On the Power of Randomization in Algorithmic Mechanism Design

Shahar Dobzinski∗
School of Computer Science and Engineering
Hebrew University of Jerusalem
shahard@cs.huji.ac.il

Shaddin Dughmi †
Department of Computer Science
Stanford University
shaddin@cs.stanford.edu

Abstract

In many settings the power of truthful mechanisms is severely bounded. In this paper we use randomization to overcome this problem. In particular, we construct an FPTAS for multi-unit auctions that is truthful in expectation, whereas there is evidence that no polynomial-time truthful deterministic mechanism provides an approximation ratio better than 2.

We also show for the first time that truthful in expectation polynomial-time mechanisms are provably stronger than polynomial-time universally truthful mechanisms. Specifically, we show that there is a setting in which: (1) there is a non-polynomial time truthful mechanism that always outputs the optimal solution, and that (2) no universally truthful randomized mechanism can provide an approximation ratio better than 2 in polynomial time, but (3) an FPTAS that is truthful in expectation exists.

∗Supported by the Adams Fellowship Program of the Israel Academy of Sciences and Humanities, and by a grant from the Israeli Academy of Sciences.
†Supported in part by NSF grant CCF-0448664.
1 Introduction

Background and Our Results

The last few years have been quite disappointing for researchers in Algorithmic Mechanism Design. Indeed, much progress has been made, but in the “wrong” direction: several papers proved that the power of polynomial time truthful mechanisms is severely bounded, either comparing to polynomial time non-truthful algorithms, or to non-polynomial time truthful mechanisms [13, 7, 8, 10, 21, 16]. This paper brings some good news: randomization might help in breaking these lower bounds to achieve better approximation guarantees.

Before studying the power of randomization in the context of mechanism design, we recall the three common notions of truthfulness:

- **Deterministic Truthfulness**: A bidder always maximizes his utility by bidding truthfully. No randomization is allowed.

- **Universal Truthfulness**: A universally-truthful mechanism is a probability distribution over deterministic truthful mechanisms. The mechanism is truthful even when the realization of the random coins is known.

- **Truthfulness in Expectation**: A mechanism is truthful in expectation if a bidder always maximizes his expected profit by bidding truthfully. The expectation is taken over the internal random coins of the mechanism.

It is well known that, in some settings (e.g. online), randomized algorithms are strictly more powerful than deterministic algorithms. In their seminal paper introducing algorithmic mechanism design, Nisan and Ronen [19] showed that universally truthful mechanisms can be more powerful than their deterministic counterparts. In this paper we show that truthfulness in expectation yields more power than universal truthfulness.

For the well-studied problem of multi-unit auctions (e.g., [17, 4, 12, 20, 14, 8, 3]), we show a polynomial time truthful in expectation mechanism with the best ratio possible from a pure algorithmic point of view:

**Theorem**: There exists a truthful in expectation FPTAS for multi-unit auctions\(^1\).

There are many truthful in expectation mechanisms in the literature [2, 1, 14, 5]. However, somewhat surprisingly, many truthful in expectation mechanisms were followed by universally truthful mechanisms with the same performance [4, 6, 9, 8]. As a result, there are few settings in which the best known truthful in expectation algorithms provide better approximation ratios than the best known universally truthful mechanisms. Furthermore, [15] shows that for digital good auctions the two notions of randomization are equivalent in power. In contrast, our second main result shows that truthful in expectation mechanisms are strictly more powerful than universally truthful ones. We study a variant of multi-unit auctions that we term **restricted multi-unit auctions**, and show a first-of-a-kind separation result:

**Theorem**: If \( \mathcal{A} \) is a universally truthful mechanism for restricted multi-unit auctions that achieves an approximation ratio of \( 2 - \epsilon \) for some constant \( \epsilon > 0 \), then \( \mathcal{A} \) has exponential communication complexity. However, there exists a truthful in expectation FPTAS for restricted multi-unit auctions.

We note that there exists a deterministic truthful mechanism that optimally solves restricted multi-unit auctions in exponential time.

---

\(^1\) An FPTAS is a \((1 + \epsilon)\)-approximation algorithm with running time that is also polynomial in \( \frac{1}{\epsilon} \).
Multi-unit Auctions

In a multi-unit auction a set of $m$ identical items is to be allocated to $n$ bidders. Each bidder $i$ has a valuation function $v_i : [m] \rightarrow \mathbb{R}^+$, where $v_i$ is non-decreasing, and normalized: $v_i(0) = 0$. The goal is the usual one of finding an allocation of the items $(s_1, \ldots, s_n)$ that maximizes the social welfare: $\Sigma_i v_i(s_i)$. All items are identical, so algorithms should run in time polynomial in the number of bits needed to represent the number $m$, and the number of bidders: $\text{poly}(n, \log m)$.

