Black-Box Reductions for Cost-Sharing Mechanism Design[☆]

Konstantinos Georgiou[a], Chaitanya Swamy[a,1]

[a] *Combinatorics & Optimization, University of Waterloo, Waterloo, ON N2L 3G1.*

Abstract

We consider the design of *strategyproof* cost-sharing mechanisms. We give two simple, but extremely versatile, *black-box reductions*, that in combination reduce the cost-sharing mechanism-design problem to the *algorithmic* problem of finding a min-cost solution for a set of players. Our first reduction shows that *any* truthful, α-approximation mechanism for the social-cost minimization (SCM) problem satisfying a technical no-bossiness condition can be morphed into a truthful mechanism that achieves an $O(\alpha \log n)$-approximation where the prices recover the cost incurred. Thus, we decouple (modulo no-bossiness) the task of truthfully computing an outcome with near-optimal social cost from the cost-sharing problem. This is fruitful since truthful mechanism-design, especially for single-dimensional problems, is a relatively well-understood and manageable task. Our second reduction nicely complements the first one by showing that any LP-relative ρ-approximation for the problem of finding a min-cost solution for a set of players yields a truthful, no-bossy, $(\rho+1)$-approximation for the SCM problem (and hence, a truthful $(\rho+1)\log n$-approximation cost-sharing mechanism).

These reductions find a slew of applications, yielding, as corollaries, the first or improved polytime cost-sharing mechanisms for a variety of problems. For example, our first reduction coupled with the celebrated VCG mechanism shows that for *any* cost-sharing problem (with a monotone cost function) one can obtain a truthful mechanism that achieves an $O(\log n)$-approximation where the prices recover the cost incurred. Other applications include $O(\log n)$-approximation mechanisms for: survivable network design problems, facility location (FL) problems including capacitated and connected FL problems, and minimum-makespan scheduling on unrelated machines. We also obtain some extensions of these results to multidimensional settings.

Key words: Algorithmic mechanism design, Cost-sharing mechanisms, Linear programming, Approximation algorithms, Black-box reductions

[☆] A preliminary version [22] appeared in the Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms, 2012. The current version has a revised Introduction and includes new results that appear in Section 5.

Email addresses: k2georgiou@math.uwaterloo.ca (Konstantinos Georgiou), cswamy@math.uwaterloo.ca (Chaitanya Swamy)

[1] Research supported partially by NSERC grant 327620-09, an NSERC Discovery Accelerator Supplement Award, and an Ontario Early Researcher Award.

1. Introduction

In a cost-sharing problem, various strategic players compete to receive a service or good. Each player has a private value or type for receiving the service, and may misreport her type if that increases her utility. The service provider has to decide which set S of players will receive the service and at what prices, incurring a publicly-known cost $C(S)$. We assume that $C(.)$ is monotone: $C(S) \leq C(T)$ if $S \subseteq T$. How can one design a protocol or *mechanism* for such a cost-sharing problem that incentivizes truthful bidding, and where the outcome computed has good social welfare and the prices charged recover the cost incurred? This is the topic of study in *cost-sharing mechanism design*. More formally, a *mechanism* consists of an algorithm that outputs a solution, and a pricing scheme for specifying prices charged to the players; the utility earned by a player is her value under the algorithmic outcome minus the price she has to pay. A mechanism is said to be *dominant-strategy incentive-compatible*, if each player maximizes her utility by declaring her true private value regardless of what the other players declare. Throughout, we will use the more-compact phrase, *strategyproofness* or *truthfulness*, to denote dominant-strategy incentive-compatibility.

Three basic desirable properties of cost-sharing mechanisms that pervade the literature on cost-sharing mechanisms are: (a) strategyproofness (SP) (or stronger incentive-compatibility notions like group-strategyproofness (GSP)); (b) *cost recovery*, i.e., the revenue of the mechanism is at least the incurred cost; and (c) *economic efficiency*. In algorithmic mechanism design, we also require that the mechanism be computationally efficient. As is common in Computer Science, we use the lens of worst-case performance guarantees to compare different cost-sharing mechanisms. Also, as is standard in Computer Science, we say that a mechanism is computationally efficient if the number of steps it takes to compute its output (i.e., both the solution and the prices charged) is polynomial in the size of the input (i.e., the number of bits required to represent it); we call such a mechanism a polytime (or polynomial-time) mechanism. The standard means of quantifying economic efficiency via social welfare maximization (i.e., maximize total value earned − cost incurred) yields a rather ill-behaved optimization problem for which no meaningful worst-case guarantees are achievable in a computationally efficient manner [20]. Therefore, following [53] (and much of the subsequent work in cost-sharing mechanism design), we formalize economic efficiency using the *social-cost minimization* (SCM) objective: minimize the sum of the cost incurred and the total value of players who do not receive service.

In light of the impossibility of achieving all three requirements (a)–(c) simultaneously [24, 50], much effort has been devoted into approaches that relax, or drop, one of these requirements. With the above notions in place, we can formalize this by seeking, for a given cost-sharing problem, a truthful mechanism that computes a solution whose social cost is within α times the optimum social cost and charges prices that recover a $\frac{1}{\beta}$-fraction of the cost incurred for *every* choice of players' private values. We call such a mechanism, a *truthful, α-approximation, β-cost-recovering mechanism*; when $\beta = 1$, we simply say cost-recovering. General (algorithmic) mechanism-design techniques for developing such cost-sharing mechanisms are much sought after, but quite rare. Before describing our results, which make progress in this direction, we briefly discuss some well-known generic constructions. The celebrated VCG [57, 15, 25] family of mechanisms satisfies requirements (a) and (c) (for any cost function) but does not yield good revenue guarantees, and moreover is often computationally intractable. (Prior to our work, no analogue of VCG that simultaneously achieves (a) and relaxed versions of (b), (c), was known for general cost-sharing problems.) At the other end, *Moulin mechanisms* [46] satisfy (a) and (b) at the expense of social-cost approximation. A Moulin-style mechanism is an iterative mechanism where prices are offered in each iteration to the current set of players; players who cannot afford to pay these prices drop out and we iterate with the remaining set of players. The mechanism halts when all (current) players accept their prices. Different variants arise depending on how exactly (i.e., simultaneously or one at a time) players are offered prices and drop out. Moulin [46] proved that if the price-sequence offered to a player is nondecreasing—a property called *cross-monotonicity*—and the prices recover the cost incurred for the current set, then the resulting mechanism is GSP (i.e., no coalition of players can benefit by lying) and satisfies cost-recovery. [53] recently showed that if the price-sequences satisfy an additional property called summability, then one can also bound the approximation of the resulting Moulin mechanism.

Moulin's result and [53] have fueled much work on the design and analysis of such price sequences (also called a *cost-sharing method*) for various problems [47, 36, 53, 49, 52, 37, 13, 28, 8]. We emphasize however that *all these results are tailored to the problem at hand*. There is no known black-box way of leveraging Moulin mechanisms in conjunction with approximation algorithms for the SCM problem to obtain truthful mechanisms with analogous approximation and cost-recovery guarantees. (An α-approximation algorithm for a minimization problem is one that returns a solution of cost at most α times the optimum on every instance.) In fact, as shown in [34], designing suitable cost shares can often be much more challenging than the underlying algorithmic problem. For example, for vertex cover, cross-monotonic cost shares cannot recover more than $O(1/|V|^{1/3})$-fraction of the cost [34], and hence no Moulin mechanism that recovers the cost can achieve approximation better than $\Omega(1/|V|^{1/3})$ (see Section 2). As a means of overcoming these limitations, Mehta et al. [44] propose acyclic mechanisms, which are also Moulin-style mechanisms. They show that for various problems, one can adapt known primal-dual algorithms for the underlying cost-minimization problem to obtain suitable cost shares, which yield acyclic mechanisms with improved guarantees. But their methods also do not yield an automatic way of obtaining suitable cost shares from primal-dual algorithms and again the construction of cost shares is problem dependent.

Our results. The above state of affairs motivates the following natural question, which is the starting point for our work: *is there a generic, that is, black-box, way of transforming approximation algorithms for the SCM problem into truthful, approximation, cost-recovering mechanisms?* Questions of this flavor lie at the heart of algorithmic mechanism design and have spurred much research in this field. Various advances in this direction have been made in the area of social-welfare-maximization (SWM) problems [40, 19, 33] with an especially crisp positive answer known for Bayesian incentive-compatibility [2, 30]. But the above question is largely unanswered for cost-sharing mechanism design, with most prior work concentrating instead on the design of suitable cost shares. In order to better understand the interplay between the three

Problem category	Problem	Previous results	**Our results**
Survivable network design problems (SNDPs) ($r_{\max}$ = maximum requirement)	Edge-disjoint SNDP	–	$O(\log n)$
	Element-Disjoint SNDP	–	$O(\log n)$
	Vertex-Disjoint SNDP	–	$O(r_{\max}^3 \log^2 n)$
	Steiner-Tree	$O(\log^2 n)$ [53]	$O(\log n)$
	Steiner-Forest	$O(\log^2 n)$ [13]	$O(\log n)$
Facility location (FL)	Uncapacitated FL	$O(\log n)$ [52, 44]	$O(\log n)$
	Soft- capacitated FL (CFL)	–	$O(\log n)$
	Multicommodity connected FL	$O(\log^2 n)$ [52]	$O(\log n)$
Covering problems	Set cover	$O(\log^2 n)$ [44]	$O(\log^2 n)$
	Vertex cover	$O(\log n)$ [44]	$O(\log n)$
Scheduling problems	Makespan minimization on unrelated machines	–	$O(\log n)$

Table 1: Summary of approximation results. n always denotes the number of players. All cited mechanisms are (at least) truthful. When citing previous work, we scale prices to ensure cost-recovery; this scaling factor thus appears in the approximation.

competing objectives in cost-sharing mechanism design (requirements (a)–(c)), we decouple the approximation and cost-recovery objectives to refine the above question into the following two fundamental questions.
– *Can one "inject" cost-recovery into a truthful, approximation mechanism?*
– *Can one convert an approximation algorithm into a truthful, approximation mechanism?*

We give two simple, but extremely versatile, *black-box reductions*, that affirmatively (almost) answer the above two questions, and in combination, reduce the cost-sharing mechanism-design problem to the *algorithmic* cost-minimization (CM) problem of finding a minimum-cost solution for a set of players. (Notice that the CM problem is *easier* than the SCM problem: a ρ-approximation algorithm for the SCM problem can be used to obtain a solution of cost at most $\rho \cdot C(S)$ for any set S of players by setting player i's value to be ∞ (i.e., very large) if $i \in S$ and 0 otherwise.)

Informal statement of reductions *(1) Any truthful, α-approximation mechanism that satisfies an additional no-bossiness property can be transformed in polytime into a truthful, $O(\alpha \log n)$-approximation, cost-recovering mechanism for n players.*
(2) For a large family of cost-sharing problems, any LP-relative ρ-approximation algorithm for the CM problem yields a polytime truthful, no-bossy, $(\rho + 1)$-approximation mechanism.

(No-bossiness is the condition that if a winning player is unaffected by changing her bid, then neither is the outcome computed.) Thus, our first reduction (Section 3) conveniently decouples (modulo no-bossiness) the task of truthfully computing a solution with near-optimal social cost and the cost-recovery requirement. We emphasize that this reduction applies to *any* (monotone) cost function. The $\log n$ factor matches the approximation lower bound proved by Dobzinski et al. [18] for truthful cost-sharing mechanisms (for subadditive cost functions), which shows that the reduction is tight (up to constant factors). This reduction is quite fruitful since (as our second reduction shows) truthful mechanism-design, especially for single-dimensional problems, is a relatively well-understood and manageable task.

One of the most widely used and remarkably successful paradigms in the design of approximation algorithms, is that of expressing a relaxation of the problem as a linear program (LP) and using this to guide the design and analysis of the approximation algorithm, either via LP rounding or via a primal-dual approach. Many approximation algorithms are thus *LP-relative* approximation algorithms, where the approximation guarantee is proved by comparing the solution cost against the optimal value of the LP-relaxation. Our second reduction (see Section 4) shows that any LP-relative approximation algorithm for the CM problem can be used to obtain a truthful, no-bossy, approximation mechanism that can be fed as input to the first reduction. Thus, in combination, our reductions yield a generic way of exporting LP-relative approximations for the CM problem into truthful, cost-recovering mechanisms with related approximation guarantees. In contrast (to our liberal requirement of having an LP-relative approximation algorithm), much of the extant work on cost-sharing mechanisms requires the use of cost shares satisfying various properties to obtain (good) cost-sharing mechanisms. (Observe that an LP-relative ρ-approximation for the SCM problem yields an LP-relative ρ-approximation for the CM problem.)

With subadditive costs, our mechanisms (in addition to individual rationality) also ensure that no player i is charged a price larger than $C(\{i\})$, a property we call *individual competitiveness* (ICT). This is desirable, as otherwise an over-charged player has an incentive to refuse participation (and try to obtain the service from elsewhere at lower cost). A related point is that we do not insist that the mechanism's revenue be at most β times the cost incurred (for some $\beta > 1$). (This condition along with cost-recovery is called β-budget balance.) The usual rationale for the upper bound is that one does not want the coalition of winning players to have an incentive to secede from the mechanism and obtain the service from elsewhere at lower cost. Since we focus on strategyproofness, we explicitly do not consider the effect of coalitions; focusing on individual players yields ICT instead as the natural requirement.

A key feature of our reductions is their generality. An immediate notable implication is that taking the VCG mechanism as input in the first reduction, we obtain, for *any* cost function, a truthful, $O(\log n)$-approximation, cost-recovering mechanism. Previously, such a result was known only for subadditive cost functions [5]. For a wide variety of cost-sharing problems, we obtain the first, or improved polytime cost-sharing mechanisms simply by plugging in a suitable LP-relative algorithm. We consider a few representative applications in Section 6, and summarize our results in Table 1. We believe that our reductions will find many more applications. Section 7 considers some extensions to multidimensional cost-sharing problems. Our results demonstrate that in contrast with our current understanding of group-strategyproof and acyclic mechanisms, strategyproofness allows for ample flexibility in cost-sharing mechanism design enabling one to effectively leverage various algorithmic results.

Our constructions are quite intuitive and easy to describe. For the first reduction, we first observe that *regardless* of the cost shares used, the allocation rule f of a Moulin-style mechanism is always monotone, and hence one can find prices $\{p_i\}$ such that $(f, \{p_i\})$ is a truthful mechanism. Now we simply initialize the Moulin mechanism with the outcome returned by the input truthful mechanism and then use the uniform cost shares $C(S)/|S|$. Since the Moulin mechanism preserves truthfulness, the resulting mechanism is truthful, while the cost shares prescribed ensure cost recovery at the expense of a $\log n$-factor loss in approximation. The second reduction proceeds by rejecting all the players who are rejected fractionally in the SCM LP, and using the LP-relative algorithm for the CM problem to compute a solution for the remaining players. Simple LP theory shows that for a broad class of LPs, this mechanism has all the desirable properties.

Related work. Moulin [46] and Moulin and Shenker [47] developed the theory of Moulin mechanisms. Subsequently, suitable cost-sharing methods were developed for various combinatorial-optimization problems, such as Steiner tree [36], Steiner forest [37], facility location [49], connected facility location [29, 42], and scheduling problems [4]. Prior to [53], such results focused on the design of cross-monotonic, approximately budget balanced (BB) cost shares; the resulting Moulin mechanisms do not however come with any (SCM-) approximation guarantees. Immorlica et al. [34] exposed an inherent limitation of this method by proving lower bounds on the BB-factor achievable by cross-monotonic cost shares for various problems. Devanur et al. [17] designed truthful, cost-recovering non-Moulin mechanisms for set cover and facility location, but do not prove any approximation guarantees.

