Truthful and Near-Optimal Mechanism Design via Linear Programming*

Ron Lavi†
ronlavi@ie.technion.ac.il

Chaitanya Swamy‡
cswamy@math.uwaterloo.ca

Abstract

We give a general technique to obtain approximation mechanisms that are truthful in expectation. We show that for packing domains, any α-approximation algorithm that also bounds the integrality gap of the LP relaxation of the problem by α can be used to construct an α-approximation mechanism that is truthful in expectation. This immediately yields a variety of new and significantly improved results for various problem domains and furthermore, yields truthful (in expectation) mechanisms with guarantees that match the best known approximation guarantees when truthfulness is not required. In particular, we obtain the first truthful mechanisms with approximation guarantees for a variety of multi-parameter domains. We obtain truthful (in expectation) mechanisms achieving approximation guarantees of $O(\sqrt{m})$ for combinatorial auctions (CAs), $(1+\epsilon)$ for multi-unit CAs with $B = \Omega(\log m)$ copies of each item, and 2 for multi-parameter knapsack problems (multi-unit auctions).

Our construction is based on considering an LP relaxation of the problem and using the classic VCG [33, 9, 18] mechanism to obtain a truthful mechanism in this fractional domain. We argue that the (fractional) optimal solution scaled down by α, where α is the integrality gap of the problem, can be represented as a convex combination of integer solutions, and by viewing this convex combination as specifying a probability distribution over integer solutions, we get a randomized, truthful in expectation mechanism. Our construction can be seen as a way of exploiting VCG in a computational tractable way even when the underlying social-welfare maximization problem is *NP*-hard.

1 Introduction

Mechanism design studies algorithmic constructions under the presence of "selfish players" who hold the inputs to the algorithm. The players are selfish in that they are interested in maximizing their own utility, and instead of revealing the true input, they may declare any false input that will increase their utilities. The goal is to design algorithms that work well with respect to the *true input*, although this information is not publicly known. Mechanism design approaches this goal by specifying along with the algorithm a pricing scheme that can be used to incentivize the players to reveal their true inputs. A *mechanism* consists of an algorithm along with a pricing scheme, whose output specifies both an algorithmic outcome and prices to be charged to the players; the utility that a player derives is the difference between his value for the algorithmic outcome and the price that he is charged. A mechanism is said to be *truthful* if a player always maximizes his utility by declaring his true input, regardless of what the other players declare. Algorithmic mechanism design [29] deals with the study of efficiently computable truthful mechanisms. The hope is that by obtaining

*A preliminary version [24] appeared in the Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science, 2005.

†Faculty of Industrial Engineering and Management, The Technion – Israel Institute of Technology. Work done while the author was a postdoctoral scholar at Caltech, Pasadena, CA 91125.

‡Dept. of Combinatorics & Optimization, University of Waterloo, Waterloo, ON N2L 3G1. Research supported partially by NSERC grant 327620-09 and an Ontario Early Researcher Award. Work done while the author was a postdoctoral scholar at Caltech, Pasadena, CA 91125.

a truthful mechanism, one can focus on the algorithmic question and set aside the strategic issue, however unfortunately not all algorithms lead to truthful mechanisms. This is best exemplified when the underlying algorithmic problem is *NP*-hard. Over the past several years, approximation algorithms have emerged as an effective algorithmic tool to deal with such computational intractability. However these algorithms have found only limited applicability in the design of truthful mechanisms, especially in "multi-parameter domains", since many approximation algorithms and techniques do not guarantee certain properties of the algorithm that are required to ensure truthfulness.

Our results. In this paper we give a general technique to convert approximation algorithms into truthful mechanisms. One of the most widely used and remarkably successful techniques in the design of approximation algorithms, is that of expressing a relaxation of the problem as a linear program (LP), and using this to design the approximation algorithm, either via LP rounding or via a primal-dual approach. We show that for a variety of domains, any approximation algorithm for the algorithmic problem, that also bounds the integrality gap of the underlying LP relaxation, can be used to obtain a randomized mechanism for the corresponding mechanism design problem (i.e., where the inputs are held by selfish agents), that is truthful in expectation, and *has performance guarantee matching that of the approximation algorithm*. Truthfulness in expectation means that a player always maximizes his *expected* utility by revealing his true input [1]. Our main result can be informally stated as follows.

Informal Main Theorem *For packing domains, given any α-approximation algorithm that proves an integrality gap of at most α for the "natural" LP relaxation, one can construct a randomized α-approximation mechanism that is truthful in expectation.*

A key feature of this result is its generality. As mentioned earlier, there are a number of problem domains for which good approximation algorithms are known for the algorithmic problem, but in the mechanism design setting there are no known truthful *and* approximation mechanisms for the corresponding problem. A common problem encountered is the fact that many commonly used approximation algorithms and techniques violate certain monotonicity properties which are required for truthfulness. In cases where truthful and approximation mechanisms are known, the construction often involves clever ways of modifying or designing the approximation algorithm so as to satisfy these properties. As a consequence, in various settings, the approximation guarantees provided by truthful mechanisms fall short of the known guarantees for the algorithmic problem when one does not require truthfulness.

Our theorem not only yields new and improved results for large problem classes such as, (multi-unit) combinatorial auctions, graph routing problems, multi-unit auctions, but in doing so, it bridges the gap between known approximation results and the guarantees obtained by truthful mechanisms. Moreover, it shows that one does not need to specifically tailor the algorithm to ensure truthfulness, and allows one to focus on the algorithmic part of the problem.

We now consider the implications of our theorem, by concentrating on the *multi-unit combinatorial auction* (MUCA) problem, which has been widely studied [27, 2, 4, 23, 6, 14]. In an MUCA, m items are to be allocated among n players and each player i has a true value $\overline{v}_i(S)$ for every subset S of items. The value functions are assumed to be non-decreasing, that is, $\overline{v}_i(S) \leq \overline{v}_i(T)$ when $S \subseteq T$, and normalized, that is, $\overline{v}_i(\emptyset) = 0$. There are $B \geq 1$ copies of each item. An allocation is valid if every item is allocated to at most B players. The goal is to find an allocation $S_1, \ldots, S_n$ of the items to the players that maximizes the sum $\sum_i \overline{v}_i(S_i)$, called the *social welfare*. When $B = 1$, the problem is called the *combinatorial auction* (CA) problem.

The problem is computationally hard even when $B = 1$, and the valuations are "short", i.e., succinctly described. The approximability threshold of this problem is known to be $O(m^{\frac{1}{B+1}})$ [4]; it is hard to approximate any better even when the players are single minded, that is, each player wants a single set S, so

$\overline{v}_i(T) = \overline{v}_i(S)$ for $T \supseteq S$ and 0 otherwise. On the positive side various $O(m^{\frac{1}{B+1}})$-approximation algorithms exist [32, 21, 6], and these algorithms also prove a corresponding integrality gap result. An interesting case is when $B = \Omega(\log m)$ for which the integrality gap becomes $(1+\epsilon)$ where $\epsilon > 0$ (more precisely $B \geq c(\epsilon) \cdot \log m$ where $c(.)$ is an increasing function of $\frac{1}{\epsilon}$; we denote this simply as $B = \Omega(\log m)$ in the sequel). However *truthfulness* results are mostly known in restricted cases, and are worse. The only result for general valuations is a truthful $O(B \cdot m^{\frac{1}{B-2}})$-approximation mechanism due to Bartal et al. [4] when $B \geq 3$. More results are known for special cases: for single-minded players, a truthful $O(\sqrt{m})$-approximation is known when $B = 1$ [27], and Briest et al. [6] recently gave a truthful $O(m^{\frac{1}{B}})$-approximation for $B \geq 1$ (this does not give a $(1+\epsilon)$-guarantee when $B = \Omega(\log m)$). Archer et al. [2] gave a randomized $(1+\epsilon)$-approximation mechanism for the "known" single-minded case when $B = \Omega(\log m)$. Babaioff et al. [3] consider the "single-value" case and (among other results) give an $O(\log^2 \overline{v}_{\max} \cdot \sqrt{m})$-guarantee; Dobzinski et al. [14] obtain an $O(\sqrt{m})$-approximation for subadditive valuations. In particular, the picture is very partial for non-single-minded players: no truthful approximation mechanism is known for general combinatorial auctions ($B = 1$), and the factors for large B are much worse than the approximation results.[1]

Using our construction, and any of the known approximation algorithms, we obtain randomized, truthful in expectation mechanisms for all values of B, where *the guarantees match the approximability threshold*. We summarize our results below for "short" valuations, where the valuations are succinctly describable and the underlying packing LP can be solved efficiently. An important such class is k-minded players: a player desires one of k sets and specifies the value of each; both the sets and the values are private. Section 4.1 gives other examples. We use "truthful" below to denote "truthful in expectation".

Combinatorial auctions ($B = 1$) We give the *first truthful mechanism* for non-single-minded valuations, and get a $O(\sqrt{m})$ guarantee.

Multi-unit combinatorial auctions We get an $O(m^{\frac{1}{B+1}})$-approximation, truthful mechanism. This improves upon the ratio of [4]. When $B = \Omega(\log m)$ we get a $(1+\epsilon)$-guarantee for valuations more general than the "known" single-minded case, answering an open question in [2].

Multi unit auctions Here the m items are identical, and the valuation specifies a value for each quantity of goods received. This generalizes Knapsack, which is the case when all players have a step function (i.e., are single minded), for which a truthful FPTAS was given by Briest et al. [6]. We give a truthful 2-approximation mechanism for multi-unit auctions, which is the *first truthful, approximation mechanism* for the multi-parameter case.

Graph routing problems These include (see Section 4.3), for example, the edge-disjoint paths problem, where we have an underlying graph and each player desires a path between his source and target nodes. This is a special case of MUCA and the previous guarantees hold.