It is not hard to see that in the “single minded” case, where each $v_i$ is a single-step function (i.e., for some $s^*, k > 0$, $v_i(s) = k$ if $s \geq s^*$ and $v_i(s) = 0$ otherwise), the problem is just a reformulation of the NP-hard Knapsack problem. The standard FPTAS for knapsack generalizes easily to multi-unit auctions.

The study of truthfulness in multi-unit auctions has a long history, starting with Vickrey’s 1961 paper [22]. The VCG mechanism is truthful and solves the problem optimally, but is not computationally-efficient (see, e.g., [18]). For the single-minded case, Mu’alem and Nisan provided a truthful polynomial-time 2-approximation algorithm, followed by an FPTAS by Briest et al [4]. In the general case, a truthful in expectation 2-approximation mechanism was presented by [14], followed by a deterministic 2-approximation mechanism in [8]. Can polynomial-time truthful mechanisms guarantee an approximation ratio better than 2? This is one of the major open questions in algorithmic mechanism design, and only a partial answer is known: a deterministic truthful mechanism with an approximation ratio better than 2 that always allocates all items must run in exponential time$^2$ [13, 8]. As mentioned before, this paper provides an FPTAS for multi-unit auctions that is truthful in expectation.

A word is in order on how the valuations are accessed. The valuation functions are objects of size $m$, whereas we are interested in algorithms that run in time polynomial in $\log m$ (the running time of the non-truthful FPTAS for multi-unit auctions). Hence, we assume that each valuation $v$ is given by a black box. For our upper bounds, the black box corresponding to $v$ needs to answer only the weak “value queries”: given $s$, what is the value of $v(s)$. Our lower bound for the power of universally truthful mechanisms assumes a black box that can answer any query based on $v$ (the “communication model”).

Main Result I: A Truthful FPTAS for Multi-unit Auctions

We now give a short description of our truthful FPTAS for multi-unit auctions. The only technique known for designing truthful deterministic mechanisms for “rich” problems like multi-unit auctions is by designing maximal-in-range (henceforth MIR) algorithms. An algorithm $A$ is called maximal-in-range if there is a set of allocations $\mathcal{R}$ (the “range”) that does not depend on the input, such that $A$ always outputs the allocation in $\mathcal{R}$ that maximizes the welfare: $A(v_1, \ldots, v_n) = \arg \max_{(s_1, \ldots, s_n) \in \mathcal{R}} \Sigma_i v_i(s_i)$. Using the VCG payment scheme together with an MIR mechanism results in a truthful mechanism. See [7] for a more formal discussion.

Therefore, one way to obtain truthful mechanisms is to identify a range that, on one hand, is “rich” enough to provide a good approximation ratio, and, on the other hand, is “simple” enough so that exact optimization over this range is computationally feasible. For multi-unit auctions, there is an MIR mechanism that provides an approximation ratio of 2 in polynomial time, but unfortunately no MIR algorithm can provide a better ratio in polynomial time [8].

In this paper we let the range $\mathcal{R}$ consist of distributions over allocations. An algorithm that always selects the allocation that maximizes the expected welfare and uses VCG payments is truthful in expectation. We term these mechanisms maximal in distributional range (MIDR).

Before discussing our truthful FPTAS, recall the spirit of the standard non-truthful FPTAS for multi-unit auctions: each valuation is simplified by rounding down each value to the nearest power of $(1 + \epsilon)$. Then, the best solution that uses only the “breakpoints” of the rounded valuations is found. This solution has a value close to the optimal unrestricted solution. However, to guarantee truthfulness via a maximal-

$^2$Indeed, it can be assumed that a non-truthful algorithm always allocates all items without loss of generality from a pure algorithmic perspective. However, this assumption has game-theoretic implications.
in-range algorithm we must find the solution with the optimal welfare in the range. Thus, we give a “weight” \( w_s \) to each allocation \( s \): when \( s \) is selected by the algorithm, the output will be \( s \) with probability \( w_s \), and with probability \( 1 - w_s \) no bidder will receive any items, effectively reducing the expected welfare of the allocation by a factor of \( w_s \). The weight of a less “structured” allocation is smaller than the weight of a more “structured” one. We show that in the optimal solution among these “weighted allocations” each bidder is assigned a bundle that is a breakpoint, or very close to a breakpoint of his valuation. Thus the optimal solution can be found efficiently using exhaustive search, when the number of bidders is a constant, since the number of breakpoints of each bidder is polynomial in the number of bits.