Roughgarden and Sundararajan [53] proposed the social-cost objective, and isolated a property of the cost-sharing method called summability that bounds the approximation of the resulting Moulin mechanism. Subsequent work designed new cost-sharing methods [52, 28, 8, 6] and/or re-analyzed previous cost-sharing methods [13, 53, 52] to also show summability bounds. [53, 8] also prove lower bounds on the summability and/or BB factor of cost-sharing methods, and [53] observed that such lower bounds, translate to poor approximation and/or poor BB for Moulin mechanisms. Mehta et al. [44] proposed acyclic mechanisms, which also require suitable cost shares, as a means of circumventing these obstacles. They show that for certain problems, primal-dual algorithms for the underlying cost-minimization problem can be easily adapted to yield acyclic mechanisms with good guarantees.

None of these results yield generic ways of translating algorithmic results for the SCM problem into analogous cost-sharing mechanisms. The design of cross-monotonic cost-shares satisfying various properties is tailored to the problem at hand and often quite intricate. [44] obtain some success, but they too are not able to automatically translate primal-dual algorithms into suitable cost shares and have to proceed in a problem-dependent way. The work of Bleischwitz et al. [5] is perhaps closest in spirit to our work. They show that for subadditive cost functions, one can obtain (not necessarily polytime) truthful (also, weakly GSP), $O(\log n)$-approximation, 1-BB mechanisms. Let Alg be a ρ-approximation algorithm Alg for the CM problem. [5] also show that if the cost function induced by Alg, denoted C_{Alg}, has a certain ordering property, then one can obtain a polytime truthful, polytime, $O(\rho \log n)$-approximation mechanism. However, this property is not known to be satisfied for various problems of interest; e.g., Steiner tree, facility location etc. Brenner and Schafer [9] show that if the cost function satisfies a different ordering property, then Alg can be used to obtain an acyclic mechanism with ρ-BB; if C_{Alg} and C satisfy some other conditions they also obtain approximation guarantees. In comparison with our requirement that Alg be an LP-relative algorithm, these conditions on Alg and C seem much more restrictive;

indeed, the applications in [5, 9] are limited to scheduling problems.

In the area of social-welfare-maximization (SWM) packing problems, more success has been obtained in devising black-box reductions. Lavi and Swamy [40] and Dughmi and Roughgarden [19] show how to translate certain algorithms into truthful-in-expectation mechanisms with the same approximation guarantee; [40] require an integrality-gap verifying approximation algorithm, whereas [19] require an FPTAS. We note that our requirement of "LP-relative ρ-approximation" is much weaker than the integrality-gap requirement in [40]. Most recently, Huang et al. [33] showed that for a symmetric single-dimensional SWM problem, any approximation algorithm can be converted to a truthful mechanism with the same approximation. If one relaxes the truthfulness condition to *Bayesian incentive compatibility*, then black-box reductions were recently obtained by Hartline and Lucier [31] in the single-dimensional setting, and [2, 30] in the multidimensional setting. None of these reductions ensure cost recovery.

2. Preliminaries

In a *cost-sharing* mechanism-design problem, we have n players with private types who compete for some service or good, and each outcome specifies a set S of players who will receive the service. Let $[n]$ denote the set $\{1, \ldots, n\}$, and $\mathcal{A} \subseteq 2^{[n]}$ denote the set of all possible outcomes. Also, there is a *publicly-known* cost-function $C : \mathcal{A} \mapsto \mathbb{R}_{\geq 0}$ that specifies the cost incurred for serving a given set of players; we use $C(i)$ to denote $C(\{i\})$. As is standard, we assume that $\mathcal{A}$ is downwards-closed and C is monotone, that is, if $T \in \mathcal{A}$ and $S \subseteq T$, then $S \in \mathcal{A}$ and $C(S) \leq C(T)$. In keeping with the vast literature on cost-sharing mechanisms, we focus for the most part on single-dimensional cost-sharing problems, wherein each player i's private type consists of a single nonnegative parameter v_i specifying her value for receiving the service. We use v to denote the tuple $(v_1, \ldots, v_n)$ and v_{-i} to denote the tuple $(v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n)$.

A (direct revelation) mechanism for a cost-sharing problem consists of an allocation rule (that is, an algorithm) $f : \mathbb{R}^n_+ \mapsto \mathcal{A}$, and a pricing scheme $p_i : \mathbb{R}^n_+ \mapsto \mathbb{R}$ for each player i. Each player i reports a type v_i (possibly deviating from his true type), and the mechanism computes the outcome $f(v)$ and charges price $p_i(v)$ to player i. We sometimes refer to the players in $f(v)$ (i.e., who receive service) as "winners". Throughout we use $\overline{v}_i$ to denote the *true type* of player i. The *utility* $u_i(\overline{v}_i; v_i, v_{-i})$ player i derives when she declares v_i and the others declare v_{-i} is $\overline{v}_i - p_i(v_i, v_{-i})$ if $i \in f(v_i, v_{-i})$ and $-p_i(v_i, v_{-i})$ otherwise, and each player aims to maximize her own utility.

We are interested in designing mechanisms where the outcome computed approximates the optimum social cost with respect to the players' true types, which is defined as $\min_{S \in \mathcal{A}}(\mathsf{SC}(\overline{v}; S) := C(S) + \sum_{i \notin S} \overline{v}_i)$, and the prices recover the cost incurred by the mechanism. More precisely, we formalize this by requiring a cost-sharing mechanism $M = (f, \{p_i\})$ to satisfy the following desirable properties:

- M is *strategyproof* (a.k.a *truthful*), which means that each player maximizes her utility by revealing her true value: for any i, v_{-i}, and any $\overline{v}_i, v'_i$, we have $u_i(\overline{v}_i; \overline{v}_i, v_{-i}) \geq u_i(\overline{v}_i; v'_i, v_{-i})$. We use the terms strategyproofness and truthfulness interchangeably from now on.

M is *individually rational* (IR) and has *no positive transfers* (NPT), i.e., $u_i(\overline{v}_i; \overline{v}_i, v_{-i}) \geq 0$ and $p_i(\overline{v}_i, v_{-i}) \geq 0$ for every $i, \overline{v}_i, v_{-i}$. (In the sequel, whenever we say truthful, we mean truthful, IR, and NPT.)

- (*Approximation*) f is an α-approximation algorithm for the social-cost minimization (SCM) problem: for every input $v = (v_1, \ldots, v_n)$, we have $\mathsf{SC}(f(v)) \leq \alpha(\min_{S \in \mathcal{A}} \mathsf{SC}(v; S))$. We drop the input v in $\mathsf{SC}(.;.)$ when this is clear from the context.
- (*Cost recovery*) The prices recover at least a $\frac{1}{\beta}$-fraction of the mechanism-designer's cost: for every input v, we have $\sum_i p_i(v) \geq C(f(v))/\beta$.

We call such a mechanism a *truthful, α-approximation, β-cost-recovering mechanism*. (We abbreviate 1-cost-recovering to cost-recovering.) Often, computing $C(S)$ for a set S turns out to be *NP*-hard. This is problematic since although the mechanism may choose a "good" set S of players, it may not be able to efficiently compute a solution of cost comparable to $C(S)$ (rendering cost-recovery, as stated above, meaningless). Therefore, we require a polytime mechanism to also specify a candidate (low-cost) solution for the set of players it serves. In such settings, the approximation and cost-recovery requirements are modified to $C_M(f(v)) + \sum_{i \notin f(v)} v_i \leq \alpha(\min_{S \in A} \mathsf{SC}(v; S))$, and $\sum_i p_i(v) \geq C_M(f(v))/\beta$, where $C_M(f(v))$ is the cost of the solution computed by M for the set $f(v)$.

We say that an allocation rule f is *implementable* if there exist prices $\{p_i\}$ such that $(f, \{p_i\})$ is a truthful mechanism. For single-dimensional problems, we have the following well-known and useful characterization of implementable allocation rules. Call an allocation rule f (for a single-dimensional problem) *monotone* if $i \in f(v_i, v_{-i})$ and $v'_i > v_i$ implies that $i \in f(v'_i, v_{-i})$ (i.e., a winning player remains a winner by increasing her bid).

Theorem 2.1 ([48, 1]) *Given an allocation rule f, there exist prices $\{p_i\}$ implementing f iff f is monotone. Suppose that f is monotone and for every i and v_{-i}, there is a well-defined threshold $t_i(v_{-i})$ such that for any input (v, v_{-i}), player i wins when $v > t_i(v_{-i})$ and loses when $v < t_i(v_{-i})$. Then, setting $p_i(v) = t_i(v_{-i})$ if player i wins and 0 otherwise for every i, gives the unique prices that implement f and ensure IR, NPT.*

We say that a cost function is subadditive if $C(S) + C(T) \geq C(S \cup T)$ for every $S, T \in \mathcal{A}$ such that $S \cup T \in \mathcal{A}$. The vast majority of cost-sharing problems that have been studied in the literature involve subadditive functions (e.g., Steiner forest, vertex cover, facility location). For subadditive cost functions, an additional desirable property that we would like to achieve is that the price charged by the mechanism to a (winning) player does not exceed the cost of serving her individually. We call this property *individual competitiveness* (ICT).

- (*Individual competitiveness*) $p_i(v) \leq C(i)$ for every (winning) player i.

(If C is not subadditive, then ICT conflicts with cost-recovery, so we impose ICT only when C is subadditive.) We view individual competitiveness as a basic sanity check: in its absence, a winner i who is charged a price larger than $C(i)$ has an incentive to secede from the mechanism and find alternate means of obtaining the service (by herself, or from a competitor) at a cost lower than the price she currently pays. This is the same rationale as the one used to motivate the core of a cooperative cost-sharing game. But since our focus is on strategyproofness we explicitly do not consider collusions among players; projecting the core-condition to individual players yields instead the ICT requirement. A related point is that our β-cost-recovery condition is subtly different from the β-budget balance condition used in the literature, wherein we require that $\frac{C(f(v))}{\beta} \le \sum_i p_i(v) \le C(f(v))$ (or equivalently, via scaling, $\sum_i p_i(v) \in [C(f(v)), \beta C(f(v))]$). As in the case of the core, the usual rationale for imposing the upper bound is that in its absence the coalition of winning players has an incentive to secede from the mechanism. As before, since we explicitly focus on individual players and not coalitions, we drop this upper-bound requirement, and insist on (approximate) cost-recovery and ICT.

Our first reduction (Section 3) requires as input a truthful, approximation mechanism that satisfies an additional technical condition called *no-bossiness*, which is defined as follows: an allocation rule f satisfies no-bossiness if for every i, v_{-i} and v, v', if $i \in f(v, v_{-i})$ and $i \in f(v', v_{-i})$, then $f(v, v_{-i}) = f(v', v_{-i})$. That is, if a winning player remains a winner by changing her bid, then the outcome computed is unaffected.[2]

Lower bounds. Dobzinski et al. showed that for the *public-excludable good* problem ($C^{\mathsf{PEG}}(S) = 1$ if $S \neq \emptyset$ and is 0 otherwise) any truthful mechanism for the SCM problem achieving β-budget balance must be $\Omega(\log n/\beta)$-approximate (see Theorem 1 in [18]). In fact, their proof actually shows that this holds for β-cost-recovering truthful mechanisms. Since C^{PEG} can be encoded as the cost function of many problems (e.g., Steiner tree, vertex cover, facility location), this implies an analogous lower bound for these cost-sharing problems.

Moulin-style mechanisms. A Moulin-style mechanism works as follows. The mechanism takes as input a cost-sharing method $\xi : 2^{[n]} \times [n] \mapsto \mathbb{R}_+$, where $\xi(S, i)$ represents intuitively the amount charged to player i when S is the set of winners. Given a current set S (initialized to $[n]$) of candidate players for receiving service, the mechanism tentatively asks each player $i \in S$ if $v_i \ge \xi(S, i)$. If this is true for all players, then the mechanism outputs S *and charges each player* $i \in S$ *the price* $\xi(S, i)$ (and 0 to the other players). Otherwise, the mechanism drops one, some, or all of the players who have $v_i < \xi(S, i)$, and iterates with the remaining set of players (we call the latter the "all-drop" rule.) Different variants arise based on the exact rule for dropping players. Moulin [46] showed that if the cost-sharing method is *cross-monotonic*—that is, $\xi(S, i) \ge \xi(T, i)$ for every $S \subseteq T, i \in S$—then all variants yield the same mechanism and this mechanism is strategyproof (in fact, group-strategyproof (GSP)). Moreover, if $\sum_{i \in S} \xi(S, i) \ge C(S)$ for all S, then the mechanism satisfies cost-recovery.

Define $\xi(S, T) := \sum_{i \in T} \xi(S, i)$. Say that a cost-sharing method ξ is competitive if $\xi(S, S) \le C(S)$ for all S; say that ξ is cost-recovering if $\xi(S, S) \ge C(S)$ for all S. Define

[2] Our no-bossiness condition is slightly different from the non-bossiness notion introduced by Satterthwaite and Sonnenschein [55]. Their notion imposes a condition on the mechanism (as opposed to only the allocation rule): a mechanism $M = (f, \{p_i\})$ satisfies non-bossiness if whenever $i \in f(v, v_{-i})$, $i \in f(v', v_{-i})$ and $p_i(v, v_{-i}) = p_i(v', v_{-i})$, we have $f(v, v_{-i}) = f(v', v_{-i})$ and $p_j(v, v_{-i}) = p_j(v', v_{-i})$ for all players j.

the budget-balance (BB) factor of a cost-sharing method ξ to be $\max_S \left\{ \frac{C(S)}{\xi(S,S)}, \frac{\xi(S,S)}{C(S)} \right\}$. Immorlica et al. [34] proved lower bounds on the BB-factor achievable by competitive, cross-monotonic ξ for various problems. Clearly, this also implies lower-bounds for cost-recovering, cross-monotonic ξ. [53] observed that the approximation of the Moulin mechanism M_ξ constructed from a competitive, cross-monotonic ξ is Ω(BB-factor of ξ). We observe that the same holds for cost-recovering ξ. Coupled with the lower bounds in [34] for various cost functions, this implies lower bounds on the approximation of every cost-recovering Moulin mechanism for these cost-functions.

Lemma 2.2 *Let ξ be cost-recovering and cross-monotonic with BB-factor β. Then, M_ξ has approximation ratio $\Omega(\beta)$.*

Proof : Let S be an inclusion-wise minimal set such that $\xi(S,S) = \beta C(S)$. So $\xi(S,i) > 0$ for all $i \in S$. Consider the following input. Set $v_i = \xi(S,i) - \epsilon > 0$ for all $i \in S$, where $\epsilon > 0$ is negligible, and $v_i = 0$ for all $i \notin S$. M_ξ will return the empty set and incur social cost $\xi(S,S) - |S|\epsilon$, whereas choosing S as the outcome yields social cost $C(S)$. Thus the approximation ratio tends to β as ϵ goes to 0. ■

3. A black-box way of injecting cost-recovery

In this section we prove the following theorem, which reduces the cost-sharing (i.e., truthful, approximation, cost-recovering) mechanism-design problem to the task of truthful and no-bossy approximation mechanism design.