With arbitrary valuations, the input itself may have exponential length (in m) which therefore renders direct revelation mechanisms infeasible. The problem remains "hard" even when one ignores computational issues: Nisan [28] shows that no algorithm with polynomial communication (in m, n) can attain an approximation ratio of $m^{\frac{1}{2}-\epsilon}$ for any $\epsilon > 0$ even when $B = 1$. Blumrosen and Nisan [5] introduce the notion of a *demand oracle*, and show that this can be used to solve the associated LP using polynomial queries. With such an oracle, our construction still works and yields *iterative* randomized, strategic mechanisms. However, the iterative structure of the mechanism changes the solution concept to an ex-post Nash equilibrium, a well-documented phenomenon in the economics literature (see e.g., [10]). So we now get truthfulness

[1] Section 5 discusses briefly some results obtained subsequent to the publication of the conference version of this paper [24]. Notably, however, our method remains the only known general method that provides a unified framework; it also yields the current-best truthful approximation mechanisms for some broad problem domains (e.g., MUCAs).

in expectation as an ex-post Nash equilibrium which roughly means the following: regardless of the other players' valuations, if they all answer truthfully, then my best response is to answer truthfully. We get the same guarantees as before, using ex-post Nash equilibrium as the solution concept.

Our construction. At a high level, our construction is quite intuitive and easy to describe. Our starting point is the classic VCG result [33, 9, 18], which provides a truthful mechanism for the underlying algorithmic problem of maximizing the social welfare, provided that the algorithmic problem can be solved *exactly*. However, in our case and in many others, this problem is *NP*-hard, and approximation algorithms need not give truthfulness. Our first step towards handling this problem is to move to a fractional domain, and to consider a *fractional mechanism* that is allowed to return a fractional feasible solution to the LP relaxation of the problem. In this new domain, since one can solve the LP in polynomial time, we can use VCG to obtain a truthful fractional mechanism. Moreover, since we are in a fractional domain, we can always scale down both the optimal LP solution and the VCG prices by α, and this clearly does not affect truthfulness.

Let x^* be an optimal LP solution. Suppose that the integrality gap of the LP relaxation is $\alpha \geq 1$ and we have an algorithm that "verifies" this gap. At the heart of our construction is a procedure that takes such an α-approximation algorithm and returns in polynomial time a *convex decomposition* of $\frac{x^*}{\alpha}$ into polynomially many integer solutions, that is, it returns λ_l values such that $\frac{x^*}{\alpha} = \sum_{l \in \mathcal{I}} \lambda_l x^l$, where $\{x^l\}_{l \in \mathcal{I}}$ is the set of all integer solutions, $\lambda_l \geq 0$, $\sum \lambda_l = 1$ (with only polynomially many positive λ_l). We show this by strengthening the decomposition technique of Carr and Vempala [7]. Now one can view this convex combination as specifying a probability distribution over the integer solutions, where exactly one solution x^l is selected with probability equal to λ_l. A simple but powerful observation, is that the randomized mechanism which returns an integer solution according to this distribution, and sets prices in such a way that the expected prices are the VCG prices scaled down by α, is *truthful in expectation*. Furthermore, the expected social welfare of the solution returned is exactly the value of the LP-optimum scaled by α; thus we get an α-approximation guarantee. The crucial step here is the decomposition procedure which allows us to move from the truthful VCG fractional mechanism to a randomized, truthful in expectation mechanism.

A notable feature is the generality and simplicity of the construction. In essence, we give a way to leverage VCG in a computationally effective manner using approximation algorithms, while maintaining truthfulness. Thus we reap the benefits of both: we get the versatility of the VCG as a mechanism design tool for devising truthful mechanisms, *and* the computational tractability of approximation algorithms. As mentioned earlier, many previous results specifically tailor the algorithm to obtain truthfulness. For example, Archer et al. [2] consider an LP-relaxation of MUCA with $\Omega(\log m)$ copies and designed a randomized rounding procedure to obtain a $(1+\epsilon)$-approximation mechanism that is truthful in expectation and also with high probability. Briest et al. [6] showed how to convert certain FPTAS's into truthful ones, and also gave primal-dual algorithms for some CAs; both results involve tailoring the algorithm so as to ensure truthfulness, but yield deterministic mechanisms. Our construction shows that if one allows randomization, then such "algorithmic artistry" is not required to ensure truthfulness, and allows the algorithm design to concentrate on the approximation component of the problem.

Organization of the paper. We begin in Section 2 by describing the mechanism-design setup, and collecting some basic definitions and facts. Section 3 describes our general method of designing a truthful mechanism based on the LP relaxation of the problem, and Section 4 applies our technique to a variety of problems. We conclude in Section 5 with a brief discussion of some recent results, and a discussion of how our ideas can be adapted to yield (partial) results also for covering problems.

2 Preliminaries

In the basic mechanism design setup, we have n players, and a set A of possible outcomes. Each player has a type given by a valuation function $v_i : A \mapsto \mathbb{R}_{\geq 0}$, where $v_i \in V_i$ and V_i is the set of all valid types of player i. Let $V = V_1 \times \cdots \times V_n$ denote the space of all players' valid types. For example, in *combinatorial auctions* we have m items; A is the set of all allocations $(S_1, \ldots, S_n)$ of items to players with $S_i \cap S_j = \emptyset$ (assuming a single copy of each item), and $v_i : A \mapsto \mathbb{R}$ is a function that (a) assigns the same value to any two outcomes that allocate the same subset to player i, so one can view v_i as specifying a value for each set S of items, (b) satisfies $v_i(\emptyset) = 0$, and (c) is monotone, i.e., $v_i(S) \leq v_i(T)$ whenever $S \subseteq T$. We use v to denote the tuple $(v_1, \ldots, v_n)$, and v_{-i} to denote the tuple $(v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n)$ that excludes i's valuation. Similarly $V_{-i} = \prod_{j \neq i} V_j$.

A (direct revelation) mechanism consists of an allocation rule (that is, an algorithm) $f : V \mapsto A$, and a pricing scheme $p_i : V \mapsto \mathbb{R}$ 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. Throughout we use $\overline{v}_i$ to denote the *true type* of player i. The *utility* that the player i derives by declaring type v_i is $\overline{v}_i(f(v_i, v_{-i})) - p_i(v_i, v_{-i})$, and each player aims to maximize his own utility. A desirable property for a mechanism to satisfy is *truthfulness*, wherein *each player maximizes his utility by reporting his true type*.

Definition 2.1 (Truthfulness) *A deterministic mechanism (f, p) is truthful if for any player i, any $v_{-i} \in V_{-i}$, and any $\overline{v}_i, v'_i \in V_i$ we have $\overline{v}_i(f(\overline{v}_i, v_{-i})) - p_i(\overline{v}_i, v_{-i}) \geq \overline{v}_i(f(v'_i, v_{-i})) - p_i(v'_i, v_{-i})$.*

A mechanism could be *randomized*, that is, it could flip coins to determine $f(v)$ and $p_i(v)$, in which case $f(v)$ and $p_i(v)$ are random variables, and a player's utility is also a random variable. For randomized mechanisms, we can consider the notion of *truthfulness in expectation* [1].

Definition 2.2 (Truthfulness in expectation) *A randomized mechanism (f, p) is truthful in expectation if for any player i, any $v_{-i} \in V_{-i}$, and any $\overline{v}_i, v'_i \in V_i$, $\mathrm{E}\big[\overline{v}_i(f(\overline{v}_i, v_{-i})) - p_i(\overline{v}_i, v_{-i})\big] \geq \mathrm{E}\big[\overline{v}_i(f(v'_i, v_{-i})) - p_i(v'_i, v_{-i})\big]$.*

Thus, if a mechanism is truthful in expectation then the expected utility of a player is maximized when he declares his true type $\overline{v}_i$ (regardless of what others declare). A randomized algorithm is often viewed as specifying a probability distribution over deterministic algorithms. It will be useful to view a randomized mechanism in a similar way. For an outcome set A, define the unit simplex $\Delta_A = \{\lambda \in \mathbb{R}^{|A|} : \sum_{a \in A} \lambda_a = 1 \text{ and } \lambda_a \geq 0 \text{ for all } a\}$. Given any randomized mechanism $M^R = (f^R, p^R)$ with outcome set A, we can define a deterministic mechanism M^D with outcome set Δ_A, which outputs the probability distribution of M^R, and charges prices that are the expected prices of M^R. To define M^D precisely, we need to extend the domain V_i to include values for outcomes in Δ_A. We do this in the obvious way: for any $v_i \in V_i$, define $v_i(\{\lambda_a\}_{a \in A}) = \sum_{a \in A} \lambda_a v_i(a)$. We abuse notation and use v_i to denote both a valuation in V_i and its corresponding extension. Now we can define M^D as follows:

1. Allocation rule: Define the function f^D by $f^D(v) = \{\lambda_a\}_{a \in A}$ where $\lambda_a = \Pr[f^R(v) = a]$,
2. Pricing scheme: For each player i, and valuation vector v, set $p_i^D(v) = \mathrm{E}\big[p_i^R(v)\big]$.

We call M^D the deterministic *support mechanism of* M^R. Observe that the criterion for truthfulness of M^D is *exactly* the criterion for the truthfulness in expectation of M^R.

Claim 2.3 *A randomized mechanism M^R is truthful in expectation iff its support mechanism M^D is truthful.*

The above claim gives a characterization of randomized mechanisms that are truthful in expectation which we use to argue the truthfulness in expectation of the randomized mechanisms we construct.