Efficiently handling any number of bidders is more involved. Given two bidders we define a “meta-bidder” by merging their valuations: the value of the meta bidder for \( s \) items is equal to the value of the optimal solution that allocates \( s \) items between the two bidders, using the weighted allocations mentioned above. We recursively define new meta-bidders given the previous ones, until we are left with only two meta bidders. Now the optimal solution can be found efficiently.

We note that MIDR mechanisms were used in [14], although only implicitly. The beautiful construction of [14] is general and applies to many settings. However, its strength and weakness is its generality: for some settings, a specifically-tailored mechanism might have more power than a mechanism obtained from the general construction of [14]. Indeed, for some of the most important settings discussed in [14] better constructions have been found [9, 6, 8]. Moreover, the LP-based techniques of [14] cannot guarantee an approximation ratio better than the integrality gap for multi-unit auctions, which is 2 (see [14]). The techniques in this paper may help in designing better mechanisms in settings where [14] performs poorly, like combinatorial auctions with submodular bidders (see below).

**Main Result II: Truthful in Expectation Mechanisms are more Powerful**

Next we prove that truthful in expectation mechanisms are strictly more powerful than universally truthful ones. Ideally, we would like to prove that universally truthful mechanisms for multi-unit auctions cannot provide an approximation ratio better than 2 in polynomial time. However, this remains open question even for deterministic mechanisms. Hence, we study a close variant of multi-unit auctions, called restricted multi-unit auctions. The FPTAS for multi-unit auctions extends almost immediately to restricted multi-unit auctions. It is more involved to show that universally truthful polynomial time mechanisms cannot provide an approximation ratio better than 2.

Roughly speaking, we first show that a polynomial-time universally truthful mechanism with an approximation ratio of \( \alpha \) must yield a polynomial-time deterministic truthful mechanism with an approximation ratio of \( \alpha \) on “many” instances. We then show that this deterministic mechanism must be an affine maximizer (a slight generalization of MIR algorithms). Finally, we prove that deterministic affine maximizers cannot provide an approximation ratio better than 2 for restricted multi-unit auctions in polynomial time for “many” instances, hence no polynomial-time universally truthful mechanism for restricted multi-unit auctions with an approximation ratio better than 2 exists.

**Open Questions**

The obvious open question is whether the techniques of this paper can be extended to improve the approximation guarantees of other questions in algorithmic mechanism design, such as variants of combinatorial auctions, or combinatorial public projects [21]. In particular:

**Main Open Question:** Is there a truthful mechanism for combinatorial auctions with subadditive (or submodular) bidders that provides a constant approximation ratio?

There are non-truthful algorithms with constant approximation ratios for combinatorial auctions with subadditive bidders[11, 23]. However, the best known polynomial-time truthful mechanisms provide a ratio that is no better than logarithmic [6, 9]. It seems that our current techniques reached a dead end: random sampling methods do not seem amenable to an approximation factor that is better than...
logarithmic, deterministic MIR mechanisms cannot provide an approximation ratio better than \(m^{1/3}\) in polynomial time [7], and the techniques of [14] are not applicable to this setting. The methods of this paper might help in constructing truthful in expectation mechanisms. Alternatively, lower bounds on the power of MIDR mechanisms would also be extremely interesting.

2 Preliminaries

2.1 The Setting

In a multi-unit auction there is a set of \(m\) identical items, and a set \(N = \{1, 2, \ldots, n\}\) of bidders. Each bidder \(i\) has a valuation function \(v_i : [m] \to \mathbb{R}^+\), which is normalized \((v_i(0) = 0)\) and non-decreasing. Denote by \(V\) the set of possible valuations. An allocation of the items \(\vec{s} = (s_1, \ldots, s_n)\) to \(N\) is a vector of non-negative integers with \(\sum s_i \leq m\). Denote the set of allocations by \(S\). The goal is to find an allocation that maximizes the welfare: \(\sum v_i(s_i)\).

The valuations are given to us as black boxes. For algorithms, the black box \(v\) will only answer the weak value queries: given \(s\), what is the value of \(v(s)\). For the impossibility result, we assume that the black box \(v\) can answer any query that is based on \(v\) (the “communication model”). Our algorithms run in time \(\text{poly}(n, \log m)\), while our impossibility result gives a lower bound on the number of bits transferred, and holds even if the mechanism is computationally unbounded.

2.2 Truthfulness

An \(n\)-bidder mechanism for multi-unit auctions is a pair \((f, p)\) where \(f : V^n \to S\) and \(p = (p_1, \cdots, p_n)\), where \(p_i : V^n \to \mathbb{R}\). \((f, p)\) might be either randomized or deterministic.