Theorem 3.1 *Given a truthful, α-approximation mechanism $M = (g, \{q_i\})$ satisfying no-bossiness, we can obtain a mechanism M' such that: (a) M' is a truthful, $O(\alpha \log n)$-approximation, cost-recovering mechanism, and is polytime computable if M is; (b) if M is ICT and C is subadditive, then M' is ICT.*

The proof follows from two constructions. The first construction is quite simple to describe and illustrates many of the ideas involved. The idea here is to simply initialize the Moulin mechanism with the output of the mechanism M and then use the uniform cost shares $\xi(S,i) = C(S)/|S|$. Since the Moulin mechanism preserves truthfulness, the resulting mechanism inherits truthfulness from M, while the uniform cost shares ensure cost-recovery while degrading the approximation by a $\log n$-factor. The resulting mechanism satisfies all the properties mentioned in Theorem 3.1 except ICT. (As mentioned earlier, for non-subadditive cost functions, we cannot hope to achieve both cost-recovery and ICT.) Next, for subadditive cost functions, we show how one can also obtain ICT by suitably refining the first construction. We describe this after detailing the first construction. For a set $S \subseteq [n]$, we define $\mathcal{R}(S) = \{i \in S : v_i < C(S)/|S|\}$.

Lemma 3.2 *Mechanism M_1 satisfies property (a) of Theorem 3.1.*

Proof : We assume here for simplicity that $C(.)$ is polytime computable, in which case it is clear that M_1 is polytime computable if M is; Remark 3.3 shows that with a slight

Mechanism $M_1 = (f_1, \{p_{1,i}\})$ Given: a truthful, α-approximation mechanism $M = (g, \{q_i\})$ satisfying no-bossiness. On input v, we do the following.

C1. Initialize $j \leftarrow 0$ and $S_0 \leftarrow g(v)$.

C2. While $\mathcal{R}(S_j) \neq \emptyset$, set $S_{j+1} \leftarrow S_j \setminus \mathcal{R}(S_j)$ and $j \leftarrow j+1$.

C3. Return S_j as the winner set. The prices, as specified via Theorem 2.1, equate to (see Lemma 3.2) $p_{1,i}(v) = \max\{q_i(v), \frac{C(S_0)}{|S_0|}, \ldots, \frac{C(S_j)}{|S_j|}\}$ if $i \in S_j$, 0 otherwise.

---

modification to the above construction, this continues to hold even otherwise. Consider any input v, and let $g(v) = S_0$ and $f_1(v) = W \subseteq S_0$. Let $OPT = \min_{S \in \mathcal{A}} \mathsf{SC}(v; S)$. (Note that M_1 always returns a feasible solution, since it returns a subset of $g(v)$, and $\mathcal{A}$ is downwards closed.)

Fix a player i who is a winner in M_1 under the input v, and let $v'_i > v_i$. Since i is a winner, we have $i \in S_0$. Observe that $g(v'_i, v_{-i}) = S_0$ since g is monotone (so i remains a winner in M) and satisfies no-bossiness. So since $v'_i > v_i$, mechanism M_1 proceeds *identically* on the inputs v and (v'_i, v_{-i}), and hence i remains a winner under the input (v'_i, v_{-i}).

We have $C(S_0) + \sum_{i \notin S_0} v_i \leq \alpha OPT$. Let $k = |S_0|$ and let $S_0 \setminus W = \{i_0, \ldots, i_m\}$, where the players are arranged in the order they were dropped (breaking ties among the players dropped in the same iteration arbitrarily). Then, we must have $v_{i_\ell} \leq C(S_0)/(k-\ell)$ since if player i_ℓ was dropped from the set S_r, then we have $v_{i_\ell} \leq C(S_r)/|S_r|$ and $|S_r| \geq |S_0| - \ell$ (since i_ℓ is the ℓ-th player to be dropped). So $v_{i_\ell} \leq C(S_0)/(k-\ell)$, and hence, it follows that $\mathsf{SC}(W) = C(W) + \sum_{i \notin S_0} v_i + \sum_{i \in S_0 \setminus W} v_i \leq C(S_0) + \sum_{i \notin S_0} v_i + C(S_0) \cdot H_k \leq \alpha(1 + H_n) OPT$.

To argue that M_1 satisfies cost-recovery, we prove that the threshold of each winner i is given by $\tau_i := \max\{q_i(v), \frac{C(S_0)}{|S_0|}, \ldots, \frac{C(S_j)}{|S_j|}\}$. This implies that the prices specified in the construction are indeed those determined by Theorem 2.1, which immediately yields cost-recovery since then $\sum_{i \in W} p_{1,i}(v) \geq \sum_{i \in W} C(W)/|W| = C(W)$. Consider some input $v' = (v'_i, v_{-i})$. Suppose that i wins in M_1 under v'. Since M is truthful and no-bossy, and $i \in g(v')$, this implies that $v'_i \geq q_i(v') = q_i(v) =$ (i's threshold value in M for v_{-i}), and $g(v') = g(v)$. Notice then that M_1 proceeds identically on both v and v'. So we must have $v'_i \geq \max\{\frac{C(S_0)}{|S_0|}, \ldots, \frac{C(S_j)}{|S_j|}\}$. Also note that player i wins in M_1 for any $v'_i > \tau_i$. This implies that i's threshold in M_1 is τ_i. ∎

Remark 3.3 When $C(.)$ is *NP*-hard to compute, we cannot necessarily compute $\mathcal{R}(S)$. Also, as discussed earlier, M_1 must now also specify a solution for the set of winners. Both issues can be handled as follows. We make the very mild assumption that a solution for S_0 also induces a solution of no greater cost for any subset of S_0. We now redefine $\mathcal{R}(S)$ as $\mathcal{R}_{S_0}(S) = \{i \in S : v_i < C_M(S_0)/|S|\}$, and the solution we return for the winner set $W = S_j$ is the one induced by S_0. Mimicking the proof of Lemma 3.2, it is easy to see that $C_{M_1}(W) + \sum_{i \notin W} v_i \leq (1 + H_n) C_M(S_0) + \sum_{i \notin S_0} v_i \leq \alpha(1 + H_n) OPT$. Also, we now have $p_{1,i}(v) = \max\{q_i(v), \frac{C_M(S_0)}{|W|}\}$ so that $C_{M_1}(W) \leq C_M(s_0) \leq \sum_{i \in W} p_{1,i}(v) \leq \sum_{i \in W} q_i(v) + C_M(S_0)$.

Also, notice that if we use $C_M(S_0)/\beta|S|$ in the definition of $\mathcal{R}_{S_0}(S)$, where $\beta \geq 1$, then the proof of Lemma 3.2 shows that we get $O(\alpha \log n/\beta)$ approximation and β-cost recovery.

One noteworthy application of the above construction is that taking M to be the VCG mechanism, which solves the SCM problem exactly and can be assumed to be no-bossy by fixing a tie-breaking rule we obtain the following very general result, which can be viewed as an analogue of VCG for cost-sharing problems. No such general result was previously known.

Corollary 3.4 *For any monotone cost function, there is a truthful, $O(\log n)$-approximation, cost-recovering mechanism.*

Individual competitiveness with subadditive cost functions. We now describe how to refine the construction of M_1 so as to obtain individual competitiveness when $C(.)$ is subadditive. A natural first attempt would be to set the threshold for a player i to remain in the current candidate set S to be $\min\{C(i), C(S)/|S|\}$ (instead of the uniform threshold $C(S)/|S|$). This however fails to ensure cost-recovery: the problem arises because one may accept a winner-set W where only a small subset $T \subseteq W$ of winners pay the price $C(W)/|W|$, which could be much smaller than $C(T)/|T|$. To rectify this, we need a more sophisticated scheme. Given the current candidate set S, we keep track of the set $\mathcal{T}(S)$ of players in S for which $C(i) < C(S)/|S|$. The players in $\mathcal{T}(S)$ are asked to pay their individual price $C(i)$ and are (permanently) accepted or rejected based on whether they can do so. If $\mathcal{T}(S) \neq \emptyset$, we update S to $S \setminus \mathcal{T}(S)$ and iterate with this set. Otherwise, the players in S are asked to pay the price $C(S)/|S|$. The players who cannot do so are rejected and we iterate with the remaining set of players. The resulting mechanism is described in detail below. Recall that $\mathcal{R}(S) = \{i \in S : v_i < C(S)/|S|\}$. We also define the sets $\mathcal{T}(S) = \{i \in S : C(i) < C(S)/|S|\}$ and $\mathcal{R}'(S) = \{i \in S : v_i < C(i)\}$.

Mechanism $M_2 = (f_2, \{p_{2,i}\})$ Given: a truthful, α-approximation mechanism $M = (g, \{q_i\})$ satisfying no-bossiness. On input v, we do the following.

- D1. Initialize $j \leftarrow 0$ and $S_0 \leftarrow g(v)$. Also, set $A \leftarrow \emptyset$, $R \leftarrow \emptyset$. (A keeps track of the set of players who will be winners, and R maintains the set of players who will be rejected.)
- D2. While $\mathcal{T}(S_j) \cup \mathcal{R}(S_j) \neq \emptyset$, proceed as follows.
  - D2.1. If $\mathcal{T}(S_j) \neq \emptyset$, then set $R \leftarrow R \cup \mathcal{R}'(\mathcal{T}(S_j))$ and $A \leftarrow A \cup \mathcal{T}(S_j) \setminus \mathcal{R}'(\mathcal{T}(S_j))$. Set $S_{j+1} \leftarrow S_j \setminus \mathcal{T}(S_j)$.
  - D2.2. Otherwise (i.e., $\mathcal{T}(S_j) = \emptyset$, $\mathcal{R}(S_j) \neq \emptyset$), set $R \leftarrow R \cup \mathcal{R}(S_j)$ and $S_{j+1} = S_j \setminus \mathcal{R}(S_j)$.
  - D2.3. Update $j \leftarrow j + 1$.
- D3. Return $A \cup S_j$ as the winner set, and let $\{p_{2,i}(v)\}$ be the corresponding prices as specified by Theorem 2.1. We show that if $q_i(v)$ is given then $p_{2,i}(v)$ can be computed in polytime (see Lemma 3.5).

Lemma 3.5 *Given the prices $\{q_i\}$, the prices $\{p_{2,i}\}$ implementing f_2 can be computed in polytime. Theorem 3.1 is satisfied by taking $M' = M_2$.*

Proof : Consider any input v, and let $g(v) = S_0$ and $f_2(v) = W \subseteq S_0$ and $OPT = \min_{S \in \mathcal{A}} \mathsf{SC}(v; S)$. The proof of monotonicity of f_2 and approximation follow by mimicking the proof of Lemma 3.2. Since M is truthful and no-bossy, if a winner i in M_2 raises her bid, then M will return the same set S_0 and hence, M_2 proceeds identically on v and the new input, and so i remains a winner. Let i_ℓ be the $(\ell + 1)$-th player dropped from S_0. Suppose $i_\ell \in S_r \setminus S_{r+1}$, so $|S_r| \geq k - \ell$. If $i_\ell \in \mathcal{R}'(\mathcal{T}(S_r))$, then $v_{i_\ell} < C(i_\ell) < C(S_r)/|S_r|$; otherwise, $i_\ell \in \mathcal{R}(S_r)$ and we again have $v_{i_\ell} < C(S_r)/|S_r|$. So in both cases, $v_{i_\ell} < C(S_r)/|S_r| \leq C(S_0)/(k - \ell)$; hence $\sum_{i \in S_0 \setminus W} v_i \leq C(S_0) \cdot H_k$ which in turn implies that $\mathsf{SC}(W) \leq \alpha(1 + H_n)OPT$.

We now show that M_2 is ICT if M is. Fix any winner $i \in W$. It suffices to argue that if $v_i \geq C(i)$ then i will be chosen as a winner in M_2. Since $v_i \geq C(i) \geq q_i(v_i, v_{-i})$, i is in S_0. Suppose $i \in S_r$. If $i \in \mathcal{T}(S_r)$, then clearly i is added to A; otherwise, we have $i \in S_{r+1}$, since $i \notin \mathcal{R}(S_r)$ as $v_i \geq C(i) \geq C(S_r)/|S_r|$. It follows that if $i \notin A$, then i is in the final set S_j. So i is a winner.

Next, we show that the prices implementing f_2 can be computed efficiently. Consider a player $i \in W$ and some input $v' = (v'_i, v_{-i})$. Suppose that i wins in M_2 under v'. As before, since M is truthful and no-bossy, we have $v'_i \geq q_i(v') = q_i(v) =$ (i's threshold value in M for v_{-i}) and $g(v') = g(v)$, so M_2 proceeds identically on v and v'. Now, i is chosen as a winner either because (1) it is added to A at some point when our candidate set was, say, S_r; or because (2) i is part of the final set S_j. Importantly, since (fixing v_{-i}) M_2 proceeds identically on every winning bid of i, *which of these two cases happen does not depend on i's (winning) bid.* If the former case happens (for every winning bid), then we must have $v'_i \geq C(i)$; also, i wins in M_2 whenever $v'_i > \max\{q_i(v), C(i)\}$. So here, we have $p_{2,i}(v) = \max\{q_i(v), C(i)\}$. In the latter case, let $\mathcal{S} \subseteq \{S_0, S_1, \ldots, S_j\}$ be the collection of sets for which $\mathcal{T}(S) = \emptyset$. Then, we must have $v'_i \geq \max_{S \in \mathcal{S}} C(S)/|S|$, and moreover, i wins in M_2 whenever $v'_i > \max\{q_i(v), \max_{S \in \mathcal{S}} C(S)/|S|\}$. So if this case happens, we have $p_{2,i}(v) = \max\{q_i(v), \max_{S \in \mathcal{S}} C(S)/|S|\}$. In particular, we note that since $S_j \in \mathcal{S}$, we have $p_{2,i}(v) \geq C(S_j)/|S_j|$.

Finally, cost-recovery follows because $\sum_{i \in W} p_{2,i}(W) \geq \sum_{i \in A} C(i) + \sum_{i \in S_j} C(S_j)/|S_j| \geq C(A) + C(S_j) \geq C(A \cup S_j)$ where the last two inequalities follow from subadditivity. ■

As before, if $C(.)$ is *NP*-hard to compute (but say $C(i)$ is polytime computable, as is often the case), then we can redefine $\mathcal{R}(S)$ and $\mathcal{T}(S)$ as $\mathcal{R}_{S_0}(S) = \{i \in S : v_i < C_M(S_0)/|S|\}$ and $\mathcal{T}_{S_0}(S) = \{i \in S : C(i) < C_M(S_0)/|S|\}$, and return the solution induced by S_0 for $A \cup S_j$. With this modification, M_2 is polytime computable if M is. Also, as before, we can trade off approximation with cost-recovery. As in Remark 3.3, we can also upper bound the revenue of this modified M_2 mechanism by $\sum_{i \in W} q_i(v) + (1 + H_n)C_M(S_0)$. This upper bound will be useful in Section 5.

4. Obtaining truthful, no-bossy, approximation in a black-box fashion

Complementing the construction described in Section 3, we now describe how to obtain a truthful, $(\rho + 1)$-approximation mechanism satisfying no-bossiness in a black-box fashion from an LP-relative ρ-approximation algorithm for the cost-minimization problem of finding a min-cost way of serving a given set of players. Combined with the reduction in Section 3, this yields a truthful, $O((\rho + 1)\log n)$-approximation cost-recovering mechanism. These reductions find numerous applications, which we discuss in Section 6.

