A Universally-truthful Approximation Scheme for Multi-unit Auctions

Berthold Vöcking
Department of Computer Science
RWTH Aachen University
voecking@cs.rwth-aachen.de

Abstract

We present a randomized, polynomial-time approximation scheme for multi-unit auctions. Our mechanism is truthful in the universal sense, i.e., a distribution over deterministically truthful mechanisms. Previously known approximation schemes were truthful in expectation which is a weaker notion of truthfulness assuming risk neutral bidders. The existence of a universally truthful approximation scheme was questioned by previous work showing that multi-unit auctions with certain technical restrictions on their output do not admit a polynomial-time, universally truthful mechanism with approximation factor better than two.

Our new mechanism employs VCG payments in a non-standard way: The deterministic mechanisms underlying our universally truthful approximation scheme are not maximal in range and do not belong to the class of affine maximizers which, on a first view, seems to contradict previous characterizations of VCG-based mechanisms. Instead, each of these deterministic mechanisms is composed of a collection of affine maximizers, one for each bidder. This yields a subjective variant of VCG in which payments for different bidders are defined on the basis of possibly different affine maximizers.

1 Introduction

The field of algorithmic mechanism design [17] provides computationally efficient mechanisms implementing social-choice functions in environments with multiple participants like, e.g., the Internet, market places, and auctions. Such a mechanism corresponds to an algorithm that aggregates the preferences of different participants into a single joint decision that maximizes social welfare or other objectives.

The mechanism should work well assuming rational selfish behavior of the participants. In order to achieve this goal, the algorithm needs to extract information from the different participants. If the mechanism is designed in such a way that it is a dominant strategy for the participants to honestly reveal their preferences, it is called *incentive compatible* or *truthful*.

The study of combinatorial auctions is central to the field of algorithmic mechanism design. The optimization problems underlying various kinds of combinatorial auctions are NP-hard but often there are known efficient approximation algorithms for these problems. Incentive compatibility, however, puts additional constraints on the design of mechanisms that rule out standard methods known from the design of approximation algorithms so that the known algorithms cannot be applied. Several previous studies show that randomization has the potential to overcome the restrictions imposed by incentive compatibility, see, e.g., [4, 7, 8, 13]. Three different notions of truthfulness are distinguished:

- *Deterministic truthfulness:* A mechanism is truthful if a bidder always maximizes his utility by bidding truthfully.
- *Universal truthfulness:* A universally truthful mechanism is a probability distribution over deterministically truthful mechanisms.
- *Truthfulness in expectation:* A mechanism is truthful in expectation if a bidder always maximizes his expected utility by bidding truthfully.

Observe that, if the random bits used by the mechanism are known, a universally truthful algorithm corresponds to a deterministically truthful algorithm. That is, the randomization does not affect the incentive compatibility but only other aspects like, e.g., the approximation guarantee. In contrast, truthfulness in expectation is a significantly weaker concept as it yields incentive compatibility only if bidders do not know the outcome of the random bits. Another, even more critical weakness of truthfulness in expectation is that it assumes bidders to be risk neutral. For a further discussion see, e.g., [7].

Multi-unit auctions are a good basis for studying the power of randomization in algorithmic mechanism design. They are arguably the most basic variant of combinatorial auctions: m identical items shall be allocated to n bidders such that social welfare is maximized. Neglecting the issue of truthfulness, the multi-unit auction problem is a generalization of the knapsack problem and admits a fully polynomial-time approximation scheme (FPTAS). In contrast, the best known polynomial-time, deterministically truthful mechanism guarantees only a 2-approximation [5].

It is known that deterministic mechanisms following Roberts' characterization [18], i.e., affine maximizers with VCG-based payments, require an exponential number of queries in order to achieve an approximation factor better than two for multi-unit auctions [5]. Only very recently, the exponential lower bound on the number of queries has even been extended towards a class of mechanisms beyond Roberts' characterization [6]. Hence, getting an approximation ratio better than two for multi-unit auctions is either impossible or requires to develop completely new ideas.

Much better results are known for randomized mechanisms. Recently, Dobzinski and Dughmi [4] were able to present a randomized FPTAS for multi-unit auctions. This FPTAS is truthful in expectation. More generally, they study the difference in the computational power between mechanisms that are truthful in expectation and mechanisms that are universally truthful. With the purpose to separate these two concepts, they show that there is a variant of multi-unit auctions with certain restrictions on the output for which there exists an FPTAS being truthful in expectation but there does not exist a polynomial-time universally truthful algorithm with an approximation factor better than two. The authors state that they ideally would like to prove this negative result for multi-unit auctions rather than only for a technical variant of these auctions.

We show, however, that proving a lower bound on the approximation ratio of universally truthful mechanisms for multi-unit auctions is not possible. In particular, we present a randomized polynomial-time approximation scheme (PTAS) that is universally truthful. Our approach uses VCG payments in a non-standard way extending the notion of VCG-based mechanisms slightly beyond the characterization of Roberts. The deterministic mechanisms underlying our randomized approximation scheme are not affine maximizers themselves but they are composed of a collection of affine maximizers, one for each bidder. This way, we can ensure truthfulness by using VCG payments for each bidder based on the bidder's affine maximizer.

The affine maximizers used by our mechanism optimize social welfare subject to additive perturbations of the valuations. The running time of the analysis employs known results from the smoothed analysis of Pareto-optimal solutions [1, 2]. Different bidders might use different scales of perturbation, depending on the bids of the other bidders. The major technical challenge is to ensure that the combination of the possibly different affine maximizers applied to different bidders leads to a feasible allocation, that is, to an allocation not exceeding the number of available items.

2 Preliminaries

In a multi-unit auction, a set of m identical items has to be allocated to n bidders. Each bidder i has a valuation function $v_i : \{0, \ldots, m\} \to \mathbb{R}_{\geq 0}$ satisfying the following standard assumptions: The valuation functions v_i are non-decreasing (*free disposal*) and $v_i(0) = 0$ (*normalization*). Let V denote the set of non-decreasing and normalized valuation functions. The set of feasible allocations is

$$A = \left\{ s = (s_1, \ldots, s_n) \in \{0, \ldots, m\}^n \;\middle|\; \sum_{i=1}^{n} s_i \leq m \right\}.$$

The valuation of bidder i for allocation s is denoted by $v_i(s)$. Let us explicitly point out that v_i depends only on s_i and not on s_{-i}, that is, $v_i(s) = v_i(s_i)$ (*no externalities*). The objective is to find an allocation $s \in A$ maximizing the *social welfare* $v(s) = \sum_{i=1}^{n} v_i(s_i)$.