**Definition 2.1** Let \((f, p)\) be a deterministic mechanism. \((f, p)\) is truthful if for all \(i\), all \(v_i, v'_i\) and all \(v_{-i}\) we have that \(v_i(f(v_i, v_{-i}), i) - p_i(v_i, v_{-i}) \geq v'_i(f(v'_i, v_{-i}), i) - p_i(v'_i, v_{-i})\).

**Definition 2.2** \((f, p)\) is universally truthful if it is a probability distribution over truthful deterministic mechanisms.

**Definition 2.3** \((f, p)\) is truthful in expectation if for all \(i\), all \(v_i, v'_i\) and all \(v_{-i}\) we have that \(E[v_i(f(v_i, v_{-i}), i) - p_i(v_i, v_{-i})] \geq E[v'_i(f(v'_i, v_{-i}), i) - p_i(v'_i, v_{-i})]\), where the expectation is over the internal random coins of the algorithm.

2.3 Maximal in Range, Maximal in Distributional range, and Affine Maximizers

**Definition 2.4** \(f\) is an affine maximizer if there exist a set of allocations \(\mathcal{R}\), a constant \(\alpha_i \geq 0\) for \(i \in N\), and a constant \(\beta_{\vec{s}} \in \mathbb{R}\) for each \(\vec{s} \in S\), such that \(f(v_1, \ldots, v_n) \in \arg\max_{\vec{s} = (s_1, \ldots, s_n) \in \mathcal{R}} (\Sigma_i (\alpha_i v_i(s_i)) + \beta_{\vec{s}})\). \(f\) is called maximal-in-range (MIR) if \(\alpha_i = 1\) for \(i \in N\), and \(\beta_{\vec{s}} = 0\) for each \(\vec{s} \in \mathcal{R}\).

**Definition 2.5** \(f\) is a distributional affine maximizer if there exist a set of distributions over allocations \(\mathcal{D}\), a constant \(\alpha_i \geq 0\) for \(i \in N\), and a constant \(\beta_D \in \mathbb{R}\) for each \(D \in \mathcal{D}\) such that: \(f(v_1, \ldots, v_n) \in \arg\max_{D \in \mathcal{D}} (E_{\vec{s} \sim D} [\Sigma_i (\alpha_i v_i(s_i)) + \beta_D])\). We say \(f\) is maximal in distributional range (MIDR) if \(\alpha_i = 1\) for all \(i\) in \(N\), and \(\beta_D = 0\) for each \(D \in \mathcal{D}\).

The following proposition is standard:

**Proposition 2.6** The following statements are true:

1. Let \(f\) be an affine maximizer (in particular, maximal in range). There are payments \(p\) such that \((f, p)\) is a truthful mechanism.

2. Let \(f\) be a distributional affine maximizer (in particular, maximal in distributional range). There are payments \(p\) such that \((f, p)\) is a truthful in expectation mechanism.
3 A Truthful FPTAS for Multi-Unit Auctions

This section provides a \((1 + \epsilon)\)-approximation algorithm for multi-unit auctions that is truthful in expectation by providing a maximal-in-distributional-range mechanism. The running time is polynomial in \(n\), \(\log m\), and \(\frac{1}{\epsilon}\). The construction has several stages. We start by defining a certain family of ranges, structured ranges. The main property of this family is that finding the optimal solution in a structured range is computationally feasible when the number of bidders is fixed. To handle any number of bidders, we use a more involved tree-like construction, where structured ranges serve as the basic building block. We start by defining structured ranges.

3.1 Structured Ranges

3.1.1 Definition

Towards constructing an FPTAS, fix the approximation parameter \(\epsilon\) such that \(0 < \epsilon < \frac{1}{2}\). It will be useful to let \(\delta = \frac{\ln \frac{1}{1 - \epsilon}}{2 \log m + 2}\). We construct ranges that consist only of weighted allocations:

**Definition 3.1** A weighted allocation \(w \cdot \vec{s}\) is a distribution over allocations of the following form: with some probability \(w\) the allocation \(\vec{s} = (s_1, \ldots, s_n)\) is chosen, and with probability \(1 - w\) the empty allocation (where each bidder is allocated an empty set) is chosen. \(w\) is called the weight of the weighted allocation.