A generic LP-model. We describe two LP-models for the cost-minimization (CM) problem and the associated SCM problem. The first model captures general covering problems without multiplicity constraints; the second allows for multiplicity constraints but captures a restricted class of general covering problems that nevertheless includes $\{0, 1\}$-covering problems. We focus on the first model below, and discuss the second model later. Consider a cost-sharing problem where the problem of finding a min-cost solution for a set S of players admits an LP-relaxation of the following form.

$$\min \quad c^T x \quad \text{s.t.} \qquad A^{(i)} x \geq b^{(i)} \quad \text{for all } i \in S, \qquad Bx \geq d, \qquad x \in \mathbb{R}^m_+. \quad \text{(C1-P)}$$

Here $A^{(i)}x \geq b^{(i)}$ denotes some constraints specific to player i that arise because i has to be served, and $Bx \geq d$ models various global constraints. We require that $d \geq 0$ and $A^{(i)}, b^{(i)} \geq 0$ for every i, and $OPT_{\text{C1-P}(S)} \leq C(S)$ for every set S (where C1-P(S) denotes (C1-P) for the set S). However, this latter requirement alone is a rather weak condition: for example, the LP where $b^{(i)} = 0$ for all i, $d = 0$ always has optimal value 0 and satisfies this condition, but this LP clearly carries no information. To ensure that (C1-P) conveys something meaningful, we stipulate that $OPT_{\text{C1-P}(S)}$ provide a "good" lower bound on $C(S)$ for every set S. More precisely, we require an algorithm Alg that for every set S, returns a solution of cost at most $\rho \cdot OPT_{\text{C1-P}(S)}$; thus, Alg is a ρ-approximation algorithm for the CM problem relative to the LP (C1-P). (Notice that the existence of such an algorithm implies that $C(S) \leq \rho \cdot OPT_{\text{C1-P}(S)}$ for every set S.) These conditions are satisfied by the LP-relaxations of many combinatorial-optimization covering problems, such as $\{0, 1\}$-covering problems (which includes various cost-sharing problems studied in [36, 49, 37, 53, 44]). We give two examples (see also Section 6).

Example 4.1 *Survivable-network design problem* (SNDP). Each player i is an (s_i, t_i) pair requiring r_i edge-disjoint paths, and multiple (unrestricted number of) copies of an edge may be included. This non-{0,1}-covering problem can be cast as (C1-P). We have an x_e variable for every edge e; the player-specific constraints are: $x(\delta(Q)) \geq r_i$ for every s_i-t_i cut Q, and there are no global constraints. There is an LP-relative 2-approximation algorithm for SNDP [35].

Example 4.2 *Makespan-minimization on unrelated machines.* Jobs are players; we have a variable $x_{\ell i}$ for every machine ℓ and job i, and a variable T. We want to minimize T subject to the player-specific constraints $\sum x_{\ell i} \geq 1$ (for every job $i \in S$), and the global constraints $T - \sum_i p_{\ell i} x_{\ell i} \geq 0,\ T - \sum_\ell p_{\ell i} x_{\ell i} \geq 0 \quad \forall i, \ell$. There is an LP-relative 4-approximation algorithm for this problem [16].

Given the LP-relaxation (C1-P) for the CM problem, the corresponding SCM problem can be encoded as follows. For each player i, we introduce a variable z_i that indicates that i does not receive service. The player-i-specific constraints then get modified to $A^{(i)}x + b^{(i)}z_i \geq b^{(i)}$. So we obtain the following LP-relaxation for the SCM problem.

$$
\begin{aligned}
\min \quad & c^T x + v^T z && && \text{(SC1-P)} \\
\text{s.t.} \quad & A^{(i)}x + b^{(i)}z_i \geq b^{(i)} && \text{for all } i && (1) \\
& Bx \geq d && && (2) \\
& x \in \mathbb{R}^m_+, \;\; z \in \mathbb{R}^n_+. && &&
\end{aligned}
$$

Observe that a feasible solution x to C1-P(S) extends to a feasible solution to (SC1-P) by setting $z_i = 1$ for $i \notin S$.

Constructing the mechanism. We use Alg to devise a polytime mechanism $M = (g, \{q_i\})$ as follows. On input v, we compute an optimal solution (x^*, z^*) to (SC1-P) using some fixed total ordering over vectors (e.g., lexicographic ordering) to break ties if there are multiple optimal solutions. We refer to (x^*, z^*) as *the* optimal solution for input v. We return $g(v) = W = \{i : z^*_i = 0\}$ as the winner set, and use Alg to compute a solution for W. Let $\{q_i\}$ be the prices implementing g (which we prove are polytime computable).

Theorem 4.3 *M is a polytime truthful, no-bossy, $(\rho + 1)$-approximation, ICT mechanism for any SCM problem where the cost-minimization problem is captured by* (C1-P).

Proof : Lemma 4.4 proves the approximation guarantees, and Lemma 4.5 shows that M is ICT and the prices implementing g are polytime computable. We argue that g is monotone and satisfies no-bossiness. Consider some input v, and let $W = g(v)$. Let (x^*, z^*) be the optimal solution to (SC1-P) for input v. Fix any winner $i \in W$. Let $v'_i > v_i$, and let (x', z') be the optimal solution computed to (SC1-P) for input $v' = (v'_i, v_{-i})$. It is easy to see that $z'_i \leq z^*_i = 0$: adding the inequalities $c^T x^* + v^T z^* \leq c^T x' + v^T z'$ and $c^T x' + v'^T z' \leq c^T x^* + v'^T z^*$ and simplifying gives $(v_i - v'_i)(z^*_i - z'_i) \leq 0$. Hence, i remains a winner under input v'.

Further, we claim that $(x', z') = (x^*, z^*)$. Observe that $(v_i - v'_i)(z^*_i - z'_i) = 0$ implies that $c^T x^* + v^T z^* = c^T x' + v^T z'$ and $c^T x' + v'^T z' = c^T x^* + v'^T z^*$. So both (x^*, z^*) and (x', z') are optimal solutions for both v and v'. So since we use a fixed tie-breaking rule, this means that $(x^*, z^*) = (x', z')$. Thus, M computes the same solution for both v and v', which means that M satisfies no-bossiness. ■

Lemma 4.4 *M achieves a $(\rho + 1)$-approximation.*

Proof : Fix an input v. Let $W = g(v)$, and (x^*, z^*) be the optimal solution to (SC1-P) for input v and OPT denote its value. The social cost of the solution computed by M for W is $C_M(W) + \sum_{i:z^*_i>0} v_i$. Notice that x^* is a feasible solution to (C1-P(W)), so the performance guarantee of Alg implies that $C_M(W) \leq \rho \cdot OPT_{\text{C1-P(W)}} \leq \rho \cdot c^T x^*$. So it suffices to argue that $\sum_{i:z^*_i>0} v_i \leq OPT$. This follows by looking at the dual of (SC1-P), and complementary slackness. The dual of (SC1-P) is

$$\max \quad \sum_i {b^{(i)}}^T \mu_i + d^T \omega \tag{SC1-D}$$

$$\text{s.t.} \quad \sum_i {A^{(i)}}^T \mu_i + B^T \omega \le c \tag{3}$$

$${b^{(i)}}^T \mu_i \le v_i \qquad \text{for all } i \tag{4}$$

$$\mu_i, \omega \ge 0.$$

Here μ_i and ω are nonnegative dual variables corresponding respectively to the primal constraints (1) and (2), and (4) is the dual constraint corresponding to the primal variable z_i. Let $(\{\mu_i^*\}, \omega^*)$ be an optimal dual solution. By complementary slackness, if $z_i^* > 0$ then ${b^{(i)}}^T \mu_i^* = v_i$ and hence, $\sum_{i: z_i^* > 0} v_i \le \sum_i {b^{(i)}}^T \mu_i^* \le OPT$; the last inequality follows from strong duality and since $d^T \omega^* \ge 0$. ■

Lemma 4.5 *M is ICT and the prices implementing g are polytime computable.*

Proof : Fix an input v and a winning player i. Let $OPT(t)$ denote the optimal value of (SC1-P) for (t, v_{-i}). The threshold value at which i wins is the smallest value t such that there is some optimal solution $(x(t), z(t))$ to (SC1-P) for (t, v_{-i}) with $z_i(t) = 0$. This is because for any $t' > t$, every optimal solution to (SC1-P) must have $z_i = 0$ (by the monotonicity proof in Theorem 4.3). So i wins under every bid $t' > t$ and loses under every bid $t' < t$. Let OPT' denote the optimal value of (SC1-P) when we force $z_i = 0$. Notice that $OPT(t) \le OPT'$, and when $z_i(t) = 0$, we have $OPT(t) = OPT'$. So the threshold value is given by $\min t \quad \text{s.t.} \quad OPT(t) \ge OPT'$. Notice that $OPT(t) \ge OPT'$ is equivalent to the condition that there exists a feasible solution $(\{\mu_i\}, \omega)$ to (SC1-D) (for (t, v_{-i})), and a feasible solution (x, z) to (SC1-P) with $z_i = 0$, such the value of $(\{\mu_i\}, \omega)$ is at least the value of (x, z). Thus, the threshold value can be computed efficiently by solving an LP.

Suppose that $(\{\mu_i^*\}, \omega^*, x^*, z^*)$ is an optimal value to this "threshold-LP". Then, observe that (μ_i^*, ω^*) is a feasible solution to the dual of C1-P($\{i\}$). So $t = {b^{(i)}}^T \mu_i^* \le {b^{(i)}}^T \mu_i^* + d^T \omega^* \le OPT_{\text{C1-P}(\{i\})} \le C(i)$. Hence, M satisfies ICT. ■

Cost-minimization problems with multiplicity constraints. The following LP-relaxation for the CM problem closely resembles (C1-P) but allows for multiplicity constraints.

$$\min \quad c^T x \tag{C2-P}$$

$$\text{s.t.} \quad A^{(i)} x \ge b^{(i)} \qquad \text{for all } i \in S \tag{5}$$

$$Bx \ge d$$

$$0 \le x \le u \in \mathbb{Z}_+^m.$$

As before, we require that $d \ge 0$ and $A^{(i)}, b^{(i)} \ge 0$ for every i, $OPT_{\text{C2-P}(S)} \le C(S)$ for every set S, and that we have an algorithm Alg that returns a solution of cost at most $\rho \cdot OPT_{\text{C1-P}(S)}$ for every set S. Further, we also require that $B \ge 0$, and for every i, if

$A^{(i)}_{re} > 0$ and $u_e > 0$ then $A^{(i)}_{re}u_e \geq b^{(i)}_r$. As before, in the corresponding LP-relaxation (SC2-P) of the SCM problem (see Appendix A), we have a variable $z_i \geq 0$ for every player i, we change the objective function to min $c^T x + v^T z$, and replace constraint (5) by $A^{(i)}x + b^{(i)}z_i \geq b^{(i)}$ for all i.

Mechanism M is constructed exactly as before; the only obvious change is that we now solve (SC2-P) (instead of (SC1-P)) to get (x^*, z^*). The proof of the following theorem is very similar to that of Theorem 4.3 and appears in Appendix A.

Theorem 4.6 *M is a polytime truthful, no-bossy, $(\rho + 1)$-approximation, ICT mechanism for any SCM problem where the cost-minimization problem is captured by* (C2-P).

Generalizing Lemma 4.5, we show that when the cost-minimization LP is (C1-P), the revenue of the constructed mechanism M is at most the optimal value of (C1-P) for the winner set $g(v)$. This will be useful in Section 5.

Lemma 4.7 *If the LP used for the CM problem is* (C1-P)*, the resulting mechanism M satisfies $\sum_{i \in g(v)} q_i(v) \leq OPT_{C1\text{-}P(g(v))} \leq OPT_{SC1\text{-}P}$ for any input v.*

Proof : Let $W = g(v)$ and fix some $i \in W$. Let (x^*, z^*) and (μ^*, w^*) be the optimal solutions to (SC1-P) and (SC1-D) for input v. Recall that $q_i(v)$ is the smallest value t such that there is some optimal solution $(x(t), z(t))$ to (SC1-P) for (t, v_{-i}) with $z_i(t) = 0$. We claim that $q_i(v) \leq {b^{(i)}}^T \mu^*_i$. Notice that (μ^*, w^*) remains a dual optimal solution for any input (t, v_{-i}) where $t > {b^{(i)}}^T \mu^*_i$ since it is feasible for the dual and satisfies complementary slackness with the primal solution (x^*, z^*). Thus, by complementary slackness every primal optimal solution for the input (t, v_{-i}) must have $z_i = 0$.

Observe that (μ^*, w^*) is a feasible solution for the dual of C1-P(W). It follows that $\sum_{j \in W} q_j(v) \leq \sum_{j \in W} {b^{(j)}}^T \mu^*_j \leq \sum_j {b^{(j)}}^T \mu^*_j + d^T \omega^* \leq OPT_{\text{C1-P}(W)}$. ■

Remark 4.8 We remark that although our construction is described in terms of an LP model for the CM problem, our ideas have wider applicability. In particular, we can also allow for a semidefinite-programming- or convex- relaxations of the CM and SCM problems that involve covering constraints. The proof of monotonicity and no-bossiness is unchanged. Examining Lemma 4.4, the only property we need is that the optimal Lagrangian multipliers (i.e., dual values) for the constraints involving the z_i variables can "pay" (approximately) for $\sum_{i: z^*_i > 0} v_i$. Lemma 4.5 easily extends: if $OPT(t)$ denotes the optimal value of the convex-program for the SCM problem for (t, v_{-i}), then the price of a winner i under v_{-i} is the minimum t such that $OPT(t) \geq$ optimum value of the SCM problem when we fix $z_i = 0$, which can be efficiently computed since $OPT(t)$ is a concave function of t.

5. Upper bounds on revenue

One criticism that may be levied against our mechanisms is that the mechanism may perhaps grossly overcharge the players in ensuring cost-recovery, and the potentially large revenue of the mechanism in not accounted for in any way, either in the SCM objective, or the cost-recovery condition. In this section, we address this criticism and show that our constructions can be adapted in simple ways to obtain revenue upper bounds.

The conventional way of ensuring bounded revenue is via the β-budget-balance condition, which requires that the revenue obtained recover the cost *and* be at most β times the cost incurred. We do not know how to obtain such a"two-sided" guarantee in general[3], but we consider an alternate way for upper bounding revenue (subject to cost-recovery) where we penalize revenue in the objective and seek to minimize the revenue + total value of excluded players. More formally, we consider the mechanism-design problem of designing a truthful mechanism $M = (f, \{p_i\})$ that (approximately) minimizes $\sum_{i \in f(v)} p_i(v) + \sum_{i \notin f(v)} v_i$, subject to the constraint $\sum_{i \in f(v)} p_i(v) \geq C(f(v))$. This is equivalent to maximizing the sum of the players' utilities, an objective considered in [32]. We call $\sum_{i \in f(v)} p_i(v) + \sum_{i \notin f(v)} v_i$, the *disutility* of the mechanism.

Since disutility minimization is not (equivalent to) social-welfare maximization, in general there is no pointwise-optimal truthful mechanism, i.e., one that has minimum disutility for every input v. But (unlike the profit-maximization objective, where the same issue arises) it is easy to choose a benchmark relative to which meaningful guarantees may be obtained. This benchmark is simply the optimum social cost for input v, which is clearly a lower bound on the disutility of a mechanism for input v.

We observe that the constructions in Sections 3 and 4 directly yield cost-recovering mechanisms with good approximation relative to this benchmark. Consider any cost-sharing problem where the CM problem is captured by (C1-P). Suppose we have an LP-relative ρ-approximation algorithm for the CM problem. Fix an input v. Let OPT be the optimal social cost for v. Let $S_0 = g(v)$ be the output of the mechanism $M = (g, \{q_i\})$ constructed in Section 4. Let $M'_2 = (f'_2, \{p'_{2,i}\})$ be the modified M_2 mechanism described at the end of Section 3. Let $W \subseteq S_0 = f'_2(v)$. As noted at the end of Section 3, we have $\sum_{i \in W} p'_{2,i}(v) \leq \sum_{i \in W} q_i(v) + (1 + H_n)C_M(S_0)$. We infer that M'_2 has disutility at most $\sum_{i \in W} q_i(v) + (1 + H_n)C_M(S_0) + \sum_{i \notin W} v_i$. Applying Lemma 4.7 this can be bounded by

$$OPT_{\text{C1-P}(S_0)} + (1 + 2H_n)C_M(S_0) + \sum_{i \notin S_0} v_i \leq (1 + \rho(1 + 2H_n))OPT.$$

Theorem 5.1 *Consider any CM problem that is modeled by the LP* (C1-P)*. Given an LP-relative ρ-approximation algorithm for the CM problem, we can obtain (in polytime) a truthful, cost-recovering mechanism with approximation $O(\rho \log n)$ for the disutility-minimization problem. This mechanism is also ICT if the cost function is subadditive.*

6. Applications

We showcase the versatility of our reductions by considering cost-sharing problems from various domains for which our constructions yield the first or improved results.