It is assumed that the valuation functions are not given explicitly but in form of a black box that can be queried by the mechanism. The black box answers so-called *weak value queries*: Given $i \in [n]$, $k \in [m]$, what is the value $v_i(k)$? The challenge is to find an approximately optimal solution without querying the valuation functions completely. An efficient algorithm is supposed to run in time polynomial in n and $\log m$, which is the established notion of *polynomial time* in this context.

A deterministic mechanism for multi-unit auctions is a pair (f, p) with $f : V^n \to A$ being a social choice function and $p = (p_1, \ldots, p_n)$, $p_i : V^n \to \mathbb{R}$ being a payment scheme. A mechanism is *(deterministically) truthful* if it is a dominant strategy for each bidder to report his true valuation, that is, for all $i \in [n]$, all v_i, v_i' and all v_{-i} it holds that bidder i's utility when bidding v_i is not smaller than the bidders utility when bidding v_i', i.e., $v_i(f(v_i, v_{-i})) - p_i(v_i, v_{-i}) \geq v_i(f(v_i', v_{-i})) - p_i(v_i', v_{-i})$.

A randomized mechanism is a probability distribution over deterministic mechanisms. It is called *universally truthful* if each of these deterministic mechanisms is truthful. A weaker form of randomized incentive compatibility is *truthfulness in expectation* which assumes that a bidder's expected utility is maximized when revealing his true valuations.

2.1 Characterizations of truthfulness. Starting point for our considerations is the well-known *direct characterization* of incentive compatible mechanisms (see, e.g., [16]).

PROPOSITION 2.1. *A mechanism (f, p) is truthful if and only if it satisfies the following conditions for every bidder i and every v_{-i}.*

a) There exists a price $q_s^{(i)}(v_{-i})$, for every $s \in A$, such that for all v_i with $f(v_i, v_{-i}) = s$, it holds $p(v_i, v_{-i}) = q_s^{(i)}(v_{-i})$.

b) The social choice function maximizes the utility for player i. That is,

$$f(v) = \underset{s \in A^{(i)}(v_{-i})}{\operatorname{argmax}} \left(v_i(s) - q_s^{(i)}(v_{-i})\right)$$

with $A^{(i)}(v_{-i}) \subseteq A$ being any non-empty subset of allocations.

These properties can be satisfied by using a social choice function corresponding to an *affine maximizer*, i.e., a function $f : V^n \to A'$, $A' \subseteq A$ of the form $f(v) = \operatorname{argmax}_{s \in A'}(\sum_{i=1}^n w_i v_i(s) + c_s)$ with constants $w_i > 0$ (multiplicative weights) and $c_s \in \mathbb{R}$ (additive perturbations). Combining an affine maximizer with a *weighted variant of VCG payments*, i.e., $p_i(v) = h_i(v_{-i}) - \sum_{k \neq i}(w_k/w_i)v_k(s) - c_s/w_i$ with $h_i : V^{n-1} \to \mathbb{R}$ being an arbitrary function, yields a truthful mechanism. For more details, we refer the reader to Nisan's *Introduction to Mechanism Design* [16].

Roberts' characterization of incentive compatible mechanisms [18] shows that only affine maximizers satisfy the conditions of the direct characterization, provided that the domain of the bidders' valuations is unrestricted, i.e. $V = \mathbb{R}^{|A|}$.

2.2 A subjective variant of VCG. The valuations of multi-unit auctions, like many other kinds of combinatorial auctions, are restricted by the assumptions of free disposal and no externalities so that there might be truthful mechanisms beyond Roberts' characterization. This motivates us to extend the scope of mechanisms in a way that allows us to apply different affine maximizers (or other, more general, quasi-linear maximizers) to different bidders for computing partial allocations that are then combined to a joint allocation. We describe our approach in its most general form.

DEFINITION 2.1. *Let $f^{(1)}, \ldots, f^{(n)}$, $f^{(i)} : V^n \to A$ be a collection of n functions such that*

$$f^{(i)}(v) = \underset{s \in A^{(i)}(v_{-i})}{\operatorname{argmax}} \left(v_i(s) + g_s^{(i)}(v_{-i})\right)$$

with $A^{(i)}(v_{-i}) \subseteq A$ being a non-empty subset of allocations and $g_s^{(i)} : V^{n-1} \to \mathbb{R}$ being an arbitrary function. The function $f : V^n \to \{0, \ldots, m\}^n$ defined by $f(v)_i = f^{(i)}(v)_i$, for $1 \le i \le n$, is called a composition of quasi-linear maximizers. *This composition is called* feasible *if $f(V^n) \subseteq A$, i.e., if it defines a feasible social choice function $f : V^n \to A$.*

Note that every affine maximizer f corresponds to a feasible composition of quasi-linear maximizers when setting $f^{(i)} = f$, $A^{(i)}(v_{-i}) = A'$ and $g_s^{(i)}(v_{-i}) = \sum_{k \neq i}(w_k/w_i)v_k(s) + c_s/w_i$, for all $i \in [n]$. In fact, our approximation scheme chooses each of the functions $f^{(1)}, \ldots, f^{(n)}$ to be an affine maximizer. However, these affine maximizers are not necessarily identical but employ possibly different additive perturbations $c_s^{(i)}$.

When comparing Definition 2.1 with the direct characterization of truthful mechanisms in Proposition 2.1, it becomes obvious that the composition of quasi-linear maximizers is more general than affine maximizers. When setting $g_s^{(i)}(v_{-i}) = -q_s^{(i)}(v_{-i})$ one observes that every mechanism that satisfies the direct characterization is a feasible composition of quasi-linear maximizers and vice versa. In other words, a deterministic mechanism (f, p) is truthful if and only if f corresponds to a feasible composition of quasi-linear maximizers.

The following proposition shows how to choose the prices p for a given feasible composition of quasi-linear maximizers f, such that the mechanism (f, p) is truthful.