Converting a support mechanism to a randomized mechanism. Our algorithms will actually construct a deterministic support mechanism. We show how to construct, from a (deterministic) support mechanism M^D, a corresponding randomized mechanism M^R (whose support mechanism is M^D) that preserves various properties of M^D. Let M^D have outcome set Δ_A, so M^R will have outcome set A. The allocation rule is easy to specify: $f^R(v) \in A$ is a random variable with distribution $f^D(v)$, i.e., we output $f^R(v) = a$ with probability λ_a where $f_D(v) = \{\lambda_a\}_{a \in A}$. So $\mathrm{E}\left[\overline{v}_i(f^R(v))\right] = \overline{v}_i(f^D(v))$ for any player i, any true type $\overline{v}_i \in V_i$ and any $v \in V$. Since the prices $p_i^R(v)$ will be set so that $\mathrm{E}\left[p_i^R(v)\right] = p_i^D(v)$ (because M^D has to be the support mechanism of M^R), the expected utility of player i is also preserved in going from M^D to M^R. One obvious way to set the prices is to deterministically set $p_i^R(v) = p_i^D(v)$ for every v, but the resulting randomized mechanism could violate *individual rationality*. Individual rationality requires that for every player i, i's utility be non-negative when he declares his *true type*, regardless of the other players' declarations. A randomized mechanism, satisfies individual rationality if for every coin toss, the utility of every player i is non-negative when he declares his true type. Setting $p_i^R(v) = p_i^D(v)$ may violate individual rationality even if M^D is individually rational, since there could be coin tosses where a player receives nothing but pays a positive amount. Perhaps the easiest way to maintain individual rationality is to set price $p_i^R(v) = v_i(a) \cdot \frac{p_i^D(v)}{v_i(f^D(v))}$ if $v_i(f^D(v)) > 0$ and outcome a results (i.e., $f^R(v) = a$), and set $p_i^R(v) = 0$ if $v_i(f^D(v)) = 0$ $(= p_i^D(v))$. Observe that $p_i^R(\overline{v}_i, v_{-i}) \leq \overline{v}_i(f^R(\overline{v}_i, v_{-i}))$ for any coin toss since $p_i^D(\overline{v}_i, v_{-i}) \leq \overline{v}_i(f^D(\overline{v}_i, v_{-i}))$, and clearly $\mathrm{E}\left[p_i^R(v)\right] = p_i^D(v)$.

We will be interested in approximating the (optimum) *social welfare* with respect to the players' true types, which is defined as $\max_{a \in A} \sum_i \overline{v}_i(a)$. The above construction of M^R from M^D preserves truthfulness (in expectation) by Claim 2.3, individual rationality, and the expected value received by a player. Thus we get the following theorem.

Theorem 2.4 *Given an individually rational, truthful, deterministic support mechanism that computes an α-approximation to the social welfare and has only polynomially many λ_as that are positive, one can obtain in polynomial time an individually rational randomized mechanism that is truthful in expectation, and computes an α-approximation to the social welfare in expectation.*

Given the above theorem, we now focus on the design of a deterministic support mechanism that outputs a probability distribution and prices, with the desired properties.

3 A general technique for constructing truthful, approximation mechanisms

In this section we describe a general technique to obtain randomized mechanisms that are truthful in expectation, and achieve approximation guarantees for the social welfare that, in several cases, match the guarantees that are achieved without worrying about truthfulness. To make our results concrete, we describe our technique for the specific setting of combinatorial auctions (CAs), although our results also hold for other packing problems when the polytope is public knowledge and the objective function is linear. In Section 4, we consider a variety of applications and use our methods to obtain truthful (in expectation) mechanisms that approximate the social welfare.

We can formulate the combinatorial auction problem as an integer program (IP) where we have a variable $x_{i,S} \in \{0, 1\}$ for each player i and set $S \neq \emptyset$ that indicates if i receives set S. Relaxing the integrality

constraints to $x_{i,S} \geq 0$ gives the following LP relaxation.

$$\begin{aligned} \max \quad & \sum_{i,S\neq\emptyset} v_i(S)x_{i,S} & & \text{(CA-P)} \\ \text{subject to} \quad & \sum_{S\neq\emptyset} x_{i,S} \leq 1 & \text{for each player } i & \quad (1) \\ & \sum_i \sum_{S:j\in S} x_{i,S} \leq 1 & \text{for each item } j & \quad (2) \\ & x_{i,S} \geq 0 & \text{for each } i, S. & \end{aligned}$$

Here v_i is i's reported valuation which satisfies $v_i(S) \leq v_i(T)$ whenever $S \subseteq T$ (and implicitly $v_i(\emptyset) = 0$). Constraints (1) state that each player is assigned at most one set, and (2) ensures that each item j is given to at most one player. Our approximation guarantee will depend on the integrality gap of (CA-P), that is, the ratio between the values of the optimal fractional and integer solutions. Our main theorem is the following.

Theorem 3.1 *Given an α-approximation algorithm, that also proves an integrality gap of at most α for* (CA-P)*, one can construct a randomized, individually rational, α-approximation mechanism that is truthful in expectation.*

We first give an overview of the construction. A classic result in mechanism design is the VCG family of mechanisms [33, 9, 18], which shows that if the underlying algorithmic problem of maximizing the social welfare, given the players' reported types, called the winner determination problem (WDP) for CAs, can be solved *exactly*, then one can construct a truthful mechanism that optimizes the social welfare (with respect to the true types). However this algorithmic problem is often *NP*-hard, and it is known that an approximation algorithm for this problem need not necessarily give a truthful mechanism.

We move to a fractional domain and consider a *fractional mechanism* that is allowed to return a fractional feasible solution to (CA-P). With this outcome set, one can solve WDP exactly in polynomial time since one can solve (CA-P) (see Section 4.1), and therefore use VCG to obtain a truthful fractional mechanism M^F which returns allocation x^*, the optimal solution to (CA-P), and prices p^F. Since we are in a fractional domain, we can scale down both the allocation and prices by some $\alpha \geq 1$ without affecting truthfulness. We give a procedure which takes an α-approximation algorithm that proves an integrality gap of α for (CA-P), and returns in polynomial time a convex decomposition of $\frac{x^*}{\alpha}$ into polynomially many integer solutions, i.e., λ_l values such that $\frac{x^*}{\alpha} = \sum_{l\in\mathcal{I}} \lambda_l x^l$, where $\{x^l\}_{l\in\mathcal{I}}$ is the set of all integer solutions to (CA-P), $\lambda_l \geq 0$, $\sum \lambda_l = 1$ (with only polynomially many $\lambda_l > 0$). This is obtained by strengthening the decomposition technique of Carr and Vempala [7]. This is the crucial step, using which we convert the truthful fractional mechanism M^F to a *truthful deterministic support mechanism* M^D that outputs $\{\lambda_l\}_{l\in\mathcal{I}}$ as the allocation and $\frac{p^F}{\alpha}$ as the prices, while losing a factor of α in the social welfare (and maintaining individual rationality). Now we use Theorem 2.4 to obtain a randomized, truthful in expectation mechanism that achieves an α-approximation to the social welfare. We summarize the construction below.

The truthful α-approximation support mechanism. Given valuations $v_1, \ldots, v_n$; an α-approximation algorithm $\mathcal{A}$ for WDP that shows an integrality gap of α for (CA-P).

1. Use VCG to get a truthful fractional mechanism M^F that outputs allocation $f^F(v) = x^*(v)$, the optimal solution to (CA-P), and prices $p^F(v)$.
2. Use $\mathcal{A}$ to obtain the convex decomposition $\frac{x^*}{\alpha} = \sum_{l\in\mathcal{I}} \lambda_l x^l$ with only polynomially many positive λ_l.
3. Return the support mechanism $M^D = (f^D, p^D)$ with $f^D(v) = \{\lambda_l\}_{l\in\mathcal{I}}$ and $p^D(v) = \frac{p^F(v)}{\alpha}$ (and use Theorem 2.4 to get the desired randomized mechanism).

Thus we show that any approximation algorithm for maximizing social welfare, that also shows an integrality gap guarantee, can be plugged in to get a randomized truthful mechanism with the same approximation guarantee. Thus, we are able to combine the versatility of VCG as a mechanism design tool for devising truthful mechanisms, *and* the computational tractability of approximation algorithms.

Details of the construction We now describe the construction in detail. Let $\mathcal{P}$ denote the feasible region of (CA-P), and $\mathbb{Z}(\mathcal{P}) \subseteq \mathcal{P}$ be the set of integer solutions of (CA-P). The integrality gap of $\mathcal{P}$ is defined as

$$IG_{\mathcal{P}} = \sup_{v=(v_1,\ldots,v_n)} \frac{\max_{x\in\mathcal{P}} \sum_{i,S} v_i(S)x_{i,S}}{\max_{x\in\mathbb{Z}(\mathcal{P})} \sum_{i,S} v_i(S)x_{i,S}}$$

where the v_is are valuations with $v_i(\emptyset) = 0, v_i(S) \leq v_i(T)$ for $S \subseteq T$. Our mechanisms require an approximation algorithm that "verifies" an integrality gap of (at most) α, by which we mean that for *any valuation vector* v the algorithm produces an integer solution of value at least $\frac{1}{\alpha}$ times the LP-optimum. We emphasize that the supremum is taken over *all* valuations v in evaluating the integrality gap, and an algorithm that verifies this gap must consequently work for all valuations. To clarify this further, whereas we may sometimes consider (designing mechanisms for) a structured class of valuations V, the integrality gap $IG_{\mathcal{P}}$ is determined only by the polytope $\mathcal{P}$, and to exploit the structure of V we need to be able to encode this structure into the polytope $\mathcal{P}$. In particular, an approximation algorithm that specifically proves a guarantee *for class* V using the optimum of (CA-P) as an upper bound (e.g., the 2-approximation algorithm for submodular valuations in [26]) *does not* suffice to bound or verify the integrality gap $IG_{\mathcal{P}}$.

The fractional mechanism defined will have outcome set $\mathcal{P}$. We extend the domain V_i to assign values to fractional solutions $x \in \mathcal{P}$: for any $v_i \in V_i$, we define $v_i(x) = \sum_S v_i(S)x_{i,S}$. We again abuse notation and use v_i to denote both the original valuation and its extension. We now define precisely the *fractional VCG mechanism* which is simply VCG in this fractional domain.