[3]If $C(.)$ is a symmetric submodular function, then we obtain 2-budget-balance. Taking the input mechanism M in mechanism M_1 of Section 3 to be the VCG mechanism, which satisfies $\sum_{i \in W} q_i(v) \leq C(W)$ and is now polytime, since $\frac{C(S)}{|S|}$ increases as $|S|$ decreases, we have $\sum_{i \in W} p_{1,i}(v) \leq \sum_{i \in W} q_i(v) + C(W) \leq 2 \cdot C(W)$.

In each case, we only need to verify that the cost-minimization problem admits an LP-relaxation of the form (C1-P) (or (C2-P)) and we have a suitable approximation algorithm for it. The cost function in all of these applications is subadditive, so all our resulting mechanisms are polytime and ICT.

Survivable network design problems (SNDPs). SNDPs are cost-minimization problems where each player is an (s_i, t_i) pair who requires r_i edge-, or element-, or vertex-disjoint paths between s_i and t_i (giving rise to EC-, ELC-, VC- SNDP respectively). We consider the setting where an unrestricted number of copies of an edge may be bought, in which case SNDP can be cast as (C1-P). Section 4 shows this for EC-SNDP. This also holds for the standard LP-relaxations for ELC-SNDP and VC-SNDP (see, e.g., [21] and [11]) as shown below. Let V be the node-set and $T = \bigcup_i \{s_i, t_i\}$. For ELC-SNDP, the player-i constraints $A^{(i)}x \geq b^{(i)}$ consist of

$$x(P, Q) \geq r_i - |V \setminus (P \cup Q)| \qquad \forall P, Q \subseteq V, P \cap Q = \emptyset, s_i \in P, t_i \in Q, P \cup Q \supseteq T.$$

For VC-SNDP, player i's constraints comprise

$$x(P, Q) \geq r_i - |V \setminus (P \cup Q)| \qquad \forall P, Q \subseteq V, P \cap Q = \emptyset, s_i \in P, t_i \in Q.$$

These problems admit LP-relative approximation algorithms with the following guarantees: 2 for EC-SNDP [35] and ELC-SNDP [21], and $O(r_{\max}^3 \log n)$ for VC-SNDP [14]. (Although not explicitly stated, the algorithm of [14] also obtains an LP-relative approximation. Also, improved LP-relative guarantees are known for the single-source and all-pairs versions of VC-SNDP, which translate to our mechanisms.)

Theorem 6.1 *There are truthful, cost-recovering mechanisms for EC-SNDP, ELC-SNDP, and VC-SNDP with approximation ratios of $O(\log n)$, $O(\log n)$ and $O(r_{\max}^3 \log^2 n)$ respectively.*

These are the *first* results for SNDP cost-sharing problems. Previously, cost-sharing mechanisms were only known for the special cases of Steiner tree and Steiner forest; even for these, our result improves upon the $O(\log^2 n)$-approximation achieved by Moulin mechanisms [53, 13] and acyclic mechanisms [44].

Facility location (FL). Various FL problems (where the clients are players) are captured by (C1-P). In all such problems, we have variables $\{y_\ell\}$ for the facilities and variables $\{x_{\ell i}\}$ for (facility, client) pairs, and $A^{(i)}x \geq b^{(i)}$ corresponds to the constraints $\sum_\ell x_{\ell i} \geq 1$. In *uncapacitated* FL (UFL), the global constraints $Bx \geq d$ are: $y_\ell - x_{\ell i} \geq 0$ for every i, ℓ; in *soft-capacitated* FL (soft-CFL), the global constraints also include $u_\ell y_\ell - \sum_i x_{\ell i} \geq 0$ for every ℓ. (Notice that although the LP for a client-set S has variables also for clients not in S, we include the $\sum_\ell x_{\ell i} \geq 1$ constraints only for players in S, so this does indeed model the min-cost problem for S.)

(C1-P) also captures *connected facility location* (ConFL) problems [27, 56, 39]. In the general version, *multicommodity ConFL*, each player i is an (s_i, t_i) pair. Serving a set S of players involves assigning the s_is and t_is to nodes called facilities, and building a network where for every (s_i, t_i) pair, the facilities catering to s_i and t_i are interconnected. The LP-relaxation of Kumar et al. [39] for multicommodity ConFL is of the type (C1-P): $\sum_{\ell} x_{\ell s_i} \geq 1$, $\sum_{\ell} x_{\ell t_i} \geq 1$ for every $i \in S$ are the player-specific constraints, and the global constraints are the remaining constraints that also involve edge variables $\{w_e\}$: $w(\delta(Q)) - (\sum_{\ell \in Q} x_{\ell s_i} - x_{\ell t_i}) \geq 0$, $w(\delta(Q)) - (\sum_{\ell \in Q} x_{\ell t_i} - x_{\ell s_i}) \geq 0$ for every ℓ, i and node-set Q.

Both UFL and soft-CFL have LP-relative $O(1)$-approximations; see, e.g., [10] and [43] and the references there in. [39] designed an $O(1)$-approximation for multicommodity ConFL relative to the LP described above. So we obtain the following results which, match the guarantees known for UFL [52, 44], yield the first results for soft-CFL, and improve upon the $O(\log^2 n)$-approximation for multicommodity ConFL [52].

Theorem 6.2 *UFL, soft-CFL, and multicommodity ConFL have truthful,* $O(\log n)$*-approximation, cost-recovering mechanisms.*

Set-cover problems. In the cost-sharing problem the players are the elements to be covered. The natural LP-relaxation for set cover (and hence vertex cover) is easily seen to be a special case of (C1-P). Set cover and vertex cover have LP-relative $O(\log n)$- and 2-approximation algorithms respectively. A direct application of Theorems 4.3 and 3.1 yields an $O(\log^2 n)$-approximation for set cover (and $O(\log n)$-approximation for vertex cover). But as observed in Remark 3.3, for any $\beta \geq 1$, we can obtain $O(\log^2 n/\beta)$-approximation and β-cost-recovery. So we obtain the following, which matches the results in [44].

Theorem 6.3 *There are truthful, cost-recovering mechanisms for vertex cover and set cover with approximation ratios of* $O(\log n)$ *and* $O(\log^2 n)$ *respectively. For set cover, we also obtain an* $O(\log n)$*-approximation,* $O(\log n)$*-cost-recovering mechanism.*

Scheduling. We now consider scheduling problems where the jobs are players and the cost of a set S of jobs is the minimum makespan incurred for scheduling jobs in S on a given set of unrelated machines. An LP-relaxation for this problem was given in Section 4 and shown to be of the form (C1-P). Correa et al. [16] devise a 4-approximation relative to this LP. Thus, we obtain the following theorem.

Theorem 6.4 *There is a truthful,* $O(\log n)$*-approximation, cost-recovering mechanism for makespan-minimization on unrelated machines.*

7. Extensions to multidimensional settings

In this section, we show that our ideas can be applied to obtain guarantees also for various *multidimensional* cost sharing problems. We consider two types of multidimensional settings.

In the first setting (Section 7.1), which we call the *multi-element* (ME) setting, each player $i \in [n]$ now controls a publicly known disjoint set of elements E_i and outcomes are now subsets of $E := \bigcup_i E_i$ (so $\mathcal{A} \subseteq 2^E$). We consider additive valuations, so player i's private type is a vector $v_i \in \mathbb{R}_+^{E_i}$ with $v_{i,e} > 0$ denoting the value i gets if element $e \in E_i$ is served; the value of a player i under an outcome $T \subseteq E$ is $v_i(T) :=$

$\sum_{e \in T \cap E_i} v_{i,e}$. Correspondingly, the social cost of outcome T is now given by $\mathsf{SC}(T) := C(T) + \sum_i v_i(E_i \setminus T)$.

The second multidimensional setting is the *multi-demand* (MD) setting considered by [44]. Here each player $i \in [n]$ has a publicly known maximum level of service $R_i \in \mathbb{Z}_+$, and an outcome specifies the level of service offered to each player i. Player i's private type is a vector $v_i \in \mathbb{R}_+^{R_i}$ with $v_{i,k} > 0$ denoting the additional value to i of level k over level $k-1$. Section 7.2 contains a precise definition of the problem.

Two related, but distinct, difficulties arise when considering multidimensional cost-sharing problems. First, we no longer have a simple condition like (value) monotonicity [48] for the implementability of an allocation rule. The implementability condition is now much more demanding, requiring it to satisfy cycle-monotonicity [51] in general, or weak-monotonicity with convex domains [3, 54]. Second, the specification of prices implementing an allocation rule is significantly more involved; prices are obtained by computing shortest paths in a certain allocation graph (see, e.g., [26, 41]) whose size is polynomial in $|\mathcal{A}|$.

We avoid these difficulties by essentially reducing the multidimensional problem to the single-dimensional problem. We consider "all-or-nothing" outcomes, where either the entire element-set or demand of a player is served or none of it is served. By suitably adapting the constructions and arguments in Sections 3 and 4, we then obtain guarantees for the multidimensional problem. However, an artifact of focusing on all-or-nothing outcomes is that our guarantees degrade with the dimensionality of the problem. A very interesting open question is to remove or reduce the dependence on the dimensionality. In this context, we note that a Moulin mechanism equipped with a cross-monotonic, γ-budget-balanced, α-summable (as defined in [53]) cost-sharing method yields a truthful, $(\alpha + \gamma)$-approximation, γ-cost-recovering mechanism for a special case of the ME problem called the LinME problem that we define in Section 7.1 (see Appendix B). But as remarked earlier, we do not have a black-box way of obtaining such cost-sharing methods (with small α, γ) and they may not even exist! A challenging open question is whether one can obtain reductions for multidimensional problems that are analogous to those in Sections 3 and 4.

7.1. The multi-element setting

Notice that since a player's valuation is additive, the multi-element SCM problem is *algorithmically* identical to the SCM problem where we consider each element $e \in E$ to be a player whose value is $v_{i,e}$ if $e \in E_i$. So we can work with the same LP-models (C1-P), (C2-P) for the CM problem (where now $S \subseteq E$) and (SC1-P), (SC2-P) for the SCM problem. Let (C-P) denote the LP used for the CM problem, and (SC-P) denote the corresponding LP for the SCM problem. As usual, an LP-relative ρ-approximation algorithm for the CM problem is an algorithm that always returns a solution of cost at most ρ times the optimal value of the (C-P).

We now say that a mechanism $(f, \{p_i\})$ is individually competitive (ICT) if $p_i(v) \leq C(f(v) \cap E_i)$ for all i; call the mechanism *weakly ICT* if $p_i(v) \leq \sum_{e \in f(v) \cap E_i} C(e)$ for all i. Intermediate in difficulty between the one-element and the ME problems is the special case of the ME problem where player i has the same value $v_i \in \mathbb{R}_+$ for getting any of her elements served; so $v_i(T) := v_i|T \cap E_i|$ and $\mathsf{SC}(T) = C(T) + \sum_i v_i|E_i \setminus T|$. We call this single-dimensional problem the *linear multi-element* (LinME) problem. An *all-or-nothing* mechanism is one where the range of the allocation rule is a subset of $\{S \subseteq E : S \cap E_i \in \{\emptyset, E_i\}$ for all $i\}$.

Theorem 7.1 *Let $k = \max_i |E_i|$ and $\lambda(v) = \max_i [v_i(E_i)/\min_{e \in E_i} v_{i,e}]$. Given an LP-relative ρ-approximation algorithm for the CM problem, we can obtain a truthful, cost-recovering mechanism with approximation ratio:*
(a) $O(\rho k \log n)$ for the ME problem with subadditive C that is weakly ICT;
(b) $O((\rho + k) \log n)$ for the LinME problem that is ICT if C is subadditive; and
(c) $O(\rho \lambda(v) \log n)$ on input v for the ME problem that is ICT if C is subadditive.

To the best of our knowledge, these are the first results for multi-element cost-sharing problems. Notice that our guarantees are weaker than those obtained for the single-dimensional problem. In light of this and the aforementioned difficulties involved in multidimensional cost-sharing mechanism-design, one can ask which of our two reductions—(1) translating truthful, approximation mechanisms to cost-sharing mechanisms, and (2) translating approximation algorithms to truthful, approximation mechanisms—becomes harder in the multidimensional setting. We give a partial answer to this question. Say that Alg is a Lagrangian-multiplier-preserving (LMP) ρ-approximation algorithm for the SCM problem, if for every input v, it returns a set $S \subseteq E$ such that $C_{\mathsf{Alg}}(S) + \rho \sum_i \sum_{e \in E_i \setminus S} v_i(e) \leq \rho OPT_{\text{(SC-P)}}$. Such LMP approximations are indeed known for various SCM problems, such as Steiner tree [23], set cover, vertex cover [38], and facility location [12]. We show that if one has an LMP ρ-approximation algorithm for the SCM problem, then one can obtain a randomized *truthful-in-expectation*, ρ-approximation mechanism for the SCM problem. A truthful-in-expectation mechanism is one where each player always maximizes her expected utility by revealing her true type.

Theorem 7.2 *Given an LMP ρ-approximation algorithm* Alg *for the SCM problem, one can obtain a truthful-in-expectation, ρ-approximation mechanism for the ME problem.*

The proof of Theorem 7.2 is based on the convex-decomposition idea used in [40] and appears in Appendix B. Unfortunately, we do not how to modify the mechanism of Theorem 7.2 so as to achieve cost-recovery, and in fact, in the multidimensional setting, we do not know of any black-box way of injecting cost-recovery into a (deterministic) truthful, approximation mechanism (that is possibly required to satisfy some additional properties).

Proof of Theorem 7.1 : Part (a) follows from the following simple reduction to the single-dimensional setting. We partition E $(= \bigcup_i E_i)$ into k sets $U_1, \ldots, U_k$ such that $|U_j \cap E_i| \leq 1$ for all i and j. (It is easy to construct such a partition by repeatedly picking a new element from the E_is to construct a new part U_j.) Each U_j induces a (one-element) single-dimensional problem where player i has the element $E_i \cap U_j$; if $E_i \cap U_j = \emptyset$, then player i does not participate. We use the constructions in Sections 3 and 4 to solve this problem. Let $T_j \subseteq U_j$ be the set served, and $p_{j,i}$ be the prices charged to player i in the j-th problem; we set $p_{j,i}(v) = 0$ if i does not participate in the

j-th problem. We return the set $\bigcup_{j=1}^{k} T_j$, and charge player i the price $\sum_{j=1}^{k} p_{j,i}$. Since $p_{j,i} \le C(T_j \cap E_i)$ for all i, j, the mechanism is weakly ICT.

Since valuations are additive, the utility of player i is the sum of her utilities in the k single-dimensional problems, and hence, the mechanism constructed is truthful. Since $\sum_{i=1}^{n} p_{j,i} \ge C(T_j)$ for each j and the cost function is subadditive, cost-recovery follows. The approximation guarantee follows from noting that the optimal solution induces a solution to the j-th problem of cost at most OPT (since C is monotone). Therefore, $C(T_j) + \sum_i \sum_{e \in (E_i \cap U_j) \setminus T_j} v_{i,e} = O(\rho \log n) OPT$, and since C is subadditive $\mathsf{SC}(\bigcup_j T_j) \le \sum_j [C(T_j) + \sum_i \sum_{e \in (E_i \cap U_j) \setminus T_j} v_{i,e}] = O(\rho k \log n) OPT$.

