\
&\leq \Phi_\sigma(S^*) + (\gamma - 1) \cdot v(S^M \setminus S^*) + (\gamma - 1)\beta \cdot C(S^*) \\
&\leq (\alpha(|S^*|) + \beta(\gamma - 1)) \cdot C(S^*) + v(U \setminus S^*) + (\gamma - 1) \cdot v(S^M \setminus S^*).
\end{aligned}$$

Rearranging terms proves the theorem. ∎

Theorem 6.1 is tight in the following sense: for every choice of values for the parameters $\alpha(|S^*|)$, β, and γ, there is a cost-sharing problem and a Moulin mechanism with these parameter values such that its welfare loss can be arbitrarily close to the upper bound in Theorem 6.1.

Like Theorem 3.4, the guarantee on additive welfare loss in Theorem 6.1 can be interpreted in several different ways. We mention only the cleanest such interpretation, in terms of minimizing the social cost objective (2).

Corollary 6.2 *Let C be a cost function defined on a universe U and χ a cross-monotonic, (β, γ)-budget-balanced, α-summable cost-sharing method for C. Then $M(\chi)$ is a $(\max\{\alpha(k)+\beta(\gamma-1), \gamma\})$-approximation algorithm for the social cost objective, where k is the size of an efficient outcome.*

For example, for a submodular cost-sharing problem with n players, dividing the corresponding Shapley cost shares by an $\sqrt{\mathcal{H}_n}$ factor yields a $(1, \sqrt{\mathcal{H}_n})$-budget-balanced and $\sqrt{\mathcal{H}_n}$-summable cost-sharing method. Corollary 6.2 implies the following guarantee for the induced Moulin mechanism (the *scaled Shapley mechanism*).

Corollary 6.3 *For every n-player submodular cost-sharing problem, the scaled Shapley mechanism is $(1, \sqrt{\mathcal{H}_n})$-budget-balanced and a $(2\sqrt{\mathcal{H}_n}-1)$-approximation algorithm for the social cost objective.*

The *scaled JV mechanism* is defined by dividing the JV cost shares by a $\Theta(\log n)$ factor.

Corollary 6.4 *For every n-player Steiner tree cost-sharing problem, the scaled JV mechanism is $(1, O(\log n))$-budget-balanced and an $O(\log n)$-approximation algorithm for the social cost objective.*

The efficiency guarantees in Corollaries 6.3 and 6.4 are better than the best possible for no-deficit Moulin mechanisms (Section 4 and Theorem 5.10).

Corollaries 6.3 and 6.4 are optimal efficiency guarantees in the following sense. First, a simple example shows that a Moulin mechanism that is no better than (β, γ)-budget-balanced is no better than a γ-approximation algorithm for the social cost.

Example 6.5 Let χ be a cross-monotonic cost-sharing method for a cost function C defined on a universe U, and suppose that χ is no better than (β, γ)-budget-balanced for C. By definition, there is a subset $S \subseteq U$ of players with $\sum_{i \in S} \chi(i, S) \leq C(S)/\gamma$. Give each player $i \in S$ the valuation $\chi(i, S)$ and other players zero valuations. With this valuation profile, the Moulin mechanism $M(\chi)$ outputs a set containing all of the players of S, with social cost at least $C(S)$. The optimal social cost is at most that of the empty set, which is at most $C(S)/\gamma$.

Second, the lower bound proofs in Section 4 and of Theorem 5.10 extend easily to show that all (β, γ)-budget-balanced Moulin mechanisms for submodular and Steiner tree cost-sharing problems are $\Omega((\log k)/\gamma)$- and $\Omega((\log^2 k)/\gamma)$-approximation algorithms for the social cost, respectively. Thus no Moulin mechanism, no matter how poor its budget-balance, obtains an $o(\sqrt{\log k})$-approximation of the social cost for submodular problems or an $o(\log k)$-approximation of the social cost for Steiner tree problems.

7 Recent Work and Future Directions

We have developed an analysis framework for quantifying efficiency loss in Moulin mechanisms, and applied this framework to identify the best efficiency guarantees achievable by such mechanisms in submodular and Steiner tree cost-sharing problems, and to rigorously quantify the feasible trade-offs between efficiency and budget-balance. We conclude by discussing some of the very recent work motivated by the conference version of this paper [41], and some possible directions for future research.