The fractional VCG mechanism *$M^F = (f^F, p^F)$ is defined as follows: the allocation rule is given by $f^F(v) = x^*(v)$ where $x^*(v)$ is an optimal solution to* (CA-P) *for valuation $v = (v_1, \ldots, v_n)$; the prices are $p_i^F(v) = \big(-\sum_{i'\neq i} v_{i'}(S)x^*_{i',S} + h_i(v_{-i})\big)$, where $h_i(v_{-i})$ is any function that does not depend on v_i.*

As usual with VCG mechanisms, one can set $h_i(v_{-i}) = \sum_{i'\neq i} v_{i'}(S)y^*_{i',S}$ above, where y^* is the optimal fractional solution when we constrain $x_{i,S} = 0$ for all S, so as to ensure that that players have non-negative utility (individual rationality), and players always have a non-negative payment (they never receive money). We will assume these prices from now on. It is a classic result that the VCG mechanism (family) is truthful, i.e., satisfies the criterion in Definition 2.1. For any $\alpha \geq 1$, we can define an *α-scaled fractional VCG mechanism* that outputs the outcome $\frac{f^F(v)}{\alpha}$ and prices $\frac{p^F(v)}{\alpha}$. For any $v \in V$ and $\overline{v}_i \in V_i$, since $\overline{v}_i\big(\frac{f^F(v)}{\alpha}\big) = \frac{\overline{v}_i(f^F(v))}{\alpha}$ (because $\overline{v}_i(x)$ is linear in x), the α-scaled fractional VCG mechanism is also clearly truthful.

Suppose that we can express $\frac{x^*(v)}{\alpha}$ as a convex combination $\sum_{l\in\mathcal{I}} \lambda_l(v)x^l$ of integral solutions where $\lambda_l(v) \geq 0, \sum_l \lambda_l(v) = 1$. (Observe that α must be at least $IG_{\mathcal{P}}$ since such a decomposition of $\frac{x^*(v)}{\alpha}$ implies that there exists an integer solution x^l of value at least $\frac{1}{\alpha} \cdot \sum_i v_i(x^*(v))$.) One then obtains a deterministic support mechanism M^D with allocation rule $f^D(v) = \{\lambda_l(v)\}_{l\in\mathcal{I}}$ and prices $p_i^D(v) = \frac{p_i^F(v)}{\alpha}$. Before detailing the decomposition procedure, we show that M^D is a truthful, α-approximation mechanism.

Lemma 3.2 *Mechanism M^D is truthful and computes an α-approximation to the social welfare.*

Proof : Essentially, we show that M^D is equivalent to the α-scaled fractional VCG mechanism and retains all its properties. For any declared valuation $v = (v_1, \ldots, v_n)$, the value player i gets in M^D is *exactly* $\frac{\overline{v}_i(f^F(v))}{\alpha}$ since $\overline{v}_i(f^D(v)) = \sum_{l\in\mathcal{I}} \lambda_l(v)\overline{v}_i(x^l)$ by definition, which is equal to $\overline{v}_i(\sum_{l\in\mathcal{I}} \lambda_l(v)x^l) =$

$\overline{v}_i\big(\frac{x^*(v)}{\alpha}\big) = \frac{\overline{v}_i(x^*(v))}{\alpha}$. Since the prices are also scaled by α, truthfulness follows from the truthfulness of M^F. Truthfulness also then implies the approximation guarantee since $\sum_i \overline{v}_i(x^*(\overline{v}))$ is an upper bound on the optimum. Note that M^D satisfies individual rationality and no player is paid by the mechanism. ■

Using Theorem 2.4 we can move from M^D to a randomized mechanism, while maintaining truthfulness, the approximation ratio, and individual rationality. Thus we obtain Theorem 3.1. The crucial property that we require here is that for any $\overline{v}_i \in V_i$, and any $v \in V$, $\sum_{l\in\mathcal{I}} \lambda_l(v)\overline{v}_i(x^l) = \frac{\overline{v}_i(x^*(v))}{\alpha}$. A *sufficient* condition for this is that $\overline{v}_i(x)$ be a linear function of x and that $\frac{x^*(v)}{\alpha}$ be decomposable as $\sum_{l\in\mathcal{I}} \lambda_l(v)x^l$, which is what we use in our construction.

3.1 Decomposing the fractional solution

We now prove the main decomposition lemma: we show that an α-approximation algorithm $\mathcal{A}$ that proves an integrality gap of α for (CA-P) can be used to express any fractional solution to (CA-P), scaled down by α, as a convex combination of integer solutions. The proof is based on the method outlined in [7] where it is shown that for minimization problems, if the integrality gap is $\beta \geq 1$, then for any fractional solution, one can obtain a convex combination of integer solutions such that β times the fractional solution *dominates* (i.e., is component-wise greater than) the convex combination. We need to modify their argument so as to ensure that we get an *exact* decomposition (for our maximization problem), which is crucial for our truthfulness argument to hold. For general maximization LPs, one cannot necessarily get such an exact decomposition of the scaled fractional solution, but only one that dominates the scaled solution. We leverage the packing structure of the problem to get an exact decomposition. Recall that $\mathcal{P}$ denotes the feasible region of (CA-P) and $\mathbb{Z}(\mathcal{P}) = \{x^l\}_{l\in\mathcal{I}}$ is the set of all integer solutions to (CA-P) where $\mathcal{I}$ is an index set for the integer solutions. We exploit the following property, which we call the *"packing property"*: if $x \in \mathbb{Z}(\mathcal{P})$ and $y \leq x$ is integral then $y \in \mathbb{Z}(\mathcal{P})$.

Fix any $x^* \in \mathcal{P}$ such that $E = \{(i,S) : x^*_{i,S} > 0\}$ has size polynomial in m, n. Any basic solution to (CA-P) satisfies this since by standard polyhedral theory such a solution has at most $m + n$ non-zero entries. We solve the linear program (P) given below to obtain the convex decomposition.

$$\begin{aligned} \min \quad & \sum_{l\in I} \lambda_l && \text{(P)} \\ \text{s.t.} \quad & \sum_l \lambda_l x^l_{i,S} = \tfrac{x^*_{i,S}}{\alpha} && \text{for all } (i,S) \in E \quad (3) \\ & \sum_l \lambda_l \geq 1 && (4) \\ & \lambda_l \geq 0 && \text{for all } l \in \mathcal{I}. \end{aligned}$$

$$\begin{aligned} \max \quad & \tfrac{1}{\alpha} \cdot \sum_{(i,S)\in E} x^*_{i,S} w_{i,S} + z && \text{(D)} \\ \text{s.t.} \quad & \sum_{(i,S)\in E} x^l_{i,S} w_{i,S} + z \leq 1 && \text{for all } l \in \mathcal{I} \quad (5) \\ & z \geq 0 \\ & w_{i,S} \text{ unconstrained} && \text{for all } (i,S) \in E. \end{aligned}$$

The primal (P) has an exponential number of variables, so we consider its dual (D). The dual has an exponential number of constraints and a *variable* $w_{i,S}$ for each constraint (3), and one can view w as a valuation. We show that a separation oracle for the dual can be obtained by using algorithm $\mathcal{A}$ with valuation w, so the ellipsoid method can be used to solve (D) and hence (P) (we also show that the optimal value is 1). One potential problem encountered is that the $w_{i,S}$ values could be negative, whereas $\mathcal{A}$ is only designed to handle non-negative valuations. However it is easy to argue that one can instead use $\mathcal{A}$ with the non-negative valuation w^+ given by $w^+_{i,S} = \max(w_{i,S}, 0)$, and this yields a separation oracle. (Alternatively, one could first compute λ_l values such that $\sum_{l\in\mathcal{I}} \lambda_l x^l \geq \frac{x^*}{\alpha}$, so the dual variables $w_{i,S}$ are now non-negative, and then use the packing property to modify these λ_l values and get an exact decomposition.)

Claim 3.3 *Let $w = \{w_{i,S}\}_{(i,S)\in E}$ be any weight vector. Define w^+ by $w^+_{i,S} = \max(w_{i,S}, 0)$. Given any integer solution $\hat{x} \in \mathbb{Z}(\mathcal{P})$, one can obtain $x^l \in \mathbb{Z}(\mathcal{P})$ such that $\sum_{(i,S)\in E} x^l_{i,S} w_{i,S} = \sum_{(i,S)\in E} \hat{x}_{i,S} w^+_{i,S}$.*

Proof : Set $x^l_{i,S} = \hat{x}_{i,S}$ if $w_{i,S} \geq 0$ and 0 otherwise. Clearly, $\sum_{(i,S)\in E} x^l_{i,S} w_{i,S} = \sum_{(i,S)\in E} \hat{x}_{i,S} w^+_{i,S}$. Since $x^l \leq \hat{x}$ is integral, by the packing property $x^l \in \mathbb{Z}(\mathcal{P})$. ■

Claim 3.4 *For any weight vector $w = \{w_{i,S}\}_{(i,S)\in E}$, one can compute in polynomial time $x^l \in \mathbb{Z}(\mathcal{P})$ such that $\sum_{(i,S)\in E} x^l_{i,S} w_{i,S} \geq \frac{1}{\alpha} \cdot \max_{x\in\mathcal{P}} \sum_{(i,S)\in E} x_{i,S} w_{i,S}$.*