Parts (b) and (c). We now reduce to the single-dimensional setting by considering all-or-nothing mechanisms. Notice that an all-or-nothing mechanism is equivalently a mechanism for the (one-element) single-dimensional cost-sharing problem where the set of outcomes is $\mathcal{A}' := \{S \subseteq [n] : \bigcup_{i \in S} E_i \in \mathcal{A}\}$ (which is downwards closed since A is), the cost function $C' : \mathcal{A}' \mapsto \mathbb{R}_+$ is defined by $C'(S) := C(\bigcup_{i \in S} E_i)$ (which is subadditive if C is), and player i's private value is $v_i(E_i)$. Thus, it suffices to design truthful, no-bossy, ICT, all-or-nothing mechanisms with approximations $(\rho+k)$ and $(\rho+1)\lambda(v)$ for the LinME and ME problems respectively. We can then inject cost-recovery by applying the construction from Section 3 on the cost function C', and appeal to Theorem 3.1 to obtain the desired guarantees. An all-or-nothing mechanism is no-bossy if $E_i \subseteq g(v_i, v_{-i})$ and $E_i \subseteq g(v'_i, v_{-i})$ then $g(v_i, v_{-i}) = g(v'_i, v_{-i})$. (Notice that this is simply the condition that, when interpreted as a mechanism for the corresponding one-element problem, the mechanism is no-bossy.) We now describe how to obtain the desired (polytime) mechanisms for the LinME and ME problems. Let Alg be the LP-relative ρ-approximation algorithm for the CM problem. Recall that (SC-P) denotes the LP for the SCM problem (which has variables $z_{i,e}$ for every $i, e \in E_i$).

The LinME problem. On input v, we compute the optimal solution (x^*, z^*) to (SC-P). Let OPT denote its value. Let $z_i^*(E_i)$ denote $\sum_{e \in E_i} z_{i,e}^*$. We return $g(v) := \bigcup_{i: z_i^*(E_i)=0} E_i$ as the outcome, and use Alg to compute a solution for $g(v)$. In the sequel, we say that "i wins" if E_i is served. Let $t_i(v_{-i})$ denote the smallest value of $v_i|E_i|$ under which i wins. We set $q_i(v) = t_i(v_i)$ if i wins and 0 otherwise for all i. Clearly $M = (g, \{q_i\})$ is an all-or-nothing mechanism. We argue that M is a polytime truthful, no-bossy, $(\rho + k)$-approximation, ICT mechanism.

Suppose that i is a winner under input v. Let (x', z') be an optimal solution for $v' = (v'_i, v_{-i})$ where $v'_i > v_i$. Then, as in the proof of Theorem 4.3, it is easy to see that $z'(E_i) \le z^*(E_i) = 0$ and hence, that $(x', z') = (x^*, z^*)$. Hence, i is also a winner under v', and $g(v) = g(v')$. The proof that M is ICT and that $t_i(v_{-i})$ is polytime computable follows by mimicking the proof of Lemma 4.5 (or Theorem 4.6). To prove the approximation, as before, it suffices to show that $\sum_{i: z^*(E_i)>0} v_i|E_i| \le k \cdot OPT$. This follows since $\sum_{i: z^*(E_i)>0} v_i|E_i| \le k(\sum_{i,e: z_{i,e}^*>0} v_i)$ and the proofs of Lemma 4.4 and Theorem 4.6 show that $\sum_{i,e: z_{i,e}^*>0} v_i \le OPT$.

The ME problem. We construct the desired mechanism M by simulating the construction in Section 4 for the cost function C'. Notice that the LP-relaxation for this modified SCM problem is an LP of the same form as (SC-P) but where we have a *single variable*

z_i for each player i, with $z_i = 0$ indicating that E_i is served and $z_i = 1$ indicating that no element of E_i is served. Let (MSC-P) denote this modified SCM-LP.

On input v, we compute the optimal solution $(\tilde{x}, \tilde{z})$ to (MSC-P). Let $\tilde{\mathsf{opt}}$ denote its value. We return $g(v) := \bigcup_{i:\tilde{z}_i=0} E_i$ as the outcome, and use Alg to compute a solution for $g(v)$. Let $t_i(v_{-i})$ denote the smallest value of $v_i(E_i)$ under which i wins. We set $q_i(v) = t_i(v_{-i})$ if i wins and 0 otherwise for all i. Recall that $\lambda(v) = \max_i[v_i(E_i)/\min_{e\in E_i} v_{i,e}]$. We abbreviate this to λ in the sequel. Since $M = (g, \{q_i\})$ simulates the construction in Section 4 for the one-element modified SCM problem, the proofs in Section 4 show that M is polytime, truthful, no-bossy, and ICT.

We next prove that M has approximation ratio $(\rho + 1)\lambda$. Let $W = g(v)$. By Lemma 4.4 and Theorem 4.6 we know that $C_M(W) + \sum_{i:\tilde{z}_i>0} v_i(E_i) \le (\rho + 1)\tilde{\mathsf{opt}}$. Let (x^*, z^*) be an optimal solution to the original SCM-LP (SC-P) for input v and OPT denote its value. Define $z_i = z^*(E_i)$. Observe that (x^*, z) is a feasible solution to (MSC-P). Since $v_i(E_i) \le \lambda v_{i,e}$ for all i, $e \in E_i$, we have $\tilde{\mathsf{opt}} \le c^T x^* + \sum_i v_i(E_i) z_i \le c^T x^* + \sum_{i,e\in E_i} \lambda v_{i,e} z^*_{i,e} \le \lambda OPT$. ■

7.2. The multi-demand setting

Recall that in the multi-demand (MD) setting, each player i has a maximum level of service R_i. An outcome is a vector $\ell = (\ell_1, \ldots, \ell_n)$ with ℓ_i specifying the level offered to player i. Player i's value under ℓ is $v_i(\ell_i) := \sum_{k=1}^{\ell_i} v_{i,k}$, where $v_i \in \mathbb{R}_+^{R_i}$ is her private type with $v_{i,k} > 0$ being the marginal value of increasing i's level to k from $k-1$. We assume that $v_{i,k}$ is non-increasing with k. The social cost of outcome ℓ is $\mathsf{SC}(\ell) := C(\ell) + \sum_i (v_i(R_i) - v_i(\ell_i))$ and ICT is the condition that the price charged to a player i is at most $C(0, \ldots, 0, \ell_i, 0, \ldots, 0)$. We assume that the outcome-set is downwards-closed and C is monotone: if $\ell' \le \ell$ and $\ell' \in \mathbb{Z}_+^n$, then ℓ' is an outcome if ℓ is, and $C(\ell') \le C(\ell)$. We say that C is subadditive if $C(\ell) + C(\ell') \ge C(\{\max(\ell_i, \ell'_i)\}_{i\in[n]})$. As in Section 7.1, we also consider the single-dimensional special case where $v_{i,k} = v_i$ for all $k \in [R_i]$ (so $\mathsf{SC}(\ell) = C(\ell) + \sum_i v_i(R_i - \ell_i)$); we call this the *linear multi-demand* (LinMD) problem.

Analogous to (SC1-P) and (SC2-P), we consider MD problems where the SCM problem is captured by one of two candidate LPs. We discuss the analogue of (SC1-P) below and describe the other LP and the changes required to the proofs in Appendix C. Variable $z_{i,k} = 1$ below indicates that player i is offered level of service at most $R_i - k$.

$$\min \quad c^T x + \sum_i \sum_{k=1}^{R_i} v_{i,R_i-k+1} z_{i,k} \qquad \text{(MD1-P)}$$

$$\text{s.t.} \quad \sum_e A^{(i)}_{re} x_e + \sum_{k=1}^{b^{(i)}_r} z_{i,k} \ge b^{(i)}_r \qquad \forall i, r \qquad (6)$$

$$Bx \ge d \qquad (7)$$

$$x \in \mathbb{R}^m_+, \quad 0 \le z_{i,k} \le 1 \qquad \forall i, k.$$

As before, we require that $d \ge 0$, $A^{(i)}, b^{(i)} \ge 0$, and $b^{(i)}_r \in [R_i]$ for all i, r. The LP-relaxation corresponding to (MD1-P) for the CM problem where we want to find the min-cost way of serving each player i at level ℓ_i is obtained by dropping all the $z_{i,k}$ variables, and replacing $b_r^{(i)}$ by $\max\{0, b_r^{(i)} - (R_i - \ell_i)\}$.

Example 7.3 As an example of an MD problem modeled by (MD1-P), let us revisit EC-SNDP when multiple copies of an edge may be picked. Each player i is an (s_i, t_i) pair who requires R_i edge-disjoint s_i-t_i paths. A feasible solution may now only provide $\ell_i \leq R_i$ edge-disjoint s_i-t_i and incur a "penalty" for player i equal to $\sum_{k=\ell_i+1}^{R_i} v_{i,k}$.

Another example is the *set-multicover* problem, where an element seeks to be covered by multiple sets (and sets may be picked multiple times).

Theorem 7.4 *Let* $R_{\max} = \max_i R_i$ *and* $\lambda(v) = \max_i[v_i(R_i)/\min_{k\in[R_i]} v_{i,k}]$. *Given an LP-relative ρ-approximation algorithm for the multi-demand CM problem, we can obtain a truthful, cost-recovering mechanism with approximation ratio: (a)* $O((\rho+R_{\max})\log n)$ *for the LinMD problem; and (b)* $O(\rho\lambda(v)\log n)$ *on input v for the MD problem. Both mechanisms are ICT if C is subadditive.*

Thus, we obtain truthful, cost-recovering, ICT mechanisms for:

- EC-SNDP and ELC-SNDP with approximation $O(R_{\max}\log n)$ for the LinMD problem and $O(\lambda(v)\log n)$ for the MD problem;
- VC-SNDP with approximation $O(R_{\max}^3\log^2 n)$ for the LinMD problem and $O(R_{\max}^3\lambda(v)\log^2 n)$ for the MD problem;
- set-multicover with approximation $O((\log n + R_{\max})\log n)$ for the MD problem and $O(\lambda(v)\log^2 n)$ for the MD problem.

These approximation factors can be improved by a factor of β at the expense of obtaining β-cost-recovery. The results known in the literature for MD problems are (prices are scaled to ensure cost-recovery; the scaling factor appears in the approximation): (i) group-strategyproof (GSP) mechanisms for EC-SNDP and fault-tolerant facility location with approximation ratios $O(\log^2 R_{\max}\log^2 n)$ and $O(R_{\max}^2\log n)$ respectively [7]; (ii) weakly GSP mechanisms for set multicover and fault-tolerant facility location with approximation ratio $O(\log n(\log n + \log R_{\max}))$ [44].

Proof : The proof (as well as the theorem statement) is along the same lines as that of parts (b) and (c) of Theorem 7.1. We again consider all-or-nothing mechanisms, that is, mechanisms which return outcomes where every player i is either fully served or not served at all. For example, the all-or-nothing MD EC-SNDP problem is precisely the single-dimensional EC-SNDP problem considered in Section 6, where serving player i entails providing R_i edge-disjoint paths and not serving i incurs penalty $v_i(R_i)$.

For a set $S \subseteq [n]$, let L_S be the vector where $L_{S,i} = R_i$ if $i \in S$ and is 0 otherwise. An all-or-nothing mechanism is thus essentially a mechanism for the single-dimensional problem specified by the outcome-set $\mathcal{A}' := \{S \subseteq [n] : L_S \in \mathcal{A}\}$ (which is downwards-closed) and cost function $C' : \mathcal{A}' \mapsto \mathbb{R}_+$ given by $C'(S) := C(L_S)$ (which is subadditive if C is), where player i's private value is $v_i(R_i)$. We describe how to obtain a truthful, no-bossy, ICT all-or-nothing mechanisms with $(\rho+R_{\max})$- and $(\rho+1)\lambda(v)$- approximations for the LinMD and MD problems respectively. We can then inject cost-recovery using the construction in Section 3 for the cost-function C'. Theorem 3.1 combined with the above results then yields the stated results. No-bossiness for the one-element problem (defined by $\mathcal{A}', C'$) translates to the following no-bossiness condition in the multi-demand setting: if $g(v_i, v_{-i})_i = R_i$ and $g(v'_i, v_{-i})_i = R_i$ then $g(v_i, v_{-i}) = g(v'_i, v_{-i})$. Let Alg be the LP-relative ρ-approximation algorithm.

The LinMD problem. On input v, we compute the optimal solution (x^*, z^*) to (MD1-P). Let OPT denote its value. Let $z_i^*(R_i)$ denote $\sum_{k \in [R_i]} z_{i,k}^*$. Let $S = \{i : z^*(R_i) = 0\}$. We return $g(v) := L_S$ as the outcome, and use Alg to compute a solution for $g(v)$. In the sequel, we say that "i wins" if i is served at level R_i. Let $t_i(v_{-i})$ denote the smallest value of $v_i R_i$ under which i wins. We set $q_i(v) = t_i(v_i)$ if i wins and 0 otherwise for all i.

The proof of no-bossiness is as in Theorem 7.1. The proof that M is ICT and that $t_i(v_{-i})$ is polytime computable follows by mimicking the proof of Lemma 4.5. To prove the approximation, we show that $\sum_{i: z^*(R_i) > 0} v_i R_i \leq k \cdot OPT$. If $z^*(R_i) > 0$ then there is some k such that $z_{i,k}^* > 0$, so we only need to show that $\sum_{i,k: z_{i,k}^* > 0} v_i \leq OPT$. Let $(\{\mu_{i,r}^*\}, \omega^*, \{\pi_{i,k}^*\})$ be an optimal dual solution, where $\mu_{i,r}^*$, ω^*, and $\pi_{i,k}^*$ correspond respectively to (6), (7) and the $z_{i,k} \leq 1$ constraint. Then by complementary slackness, $\sum_{i,k: z_{i,k}^* > 0} v_i = \sum_{i,k: z_{i,k}^* > 0} \big(\sum_{r: b_r^{(i)} \geq k} \mu_{i,r}^* - \pi_{i,k}^* \big) \leq \sum_{i,r} b_r^{(i)} \mu_{i,r}^* - \sum_{i,k} \pi_{i,k}^* \leq OPT$.

The MD problem. As in the ME setting, we construct the mechanism M by simulating the construction in Section 4 for C'. The LP-relaxation for this modified SCM problem is (MD1-P) with the constraints $z_{i,k} = z_{i,k'}$ for all $i, k, k' \in [R_i]$. Let (MMD-P) denote this modified SCM-LP. Observe that (MMD-P) is of the same form as (SC1-P) (for the corresponding all-or-nothing problem).