The weight of a weighted allocation of \(m\) items to \(n\) bidders is determined using the following function \(\text{weight}_\epsilon: S \rightarrow [1 - \epsilon, 1]\), where \(S\) is the set of all allocations of \(m\) items to \(n\) bidders. Let \(t\) be the maximal (non-negative) integer such that for each \(i\), \(s_i\) is a multiple of \(2^t\). Let \(p = (1 + 2\delta)\) if \(s_i \neq 0\) for \(n - 1\) bidders, let \(p = 1\) otherwise. Now define \(\text{weight}_\epsilon(\vec{s}) = (1 - \epsilon)(1 + 2\delta)^t p\). Observe that the number of different outputs of weight \(\epsilon\) is \(O(\log m)\), where increasing \(t\) by 1 makes the output value increase by a factor of \((1 + 2\delta)\).

A structured range consists only of weighted allocations whose weight is determined by \(\text{weight}_\epsilon\). Informally, in a structured range “structurally simpler” allocations have greater weight.

**Definition 3.2** An \(\epsilon\)-structured range for \(n\) bidders and \(m\) items is the following set of weighted allocations: \(\{\text{weight}_\epsilon(\vec{s}) \cdot \vec{s}\}_{\vec{s} \in S}\), where \(S\) is the set of allocations for these \(n\) bidders. We sometimes call an \(\epsilon\)-structured range a structured range when \(\epsilon\) is clear from context.

The optimal solution in a structured range provides a good approximation to the (unweighted) optimal solution: every allocation is in the range with weight at least \((1 - \epsilon)\). In particular, the optimal (unweighted) allocation is in the range with a weight of at least \((1 - \epsilon)\).

**Lemma 3.3** The expected approximation ratio of an algorithm that optimizes over an \(\epsilon\)-structured range is at least \((1 - \epsilon)\).

3.1.2 The Optimal Solution is in Neighborhoods of Breakpoints

The important property of structured ranges is that finding an optimal weighted allocation is computationally easy. We show that the optimal solution consists of bundles that are near breakpoints, where breakpoints are bundles at which the valuation first exceeds a power of \(1 + \delta\).

**Definition 3.4** A bundle \(s\) is called a \(b\)-breakpoint of a valuation \(v\) if it is the smallest multiple of \(b\) s.t. \(v(s) \geq (1 + \delta)^l v(b)\), for some integer \(l \geq 0\). The bundle 0 is also considered a \(b\)-breakpoint of \(v\), for all \(b\).

**Definition 3.5** Let \(s\) be a \(b\)-breakpoint of \(v\). The neighborhood of \(s\) consists of the bundles \(s, s + b, s + 2b, \ldots, s + n \cdot b\).
Lemma 3.6 Let weight\(_w(\bar{a}) \cdot \bar{o}\) be a weighted allocation that maximizes the welfare in a structured range among all allocations with \(\Sigma_i s_i \leq s\), for some \(m \geq s \geq 0\). Let \(t\) be the maximal non-negative integer s.t. each \(o_i\) is a multiple of \(2^t\). Then, each \(o_i\) is in the neighborhood of a \(2^t\)-breakpoint.

Proof: Suppose towards a contradiction that there is some bidder \(k\), where \(o_k\) is not in the neighborhood of a \(2^t\)-breakpoint. Round down each \(o_i\) (that is not a \(2^t\)-breakpoint) to the nearest \(2^t\)-breakpoint of \(v_i\) denoted \(\overline{a}_i\). Let \(c_i\) be such that \(\overline{a}_i = c_i \cdot 2^t\). For each \(i\), define \(c'_i\) as follows:

\[
c'_i = \begin{cases} 
c_i + 1, & c_i \text{ is odd;} 
c_i, & c_i \text{ is even.}
\end{cases}
\]

For each \(i\), let \(a_i = c'_i \cdot 2^t\). We will show that the expected welfare of weight\(_w(\bar{a}) \cdot \bar{\sigma}\) is greater than that of the optimal solution weight\(_w(\bar{a}) \cdot \bar{\sigma}\). Note that \(\Sigma_i a_i \leq s\): on one hand by our assumption, \(o_k - \overline{a}_k \geq n \cdot 2^t\). On the other hand, \(\Sigma_i (a_i - \overline{a}_i) \leq n \cdot 2^t\), since for each \(i\), \(a_i - \overline{a}_i = 2^t\) if \(\overline{a}_i\) is odd, else \(a_i - \overline{a}_i = 0\). (In a sense, we “take” items from bidder \(k\) and spread them among the other bidders.)