Proof : Let $O^* = \max_{x\in\mathcal{P}} \sum_{(i,S)\in E} x_{i,S} w_{i,S}$. Let w^+ be as defined in Claim 3.3. Clearly w^+ is component-wise greater than w and $\sum_{i,S} x_{i,S} w^+_{i,S} = \sum_{(i,S)\in E} x_{i,S} w^+_{i,S}$ for any x. If $\mathcal{A}$ only expects a non-negative valuation as input, then we can simply run $\mathcal{A}$ on w^+ to get an integer solution $\hat{x}$ such that $\sum_{(i,S)\in E} \hat{x}_{i,S} w^+_{i,S} \geq \frac{O^*}{\alpha}$. But in our case $\mathcal{A}$ requires a valuation that satisfies monotonicity, so we cannot directly feed it w^+. However one can simply "monotonize" w^+: for each i define valuation $\tilde{v}_i$ by $\tilde{v}_i(S) = \max_{T\subseteq S:(i,T)\in E} w^+_{i,T}$ where the maximum is 0 if there is no $T \subseteq S$ such that $(i,T) \in E$. We use $\mathcal{A}$ on valuation $\tilde{v}$ to compute an α-approximate integer solution $\tilde{x}$, so $\sum_{i,S} \tilde{x}_{i,S} \tilde{v}_{i,S} \geq \frac{O^*}{\alpha}$ since $\tilde{v}_{i,S} \geq w^+_{i,S}$ for every (i,S). It is easy to transform $\tilde{x}$ to $\hat{x}$ so that $\sum_{(i,S)\in E} \hat{x}_{i,S} w^+_{i,S} = \sum_{i,S} \tilde{x}_{i,S} \tilde{v}_{i,S}$. We set $\hat{x}_{i,S} = \tilde{x}_{i,S}$ if $(i,S) \in E$ or $\tilde{x}_{i,S} = 0$, otherwise set $\hat{x}_{i,T'} = 1$ for $T' = \arg\max_{T\subseteq S:(i,T)\in E} w_{i,T}$, and $\hat{x}_{i,S} = 0$. Finally, we use Claim 3.3 to convert $\hat{x}$ to a solution $x^l \in \mathbb{Z}(\mathcal{P})$, which gives the desired integer solution. ■

Lemma 3.5 *One can obtain in polynomial time a decomposition $\frac{x^*}{\alpha} = \sum_{l\in\mathcal{I}} \lambda_l x^l$ where $\lambda_l \geq 0$ for every l (with only polynomially many positive λ_l) and $\sum_{l\in\mathcal{I}} \lambda_l = 1$.*

Proof : We first show that the optimal value of (D), and hence of (P) by strong duality, is exactly 1. So an optimal solution to (P) yields the convex decomposition. Setting $z = 1, w_{i,S} = 0$ for all $(i,S) \in E$ gives a feasible solution with value 1. We claim that any feasible solution (w,z) has value at most 1. Suppose $\frac{1}{\alpha} \cdot \sum_{(i,S)\in E} x^*_{i,S} w_{i,S} + z > 1$. Using Claim 3.4 one can find an integer solution x^l such that $\sum_{(i,S)\in E} x^l_{i,S} w_{i,S} \geq \frac{1}{\alpha} \cdot \sum_{(i,S)\in E} x^*_{i,S} w_{i,S} > 1 - z$, contradicting the feasibility of (w,z).

The above argument shows that we can add the inequality $\frac{1}{\alpha} \cdot \sum_{(i,S)\in E} x^*_{i,S} w_{i,S} + z \geq 1$ to the dual program (D) without altering anything. We will run the ellipsoid method on (D) to identify a dual program with a polynomial-size set of inequalities (5), that is equivalent to (D). These inequalities will be the violated inequalities returned by the separation oracle during the execution of the ellipsoid method, that are used to cut the ellipsoid. Taking the dual of this compact program gives a primal program with a polynomial number of variables and constraints which we can solve to get λ_l values that sum to 1; this gives the desired decomposition. The separation oracle that we use is as follows: at a point (w,z), if $\frac{1}{\alpha} \cdot \sum_{(i,S)\in E} x^*_{i,S} w_{i,S} + z > 1$, then we can find an x^l and a corresponding violated constraint using Claim 3.4; otherwise we use the half space $\frac{1}{\alpha} \sum_{(i,S)\in E} x^*_{i,S} w_{i,S} + z \geq 1$ to cut the current ellipsoid. ■

4 Applications

4.1 General and multi-unit combinatorial auctions

In multi-unit combinatorial auctions (MUCAs) we have m items with B copies of each item, and n players, and each player desires at most one copy of an item. With $B = 1$ we get the regular combinatorial auction. The LP relaxation for MUCA is similar to (CA-P), but the RHS of constraints (2) changes to B. The

integrality gap of this relaxation is a function of B: (i) the integrality gap is $O(m^{\frac{1}{B+1}})$ for any $B \geq 1$; (ii) the integrality gap is $(1+\epsilon)$ when $B = \Omega(\log m)$ (more precisely $B \geq c(\epsilon) \cdot \log m$ where $c(.)$ increases with $\frac{1}{\epsilon}$) for any $\epsilon > 0$. Moreover there are algorithms that "verify" this gap, e.g., the derandomization of "standard" randomized rounding [32], the rounding approach of [21], or the recent primal-dual algorithm of [6].

We can use the above approximation algorithms to construct randomized, strategic, approximation mechanisms for CAs and MUCAs, with arbitrary valuation functions. There are two (linked) issues here: (1) the representation of the input, and (2) the computational effort needed to solve (CA-P). With general valuations, representation becomes an issue since the length of the valuation can be exponential in m, which therefore also renders direct revelation mechanisms intractable. Two ways to address this have been considered: (a) assuming that we have "short" valuations that can be succinctly described (with a suitable bidding language), thus allowing for direct revelation mechanisms; (b) assuming that we have some oracle access to the valuation, so the mechanism now needs to be an iterative mechanism. It is well known in the economics literature that the issue of "one-shot" vs. iterative mechanisms also affects truthfulness properties. To specify our results precisely and address (1) and (2), we consider these two settings separately.

Short valuations. We first consider the case when the valuations are succinctly describable. An important such valuation class is that of single-minded bidders where each player wants just one set (or any superset) and specifies that set along with its value. More generally players could specify $k \geq 1$ subset-value pairs $(S_1, w_1), \ldots (S_k, w_k)$, which yields two well-studied classes of valuations; (a) an *XOR* valuation where the player desires *one* of these k subsets (the "k-minded case"), so $v(T) = \max_{i:S_i \subseteq T} w_i$; (b) an *OR* valuation where the player is willing to receive *any* collection of disjoint sets from $S_1, \ldots S_k$ and the values add up, so $v(T) = \max_{\text{disjoint sets } S_{i_1}, \ldots, S_{i_\ell} \subseteq T} \big(v(S_{i_1}) + \cdots + v(S_{i_\ell})\big)$. It is easy to see that for both these classes of valuations, one can devise a demand oracle for the valuation from the above description with running time polynomial in k for an *XOR* valuation, and exponential in k for an *OR* valuation. Hence, one can solve the LP (CA-P) in polynomial time [5] (see also "General valuations" below) in the following two cases: (a) *XOR* valuations, when $k = \mathsf{poly}(m, n)$; and (b) *OR* valuations, when $k = O\big(\log(m+n)\big)$. We use "short" valuations below to encode both, succinctness of descriptions, and the requirement that (CA-P) can be solved in time polynomial in m and n.

Theorem 4.1 *For MUCAs with "short" valuations and B copies of each item, we obtain randomized, truthful in expectation mechanisms with the following approximation factors: (i) a factor of $O(m^{\frac{1}{B+1}})$ for any $B \geq 1$; and (ii) a $(1+\epsilon)$-guarantee for any $\epsilon > 0$ when $B = \Omega(\log m)$.*

We obtain the *first truthful mechanisms* with non-trivial approximation guarantees for *all values of B*, and with short valuations, improve upon the deterministic $O(B \cdot m^{\frac{1}{B-2}})$-approximation ratio of Bartal et al. [4] for general valuations. In particular, when $B = \Omega(\log m)$ we get the *first truthful (in expectation) $(1+\epsilon)$-approximation mechanism* for short multi-parameter valuations, answering an open question in [2] about devising mechanisms for valuations more general than the "known" single-minded case (i.e., the sets are public). More work has been directed toward the single-minded case defined in [27] who gave a deterministic mechanism with an $O(\sqrt{m})$ guarantee when $B = 1$. Archer et al. [2] obtained a $(1+\epsilon)$-approximation mechanism when $B = \Omega(\log m)$, for the "known" single-minded case that is truthful both in expectation and with high probability. Recently, Briest et al. [6] gave a deterministic $O\big(m^{\frac{1}{B}}\big)$-approximation mechanism for the unknown single-minded case. We give a unified way to obtain randomized, truthful in expectation mechanisms and get approximation guarantees that match the known inapproximability results [4].

General valuations. We now consider the setting with general valuations with an oracle access to the valuations. Two kinds of oracle access were considered by [30, 5]: *value queries*, where the query asks

for the value of a set S, and *demand queries*, where the query specifies item prices $p_1, \ldots, p_m$, and asks a player i to return a subset S that maximizes $v_i(S) - \sum_{j \in S} p_j$. Demand queries are very natural from an economic perspective and are known to be strictly more powerful than value queries [30, 5]. Blumrosen and Nisan [5] show that LP (CA-P) can be solved with a polynomial number of demand queries using the ellipsoid method, since they yield a separation oracle for the dual of (CA-P). This addresses issue (2).