PROPOSITION 2.2. *Let $f = (f^{(1)}, \ldots, f^{(n)})$ be a feasible composition of quasi-linear maximizers. Let $h_i : V^{n-1} \to \mathbb{R}$ be an arbitrary function that does not depend on v_i. For every $i \in [n]$, define $p_i(v) = h_i(v_{-i}) - g_s^{(i)}(v_{-i})$ with $s = f^{(i)}(v)$ denoting the allocation selected by function $f^{(i)}$. Then (f, p) is truthful.*

Proof. The utility of bidder i for an allocation s chosen by f_i is $v_i(s) - p_i(v) = v_i(s) + g_s^{(i)}(v_{-i}) - h_i(v_{-i})$. If v_{-i} is fixed arbitrarily, then this expression is maximized if $v_i(s) + g_s^{(i)}(v_{-i})$ is maximized, which is what happens when i reports v_i truthfully as $f^{(i)}$ picks the allocation that maximizes $v_i(s) + g_s^{(i)}(v_{-i})$. □

Let us remark that when h_i is chosen according to Clarke's pivot rule, i.e., $h_i(v_{-i}) = v(f^{(i)}(0, v_{-i}))$, then the mechanism is *individually rational*, i.e., bidders always get non-negative utility, and has *no positive transfer*, i.e., payments are always non-negative. This approach corresponds to a *subjective variant of VCG.*

2.3 Previous work. Neglecting the issue of truthfulness, the multi-unit auction problem is a generalization of the NP-hard knapsack problem and admits a fully polynomial time approximation scheme (FPTAS). In particular, in the *single minded* case, in which each v_i is a step function with only one step, the computational problem behind multi-unit auctions corresponds to the knapsack problem. It is well known that the knapsack problem admits an FPTAS [11]. This FPTAS, however, is not incentive compatible. In the single-minded case truthfulness is achieved if algorithms satisfy certain monotonicity properties. Mu'alem and Nisan [14] present a generic approach for combining monotone algorithms. They are able to present a 2-approximation for single-minded multi-unit auctions. Briest et al. [3] extend their approach and, this way, derive a truthful FPTAS for the single-minded case.

Monotonicity yields truthfulness only for single-minded bidders. All known computationally efficient, deterministically truthful mechanisms beyond the single-minded case are *maximal-in-range* (MIR), i.e., optimize over a subrange $A' \subseteq A$. This subrange is chosen in such a way that an optimal allocation is found in polynomial time. Using the MIR approach, Dobzinski and Nisan [5] present a PTAS for multi-unit auctions with k-minded bidders, i.e., each v_i is a step function with up to k steps. The PTAS is complemented by a proof showing that the MIR approach cannot be used to derive an FPTAS for k-minded bidders, unless $\mathsf{P} = \mathsf{NP}$.

For the case of valuations given by black boxes, Dobzinski and Nisan [5] give a polynomial time, deterministically truthful 2-approximation mechanism based on the MIR approach. In addition they prove, that any algorithm that is based on this approach and achieves an approximation ratio better than 2 needs an exponential number of queries. This negative result can be extended to VCG-based algorithms employing affine maximizers.

Very recently, Dobzinski and Nisan [6] introduced a novel kind of auction mechanisms for multi-unit auctions called *triage auction*. Remarkably, the triage auction is not VCG-based but admits approximation ratios better than two. Unfortunately, however, it does not allow for an efficiently computable implementation. Interestingly, it follows from the analysis of the triage auctions that every mechanism that is *scalable*, i.e., multiplying all valuations by the same positive constant does not change the allocation, and guarantees an approximation ratio better than 2 needs an exponential number of queries. Hence, an efficiently computable, deterministically truthful mechanism that achieves a better approximation ratio than 2 cannot be scalable.

In a seminal work, Lavi and Swami [13] have introduced a very general framework showing that any approximation algorithm witnessing an LP integrality gap can be transformed into an algorithm that is truthful in expectation. Their algorithm follows an approach called maximal-in-distributional-range (MIDR) that optimizes over a range of distributions over allocations and applies VCG prices to these distributions. For the multi-unit auction problem the integrality gap is 2 and, hence, the framework of Lavi and Swami gives a 2-approximation.

Recently, Dobzinski and Dughmi [4] have presented an improved MIDR algorithm for the multi-unit auction. In particular, they present an FPTAS for multi-unit auctions that is truthful in expectation. Dughmi and Roughgarden [8] present a more general result showing that every packing problem that admits an FPTAS also admits a truthful-in-expectation randomized mechanism that is an FPTAS.

2.4 Outline. Before presenting the randomized mechanism, we introduce two tools addressing the key issues that need to be solved. In Section 3, we present an affine maximizer, called Δ-perturbed maximizer, that perturbs all valuations by adding "relatively small" numbers. The scale of these perturbations is controlled by the parameter Δ. On the one hand, if Δ is chosen small enough then the optimal allocation with respect to the perturbed bids yields a good approximation of the optimal allocation with respect to the original bids. On the other hand, if Δ is sufficiently large then the running time of the Δ-perturbed maximizer can be bounded polynomially. The proof for the running time of this algorithm applies results from the smoothed analysis of Pareto-optimal solution [1, 2].

Ideally, Δ should be chosen as a function of the maximum bid $v_{\max}$. Unfortunately, truthfulness precludes explicitly using $v_{\max}$ as it depends on the valuations of all bidders. Instead, for each bidder i, our mechanism computes a lower bound L_i on $v_{\max}$ that is independent of the valuations v_i. It is crucial for the correctness of our mechanism that all bidders agree on the same lower bound if the largest and the second largest bids are relatively close. In Section 4, we formally investigate this consensus problem in terms of a two player game and present a randomized solution to the problem that is based on and extends previous results about randomized consensus in mechanism design [9, 10].

In Section 5, we introduce our mechanism using the previously presented tools. In particular, universal truthfulness is achieved by applying the subjective variant of VCG presented in Section 2.2. We bound the expected approximation ratio and the expected running time of the algorithm. Finally, in Section 6, we present a variant of the algorithm with a deterministic running time bound.

3 The Δ-perturbed maximizer

Our mechanism calls an affine maximizer with different kinds of additive perturbations as a subroutine. The Δ-perturbed maximizer, for $\Delta > 0$, takes as input the bids and perturbs these bids by adding two terms to each bid. That is, for $1 \le i \le n$, $0 \le j \le m$, $v_i(j)$ is increased by an integer $q(j) = O(\log m)$ and a random number $x_i^j \in [0,1]$ both multiplied by Δ. Thus, the parameter Δ controls the scale of the perturbation.