We now calculate the expected welfare of \(\bar{a}\). All \(c'_i\)’s are even, thus each \(a_i\) is a multiple of \(2^{t+1}\). In addition, the number of bidders that receive no items is larger in \(\bar{a}\). Hence weight\(_w(\bar{a}) \geq (1 + 2\delta)\) weight\(_w(\bar{\sigma})\), by the definition of weight\(_w\). Since for each \(i\), \(a_i \geq \overline{a}_i\), the monotonicity of the valuations implies that:

\[
\text{weight}_w(\bar{a}) \cdot \Sigma_i v_i(a_i) \geq (1 + 2\delta) \text{ weight}_w(\bar{\sigma}) \cdot \Sigma_i v_i(a_i)
\]

\[
\geq (1 + 2\delta) \text{ weight}_w(\bar{\sigma}) \cdot \Sigma_i v_i(\overline{a}_i)
\]

Using the definition of a breakpoint, the value of each \(v_i(o_i)\) is close to the value of \(v_i(\overline{a}_i)\):

\[
(1 + 2\delta) \text{ weight}_w(\bar{\sigma}) \cdot \Sigma_i v_i(\overline{a}_i) \geq (1 + 2\delta) \text{ weight}_w(\bar{\sigma}) \cdot \Sigma_i v_i(\overline{a}_i)
\]

\[
> \text{ weight}_w(\bar{\sigma}) \cdot \Sigma_i v_i(o_i)
\]

I.e., the expected welfare of the distribution weight\(_w(\bar{a}) \cdot \bar{a}\) is greater than that of the optimal one: weight\(_w(\bar{a}) \cdot \bar{\sigma}\). A contradiction.

\[\square\]

3.2 Warmup: An FPTAS for a Fixed Number of Bidders

As a warm-up we provide an MIDR \((1+\epsilon)\)-approximation algorithm for a fixed number of bidders (Figure 1). The idea is to fully optimize over a structured range. Specifically, the number of value queries the algorithm makes is poly \(n, \log m, \frac{1}{\delta}, \log \max_i \frac{v_i(m)}{v_i(1)}\) = poly \(n, \log m, \frac{1}{\delta}, \log \max_i \frac{v_i(m)}{v_i(1)}\). This algorithm has two drawbacks. The first major one is that while the number of value queries is small, the running time of the algorithm is exponential in \(n\): exhaustive search is used to find the best allocation that consists only of bundles in the neighborhood of breakpoints. However, for a fixed number of bidders the exhaustive search (and hence the algorithm) is also computationally efficient\(^3\).

The second drawback is that the running time and the number of value queries depends (polynomially) on \(\log \max_i \frac{v_i(m)}{v_i(1)}\) (in a sense, the algorithm is only “weakly polynomial”). This can be fixed (using “significant breakpoints” – see the next section), but to keep the presentation of this warmup simple we do not address this here. We stress that the FPTAS for a general number of bidders we present in the next subsection is “strongly polynomial”. I.e., runs in time poly \(n, \log m, \frac{1}{\epsilon}\).

Lemma 3.7 The number of value queries the algorithm makes is poly \(\log m, n, \frac{1}{\epsilon}, \log \max_i \frac{v_i(m)}{v_i(1)}\).
1. For each $t = 0, \ldots, \lfloor \log m \rfloor$:
   - Find the $2^t$-breakpoints of each bidder $i$.
   - Let $\bar{s}^t = (s_1^t, \ldots, s_n^t)$ be the allocation maximizing welfare among all allocations $(s_1, \ldots, s_n)$ where for each $i$, $s_i$ is in the neighborhood of a $2^t$-breakpoint of $v_i$.

2. Let $\text{weight}_\epsilon(\bar{s}) \cdot \bar{s}$ be the welfare maximizing distribution in $\{\text{weight}_\epsilon(\bar{s}^0) \cdot \bar{s}^0, \ldots, \text{weight}_\epsilon(\bar{s}^{\lfloor \log m \rfloor}) \cdot \bar{s}^{\lfloor \log m \rfloor}\}$. Return $\text{weight}_\epsilon(\bar{s}) \cdot \bar{s}$.

---

**Figure 1:** FPTAS for Fixed Number of Bidders

**Proof:** For each one of the possible $|\log m|$ values of $t$, we find all $O(\log_{1+\delta} \max_i \frac{v_i(m)}{v_i(1)})$ breakpoints of each bidder $i$, and then query the $n$ bundles in the neighborhood of the breakpoints. Binary search finds a single breakpoint in $O(\log m)$ queries, thus the total number of queries is $O(n^2 \cdot \log^2 m \log_{1+\delta} \max_i \frac{v_i(m)}{v_i(1)}) = poly \left( \log m, n, \frac{1}{\epsilon}, \log \max_i \frac{v_i(m)}{v_i(1)} \right)$. 