As mentioned previously, direct revelation mechanisms become ineffective with general valuations. Using Theorem 3.1 we can construct strategic, iterative mechanisms for general valuations using demand queries. It is well known in the economics literature (see e.g., [10]) that moving from direct-revelation to iterative mechanisms often leads to a weakening of the solution concept, from dominant strategies to *ex-post Nash equilibrium*. Our mechanisms yield truthfulness in expectation as an ex-post Nash equilibrium, which means the following: for any player i, and *any type profile* $v_{-i} \in V_{-i}$ of the other players, if the other players act according to their types in this profile in the iterative mechanism, then i's best response (for maximizing expected utility) is to be truthful. In particular, this means that truthful revelation is a Nash equilibrium with no *ex-post regret*, that is, even if a player were told beforehand the types that the other players would use to act in the iterative mechanism, he would have no incentive to deviate from his truth-telling strategy. As an example, consider the regular second-price auction with two players with the following iterative flavor: player X bids first *and then* player Y states his bid; the player with highest bid wins and pays the second highest bid value. Truthfulness fails to be a dominant strategy. Suppose Y chooses the strategy "if X bids above 5, then I will say 20, otherwise I will say 2", then, if X's true value is higher than 5, his best response is to declare 5. However truthfulness is still an ex-post Nash equilibrium: if Y fixes *any* private value and bids that, then regardless of Y's private value X's best response is to tell the truth.

Theorem 4.2 *For MUCAs with "general" valuations and demand oracles, and B copies of each item, we obtain randomized mechanisms for which truthfulness in expectation is an ex-post Nash equilibrium, with the following guarantees: (i) an $O(m^{\frac{1}{B+1}})$-guarantee for any $B \geq 1$; and (ii) a $(1+\epsilon)$-guarantee for any $\epsilon > 0$ when $B = \Omega(\log m)$.*

Very little is known about truthful and approximation mechanisms with general valuations. For $B \leq 2$ and general valuations, no strategic mechanisms were known previously. Bartal et al. [4] gave a deterministic, truthful $O(B \cdot m^{\frac{1}{B-2}})$-approximation for general valuations using demand oracles, and Dobzinski et al. [14] gave a deterministic, truthful $O(\sqrt{m})$-approximation for $B = 1$ and subadditive valuations using value oracles. We improve upon the performance guarantee of [4] for all B, and get the same guarantee as [14] for general valuations but using demand oracles, randomization, and a different solution concept.

4.2 General multi-unit auctions and multi-parameter knapsack

In a *multi-unit auction*, there are m *identical* items. and each player i has a value $v_i(j)$ for getting j items, $1 \leq j \leq m$, where the $v_i(.)$s are non-decreasing functions. The goal is to distribute the m items, allocating x_i items to player i, so as to maximize $\sum_i v_i(x_i)$. This problem is solvable in time polynomial in m, so if each $v_i(j)$ is explicitly specified then one can implement VCG. We will consider valuations that are specified more succinctly, e.g., step functions, piecewise linear functions with a "small" number of pieces, *XOR*, *OR* valuations or their combinations etc., or valuations given via an oracle access. We want mechanisms that run in time polynomial in n and the size of the specification.

If all value functions are step functions (single-minded players), then this is exactly the knapsack problem, for which Briest et al. [6] gave a truthful FPTAS. Kothari et al. [22] gave an "approximately truthful" FPTAS for the special case of "marginal decreasing valuations". We obtain the *first* truthful, approximation mechanisms for general multi-unit auctions (i.e., multi-parameter knapsack problems). We consider the

following natural LP relaxation for this problem:

$$\max \quad \Big\{\sum_{i,j} v_i(j)x_{ij} \quad \text{subject to} \quad \sum_j x_{ij} \le 1 \quad \text{for all } i; \quad \sum_{i,j} j \cdot x_{ij} \le m; \;\; x_{ij} \ge 0 \;\; \forall i,j\Big\}. \qquad \text{(KN-P)}$$

Here variable x_{ij} indicates if player i gets j items. When the $v_i(.)$s are step functions this LP reduces to the standard LP relaxation for Knapsack, for which a simple greedy algorithm proves an integrality gap of at most 2 (see, e.g., [20]). We give a greedy algorithm that generalizes this algorithm and bounds the integrality gap of (KN-P) by 2. Using this, we obtain the *first truthful, approximation mechanisms for non-single-minded players* and get a guarantee of 2. Lavi et al. [23] showed that with two players, if all items must always be allocated, then with *OR* valuations, no *deterministic* truthful mechanism can obtain an approximation ratio better than 2. As in Section 4.1, "short" valuations will mean that the valuation is given succinctly and we can solve the LP in polynomial time. Since items are identical, a demand oracle here fixes a price $p \ge 0$ and asks player i to return a quantity j that maximizes $v_i(j) - p \cdot j$.

Theorem 4.3 *There is a randomized, 2-approximation mechanism for multi-unit auctions, which (i) is truthful in expectation for "short" valuations; and (ii) has truthfulness in expectation as an ex-post Nash equilibrium for general valuations with demand oracles.*

We now give the greedy algorithm which proves an integrality gap of 2 for (KN-P). Let OPT be the value of the LP-optimum. Define the marginal value function of i given quantity j, $v_i(\cdot|j)$, as follows: for any $x \ge 0$, $v_i(x|j) = v_i(x+j) - v_i(j)$.

1. Initialize $M \leftarrow m$. Each player i starts off with $q_i = 0$ items. While $M > 0$ we repeatedly do the following: (a) find i^*, j^* that maximizes $v_i(j|q_i)/j$; (b) if $j^* \le M$, set $q_{i^*} \leftarrow q_{i^*} + j^*$ (i.e., allocate j^* more items to i^*); (c) decrement M by j^*.

2. Let $\mathsf{Greedy} = \sum_i v_i(q_i)$ be the value of this allocation, and let $\mathsf{Max} = \max_i v_i(m)$. If $\mathsf{Max} > \mathsf{Greedy}$, allocate all items to the player with the maximum $v_i(m)$, otherwise return $(q_1, \ldots q_n)$ as the allocation.

Lemma 4.4 *The above algorithm finds an allocation with value at least $OPT/2$.*

Proof : We will exhibit a feasible dual solution with value at most twice the value obtained by the algorithm, hence the claim will follow. The dual program is

$$\min \quad \Big\{\sum_i \alpha_i + m \cdot \beta \quad \text{subject to} \quad \alpha_i \ge v_i(j) - j \cdot \beta \;\; \forall i,j; \quad \beta, \alpha_i \ge 0 \;\; \forall i\Big\}.$$

Suppose the algorithm stops at the *end* of iteration $s+1$, that is, *after* M it is decremented in iteration $s+1$, we have $M \le 0$. Let (i_l, j_l) be the (i^*, j^*) pair in iteration l for $l = 1, \ldots, s+1$. Recall that q_i is the number of items allocated to player i. Let $q_{i,k}$ denote the quantity $\sum_{l=1:i=i_l}^{k} j_l$, and let $p_i = q_{i,s}$. So $\sum_{l=1}^{s+1} j_l = \sum_i p_i + j_{s+1} \ge m$. Note that $p_i \le q_i$, and it could be that $p_i < q_i$ — this can happen only if $\sum_{l=1}^{s+1} j_l = m$ and $i = i_{s+1}$, so $q_i = p_i + j_{s+1}$. We will exhibit a feasible dual solution of value at most $2\max\{\sum_i v_i(p_i), \mathsf{Max}\}$. Define $\beta = v_{i_{s+1}}(j_{s+1}|q_{i_{s+1}})/j_{s+1}$, and $\alpha_i = v_i(p_i) - p_i \cdot \beta$ for every i. Before verifying feasibility, we show that this proves the lemma:

$$\sum_i \alpha_i + m \cdot \beta \le \sum_i (\alpha_i + p_i \cdot \beta) + j_{s+1} \cdot \beta = \sum_i v_i(p_i) + v_{i_{s+1}}(j_{s+1}|q_{i_{s+1}}) \le 2\max\Big\{\sum_i v_i(p_i), \mathsf{Max}\Big\}.$$

Consider any iteration $l = 1, \ldots, s$. Let $i = i_l, q = q_{i,l-1}$ and $i' = i_{l+1}, q' = q_{i',l}$. We claim that $v_i(j_l|q)/j_l \ge v_{i'}(j_{l+1}|q')/j_{l+1}$. This is clear from the definition of (i^*, j^*) if $i \ne i'$, otherwise this follows since $q' = q + j_l$, and $v_i(j_l|q)/j_l \ge v_i(j_l + j_{l+1}|q)/(j_l + j_{l+1}) = \big(v_i(j_l|q) + v(j_{l+1}|q')\big)/(j_l + j_{l+1})$.

Now fix any i. For $j \geq p_i$, by the definition of β we have $\beta \geq v_i(j - p_i|p_i)/(j - p_i)$ or equivalently, $\alpha_i \geq v_i(j) - j \cdot \beta$. Consider the piecewise-linear function f_i passing through the points $\big(q_{i,l}, v_i(q_{i,l})\big), l = 0, \ldots, s$ (where $q_{i,0} = 0$). By our earlier claim, $f_i(.)$ is a *concave* (increasing) function, and each segment of $f_i(.)$ has slope at least β. These two facts imply that the line passing through $\big(p_i, f_i(p_i)\big)$ with slope β lies above the $f_i(.)$ curve in $[0, p_i]$. Also note that for any $j \leq p_i$, $f_i(j) \geq v_i(j)$ since if $j \in [q_{i,l}, q_{i,l+1}]$ for some l, then $f_i(j) = v_i(q_{i,l}) + (j - q_{i,l}) \cdot \frac{v_i(j_{l+1}|q_{i,l})}{j_{l+1}} \geq v_i(q_{i,l}) + (j - q_{i,l}) \cdot \frac{v_i(j - q_{i,l}|q_{i,l})}{j - q_{i,l}} = v_i(j)$. Combining these facts, we get that for any $j \leq p_i$, $f_i(p_i) - (p_i - j) \cdot \beta \geq f_i(j) \geq v_i(j)$ which when rearranged gives $\alpha_i \geq v_i(j) - j \cdot \beta$. ■

4.3 Graph routing problems

We now consider two closely related problems, *edge-disjoint-paths* (EDP) and *all-or-nothing multi-commodity flow* (ANF). In both EDP and ANF we have a graph $G = (V, E)$ with capacities $u_e \geq 1$ on the edges. The players are (s_i, t_i) source-sink pairs where $s_i, t_i \in V$. Each player has a value w_i. Both (s_i, t_i) and the value w_i are private information; a special case is the "known" case where the (s_i, t_i) pairs are public and only w_i is private. The social-welfare maximization problem is to select a maximum-weight routable set. In EDP, a set is routable if each (s_i, t_i) pair in the set can be assigned an s_i-t_i path and at most u_e paths use edge e; in ANF, for each (s_i, t_i) pair in the set we need to route one unit of flow from s_i to t_i respecting edge capacities, possibly splitting the flow across several paths. For the mechanism-design problem, we define the outcome set carefully to ensure that only players' values for an outcome, and not the set itself, depends on his private information. For EDP, an outcome is an allocation of edges (as paths) to players such that each edge e is used at most u_e times (and player i receives value w_i if he is allocated an s_i-t_i path); for ANF, an outcome is a flow-vector allocation to each player, such that the total flow through each edge e is at most u_e (and player i receives value w_i if his flow-vector sends one unit of flow from s_i to t_i).