On input v, we compute the optimal solution $(\tilde{x}, \tilde{z})$ to (MMD-P). Let $\tilde{\mathsf{opt}}$ denote its value. Let $S = \{i : \tilde{z}_{i,1} = 0\}$. We return $W = g(v) := L_S$ as the outcome, and use Alg to compute a solution for $g(v)$. Let $t_i(v_{-i})$ denote the smallest value of $v_i(R_i)$ under which i wins. We set $q_i(v) = t_i(v_{-i})$ if $i \in S$ and 0 otherwise for all i. Recall that $\lambda := \lambda(v) = \max_i [v_i(R_i) / \min_{k \in [R_i]} v_{i,k}]$. Since $M = (g, \{q_i\})$ simulates the construction in Section 4 for the (one-element) modified SCM problem, the proofs in Section 4 show that M is polytime, truthful, no-bossy, ICT, and $C_M(W) + \sum_{i \notin S} v_i(R_i) \leq (\rho + 1)\tilde{\mathsf{opt}}$. Let (x^*, z^*) be an optimal solution to the original SCM-LP (MD1-P) for input v and OPT denote its value. Observe that since the objective-function coefficients of the $z_{i,k}$s are nondecreasing with k, we may assume that $z_{i,1}^* \geq z_{i,2}^* \geq \ldots \geq z_{i,R_i}^*$. Define $z_{i,k} = z_{i,1}^*$ for all $k \in [R_i]$. Observe that (x^*, z) is a feasible solution to (MMD-P). Since $v_i(R_i) \leq \lambda v_{i,k}$ for all i, $k \in [R_i]$, we have that $\tilde{\mathsf{opt}} \leq c^T x^* + \sum_i v_i(R_i) z_{i,1}^* \leq c^T x^* + \sum_{i,k \in [R_i]} \lambda v_{i,k} z_{i,k}^* \leq \lambda OPT$. ■

8. Conclusions and Discussion

We investigate whether there is a black-box way of transforming approximation algorithms for the social-cost-minimization (SCM) problem into truthful, approximation, cost-recovering mechanisms. We provide two reductions that in combination reduce the cost-sharing mechanism-design problem to the algorithmic cost-minimization (CM) problem (of finding a minimum-cost solution for a given set of players) for a large class of problems, thereby affirmatively answering the above question. Our first reduction shows that for any (monotone) cost function, cost-recovery can be injected into any truthful, approximation mechanism satisfying an additional no-bossiness condition while losing an $O(\log n)$-factor in the approximation ratio. Complementing this, our second reduction shows that for a broad class of cost-sharing problems, any LP-relative ρ-approximation algorithm for the CM problem yields a truthful, no-bossy, $(\rho + 1)$-approximation mechanism.

Our reductions apply to the single-dimensional setting where each player's private type consists of a single nonnegative parameter. Although, we also obtain some results for multidimensional settings in Section 7, our guarantees for multidimensional problems are weak and suffer from the "curse of dimensionality". An extremely interesting and challenging research direction to emerge from our work is to obtain improved guarantees for multidimensional cost-sharing problems and to investigate whether one can obtain similar black-box reductions for multidimensional settings. Such reductions would substantially advance our understanding of cost-sharing mechanism design, and mechanism design in general, since very few reductions from mechanism design to algorithm design are known for multidimensional mechanism-design problems.

As Theorem 7.2 indicates, it might be possible to translate approximation algorithms into truthful, approximation mechanisms; however, injecting cost recovery seems to be a much harder task. One technical difficulty that arises here is that given an input truthful, approximation mechanism M (to which we want to add cost recovery), we do not have a good handle on how the output of M changes when a player changes her bid. In the single-dimensional setting, we avoided this difficulty by imposing the no-bossiness condition on M, which (along with monotoniciy) conveniently determines how M's output (which is the input to M_1) changes under single-player bid changes. In the multidimensional setting, it seems rather difficult to define a meaningful notion of no-bossiness that is both powerful enough to give us a good handle on how the output of the input mechanism M changes, and weak enough that one can devise a truthful, approximation mechanism satisfying this condition. The natural extension of the no-bossiness condition defined in Section 2 is not useful: for example, in the multi-element setting, this would say that if the chosen set of elements of player i does not change when her (reported) type changes then the mechanism's output does not change; but the chosen set of elements of player i could of course change and then we obtain no information about how the mechanism's output changes. One could define more stringent no-bossiness conditions but this further complicates the task of designing a truthful, approximation mechanism satisfying the stipulated no-bossiness condition. Defining a meaningful and tractable no-bossiness notion for randomized mechanisms seems to be an even more difficult proposition.

In light of thse difficulties, it would be insightful to understand how critical is the role played by no-bossiness in the single-dimensional setting. In particular, is it possible to obtain a reduction of the type described in Section 3 (where we inject cost recovery) without imposing no-bossiness for the input mechanism? Also, the limitations imposed by no-bossiness are not well understood. For example, can a truthful, approximation mechanism be made no-bossy efficiently without degrading its approximation factor by much? We leave these as open questions.

References

[1] A. Archer and É. Tardos. Truthful mechanisms for one-parameter agents. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 482–491, 2001.

[2] X. Bei and Z. Huang. Bayesian incentive compatibility via fractional assignments. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 720–733, 2011.

[3] S. Bikhchandani, S. Chatterjee, R. Lavi, A. Mu'alem, N. Nisan, and A. Sen. Weak monotonicity characterizes deterministic dominant-strategy implementation. *Econometrica*, 74:1109–1132, 2006.

[4] Y. Bleischwitz and B. Monien. Fair cost-sharing methods for scheduling jobs on parallel machines. *J. Discrete Algorithms*, 7(3):280–290, 2009.

[5] Y. Bleischwitz, B. Monien, and F. Schoppmann. To be or not to be (served). In *Proceedings of the 3rd International Workshop on Internet and Network Economics (WINE)*, pages 515–528, 2007.

[6] Y. Bleischwitz and F. Schoppmann. New efficiency results for makespan cost sharing. *Inf. Process. Lett*, 107(2):64–70, 2008.

[7] Y. Bleischwitz and F. Schoppmann. Group-strategyproof cost sharing for metric fault tolerant facility location. In *Proceedings of the 1st International Symposium on Algorithmic Game Theory (SAGT)*, pages 350–361, 2008.

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

[9] J. Brenner and G. Schäfer. Singleton acyclic mechanisms and their applications to scheduling problems. In *Proceedings of the 1st International Symposium on Algorithmic Game Theory (SAGT)*, pages 315–326, 2008.

[10] J. Byrka and K. Aardal. An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem. *SIAM J. Comput.*, 39(6):2212–2231, 2010.

[11] T. Chakraborty, J. Chuzhoy, and S. Khanna. Network design for vertex connectivity. In *Proceedings of the 40th Annual ACM Symposium on Theory of computing (STOC)*, pages 167–176, 2008.

[12] M. Charikar, S. Khuller, D. Mount, and G. Narasimhan. Algorithms for facility location problems with outliers. In *Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 642–651, 2001.

[13] S. Chawla, T. Roughgarden, and M. Sundararajan. Optimal cost-sharing mechanisms for steiner forest problems. In *Proceedings of the 2nd International Workshop on Internet and Network Economics (WINE)*, pages 112–123, 2006.

[14] J. Chuzhoy and S. Khanna. An $O(k^3 \log n)$-approximation algorithm for vertex-connectivity survivable network design. In *Proceedings of the 50th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 437–441, 2009.

[15] E. Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, 1971.

[16] J. Correa, M. Skutella, and J. Verschae. The power of preemption on unrelated machines and applications to scheduling orders. In *Proceedings of 12th APPROX*, pages 84–97, 2009.

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

[18] S. Dobzinski, A. Mehta, T. Roughgarden, and M. Sundararajan. Is Shapley cost sharing optimal? In *Proceedings of the 1st International Symposium on Algorithmic Game Theory (SAGT)*, pages 327–336, 2008.

[19] S. Dughmi and T. Roughgarden. Black-box randomized reductions in algorithmic mechanism design. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 775–784, 2010.

[20] J. Feigenbaum, C. Papadimitriou, and S. Shenker. Sharing the cost of multicast transmissions. *J. Comput. Syst. Sci*, 63(1):21–41, 2001.

[21] L. Fleischer, K. Jain, and D. Williamson. Iterative rounding 2-approximation algorithms for minimum-cost vertex connectivity problems. *J. Comput. Syst. Sci*, 72(5):838–867, 2006.

[22] K. Georgiou and C. Swamy. Black-box mechanisms for cost-sharing mechanism design. In *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 896–913, 2012.

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

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

[25] T. Groves. Incentives in teams. *Econometrica*, 41(4):617–31, July 1973.

[26] H. Gui, R. Muller, and R. V. Vohra. Characterizing dominant strategy mechanisms with multi-dimensional types. Working paper, 2004.

[27] A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi, and B. Yener. Provisioning a virtual private network: a network design problem for multicommodity flow. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 389–398, 2001.

[28] 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.

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

[30] J. Hartline, R. Kleinberg, and A. Malekian. Bayesian incentive compatibility via matchings. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 734–747, 2011.

[31] J. Hartline and B. Lucier. Bayesian algorithmic mechanism design. In *Proceedings of the 42nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 301–310, 2010.

[32] J. Hartline and T. Roughgarden. Optimal mechanism design and money burning. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC)*, pages 75–84, 2008.

[33] Z. Huang, L. Wang, and Y. Zhou. Black-box reduction in mechanism design. In *Proceedings of the 14th APPROX*, pages 254-265, 2011.

[34] N. Immorlica, M. Mahdian, and V. Mirrokni. Limitations of cross-monotonic cost-sharing schemes. *ACM Transactions on Algorithms*, 4(2), 2008.

[35] K. Jain. A factor 2 approximation algorithm for the generalized Steiner network problem. *Combinatorica*, 21(1):39–60, 2001.

[36] K. Jain and V. Vazirani. Applications of approximation algorithms to cooperative games. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 364–372, 2001.

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

[38] J. Könemann, O. Parekh, and D. Segev. A unified approach to approximating partial covering problems. *Algorithmica*, 59(4):489–509, 2011.

[39] A. Kumar, A. Gupta, and T. Roughgarden. A constant-factor approximation algorithm for the multicommodity. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 333–344, 2002.

[40] R. Lavi and C. Swamy. Truthful and near-optimal mechanism design via linear programming. *Journal of the ACM*, 58(6):25, 2011.

[41] R. Lavi and C. Swamy. Truthful mechanism design for multi-dimensional scheduling via cycle monotonicity. *Games and Economic Behavior*, 67(1):99–124, 2009.

[42] S. Leonardi and G. Schäfer. Cross-monotonic cost sharing methods for connected facility location games. *Theor. Comput. Sci*, 326(1-3):431–442, 2004.

[43] M. Mahdian, Y. Ye, and J. Zhang. Approximation algorithms for metric facility location problems. *SIAM J. Comput.*, 36(2):411–432, 2006.

[44] A. Mehta, T. Roughgarden, and M. Sundararajan. Beyond Moulin mechanisms. *Games and Economic Behavior*, 67:125–155, 2009.

[45] H. Minoeei and C. Swamy. Truthful mechanism design for multidimensional covering problems. In *Proceedings of the 8th Workshop on Internet and Network Economics (WINE)*, pages 448–461, 2012.

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

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

[48] R. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6:58–73, 1981.

[49] M. Pál and É Tardos. Group strategyproof mechanisms via primal-dual algorithms. In *Proceedings of the 43rd Annual ACM Symposium on Foundations of Computer Science (FOCS)*, pages 584–593, 2003.

[50] K. Roberts. The characterization of implementable choice rules. In *Aggregation and Revelation of Preferences*, pages 321–349, 1979.

[51] J. C. Rochet. A necessary and sufficient condition for rationalizability in a quasi-linear context. *Journal of Mathematical Economics*, 16:191–200, 1987.

[52] T. Roughgarden and M. Sundararajan. Optimal efficiency guarantees for network design mechanisms. In *Proceedings of the 13rd International Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 469–483, 2007.

[53] T. Roughgarden and M. Sundararajan. Quantifying inefficiency in cost-sharing mechanisms. *Journal of the ACM*, 56(4), 2009.

[54] M. Saks and L. Yu. Weak monotonicity suffices for truthfulness on convex domains. In *Proc. 6th EC*, pages 286–293, 2005.

[55] M. Satterthwaite and H. Sonnenschein. Strategy-proof allocation mechanisms at differentiable points. *Review of Economic Studies*, XLVIII:587–597, 1981.

[56] C. Swamy and A. Kumar. Primal-dual algorithms for connected facility location problems. *Algorithmica*, 40(4):245–269, 2004.

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

A. Proof of Theorem 4.6

The LP-relaxation of the SCM problem obtained from (C2-P) is as follows.

$$
\begin{aligned}
\min \quad & c^T x + v^T z && && \text{(SC2-P)}\\
\text{s.t.} \quad & A^{(i)} x + b^{(i)} z_i \geq b^{(i)} && \text{for all } i\\
& Bx \geq d\\
& 0 \leq x \leq u \in \mathbb{Z}^m_+, \quad z \in \mathbb{R}^n_+.
\end{aligned}
$$

The argument for the monotonicity and no-bossiness of the allocation rule is exactly as in the proof of Theorem 4.3. The proof that prices are polytime computable mimics the proof of Lemma 4.5.

We now prove the approximation ratio and that M is ICT. Fix an input v. Let $W = g(v)$ and (x^*, z^*) be the optimal solution to (SC2-P) for input v. As in the proof of Lemma 4.4, it suffices to show that $\sum_{i:z^*_i>0} v_i \leq OPT = OPT_{\text{SC2-P}}$. Again the key is to consider the dual of (SC2-P).

$$
\begin{aligned}
\max \quad & \sum_i {b^{(i)}}^T \mu_i + d^T \omega - u^T \theta && && \text{(SC2-D)}\\
\text{s.t.} \quad & \sum_i {A^{(i)}}^T \mu_i + B^T \omega - \theta \leq c && && (8)\\
& {b^{(i)}}^T \mu_i \leq v_i && \text{for all } i && (9)\\
& \mu_i, \omega \geq 0, \quad \theta \in \mathbb{R}^m_+.
\end{aligned}
$$

Here θ is the new dual variable corresponding to the $x \leq u$ multiplicity constraints. Let $(\{\mu^*_i\}, \omega^*, \theta^*)$ be an optimal dual solution. Then $\sum_{i:z^*_i>0} v_i = \sum_{i:z^*_i>0} {b^{(i)}}^T \mu^*_i$. Note that $\theta^*_e > 0$ implies that $x^*_e = u_e$, so $\sum_{e:\theta^*_e>0} u_e c_e = \sum_{e:\theta^*_e>0} u_e (\sum_{i,r} A^{(i)}_{re} \mu^*_{ir} + \sum_r B_{re} \omega^*_r - \theta^*_e)$. So we have $\sum_{i:z^*_i>0} {b^{(i)}}^T \mu^*_i + \sum_{e:\theta^*_e>0} u_e c_e = \sum_{i,r} \mu^*_{ir} \kappa_{ir} + \sum_{r,e:\theta^*_e>0} x^*_e B_{re} \omega^*_r - u^T \theta^*$ where $\kappa_{ir} = b^{(i)}_r + \sum_{e:\theta^*_e>0} u_e A^{(i)}_{re}$ if $z^*_i > 0$ and $\kappa_{ir} = \sum_{e:\theta^*_e>0} u_e A^{(i)}_{re}$ otherwise.

We claim that $\mu^*_{ir} \kappa_{ir} \leq \mu^*_{ir} b^{(i)}_r$. This is true if $z^*_i = 0$ since then $\mu^*_{ir} \kappa_{ir} = \mu^*_{ir} (\sum_{e:\theta^*_e>0} u_e A^{(i)}_{re}) \leq \mu^*_{ir} (\sum_e x^*_e A^{(i)}_{re}) = \mu^*_{ir} b^{(i)}_r$. If $u_e \geq 1$, $z^*_i > 0$, and $\mu^*_{ir} > 0$, then if $A^{(i)}_{re} > 0$ we must have $\theta^*_e = 0$. Otherwise, we have $x^*_e = u_e \geq 1$, and hence, $\sum_{e'} A^{(i)}_{re'} x^*_{e'} + b^{(i)}_r z^*_i > b^{(i)}_r$ since $A^{(i)}_{re} u_e \geq b^{(i)}_r$, which contradicts complementary slackness. Therefore, we have $\mu^*_{ir} \kappa_{ir} \leq \mu^*_{ir} b^{(i)}_r$. Also, $\sum_{r,e:\theta^*_e>0} x^*_e B_{re} \omega^*_r \leq \sum_{r,e} x^*_e B_{re} \omega^*_r = d^T \omega^*$. So $\sum_{i:z^*_i>0} v_i \leq \sum_{i,r} \mu^*_{ir} \kappa_{i,r} + \sum_{r,e:\theta^*_e>0} x^*_e B_{re} \omega^*_r - u^T \theta^* \leq \sum_i {b^{(i)}}^T \mu^*_i + d^T \omega^* - u^T \theta^* = OPT$.