The following is an immediate corollary of Lemma 3.6 (setting $s = m$ in the statement of the lemma):

**Lemma 3.8** The algorithm finds a welfare maximizing distribution in an $\epsilon$-structured range.

Using Lemma 3.3 we now have:

**Theorem 3.9** There is a $(1 - \epsilon)$-approximation algorithm for multi-unit auctions that is truthful in expectation and makes $poly \left( n, \log m, \frac{1}{\epsilon}, \log \max_i \frac{v_i(m)}{v_i(1)} \right)$ value queries.

---

### 3.3 The Main Result: An FPTAS for a General Number of Bidders

This section presents a computationally-efficient FPTAS for a general number of bidders. We use structured ranges as a basic building block.

Let $u$ and $v$ be valuations of two bidders. Informally speaking, define a “meta bidder” with valuation $w$ as follows: the value $w(s)$ is set to the optimal expected welfare of allocating at most $s$ items in a structured range for $m$ items and the two bidders. Note that, by lemma 3.6 and a straightforward modification of the algorithm in the previous section, each query to $u$ can be calculated in time $poly \left( \log m, \frac{1}{\epsilon} \right)$. Apply this construction recursively, combining two meta bidders into one “larger” meta bidder (see Figure 2).

We end up with two meta-bidders at the top level, a case solved in the previous section. We will show that this algorithm runs in time $poly \left( n, \log m, \frac{1}{\epsilon} \right)$, and that the solution obtained well-approximates the optimal welfare. We will also show that this algorithm optimizes over some range of distributions, yielding truthfulness in expectation. Throughout this section we assume, without loss of generality, that $n$ is a power of 2 (as we can always add bidders that have a value of 0 for each bundle).

### 3.3.1 The Range

Before defining the range of the algorithm, we define a binary hierarchical division of the bidders, illustrated in Figure 2. We think of each node in the hierarchy as a meta-bidder representing a subset of the bidders. The root meta-bidder, representing $\{1, \ldots, n\}$, is composed of two meta-bidders representing $\{1, \ldots, \frac{n}{2}\}$ and $\{\frac{n}{2}+1, \ldots, n\}$, respectively. Applying the division recursively, meta-bidder $\{a, \ldots, b\}$ has two children: $\{a, \ldots, \frac{a+b-1}{2}\}$ is the left child and $\{\frac{a+b-1}{2}+1, \ldots, b\}$ is the right child. The leaves correspond to single bidders. Number the tree levels from 1 to $\log n$, where the leaves are in level $\log n$, and the top vertex is in level 1. The following definitions will prove helpful in defining the range of the algorithm $\mathcal{R}$. 
Definition 3.10 Let $A$ and $B$ be two disjoint set of bidders. Let $D_A$ and $D_B$ be two distributions over allocations to bidders in $A$ and $B$, respectively. Let $w \in [0,1]$. A weighted composition $w \cdot (D_A \oplus D_B)$ is the following distribution over allocations to bidders in $A \cup B$: with probability $w$, allocate to bidders in $A$ according to $D_A$ and to bidders in $B$ according to $D_B$, and with probability $(1 - w)$ allocate no items to all bidders in $A \cup B$.

Definition 3.11 The set of components of a weighted composition distribution $w \cdot (D_A \oplus D_B)$ includes $D_A$ and $D_B$. If $D_A$ and $D_B$ are weighted composition distributions then the components of $D_A$ and $D_B$ are also contained in the set of components of $w \cdot (D_A \oplus D_B)$.

Let $\epsilon' = \frac{\ln \frac{1}{\epsilon n}}{\log n}$. In the remainder of this section, every reference to a structured range refers to an $\epsilon'$-structured range on $m$ items and two bidders, though we often refer to a subset of the structured range that allocates at most $k$ items, for some $k \leq m$. For each vertex $T$ in the hierarchical tree with two leaf children $l$ and $r$, let $R_T^k$ be the set of all weighted allocations of at most $k$ items in a $\epsilon'$-structured range of bidders $l$ and $r$. For a vertex $T$ with two non-leaf children $L$ and $R$, let

$$R_T^k = \{ \text{weight}_\epsilon(a, b) \cdot (D_L \oplus D_R) \mid a + b \leq k, D_L \in R_L^a, D_R \in R_R^b \}$$

Now let the range of the algorithm be $R = R^m_\epsilon$.

Consider some distribution $D \in R_U^1$, (for some $U$ and $t$). Let $T$ be a descendant of $U$ in the hierarchical tree. There is a unique $k \leq t$ such that some $D_T^k \in R_T^k$ is a component of $D$. The distribution $D_T^k$ is said to be induced by $D$ on $T$, and $k$ is said to be the number of items associated with $T$ in $D$. Let $s_i$ be the number of items associated with bidder $i$ in $D$. The (feasible) allocation $(s_1, \ldots, s_n)$ is called the base allocation of $D$. Notice that for every allocation $\vec{s}$ there is at least one allocation $D \in R$ such that $\vec{s}$ is the base allocation of $D$. Moreover, distribution $D$ always allocates either $s_i$ items or 0 items to bidder $i$.