The most obvious open problems suggested by our analysis framework are to establish matching upper and lower bounds on the best-possible efficiency guarantees achievable by Moulin mechanisms for additional classes of fundamental cost-sharing problems. Recent work has accomplished this for all of the classes of cost-sharing problems for which $O(1)$-budget-balanced Moulin mechanisms are known: facility location problems [42]; Steiner forest network design problems [9] and a prize-collecting generalization [19]; rent-or-buy network design problems [42]; and scheduling problems [8].

An important direction for further work is the design of non-Moulin cost-sharing mechanisms. Mehta, Roughgarden, and Sundararajan [33] recently extended Moulin mechanisms to a wider class they call "acyclic mechanisms", and prove that for several classes of cost-sharing problems, acyclic mechanisms can obtain approximate budget-balance and/or efficiency guarantees superior to those possible for Moulin mechanisms. Open problems include finding applications of acyclic mechanisms to new classes of cost-sharing problems, and generalizing acyclic mechanisms even further. On the negative side, Dobzinski et al. [13] recently proved that the $\Omega(\log n)$ approximate efficiency lower bound of Section 4 is inescapable, even for mechanisms that are only truthful in expectation and approximately budget-balanced.

A final direction is to completely characterize different classes of strategyproof cost-sharing mechanisms. Thus far, Moulin [35] characterized the groupstrategyproof (GSP) and 1-budget-balanced mechanisms (recall from Remark 2.11 that all Moulin mechanisms are GSP); Immorlica, Mahdian, and Mirrokni [23] partially characterized GSP mechanisms without any budget-balance assumptions; and Juarez [26] very recently made progress toward characterizing "weakly GSP" cost-sharing mechanisms, a class that includes the acyclic mechanisms of [33]. Leveraging these characterizations to obtain matching upper and lower bounds on the best-possible budget-balance and efficiency guarantees achievable by the corresponding class of cost-sharing mechanisms is a worthy challenge for future research.

References

[1] N. Andelman, M. Feldman, and Y. Mansour. Strong price of anarchy. *Games and Economic Behavior*, 2009. To appear.

[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] S. Arora and C. Lund. Hardness of approximations. In D. S. Hochbaum, editor, *Approximation Algorithms for NP-Hard Problems*, chapter 10, pages 399–446. PWS Publishing Company, 1997.

[4] Y. Bachrach, E. Markakis, A. D. Procaccia, J. S. Rosenschein, and A. Saberi. Approximating power indices. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 943–950, 2008.

[5] 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.

[6] Y. Bleischwitz and B. Monien. Fair cost-sharing methods for scheduling jobs on parallel machines. In *Proceedings of the 6th Italian Conference on Algorithms and Complexity (CIAC)*, volume 3998 of *Lecture Notes in Computer Science*, pages 175–186, 2006.

[7] A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

[8] J. Brenner and G. Schäfer. Cost sharing methods for makespan and completion time scheduling. In *Proceedings of the 24th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 4393 of *Lecture Notes in Computer Science*, pages 670–681, 2007.

[9] S. Chawla, T. Roughgarden, and M. Sundararajan. Optimal cost-sharing mechanisms for network design. In *Proceedings of the Second Annual International Workshop on Internet and Network Economics (WINE)*, volume 4286 of *Lecture Notes in Computer Science*, pages 112–123, 2006.

[10] S. A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on the Theory of Computing (STOC)*, pages 151–158, 1971.

[11] R. Deb and L. Razzolini. Auction-like mechanisms for pricing excludable public goods. *Journal of Economic Theory*, 88(2):340–368, 1999.

[12] N. R. Devanur, M. Mihail, and V. V. Vazirani. Strategyproof cost-sharing mechanisms for set cover and facility location games. *Decision Support Systems*, 39(1):11–22, 2005.

[13] S. Dobzinski, A. Mehta, T. Roughgarden, and M. Sundararajan. Is Shapley cost-sharing optimal? In *Proceedings of the First International Symposium on Algorithmic Game Theory (SAGT)*, volume 4997 of *Lecture Notes in Computer Science*, pages 327–336, 2008.

[14] J. Edmonds. Optimum branchings. *Journal of Research of the National Bureau of Standards, Series B*, 71(4):233–240, 1967.