Definition 3.1. (Δ-perturbed maximizer)
Let $\Delta > 0$. For $1 \le i \le n$, $0 \le j \le m$, let x_i^j be a random variable chosen independently, uniformly at random from $[0,1]$. For $k \in \{1, \dots, m\}$, let $q(k)$ denote the largest integer such that k is a multiple of $2^{q(k)}$. Set $q(0) = \lfloor \log m \rfloor + 1$. For $1 \le i \le n$, $0 \le j \le m$, define

$$v'_i(j) = v_i(j) + (2q(j) + x_i^j)\Delta\,.$$

The Δ-perturbed maximizer *selects an allocation $s \in A$ maximizing $v'(s) = \sum_{i=1}^n v'_i(s_i)$.*

Observe that $0 \le q(k) \le \lfloor \log m \rfloor + 1$, for $0 \le k \le m$. Hence, for every $s \in A$, $0 \le v'(s) - v(s) \le (2\lfloor \log m \rfloor + 3)\Delta n$. This yields

Lemma 3.1. *The Δ-perturbed maximizer selects an allocation maximizing the social welfare up to an additive error of $(2\lfloor \log m \rfloor + 3)\Delta n$.* □

Lemma 3.2. *The Δ-perturbed maximizer sets $s_i = 0$, for every i with $v_i(m) < \Delta$.*

Proof. For $1 \le k \le m$, it holds $q(k) \le \lfloor \log m \rfloor \le q(0) - 1$. Combining this equation with $v_i(k) \le v_i(m) < \Delta$ and $x_i^k \le 1$ gives

$$\begin{aligned} v'_i(0) &= 2q(0)\Delta + x_i^0 \\ &\ge 2(q(k)+1)\Delta \\ &> 2q(k)\Delta + v_i(k) + x_i^k\Delta \\ &= v'_i(k)\,, \end{aligned}$$

for every $0 \le k \le m$. Consequently, setting $s_i = 0$ maximizes v'_i and, hence, v', too. □

Next we turn our attention to the efficient computation of the Δ-perturbed maximizer. As the running time of our algorithm shall be bounded polynomially in $\log m$, it is not possible to evaluate the valuation function and the random variables x_i^j completely. We will show that it is sufficient to consider only a small number of breakpoints of the valuation functions in order to find the allocation maximizing v'. This property is based on the additve perturbations of the valuations by the $2q(k)\Delta$ term in the definition of v'.

The breakpoints are defined as follows. Let $V_i = (v_i(0), v_i(1), \dots, v_i(m))$ denote the non-decreasing sequence of bids of bidder i. We partition V_i into non-decreasing subsequences $V_i^q = (v_i(k))_{k=0,\dots,m|q(k)=q}$, for $0 \le q \le \lfloor \log m \rfloor + 1$. The *$q$-breakpoints* of bidder i are defined to be the smallest indices of the sequence V_i^q such that the value of this index is at least $0, \Delta, 2\Delta, \dots$ The *breakpoints* of bidder i are defined to be the union of the q-breakpoints, for $0 \le q \le \lfloor \log m \rfloor + 1$.

Lemma 3.3. *Any allocation s maximizing $v'(s)$ satisfies that s_i, for $1 \le i \le n$, is a breakpoint of V_i.*

Proof. For the purpose of a contradiction, assume that s maximizes v' and s_i is not a breakpoint. Let $q = q(s_i)$. Observe that $s_i > 0$ as index 0 is a breakpoint. This implies that $q \le \lfloor \log m \rfloor$ and s_i is a multiple of 2^q, but not a multiple of 2^{q+1}. Now consider the allocation s' with $s'_j = s_j$, for $j \ne i$, and $s'_i = s_i - 2^q$. This way, s'_i is a multiple of 2^{q+1} such that $q' = q(s'_i) \ge q + 1$. This gives

$$v'_i(s_i) = v_i(s_i) + 2q\Delta + x_i^{s_i}\Delta \le v_i(s_i) + 2q'\Delta - \Delta\,.$$

As s_i is not a breakpoint, it is not the first element in V_i^q and the difference between $v_i(s_i)$ and the value of its predecessor in this list is less than Δ. The element placed before s_i in this list is $s_i - 2^{q+1}$. This gives

$$v_i(s_i) < v_i(s_i - 2^{q+1}) + \Delta \le v_i(s'_i) + \Delta\,.$$

By combining these equations, we obtain

$$v'_i(s_i) < v_i(s'_i) + 2q'\Delta \le v'_i(s'_i)\,.$$

That is, we have constructed a feasible allocation s' with $v'(s) < v'(s')$ which contradicts our assumption and thus proves the lemma. □

Finding the optimal solution with respect to the perturbed maximizer corresponds to the *multiple-choice knapsack problem*: For simplicity in notation, assume that there is the same number b of breakpoints for each bidder. This way, there are nb objects, each of which corresponds to one of the breakpoints. Let us use the tuple (i,j) for denoting a breakpoint of bidder i at which j items are allocated to bidder i. An object k corresponding to a breakpoint (i,j) has *weight* $w_k = j$ and *profit* $p_k = v'_i(j)$. The objects are divided into n classes, one for each bidder. The algorithm has to pick a subset $S \subseteq [nb]$ containing at most one object from each of the n classes such that the sum of the weights $w(S)$ is at most m (the capacity of the knapsack) and the sum of the profits $p(S)$ is maximized.

We use the dynamic programming framework of Nemhauser and Ullmann [15], see also [12], for enumerating all the Pareto-optimal subsets, where Pareto-optimality is defined as follows. A subset $S \subseteq [nb]$ is called feasible if it contains at most one object from each class. Subset S is said to *dominate* subset S' if $w(S) \leq w(S')$ and $p(S) > p(S')$. (Since profits are perturbed by adding continuous random variables, we can safely assume that different subsets have different profits, which simplifies the presentation.) The *Pareto-optimal* subsets are those feasible subsets that are not dominated by any other feasible subset.

The running time of the dynamic program is $O(b\sum_{i=1}^{n} k_i)$ where k_i denotes the number of Pareto-optimal sets restricted to the feasible subsets over the classes (bidders) 1 to i. Now we exploit that the Δ-perturbed maximizer adds a random number x_i^j times Δ to each profit (valuation). Recall that these numbers are chosen independently uniformly at random from [0,1]. This perturbation model fits into the framework of smoothed analysis for the knapsack and other binary optimization problems introduced in [2]. This way, we obtain an upper bound on the expected number of Pareto-optimal subsets.

LEMMA 3.4. ([1, 2]) *For every $i \in [n]$, $\mathbf{E}[k_i] = O(b^2n^2P/\Delta)$ with P denoting the maximum profit.* $\square$

Hence, by linearity of expectation, the expected running time for enumerating the Pareto-optimal subsets is $O(b^3n^3P/\Delta)$. In order to bound the running time of the Δ-perturbed maximizer, we could now substitute $v_{\max}$ for P. However, it will turn out that this is not good enough in order to get a polynomial running time bound for the mechanism. In particular, the mechanism chooses Δ proportional to the *second largest bid* $\max_{i\neq i^*} v_i(m)$. In order to get a polynomial running time bound for the mechanism, we need thus a running time bound for the Δ-perturbed maximizer in terms of the second largest rather than the largest bid. We achieve this goal by enumerating the Pareto-optimal allocations among all bidders except i^* and then supplementing each of these allocations with items for bidder i^*.