Let t^* be the price of a winning player i. For any $t < t^*$, we know that every optimal solution $(x(t), z(t))$ to (SC2-P) for (t, v_{-i}) must have $z_i(t) > 0$. Let $(\{\mu^*_i\}, \omega^*, \theta^*)$ be an optimal solution to (SC2-D), Then $t = {b^{(i)}}^T \mu^*_i$ and the argument above shows that $t + \sum_{j \neq i} {b^{(j)}}^T \mu^*_j \leq \sum_j {b^{(j)}}^T \mu^*_j + d^T \omega^* - u^T \theta^*$, that is, $t \leq {b^{(i)}}^T \mu^*_i + d^T \omega^* - u^T \theta^*$. Finally, observe $(\mu^*_i, \omega^*, \theta^*)$ is a feasible solution to the dual of C2-P($\{i\}$), so $t \leq OPT_{\text{C2-P}(\{i\})} \leq C(i)$. This holds for all $t < t^*$, so $t^* \leq C(i)$. This proves that M is ICT.

B. Omitted details from Section 7.1

Proof of Theorem 7.2 : The construction exploits the convex-decomposition idea used in [40]. This observation is also used in [45]. On input v, we compute an optimal solution (x^*, z^*) to (SC-P). Let OPT denote its value. Note that since we solve (SC-P) optimally, we can use the VCG prices $\{p_i\}$ (which can be computed efficiently) to obtain a fractional truthful mechanism. More precisely, this means that if (x', z') is the optimal solution to (SC-P) for input (v'_i, v_{-i}), then we have $\sum_{e\in E_i} v_{i,e}(1 - z^*_{e,i}) - p_i(v) \geq \sum_{e\in E_i} v_{i,e}(1 - z'_{e,i}) - p_i(v'_i, v_{-i})$. We show that using Alg one can obtain a convex combination of polynomially many integral solutions to (SC-P) such that $\Pr[\text{element } e \in E_i \text{ is not served}] = z^*_{e,i}$ for all $i, e \in E_i$ and the expected cost is at most $\rho \cdot OPT$. Our randomized allocation rule chooses an integral solution with probability equal to its weight in this convex combination. Let X denote the random set served. As in [40], we can come up with random prices $\{Q_i\}$ such that $Q_i(v) \leq v_i(X)$ and $\mathrm{E}[Q_i(v)] = p_i(v)$. This randomized mechanism clearly achieves a ρ-approximation. It is truthful in expectation because a player's expected utility is the same as her utility in the fractional truthful mechanism. Formally, if player i's true type is $\bar{v}_i$ and the others report v_{-i}, her expected utility $\mathrm{E}[u_i(\bar{v}_i; \bar{v}_i, v_{-i})]$ when she reports $\bar{v}_i$ is $\sum_{e\in E_i} \bar{v}_{e,i}(1 - \Pr[e \text{ is not served}]) - p_i(\bar{v}_i, v_{-i}) = \sum_{e\in E_i} \bar{v}_{e,i}(1 - z^*_{e,i}) - p_i(\bar{v}_i, v_{-i})$, and if she reports v'_i, we have $\mathrm{E}[u_i(\bar{v}_i; v'_i, v_{-i})] = \sum_{e\in E_i} \bar{v}_{e,i}(1 - z'_{e,i}) - p_i(v'_i, v_{-i}) \leq \mathrm{E}[u_i(v_i; v_i, v_{-i})]$.

Let $\{(x^{(l)}, z^{(l)})\}_{l\in\mathcal{I}}$ denote the collection of all integral solutions to (SC-P) where $z_{e,i} \in \{0, 1\}$ for all $i, e \in E_i$. Examining (SC1-P) and (SC2-P) we see that if (x, z) is a feasible solution, then so is (x, z') where $z' \geq z$. Consider the following LP. The index l below indexes integral solutions and ranges over $\mathcal{I}$.

$$\begin{aligned} \max \quad & \sum_l \gamma_l && \text{(P)} \\ \text{s.t.} \quad & \sum_l \gamma_l \leq 1 && (10) \\ & \sum_l \gamma_l c^T x^{(l)} \leq \rho c^T x^* && (11) \\ & \sum_l \gamma_l z^{(l)} \leq z^* && (12) \\ & \gamma \geq 0. \end{aligned}$$

$$\begin{aligned} \min \quad & \rho(\kappa c^T x^* + w^T z^*) + \delta && \text{(D)} \\ \text{s.t.} \quad & \kappa c^T x^{(l)} + \rho w^T z^{(l)} + \delta \geq 1 && \forall l \\ & \kappa, \delta, w \geq 0. \end{aligned}$$

We claim that a feasible solution to (P) with value 1 (which is therefore an optimal solution) can be modified so that (12) holds at equality. To see this, suppose that $\sum_l \gamma_l z^{(l)}_{i,e} < z^*_{i,e}$. Then, we can take some l such that $\gamma_l > 0$ and $z^{(l)}_{i,e} = 0$ and transfer some weight from this integral solution to the integral solution $(x' = x^{(l)}, z')$, where $z'_{i,e} = 1$ and $z'_{i',e'} = z^{(l)}_{i',e'}$ for all $(i', e') \neq (i, e)$. By repeatedly doing this, we can obtain a convex combination where (12) holds at equality. To show that the optimal value of (P) is 1 and that it can be solved efficiently, we move to the dual (D). Here, δ and κ are the dual variables corresponding to (10) and (11) respectively, and $\rho w_{e,i}$ is the dual variable corresponding to the (i, e)-th constraint of (12). Suppose there is some (κ, δ, w) for which the objective value of (D) is less than 1. Then, since Alg is an LMP ρ-approximation algorithm, we can run it on the input $(c, w/\kappa)$ to obtain an integer solution $(x^{(l)}, z^{(l)})$ such that $c^T x^{(l)} + \rho w^T z^{(l)}/\kappa \leq \rho(c^T x^* + w^T z^*/\kappa) < (1-\delta)/\kappa$. But this means that the corresponding constraint of (D) is violated. This shows that $OPT_{(\text{D})} = OPT_{(\text{P})} \geq 1$, and hence is exactly one. By using the ellipsoid method with Alg providing a separation oracle, we can get an LP with only polynomially many constraints that is equivalent to (D); taking its dual yields an LP of the form (P) with only polynomially many variables. Solving this, and then tweaking the γ values so that (12) holds at equality yields the desired convex combination. ■

The performance of Moulin and acyclic mechanisms. Let $\xi : 2^{[n]} \times [n] \mapsto \mathbb{R}_+$ be a cost-sharing method for a cost function C. The following notion of summability was introduced in [53]: say that ξ is *α-summable* if for every set $S \subseteq [n]$ and every permutation $\{s_1, \ldots, s_n\}$ of the elements of S, we have $\sum_{i=1}^{n} \xi(\{s_1, \ldots, s_i\}, s_i) \leq \alpha \cdot C(S)$. Say that ξ is (β, γ)-budget-balanced if $C(S)/\gamma \leq \xi(S, S) \leq \beta C(S)$ for all $S \subseteq [n]$.

Let g_ξ be the allocation rule of the Moulin mechanism M_ξ obtained when we treat each element $e \in E_i$ as a player with value $v_{i,e}$.

Lemma B.1 *Let ξ be a cross-monotonic, (β, γ)-budget-balanced, α-summable cost-sharing method. Then, g_ξ is an implementable $O(\alpha + \beta(\gamma - 1) + \gamma)$-approximation algorithm for the LinME problem, and the prices implementing it yield γ-cost-recovery.*

Proof : The approximation guarantee follows from Corollary 6.2 in [53]. To prove implementability, fix player i and v_{-i}. Consider the runs of M_ξ on the inputs $v = (v_i, v_{-i})$ and $v' = (v'_i, v_{-i})$ where $v'_i > v_i$. Let $S^\ell(v)$ and $S^\ell(v')$ denote the current set of elements in iteration ℓ of M_ξ in the runs on v and v' respectively. We need to show that $|g_\xi(v') \cap E_i| \geq |g_\xi(v) \cap E_i|$. This follows from the observation that $S^\ell(v') \supseteq S^\ell(v)$, which follows easily by induction since ξ is cross-monotonic. In fact, we have the stronger property that $g_\xi(v) \subseteq g_\xi(v')$.

Let $q_i(r, v_{-i})$ be the smallest value t such that $|g_\xi(t, v_{-i}) \cap E_i| \geq r$; this is ∞ if there is no such t and 0 for $r = 0$. Let $T_{i,r} = g_\xi(q_i(r, v_{-i}), v_{-i})$. So by definition $|T_{i,r} \cap E_i| \geq r$. Also, $T_{i,r} \subseteq T_{i,r'}$ for all $r' \geq r$ since $q_i(r, v_{-i}) \leq q_i(r', v_{-i})$. The price charged to player i on input v is $\sum_{r=1}^{|g_\xi(v) \cap E_i|} q_i(r, v_{-i})$. So since $\xi(S, S) \geq C(S)/\gamma$ for all $S \subseteq [n]$, it suffices to show that $q_i(r, v_{-i}) \geq \max_{e \in T_{i,r}} \xi(T_{i,r}, e)$. Suppose this was false and there is some $e \in T_{i,r}$ such that $q_i(r, v_{-i}) < \xi(T_{i,r}, e)$. Then, under the input $(q_i(r, v_{-i}), v_{-i})$, the mechanism would not return $T_{i,r}$ as the output, which is a contradiction. ■

The following examples show that slight extensions of the above lemma—enlarging the domain of problems, or weakening the conditions on g_ξ—make it false.

Example B.2 *g_ξ is not implementable for the ME problem.* We have one player who owns two elements a, b. The cost function is $C(\{b\}) = C(\{a, b\}) = 2$; $C(\{a\}) = 1$. The cost-sharing method is: $\xi(\{a, b\}, e) = 1$ for $e = a, b$; $\xi(\{b\}, b) = 2$; $\xi(\{a\}, a) = 1$. Note that C is submodular and ξ is $(1, 1)$-budget-balanced. Consider the inputs $v = (0.9, 1.9)$ and $v' = (1, 1)$. We have $g_\xi(v) = \emptyset$ and $g_\xi(v') = \{a, b\}$ but this violates weak-monotonicity since $v'(\{a, b\}) < v(\{a, b\})$.

Acyclic mechanisms were defined in [44] in order to leverage cost-sharing methods that are obtained from primal-dual algorithms but need not be cross-monotonic. We show that the allocation rule of an acyclic mechanism need not be implementable for the LinME problem.

Example B.3 We have one player who owns three elements $E = \{a, b, c\}$. The offer function is such that elements are always considered in the order a, b, c. Let C be an additive cost function with $C(\{a\}) = 1$, $C(\{b\}) = C(\{c\}) = 1 + 2\epsilon$. The cost-sharing method is: $\xi(S, a) = 1$ for all $S \ni a$; $\xi(E, b) = \xi(E, c) = 1 + 2\epsilon$; $\xi(\{b, c\}, b) = \xi(\{b, c\}, c) = 1 - \epsilon$; and $\xi(\{a, c\}, c) = 1 + 2\epsilon = \xi(\{c\}, c)$. Note that these ξ values are compatible with ξ being valid for τ (see [44]), and this condition fixes all the remaining ξ values. Note that ξ is $(1 - O(\epsilon), 1)$-budget-balanced. On input $v = 1 - \epsilon$, the acyclic mechanism drops a in the first iteration, and returns $\{b, c\}$. On the input $v' = 1 + \epsilon$, the acyclic mechanism retains a, drops b, c in the subsequent iterations, and returns $\{a\}$. This contradicts monotonicity.

C. Omitted details from Section 7.2

The second LP-relaxation for the SCM problem, which generalizes (SC2-P), is as follows.

$$\min \qquad c^T x + \sum_i \sum_{k=1}^{R_i} v_{i,R_i-k+1} z_{i,k} \tag{MD2-P}$$

$$\text{s.t.} \qquad \sum_e A^{(i)}_{re} x_e + \sum_{k=1}^{b^{(i)}_r} z_{i,k} \geq b^{(i)}_r \qquad \forall i, r$$

$$Bx \geq d$$

$$0 \leq x \leq u \in \mathbb{Z}^m_+, \ \ 0 \leq z_{i,k} \leq 1 \qquad \forall i, k.$$

We require that $d \geq 0$, $A^{(i)}, b^{(i)} \geq 0$, and $b^{(i)}_r \in [R_i]$ for all i, r. We also require that $B \geq 0$, and for every i, if $A^{(i)}_{re} > 0$ and $u_e > 0$ then $A^{(i)}_{re} u_e \geq b^{(i)}_r$. The corresponding LP-relaxation for the CM problem where we want to find the min-cost way of serving each player i at level ℓ_i is obtained by dropping all the $z_{i,k}$ variables, and replacing $b^{(i)}_r$ by $\max\{0, b^{(i)}_r - (R_i - \ell_i)\}$.

We briefly describe the modifications to the proof of Theorem 7.4 if the SCM LP is described by (MD2-P). The mechanism M for the LinMD and MD problems is constructed exactly as before. For the MD problem, the modified SCM-LP (MMD-P) is now of the same form as (SC2-P), and the proof is unchanged with the relevant arguments from Section 4 now supplied by Theorem 4.6.

For the LinMD problem, no-bossiness follows as in Theorem 7.1. The proof that M is ICT and prices are polytime computable follows again by mimicking suitable arguments from Section 4. To prove the approximation, we again prove that $\sum_{i:z^*(R_i)>0} v_i R_i \leq k \cdot OPT$ by showing that $\sum_{i,k:z^*_{i,k}>0} v_i \leq OPT$. Let $(\{\mu^*_{i,r}\}, \omega^*, \theta^*, \{\pi^*_{i,k}\})$ be an optimal dual solution, where θ^*_e is value of the new dual variable corresponding to the $x_e \leq u_e$ constraint. Now, as in the proof of Theorem 4.6, we get that

$\sum_{i,k:z^*_{i,k}>0} v_i + \sum_{e:\theta^*_e>0} u_e c_e = \sum_{i,r} \mu^*_{i,r}\kappa_{i,r} + \sum_{r,e:\theta^*_e>0} x^*_e B_{re}\omega^*_r - u^T\theta^* - \sum_{i,k} \pi^*_{i,k}$ where $\kappa_{i,r} = |\{k \leq b^{(i)}_r : z^*_{i,k} > 0\}| + \sum_{e:\theta^*_e>0} u_e A^{(i)}_{re}$. If there is no $k \leq b^{(i)}_r$ with $z^*_{i,k} > 0$ then $\mu^*_{i,r}\kappa_{i,r} \leq \mu^*_{i,r}(\sum_e x^*_e A^{(i)}_{re}) = \mu^*_{i,r} b^{(i)}_r$. Otherwise, if $u_e \geq 1$, $\mu^*_{i,r} > 0$ and $A^{(i)}_{re} > 0$, we must have $\theta^*_e = 0$ (or else $\sum_e A^{(i)}_{re} x^*_e + \sum_{k \leq b^{(i)}_r} z^*_{i,k} > b^{(i)}_r$). It follows that $\mu^*_{i,r}\kappa_{i,r} \leq \mu^*_{i,r} b^{(i)}_r$, and hence, $\sum_{i,k:z^*_{i,k}>0} v_i + \sum_{e:\theta^*_e>0} u_e c_e \leq \sum_{i,r} \mu^*_{i,r} b^{(i)}_r + d^T\omega^* - u^T\theta^* - \sum_{i,k} \pi^*_{i,k} = OPT$.