[15] J. Feigenbaum, A. Krishnamurthy, R. Sami, and S. Shenker. Hardness results for multicast cost sharing. *Theoretical Computer Science*, 304(1-3):215–236, 2003.

[16] J. Feigenbaum, C. Papadimitriou, and S. Shenker. Sharing the cost of multicast transmissions. *Journal of Computer and System Sciences*, 63(1):21–41, 2001.

[17] J. Feigenbaum and S. J. Shenker. Distributed algorithmic mechanism design: Recent results and future directions. In *Proceedings of the 6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pages 1–13, 2002.

[18] J. Green, E. Kohlberg, and J. J. Laffont. Partial equilibrium approach to the free rider problem. *Journal of Public Economics*, 6(4):375–394, 1976.

[19] A. Gupta, J. Könemann, S. Leonardi, R. Ravi, and G. Schäfer. An efficient cost-sharing mechanism for the prize-collecting Steiner forest problem. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1153–1162, 2007.

[20] A. Gupta, A. Srinivasan, and É. Tardos. Cost-sharing mechanisms for network design. *Algorithmica*, 50(1):98–119, 2008.

[21] S. Hart and A. Mas-Colell. Potential, value, and consistency. *Econometrica*, 57(3):589–614, 1989.

[22] J. D. Hartline. *Optimization in the Private Value Model: Competitive Analysis Applied to Auction Design*. PhD thesis, University of Washington, 2003.

[23] N. Immorlica, M. Mahdian, and V. S. Mirrokni. Limitations of cross-monotonic cost-sharing schemes. *ACM Transactions on Algorithms*, 4, 2008. Article 24.

[24] 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.

[25] K. Jain and V. Vazirani. Equitable cost allocations via primal-dual-type algorithms. *SIAM Journal on Computing*, 38(1):241–256, 2008.

[26] R. Juarez. Group strategyproof cost sharing: the role of indifferences. Working paper, 2007.

[27] K. Kent and D. Skorin-Kapov. Population monotonic cost allocation on MST's. In *Operational Research Proceedings KOI*, pages 43–48, 1996.

[28] J. Kleinberg and É. Tardos. Approximation algorithms for classification problems with pairwise relationships: metric labeling and markov random fields. *Journal of the ACM*, 49(5):616–639, 2002.

[29] J. Könemann, S. Leonardi, G. Schäfer, and S. van Zwam. A group-strategyproof mechanism for the Steiner forest game. *SIAM Journal on Computing*, 37(5):1319–1341, 2008.

[30] S. Leonardi and G. Schäfer. Cross-monotonic cost-sharing methods for connected facility location. *Theoretical Computer Science*, 326(1-3):431–442, 2004.

[31] F. Maniquet and Y. Sprumont. Fair production and allocation of an excludable nonrival good. *Econometrica*, 72(2):627–640, 2004.

[32] A. Mas-Colell, M. D. Whinston, and J. R. Green. *Microeconomic Theory*. Oxford University Press, 1995.

[33] A. Mehta, T. Roughgarden, and M. Sundararajan. Beyond Moulin mechanisms. *Games and Economic Behavior*, 2009. To appear.

[34] D. Monderer and L. S. Shapley. Potential games. *Games and Economic Behavior*, 14(1):124–143, 1996.

[35] H. Moulin. Incremental cost sharing: Characterization by coalition strategy-proofness. *Social Choice and Welfare*, 16(2):279–320, 1999.

[36] H. Moulin and S. Shenker. Strategyproof sharing of submodular costs: Budget balance versus efficiency. *Economic Theory*, 18(3):511–533, 2001.

[37] S. Muthukrishnan. Data streams: Algorithms and applications. *Foundations and Trends in Theoretical Computer Science*, 1(2):117–236, 2005.

[38] M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, 1994.

[39] 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.

[40] K. Roberts. The characterization of implementable choice rules. In J. J. Laffont, editor, *Aggregation and Revelation of Preferences*. North-Holland, 1979.

[41] 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.

[42] T. Roughgarden and M. Sundararajan. Optimal efficiency guarantees for network design mechanisms. In *Proceedings of the 12th Conference on Integer Programming and Combinatorial Optimization (IPCO)*, volume 4513 of *Lecture Notes in Computer Science*, pages 469–483, 2007.

[43] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16(1):8–37, 1961.