LEMMA 3.5. *The Δ-perturbed maximizer can be computed in expected time $O((n\log m)^3(P/\Delta)^4 + \log^4 m)$ with P denoting $\max_{i\neq i^*} v_i(m)$.*

Proof. As a first step the algorithm computes the breakpoints for all bidders $i \neq i^*$. The number of breakpoints for each of these bidders is at most $b = (\lfloor \log m \rfloor + 2)(P/\Delta + 1)$. Each breakpoint can be computed by using a binary search. The running time for computing all breakpoints is thus $O((P/\Delta)\log^2 m)$.

Next the algorithm enumerates the Pareto-optimal allocations for the bidders in $[n] \setminus i^*$. This takes time $O(b^3n^3(P/\Delta))$ in expectation. The optimal solution including bidder i^* is composed of one of these solutions supplemented by a number of items for i^*. In particular, the algorithm needs to check only the maximum number of items from list $V_{i^*}^q$, for each $q \in \{0, \ldots, \lfloor \log m \rfloor + 1\}$, that can be added to each of the computed Pareto-optimal solutions without exceeding the total number of available items m. As there are $O(b^2n^2(P/\Delta))$ Pareto-optimal solutions in expectation and $q = O(\log m)$ lists each of which can be searched in time $O(\log m)$, this takes time $O(b^2n^2(P/\Delta)\log^2 m)$ in expectation.

As $b = (\lfloor \log m \rfloor + 2)(P/\Delta + 1)$, this gives an upper bound of $O((n\log m)^3(P/\Delta)^4 + \log^4 m)$ on the expected running time. $\square$

4 Consensus with drop-outs

Suppose there are two players each of which has access to a real number. In particular, let $a_1 \in \mathbb{R}$ be a number that is available to player 1 but not visible to player 2 and $a_2 \in \mathbb{R}$ be a number that is available to player 2 but not visible to player 1. In the setting that we are considering, it is known that $|a_1 - a_2| \leq 1$. Apart from these numbers, the two players are undistinguishable and we seek for a symmetric algorithm that allows them to agree on a common lower bound $b \leq \min\{a_1, a_2\}$ without allowing them to communicate about their numbers. Ideally, we would seek for a *consensus function* $\ell : \mathbb{R} \to \mathbb{R}$ satisfying

$$\forall a_1, a_2 \in \mathbb{R}, |a_1 - a_2| \leq 1 : \ell(a_1) = \ell(a_2) \leq \min\{a_1, a_2\} .$$

Unfortunately, however, such a consensus function does not exist: For the purpose of a contradiction, assume that the function does exist and let $\ell(x) = b$, for some $b < x$. This implies $\ell(y) = b$, for all $y \leq x$ by inductively considering the intervals $[x-1, x], [x-2, x-1], [x-3, x-2] \ldots$. Hence, for $y < b$, we obtain $\ell(y) = b > y$, which contradicts our assumption.

In order to solve the consensus problem, one needs to introduce randomization. Goldberg and Hartline [9, 10] present a consensus function failing with some bounded probability. However, using their approach in our mechanism for multi-unit auctions leads to infeasible allocations and, for this reason, destroys incentive compatibility even though the failure probability can be made arbitrary small.

In the following, we present a randomized consensus function in which a player can detect whether the consensus might fail. In this case the player "drops out". In our mechanism, a bidder corresponding to such a player receives the empty allocation. This way, we can ensure the feasibility of our mechanism.

The randomized variant of the consensus function has an additional parameter τ that is chosen uniformly at random from $[0,1]$. Both players use the same random number τ. The consensus function is defined as follows.

DEFINITION 4.1. *Let* $0 < \epsilon \leq 1$. *An* ϵ-drop-out consensus function $\ell : [0,1] \times \mathbb{R} \to \mathbb{R} \cup [\bot]$ *satisfies:*

1. *For every* $a \in \mathbb{R}$ *and* τ *chosen uniformly at random from* $[0,1]$, $\mathbf{Pr}\left[\ell(\tau,a) = \bot\right] = \epsilon$.
2. *For every* $a \in \mathbb{R}$ *and* $\tau \in [0,1]$ *with* $\ell(\tau,a) \neq \bot$, *it holds* $\ell(\tau,a) \leq a$.
3. *For every* $a_1 < a_2 \in \mathbb{R}$ *and every* $\tau \in [0,1]$ *with* $\ell(\tau,a_1) \neq \bot$ *and* $\ell(\tau,a_2) \neq \bot$, *it holds: If* $\ell(\tau,a_1) \neq \ell(\tau,a_2)$ *then* $a_1 < \ell(\tau,a_2) - 1$.

The additive gap *of* ℓ *is defined by* $\max\{a - \ell(\tau,a) \mid \tau \in [0,1], a \in \mathbb{R}, \ell(\tau,a) \neq \bot\}$.

More intuitively, the three properties can be summarized as follows: If one of the players obtains $\bot$ when evaluating ℓ, then this player *drops out.* Property 1 states that each of the two players drops out with probability ϵ. Property 2 states that if a player does not drop out then $\ell(\tau,a)$ is a lower bound on the player's number a. Property 3 states that the consensus is successful if none of the players drops out. In this case, either both players agree on the same lower bound $\ell(\tau,a_1) = \ell(\tau,a_2) \leq \min\{a_1,a_2\}$ or the smaller of the two numbers is *well separated* from the lower bound of the other one, i.e., $a_1 < \ell(\tau,a_2) - 1$. Observe that the latter statement can only be true if $a_2 > a_1 + 1$.

These properties are crucial for the correctness and the truthfulness of our mechanism. The running time of the mechanism depends exponentially on the gap of ℓ. Hence, we seek for a function ℓ whose gap is as small as possible.

LEMMA 4.1. *For every* $0 < \epsilon \leq 1$, *there exists an* ϵ-*drop-out consensus function* ℓ *with additive gap* $\frac{1}{\epsilon} - 1$.

Proof. We define a function that maps a given number $a \in \mathbb{R}$ to a multiple of $\frac{1}{\epsilon}$ perturbed by a suitable offset depending on the outcome of τ. In particular, for $\tau \in [0,1]$ and $k \in \mathbb{Z}$, define

$$x_\tau(k) = (k + \tau) \cdot \frac{1}{\epsilon}\,.$$

For fixed $\tau \in [0,1]$, the sequence $(x_\tau(k))_{k\in\mathbb{Z}}$ partitions the set of real numbers into intervals of length $\frac{1}{\epsilon}$, called τ-*intervals.* Let k be the largest integer such that $x_\tau(k) \leq a$. We define

$$\ell(\tau,a) \;=\; \begin{cases} x(\tau,k) & \text{if } x(\tau,k+1) - a > 1, \\ \bot & \text{if } x(\tau,k+1) - a \leq 1. \end{cases}$$

That is, a is mapped to the left boundary of the τ-interval in which it is contained; unless its number a has distance less than one to the right boundary of its τ-interval, in which case the player drops out. Obviously, the additive gap of ℓ corresponds to the length of the τ-intervals minus one and is thus $\frac{1}{\epsilon} - 1$.