We apply our techniques to obtain two types of results for EDP and ANF. Let $B = \min_e u_e$. First, we observe that Theorem 4.1 yields a truthful (in expectation) $O(m^{\frac{1}{B+1}})$-approximation mechanism for both problems. Next, we show that for the *known* case, we can translate the improved guarantees of [8] for EDP and ANF on certain graph classes to the mechanism-design setting to obtain truthful mechanisms with *matching* approximation guarantees for EDP and ANF (on the appropriate graph classes).

Both EDP and ANF admit the following flow-path based LP relaxation. We use P below to index the collection Path of all (simple) paths in G.

$$\begin{aligned} \max \quad & \sum_{i,P} v_i(P) f_{i,P} && \text{(Route-P)} \\ \text{subject to} \quad & \sum_{P} f_{i,P} \leq 1 && \text{for each player } i \\ & \sum_{i} \sum_{P: e \in P} f_{i,P} \leq u_e && \text{for each edge } e \in E \\ & f_{i,P} \geq 0 && \text{for each } i, P. \end{aligned}$$

Let Path_i denote the collection of s_i-t_i paths. Here $f_{i,P}$ denotes the amount of flow on path P, and $v_i(P) = w_i$ if $P \in \mathsf{Path}_i$, and is 0 otherwise. It is easy to see that an integral solution corresponds to an EDP solution, and a solution where $\sum_P f_{i,P}$ is integral for each player i corresponds to an ANF solution. Observe that EDP is a structured MUCA with short valuations (since one can construct a demand oracle for player i, given $(s_i, t_i; w_i)$), and (Route-P) is simply a specialization of (CA-P) (where items are edges, and a set S that does not contain an s_i-t_i path has $v_i(S) = 0$). Also, a solution to EDP is also clearly a solution to ANF. Thus, Theorem 4.1 yields a randomized, truthful $O(m^{\frac{1}{B+1}})$-approximation for both problems.

Chekuri et al. [8] use the above LP to obtain approximation guarantees of (i) $O(\log m)$ for EDP on planar graphs when $B \geq 2$; and (ii) $O(\log^2 m)$ for ANF on general graphs, and $O(\log m)$ for planar graphs. (The guarantees are stated for $w_i = 1$, but it is implicit that the algorithms generalize to arbitrary w_i.) Notice however that this *does not* imply analogous bounds on the integrality gap of (Route-P) for these graph classes since the valuation functions arising in EDP and ANF are *structured valuations*. (In fact, it is not hard to cast MUCA as EDP, so the integrality gap of (Route-P) with $u_e = B$ for all e is $\Theta(m^{\frac{1}{B+1}})$.) However, we show that in the known case, our techniques can be used to obtain matching approximation guarantees in the mechanism-design setting.

The key observation is that since each player i's (true) value depends (linearly) only on the *amount of flow* routed from s_i to t_i (which are publicly known) and not on the actual flow-paths that are chosen, it suffices to obtain a "partial" decomposition such that the (total flow routed for an (s_i, t_i) pair by the convex combination) = (total flow routed for (s_i, t_i) by the LP)/α. To see this formally, recall that (see Lemma 3.2) to translate the fractional VCG mechanism to (a support mechanism, and hence) a truthful-in-expectation mechanism, we require a convex combination $\sum_{l \in \mathcal{I}} \lambda_l(v) x^l$ that satisfies $\sum_{l \in \mathcal{I}} \lambda_l(v) \overline{v}_i(x^l) = \overline{v}_i(f^*(v))/\alpha$ for each player i, where $\overline{v}_i(f)$ is the true value of player i for a feasible solution f to (Route-P), and $f^*(v)$ is an optimal solution to (Route-P) for the valuation-profile v. In the known case, $\overline{v}_i(f) = \overline{w}_i\big(\sum_{P \in \mathsf{Path}_i} f_{i,P}\big)$, and so a "partial" decomposition of $f^*(v)$ suffices to yield this property; but if (s_i, t_i) is private information, then this is no longer true, and one needs to resort to a "complete" decomposition (which requires an integrality-gap-verifying algorithm for (Route-P)).

We now argue briefly that an α-approximation algorithm for EDP or ANF based on the LP (Route-P) can be used to obtain a partial decomposition (with the same α), enabling us to translate the approximation guarantees of [8] to the mechanism-design setting. Similar results are known with demands d_i when $\max d_i \leq B$ and these also transfer to the mechanisms. For notational convenience, given a solution f to (Route-P), we introduce auxilliary variables $x_i := \sum_{P \in \mathsf{Path}_i} f_{i,P}$ to denote the total flow routed for player i. The set of "integer" solutions $\{(f^l, x^l)\}_{l \in \mathcal{I}}$ that the decomposition uses is (i) the set of integer solutions to (Route-P) for EDP; and (ii) the set of solutions to (Route-P) with integral x_is for ANF. (Note that for EDP, the corresponding set satisfies the packing property, and for ANF, the projection of the set to the x-space satisfies the packing property.) To partially decompose a solution (f^*, x^*), we solve the following primal and dual LPs.

$$\begin{array}{llll} \min & \sum_{l \in I} \lambda_l & & \text{(P')} \\ \text{s.t.} & \sum_l \lambda_l x_i^l = \frac{x_i^*}{\alpha} & \text{for all } i & \\ & \sum_l \lambda_l \geq 1 & & \\ & \lambda_l \geq 0 & \text{for all } l \in \mathcal{I}. & \end{array} \qquad \begin{array}{llll} \max & \frac{1}{\alpha} \cdot \sum_i x_i^* \pi_i + z & & \text{(D')} \\ \text{s.t.} & \sum_i x_i^l \pi_i + z \leq 1 & \text{for all } l \in \mathcal{I} & (6) \\ & z \geq 0 & & \\ & \pi_i \text{ unconstrained} & \text{for all } i \in E. & \end{array}$$

Arguments that are almost identical to those in Section 3.1 apply to this pair of LPs. Their optimal value is 1, and it is not hard to see that an LP-based α-approximation algorithm for EDP or ANF can be used to find a violated inequality (6) when $\frac{1}{\alpha} \cdot \sum_i x_i^* \pi_i + z > 1$. So we obtain a separation oracle for (D') as before, and this can be used to obtain a compact dual LP, and hence, a compact primal LP (with optimal value 1) whose solution yields the partial decomposition.

Previous truthful mechanisms for EDP were: for the "known" case, a deterministic $O(\sqrt{m})$-approximation when $d_i = 1$ using the greedy algorithm in [27], and a deterministic $O(m^{\frac{1}{B-1}})$-approximation mechanism [6] when $B > 1$; in the "unknown" case [4] gave an $O\big(B \cdot (\frac{m}{\min d_i})^{\frac{1}{B-2}}\big)$-approximation when $B > 2$. [3] give a deterministic $O(\log^2 w_{\max} \cdot \sqrt{m})$-approximation mechanism for $B = 1$ under the concept of undominated strategies.

5 Conclusions and discussion

We have presented a general method for devising randomized, truthful-in-expectation mechanisms for packing problems that allows one to exploit approximation algorithms and obtain matching approximation guarantees for the mechanism-design problem. Given an α-approximation algorithm that also bounds the integrality gap of the LP-relaxation of a packing problem by α, we show how to design a truthful mechanism with the same approximation ratio. Our method proceeds by decomposing the optimal *LP-solution* scaled down by α into a convex combination of integer solutions, which can then be viewed as a probability distribution over integer solutions. We show that the payments of the fractional VCG mechanism (whose outcome is the LP-optimum) can be modified and combined with the above randomized algorithm to yield a randomized mechanism that is truthful in expectation. We demonstrate the usefulness of our technique on a variety of problem domains: combinatorial auctions, multi-unit auctions, multi-unit combinatorial auctions, and graph-routing problems (edge-disjoint-paths (EDP), all-or-nothing multicommodity flow (ANF)). In many cases, the approximation guarantees we obtain are the best possible if one requires polynomial-time computation (unless *P=NP*).

It is instructive to note that our method can also be viewed as constructing a *maximal-in-distributional-range* (MIDR) mechanism (see [12]). That is, instead of presenting our method as one where we first obtain the fractional VCG mechanism and then translate this to a randomized mechanism, we can view our construction as one that uses an α-integrality-gap-verifying approximation algorithm to *exactly optimize* the social welfare over a subset of the convex hull of integer solutions, and hence, implement VCG over a suitable (fixed) outcome set. To see this, let $\mathcal{P}_I$ denote the convex hull of integer points in $\mathcal{P}$ (i.e., $\mathbb{Z}(\mathcal{P})$), where $\mathcal{P}$ is the polytope of feasible solutions to the LP-relaxation of the packing problem. Observe that the outcome of any randomized or support mechanism corresponds to a point in $\mathcal{P}_I$. Our decomposition lemma (Lemma 3.5) shows that $\mathcal{P}/\alpha \subseteq \mathcal{P}_I$; also, we can efficiently optimize over $\mathcal{P}/\alpha$ since this is equivalent to optimizing over $\mathcal{P}$. Thus, our randomized (or deterministic support) mechanism (which returns the optimum over $\mathcal{P}/\alpha$ as its outcome) is essentially VCG over the outcome-set $\mathcal{P}/\alpha$. (And the α-approximation ratio follows since the optimum value over $\mathcal{P}/\alpha$ is within an α-factor of the optimum over $\mathcal{P}$.)