We have to prove that ℓ satisfies the properties from Definition 4.2. First observe that $x(\tau,k+1)$ is a random number that is picked uniformly at random from the interval $(a, a + \frac{1}{\epsilon}]$, As a consequence,

$$\mathbf{Pr}\left[\ell(\tau,a) = \bot\right] = \mathbf{Pr}\left[x(\tau,k+1) \in (a,a+1]\right] = \epsilon\,,$$

which corresponds to the first property.

Now suppose $\ell(\tau,a) \neq \bot$. In this case $\ell(\tau,a) = x(\tau,k) \leq a$ which follows directly from the definition of $\ell(\tau,a)$ and $x(\tau,k)$. Hence, the second property holds as well.

Finally, we prove the third property. Fix $\tau \in \mathbb{R}$. Suppose $\ell(\tau,a_1) \neq \bot$, $\ell(\tau,a_2) \neq \bot$ and $\ell(\tau,a_1) \neq \ell(\tau,a_2)$, for some $a_1 < a_2 \in \mathbb{R}$. In this case, a_1 and a_2 lie in different τ-intervals. In particular, if $\ell(\tau,a_1) = x(\tau,k)$ then $\ell(\tau,a_2) \geq x(\tau,k+1)$. Now the definition of ℓ gives $x(\tau,k+1) - a_1 > 1$ and, hence, $\ell(\tau,a_2) - a_1 > 1$, which corresponds to the third property. ☐

In our mechanism, we use the following multiplicative variant of the drop-out consensus function. In this variant, the players have access to positive real numbers a_1 and a_2. The goal is to compute a lower bound $L > 0$ on these numbers such that the multiplicative gap $\max\{a_1/L, a_2/L\}$ is as small as possible, where the lower bound needs to hold only if the ratio $\max\{a_1/a_2, a_2/a_1\}$ is bounded from above by some given parameter $N > 1$.

DEFINITION 4.2. *Let* $0 < \epsilon \leq 1$ *and* $N > 1$. *An* (ϵ, N)-drop-out consensus function $L : [0,1] \times \mathbb{R}_{>0} \to \mathbb{R}_{>0} \cup [\bot]$ *satisfies:*

1. *For every* $a > 0$ *and* τ *chosen uniformly at random from* $[0,1]$, $\mathbf{Pr}\left[L(\tau,a) = \bot\right] = \epsilon$.
2. *For every* $a > 0$ *and* $\tau \in [0,1]$ *with* $L(\tau,a) \neq \bot$, *it holds* $L(\tau,a) \leq a$.
3. *For every* $a_1 > 0, a_2 > a_1$ *and every* $\tau \in [0,1]$ *with* $L(\tau,a_1) \neq \bot$ *and* $L(\tau,a_2) \neq \bot$, *it holds: If* $L(\tau,a_1) \neq L(\tau,a_2)$ *then* $a_1 < L(\tau,a_2)/N$.

The multiplicative gap *of* L *is defined by* $\max\{a/L(\tau,a) \mid \tau \in [0,1], a > 0, L(\tau,a) \neq \bot\}$.

A multiplicative (ϵ, N)-drop-out consensus function L for numbers $a_1, a_2 > 0$ can be derived from an additive ϵ-drop-out consensus function ℓ for numbers $a_1', a_2' \in \mathbb{R}$ by setting $a_1' = \log_N a_1$ and $a_2' = \log_N a_2$. If the additive gap of ℓ is $g > 0$ then the multiplicative gap of L is N^g. This gives

LEMMA 4.2. *For every $0 < \epsilon \le 1$ and $N > 1$, there exists an (ϵ, N)-drop-out consensus function L with multiplicative gap $N^{1/\epsilon-1}$.*

5 Description of the mechanism

Now we describe our randomized mechanism for multi-unit auctions. It yields a $(1-4\epsilon)$-approximation of the social welfare, in expectation, where $0 < \epsilon \le 1$ is a parameter.

The algorithm uses the following set of random numbers: τ is a number chosen uniformly at random from $[0, 1]$. This random number is used within all calls to the consensus function. For $1 \le i \le n$ and $0 \le j \le m$, let x_i^j denote a random number that is chosen uniformly at random from $[0, 1]$, too. The random numbers x_i^j are used within different calls to the Δ-perturbed maximizer. Let us stress that all calls to the Δ-perturbed maximizer use the same set of random variables $\{x_{i,j}\}$ but possibly different values of $\Delta > 0$. The random numbers τ and x_i^j are supposed to be stochastically independent.

Recall that we want to achieve a running time bound being polynomial in $\log m$. For this reason, the algorithm cannot even generate all random numbers x_i^j. Let us point out, however, that the algorithm needs to access x_i^j only when it accesses the valuation $v_i(j)$. Hence x_{ij} can be generated when $v_i(j)$ is accessed for the first time. In our running time analysis, we assume for simplicity that the random numbers x_{ij} are provided by a black box like the valuations $v_i(j)$.

Our mechanism for multi-unit auctions proceeds as follows. Let $N = 2(\log m + 3)n/\epsilon$. For every bidder, $i \in [n]$, the mechanism computes the number s_i of items allocated to bidder i in the following way.

- At first, the mechanism calls the (ϵ, N)-drop-out function L (as described in Definition 4.2) with parameter $v_{\max}^{(-i)} = \max_{j\in[n], j\neq i} v_j(m)$. In particular, let $L_i = L(\tau, v_{\max}^{(-i)})$.
- If $L_i = \perp$ then the algorithm sets $s_i = 0$.
- If $L_i \neq \perp$ then L_i is a lower bound on $v_{\max}^{(-i)}$ and, hence, on $v_{\max} = \max_{i\in[n]} v_i(m)$ as well. The algorithm calls the Δ_i-perturbed maximizer for $\Delta_i = L_i/N$. This call yields an allocation $s^{(i)} \in \{0, \ldots, m\}^n$. The mechanism sets $s_i = s_i^{(i)}$.

We described the mechanism in a way using up to n calls to the Δ-perturbed maximizer, one for each bidder. We claim, however, that only one or two calls are necessary since there are at most two different outcomes for the parameter Δ. To see this, let $i^* = \mathrm{argmax}_{i\in[n]} v_i(m)$ be a bidder making the highest bid. Observe that for all bidders $i \neq i^*$, $v_{\max}^{(-i)} = v_{\max}$. That is, the algorithm computes the same lower bound L_i, for all $i \neq i^*$. Suppose $L_i \neq \perp$. Then all these bidders derive the same parameter Δ_i and thus compute the same allocation $s^{(i)}$. Only bidder i^* might compute a different allocation $s^{(i^*)}$. We need to show that the combination of $s^{(i^*)}$ and $s^{(i)}$, for $i \neq i^*$, gives a feasible solution, that is, it holds

LEMMA 5.1. $\sum_{i=1}^n s_i \le m$.

Proof. First suppose that bidders i^* and $i \neq i^*$ do not drop out within the consensus function and both types of bidders agree on the same value $L_i = L_{i^*}$. In this case, the mechanism calls the Δ-perturbed maximizer with the same parameters for all bidders. As a consequence, $s^{(i)} = s^{(i^*)}$, for all $i \in [n]$, which implies $\sum_{i=1}^n s_i = \sum_{i=1}^n s_i^{(i^*)} \le m$ since $s^{(i^*)}$ is a feasible allocation.

Next suppose that at least one of the bidders drops out in the consensus. That is, $L_{i^*} = \perp$ or $L_i = \perp$, for all $i \neq i^*$. In this case, the algorithm calls the Δ-perturbed maximizer with the same parameters for all the bidders that do not drop out, which again ensures feasibility.

It remains proving feasibility for the case that none of the bidders drops out and $L_i \neq L_{i^*}$, for $i \neq i^*$. Observe that $L_i \neq L_{i^*}$ implies $L_i > L_{i^*}$ and thus $v_{\max}^{(-i)} > v_{\max}^{(-i^*)}$. Thus, Property 3 in Definition 4.2 of the consensus function gives $v_{\max}^{(-i^*)} < L_i/N = \Delta_i$. In words, the maximum bid among the bidders $i \neq i^*$ is upper-bounded by Δ_i. Now Lemma 3.2 yields that the Δ_i-perturbed maximizer sets $s_i = 0$, for all $i \neq i^*$, which implies $\sum_{i=1}^n s_i = s_{i^*} \le m$. $\square$

Next we investigate the approximation factor. Let *opt* denote the optimal welfare for the given valuations.

LEMMA 5.2. *The expected social welfare of the computed allocation is at least $(1 - 4\epsilon)opt$.*

Proof. All bidders $i \neq i^*$ use the same parameter L_i in the ϵ-drop-out consensus function. If one of them drops out, then all of them drop out. Property 1 in Definition 4.2 shows that the probability that one of them drops out is ϵ. The probability that bidder i^* drops out is ϵ, too. Hence, with probability $1-2\epsilon$ none of the bidders drops out.

In the following, we prove that the approximation ratio is at least $1-2\epsilon$ if none of the bidders drops out, which altogether gives an approximation factor of at least $(1-2\epsilon)^2 \ge (1-4\epsilon)$ and, thus, proves the lemma.

Let us first analyze the social welfare of the allocation $s^{(i^*)}$ that is calculated by the perturbed maximizer of bidder i^*. By Lemma 3.1, allocation $s^{(i^*)}$ approximates *opt* up to an additive error of at most

$$(2\log m + 3)n\Delta_{i^*} = (2\log m + 3)n\,\frac{L_{i^*}}{N}\,.$$

As $N = 2(\log m + 3)n/\epsilon$ and $L_{i^*} \leq v_{\max} \leq opt$, the additive error is at most $\epsilon L_{i^*} \leq \epsilon\, opt$. Consequently, $v(s^{(i^*)}) \geq (1-\epsilon)opt$.

Suppose none of the bidders drops out and $L_i = L_i^*$, for $i \neq i^*$. In this case, the mechanism uses the same set of parameters L_{i^*} and $\Delta_{i^*} = L_{i^*}/N$ for all bidders. In particular, the allocation s computed by the mechanism corresponds to the allocation $s^{(i^*)}$ selected by the perturbed mechanism of bidder i^*. In this case, the mechanism achieves thus a social welfare of at least $(1-\epsilon)opt$.

Now suppose none of the bidders drops out and $L_i \neq L_i^*$. We have studied this case already in the proof of Lemma 5.1 where we have shown that

a) the mechanism sets $s_{i^*} = s_{i^*}^{(i^*)}$ and $s_i = 0$, for $i \neq i^*$; and

b) it holds $v_{\max}^{(-i^*)} < L_i/N$.

Thus, the mechanism achieves social welfare

$$v(s^{(i^*)}) - \sum_{i\neq i^*} v_i(s_i^{(i^*)}) \geq (1-\epsilon)opt - \sum_{i\neq i^*} v_{\max}^{(-i^*)}\,.$$

Now

$$\sum_{i\neq i^*} v_{\max}^{(-i^*)} < (n-1)\,\frac{L_i}{N} \leq \epsilon\, opt$$

since $L_i \leq v_{\max} \leq opt$ and $N \geq n/\epsilon$. Thus, if none of the bidders drops out, the social welfare of the computed solution is at least $(1-2\epsilon)opt$, which completes the proof of the lemma. □

We combine the algorithm with the subjective variant of VCG payments as described in Section 2.2.

Lemma 5.3. *The described mechanism is universally truthful.*

Proof. Fix the random variables τ and x_i^j. The outcome of the consensus function $L(\tau, v_{\max}^{(-i)})$ does not depend on the valuation of bidder i and, hence, cannot be influenced by the bidder. Bidders with $L(\tau, v_{\max}^{(-i)}) = \perp$ receive the empty allocation and, hence, have no incentive to lie. Thus, we can focus on the bidders with $L(\tau, v_{\max}^{(-i)}) \neq \perp$.

The allocation for a bidder i with $L(\tau, v_{\max}^{(-i)}) \neq \perp$ is selected by the Δ_i-perturbed maximizer. The parameter Δ_i is a function of $v_{\max}^{(-i)}$ and does not depend on v_i. In particular, the Δ_i-perturbed maximizer satisfies the conditions described in Definition 2.1 when setting $g_s^{(i)}(v_{-i}) = (2q(s_i) + x_i^i)\Delta_i + \sum_{j\neq i}(v_i(s_j) + (2q(s_j) + x_i^j)\Delta_i)$. As a consequence, Proposition 2.1 yields that the composition of these affine maximizers together with the subjective variant of VCG payments is truthful. □

The running time of the bidder is dominated by the time needed for computing the perturbed maximizer, once with parameter Δ_{i^*} and once with parameter Δ_i, for $i \neq i^*$. Lemma 3.5 gives the expected running time in terms of this parameter, it is $O((n\log m)^3(P/\Delta)^4 + \log^4 m)$ with $P = v_{\max}^{(-i^*)}$ denoting the second largest bid. As $\Delta_{i^*} \leq \Delta_i$, we only need to consider the case $\Delta = \Delta_{i^*}$.