Subsequent to the publication of the conference version of this paper [24], various other solutions have been obtained for combinatorial auctions and multi-unit auctions. For combinatorial auctions, [15, 11] obtain the same approximation guarantees that we obtain but under the stronger solution concept of universal truthfulness. For multi-unit auctions, [13] give a deterministic truthful 2-approximation, and [12] obtain an FPTAS that is truthful in expectation (also via an MIDR mechanism). These mechanisms are however *tailored* to the specific problem considered; in contrast, our method is the only (currently known) *general* method to convert approximation algorithms to truthful mechanisms (with matching approximation guarantees) for multidimensional problem domains. It is also worth noting that our mechanisms have remained the state-of-art for other problem domains, most notably, multi-unit combinatorial auctions, where no other truthful mechanism achieves similar approximation guarantees.

One of the insights to emerge from our work is that fractional truthful mechanisms combined with a suitable rounding scheme can be an effective tool in truthful mechanism design even for multidimensional domains, The versatility and usefulness of this approach has been reinforced in various subsequent works, such as [25, 17, 16, 19]. The latter three papers [17, 16, 19] all devise truthful-in-expectation mechanisms for their respective problems (stochastic market planning, generalized assignment problem, secondary spectrum auctions) by first devising a fractional truthful mechanism and then adapting our rounding ideas to convert this into the desired randomized mechanism. Lavi and Swamy [25] consider a special case of the minimum-makespan scheduling problem on unrelated machines. In their case, constructing a fractional truthful mechanism is non-trivial since makespan minimization is not a social-welfare-maximization problem, and the rounding scheme is tailored to the problem at hand and different from the one proposed here.

5.1 Covering problems

While we focus on packing problems, another broad class of optimization problems that has natural game-theoretic aspects is covering problems. Our techniques can be adapted to yield some partial results for this class of problems. As an illustrative example, consider the *set cover* problem, which can be used to abstract the following setting. A buyer needs to compose a team of workers that together cover a set of capabilities. Each worker offers some of these capabilities, and incurs a certain cost if hired as part of the team. The optimization problem is to select a minimum-cost set of workers that cover the required capabilities. In the mechanism-design setting (sometimes referred to as a *procurement auction*), the workers' costs and/or capabilities are private information, and one therefore seeks to construct a truthful mechanism that achieves a "good" approximation ratio. (As in the case of graph routing problems, we need to define the outcome set carefully so that only players' values for an outcome and not the set itself depends on private information.) If the integrality gap of the LP relaxation of the covering problem is $\alpha > 1$ (as verified by an α-approximation algorithm), then it is not hard to see that (by dovetailing the arguments in Section 3.1), given an LP-solution x, one can decompose $\alpha \cdot x$ into a convex combination of integer solutions where *multiple copies* of a worker may be purchased. That is, these integer solutions (and hence, the random integer solution specified by the convex combination) are only feasible for the original problem provided a worker may be hired multiple times; in such cases our method yields an α-approximation, truthful (in expectation) mechanism.

An example of such a setting is the *multiset multicover problem with multiplicities*, where each capability must be covered a certain number of times, each worker's private information is a vector specifying the amounts of the different capabilities he offers, and there is no limit on how many copies of a worker may be chosen. Another example is the class of *network design problems*, where each worker owns edges of an underlying known graph: the identity of the edges owned and their costs are private, and the goal is to select a minimum-cost edge-set satisfying certain connectivity requirements, possibly with other side constraints such as degree bounds. Such network design problems often come in two flavors, one where there is a bound on the number of times an edge may be selected, and one where there is no such bound. Our technique yields truthful, approximation mechanisms for the latter class of problems with guarantees matching that of the best known LP-based (integrality-gap-verifying) approximation algorithm for the problem.

We also remark that for *single-dimensional* covering problems—e.g., network design problems where each edge is a separate player and its cost is private information—we can adapt our proofs to show that one can decompose $\min\{\alpha \cdot x, \mathbf{1}\}$ into a convex combination of *feasible* integer solutions to the problem (that is, we no longer need to buy multiple copies). (Here $\mathbf{1}$ denotes the all ones vector and $\min$ denotes component-wise minimum.) The resulting randomized mechanism then satisfies *value monotonicity*, a necessary and sufficient condition for truthfulness, and we again get a general result: *given an LP-based α-approximation for a covering problem, one can get an α-approximation truthful mechanism for the analogous single-dimensional mechanism-design problem.* As an application, we mention that this immediately yields the first truthful, approximation mechanisms for the single-dimensional mechanism-design version of *survivable network design* (possibly with degree constraints) with approximation guarantees matching those of the algorithmic problem.

A very interesting open problem is to design a truthful approximation mechanisms for multidimensional covering problems when such multiplicities are not allowed.

Acknowledgment

We thank Bruce Shepherd for very useful discussions about the results in [8].

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] A. Archer, C. Papadimitriou, K. Talwar, and É. Tardos. An approximate truthful mechanism for combinatorial auctions with single parameter agents. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 205–214, 2003.

[3] M. Babaioff, R. Lavi, and E. Pavlov. Single-value combinatorial auctions and implementation in undominated strategies. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1054–1063, 2006.

[4] Y. Bartal, R. Gonen, and N. Nisan. Incentive compatible multi-unit combinatorial auctions. In *Proceedings of the 9th Conference on Theoretical Aspects of Rationality and Knowledge (TARK)*, pages 72–87, 2003.

[5] L. Blumrosen and N. Nisan. On the computational power of iterative auctions. In *Proceedings of the 6th ACM Conference on Electronic Commerce (EC)*, pages 29–43, 2005.

[6] P. Briest, P. Krysta, and B. Vöcking. Approximation techniques for utilitarian mechanism design. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 39–48, 2005.

[7] R. Carr and S. Vempala. Randomized metarounding. *Random Structures and Algorithms*, 20(3):343–352, 2002.

[8] C. Chekuri, S. Khanna, and F.B. Shepherd. Multicommodity flows, well-linked terminals and routing problems. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 183–192, 2005.

[9] E. Clarke. Multipart pricing of public goods. *Public Choice*, 8:17–33, 1971.

[10] V. Conitzer and T. Sandholm. Computational criticisms of the revelation principle. In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC)*, pages 262–263, 2004.

[11] S. Dobzinski. Two randomized mechanisms for combinatorial auctions. In *Proceedings of the 11th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, pages 89–103, 2008.

[12] S. Dobzinski and S. Dughmi. Truthful randomized mechanisms for combinatorial auctions. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 505–514, 2009.

[13] S. Dobzinski and N. Nisan. Mechanisms for multi-Unit auctions. In *Proceedings of the 8th ACM Conference on Electronic Commerce (EC)*, pages 346–351, 2007.

[14] S. Dobzinski, N. Nisan, and M. Schapira. Approximation algorithms for combinatorial auctions with complement-free bidders. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 610–618, 2005.

[15] S. Dobzinski, N. Nisan, and M. Schapira. Truthful randomized mechanisms for combinatorial auctions. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC)*, pages 652–661, 2006.

[16] S. Dughmi and A. Ghosh. Truthful assignment without money. In *Proceedings of the 11th ACM Conference on Electronic Commerce (EC)*, 2010. To appear.

[17] D. Golovin. Stochastic packing-market planning. In *Proceedings of the 8th ACM Conference on Electronic Commerce (EC)*, pages 172–181, 2007.

[18] T. Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973.

[19] M. Hoefer, T. Kesselheim, and B. Vöcking. Combinatorial auctions for secondary spectrum auctions. *Manuscript*, 2010.

[20] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack problems*. Springer-Verlag, 2004.

[21] S. Kolliopoulos and C. Stein. Approximating disjoint-path problems using greedy algorithms and packing integer programs. *Mathematical Programming, Series A*, 99:63–87, 2004.

[22] A. Kothari, D. Parkes, and S. Suri. Approximately-strategyproof and tractable multi-unit auctions. In *Proceedings of the 4th ACM Conference on Electronic Commerce (EC)*, pages 166–175, 2003.

[23] R. Lavi, A. Mu'alem, and N. Nisan. Towards a characterization of truthful combinatorial auctions. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 574–583, 2003.

[24] R. Lavi and C. Swamy. Truthful and near-optimal mechanism design via linear programming. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 595–604, 2005.

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

[26] B. Lehmann, D. Lehmann, and N. Nisan. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior*, 55(2):270–296, 2006.

[27] D. Lehmann, L. O'Callaghan, and Y. Shoham. Truth revelation in approximately efficient combinatorial auctions. *Journal of the ACM*, 49(5):1–26, 2002.

[28] N. Nisan. The communication complexity of approximate set packing and covering. In *Proceedings of the 29th International Colloquium on Automate, Languages and Programming (ICALP)*, pages 868–875, 2002.

[29] N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35:166–196, 2001.

[30] N. Nisan and I. Segal. The communication requirements of efficient allocations and supporting prices. *Journal of Economic Theory*, 129(1):192–224, 2006.

[31] N. Nisan. Bidding languages for combinatorial auctions. In P. Cramton, Y. Shoham and R. Steinberg, editors, *Combinatorial Auctions*, The MIT Press, 2005.

[32] P. Raghavan. Probabilistic construction of deterministic algorithms: approximating packing integer problems. *Journal of Computer and Systems Sciences*, 37:130–143, 1988.

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