We use the multiplicative gap of the consensus function given in Lemma 4.2 in order to upper-bound P. It holds

$$P = v_{\max}^{(-i^*)} \leq N^{1/\epsilon - 1} L_{i^*} = N^{1/\epsilon}\Delta_{i^*}\,.$$

Thus, $P/\Delta_{i^*} \leq N^{1/\epsilon}$. This gives an expected running time of $O((n\log m)^3 N^{4/\epsilon})$ with $N = O(n\log m/\epsilon)$. Combining this running time bound with the other results in this section gives

Theorem 5.1. *There exists a randomized polynomial time approximation scheme for multi-unit auctions that is universally truthful.*

6 Deterministic running time bound

Finally, let us investigate how the bound on the expected running time can be turned into a deterministic running time bound. That is, only the bound on the approximation guarantee is of stochastic nature. Without the aspect of truthfulness this would be completely obvious, as one could simply stop the algorithm when it exceeds the bound on the expected running time by a factor of $1/\epsilon$, for $0 < \epsilon < 1$, and assign the empty allocation to all bidders. By the Markov inequality this would happen only with probability ϵ and, hence, decrease the approximation factor only by $1-\epsilon$. When using this approach, however, bidders might have an incentive to lie in order to improve the running time and, this way, their utility.

Instead, for every bidder i, we define a criterion C_i that is independent of v_i and is used to decide whether s_i should be set to 0 since the running time bound of the algorithm cannot be guaranteed. The criterion C_i is defined as follows: The mechanism calculates the set of Pareto-optimal solutions with respect to the Δ_i-perturbed maximizer over all bidders except bidder i and the bidder i' with largest bid among the other bidders, following the algorithm described in Section 3. If the running time of this computation exceeds the bound on the expected running time by n/ϵ, the mechanism sets $s_i = 0$. Otherwise, the mechanism computes the solution for this bidder using the Δ_i-perturbed maximizer.

This mechanism is universally truthful as the criterion C_i for bidder i does not depend on v_i. Our approach decreases the approximation factor at most by a factor of $1 - \epsilon$ since the probability that the calculation for at least one of the bidders needs to be stopped is at most ϵ.

Finally, we show that C_i guarantees a polynomial running time bound for the Δ_i-perturbed maximizer. Either i or i' corresponds to the bidder with maximum bid, called i^*. The algorithm for the Δ_i-perturbed maximizer, as described in Section 3, computes the Pareto-optimal allocations over the bidders in $[n] \setminus \{i^*\}$ and, hence, for one more bidder than in the test for criterion C_i. Note, however, that one additional bidder can increase the number of Pareto-optimal solutions only by a factor of b, where b is the polynomially bounded number of breakpoints of the bidder. Thus, the criterion guarantees a deterministic polynomial bound on the running time of the algorithm, which gives

THEOREM 6.1. *There exists a randomized approximation scheme for multi-unit auctions that is universally truthful and admits a deterministic polynomial bound on the running time.*

7 Acknowledgments

I would like to thank Tim Roughgarden for bringing [9, 10] to my attention and an anonymous referee for enlightening comments on Definition 2.1.

References

[1] René Beier, Heiko Röglin, and Berthold Vöcking. The smoothed number of Pareto optimal solutions in bicriteria integer optimization. In *Proc. of the 12th International Integer Programming and Combinatorial Optimization Conference (IPCO)*, pages 53–67, 2007.

[2] René Beier and Berthold Vöcking. Random knapsack in expected polynomial time. *Journal of Computer and System Sciences (JCSS)*, 69(3), 2004.

[3] Patrick Briest, Piotr Krysta, and Berthold Vöcking. Approximation techniques for utilitarian mechanism design. In *Proc. of the 37th ACM Symp. on Theory of Computing (STOC)*, pages 39–48, 2005.

[4] Shahar Dobzinski and Shaddin Dughmi. On the power of randomization in algorithmic mechanism design. In *Proc. of the 50th IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 505–514, 2009.

[5] Shahar Dobzinski and Noam Nisan. Mechanisms for multi-unit auctions. In *Proc. of the 8th ACM Conference on Electronic Commerce (EC)*, 2007.

[6] Shahar Dobzinski and Noam Nisan. Multi-unit auctions: beyond Roberts. In *Proc. of the 12th ACM Conference on Electronic Commerce (EC)*, pages 233–242, 2011.

[7] Shahar Dobzinski, Noam Nisan, and Michael Schapira. Truthful randomized mechanisms for combinatorial auctions. In *Proc. of the 38th ACM Symp. on Theory of Computing (STOC)*, pages 644–652, 2006.

[8] Shaddin Dughmi and Tim Roughgarden. Black-box randomized reductions in algorithmic mechanism design. In *Proc. of the 51st IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 775–784, 2010.

[9] Andrew V. Goldberg and Jason D. Hartline. Competitiveness via consensus. In *Proc. of the 14th ACM Symp. on Discrete Algorithms (SODA)*, pages 215–222, 2003.

[10] Andrew V. Goldberg and Jason D. Hartline. Collusion-resistant mechanisms for single-parameter agents. In *SODA*, pages 620–629, 2005.

[11] Oscar H. Ibarra and Chul E. Kim. Fast approximation algorithms for the knapsack and sum of subset problems. *Journal of the ACM*, 22(4):463–468, 1975.

[12] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems.* Springer, 2004.

[13] Ron Lavi and Chaitanya Swamy. Truthful and near-optimal mechanism design via linear programming. In *Proc. of the 46th IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 595–604, 2005.

[14] Ahuva Mu'alem and Noam Nisan. Truthful approximation mechanisms for restricted combinatorial auctions. In *Proceedings of the 18th National Conference on Artificial Intelligence and 14th Conference on Innovative Applications of Artificial Intelligence (AAAI/IAAI)*, pages 379–384, 2002.

[15] G. Nemhauser and Z. Ullmann. Discrete dynamic programming and capital allocation. *Management Science*, 15(9):494–505, 1969.

[16] Noam Nisan. Introduction to mechanism design (for computer scientists). In Noam Nisan, Tim Roughgarden, Éva Tardos, and Vijay V. Vazirani, editors, *Algorithmic Game Theory*, pages 209–241. Cambridge University Press, 2007.

[17] Noam Nisan and Amir Ronen. Algorithmic mechanism design (extended abstract). In *Proc. of the 31st ACM Symp. on Theory of Computing (STOC)*, pages 129–140, 1999.

[18] Kevin Roberts. The characterization of implementable choice rules. In *In Jean-Jacques Laffont (Ed.), Aggregation and Revelation of Preferences. Papers presented at the first European Summer Workshop of the Economic Society*, pages 321–349, North-Holland, 1979.