![Meta-Bidder Hierarchy](image)

Figure 2: Meta-Bidder Hierarchy

An algorithm that optimizes over $R$ is truthful in expectation. It also provides a good approximation:

Lemma 3.12 An algorithm that optimizes over $R$ provides an approximation ratio of at least $1 - \epsilon$.

Proof: The lemma follows from the following claim:

Claim 3.13 Let $D \in R$. Let $\vec{s}$ be its base allocation. $\Pr_D[\text{bidder } i \text{ receives } s_i] \geq 1 - \epsilon$.

Proof: If $s_i = 0$ the claim is true. Otherwise, for a non-leaf vertex $T$ with children $L$ and $R$, let $w_T \cdot (D_L \oplus D_R)$ denote the weighted composition distribution induced by $D$ on $T$. $w_T$ is the probability that the child vertices of $T$ receive a non-empty bundle. By definition, $w_T \geq 1 - \epsilon'$. The probability that a bidder $i$ receives $s_i$ is the probability that all non-leaf vertices containing $i$ “allocate” items to their
children. Let $P$ be the set of all non-leaf vertices on the path in the hierarchal tree from the leaf $\{i\}$ to the root $N$. We can bound this probability as follows:

$$
\prod_{T \in P} w_T \geq (1 - \epsilon')^{\log n} = \left( 1 - \frac{\ln \frac{1}{1-\epsilon}}{\log n} \right)^{\log n} \geq 1 - \epsilon
$$

}\end{proof}

Observe that the optimal (unweighted) allocation $(o_1, \ldots, o_n)$ is induced by some $D \in \mathcal{R}$. Hence, each bidder $i$ receives $o_i$ with probability at least $(1 - \epsilon)$ in $D$. The expected welfare of $D$ is therefore at least $(1 - \epsilon)$ times the welfare of the optimal unweighted allocation. The algorithm optimizes over all distributions in the the range $\mathcal{R}$, and thus returns an allocation with at least that expected welfare. \end{proof}

We now make the meta-bidder intuition formal.

**Definition 3.14** Let $\mathcal{D}$ be a range of distributions of $m$ items to a set $U$ of bidders. Let $\mathcal{D}^k$ be the set of all distributions in $\mathcal{D}$ that never allocate more than $k$ items. Let $S^k_U$ be the set of all allocations that allocate at most $k$ items to bidders in $U$. Define $v$ to be the valuation of a meta-bidder over $U$ using $D$ as follows:

$$
v(k) = \max_{D \in \mathcal{D}^k} \sum_{s \in S^k_U} \Pr_D[s \text{ is chosen}] \cdot \sum_{i \in U} v_i(s_i)
$$

Note that a valuation of a meta-bidder is monotone and normalized. We now define a meta-bidder for each vertex $T$. The definition is recursive. For a vertex $T$ with two leaf children $l$ and $r$, let $v_T$ be the valuation of the meta-bidder $T$ defined over $\{l, r\}$ using the range of distributions $\mathcal{D} = \cup_k \mathcal{R}^k_T$ (i.e., an $\epsilon'$-structured range). For a vertex $T$ with two meta-bidder children $L$ and $R$, let $v_T$ be the valuation of the meta-bidder $T$ defined over $\{L, R\}$ using the the $\epsilon'$-structured range of allocations to (meta-) bidders $L$ and $R$. The meta-bidder idea is powerful for precisely the following reason:

**Lemma 3.15** Let $O$ be a welfare maximizing distribution in $\mathcal{R}^n_T$. Let $O_U$ be the distribution of allocations to bidders in $U$ induced by $O$ on $U$, where $U$ is a descendant of $T$ in the hierarchical tree or $U = T$. Denote the expected welfare of $O_U$ by $|U|$. Let $o_U$ be the bundle that is associated with $U$ in $O$. Then, $v_U(o_U) = |U|$.

\begin{proof} The proof is by induction on the level of $U$. For a vertex $U$ with two leaf children $l$ and $r$, $v_U(o_U)$ equals the value of the welfare maximizing solution that allocates at most $o_U$ items in a structured range. On the other hand, observe that $O$ induces a weighted allocation that allocates at most $o_U$ items in a structured range, and that $O$ is welfare maximizing. Hence, the expected welfare of $O_U$ equals $v_U(o_U)$.

Consider a vertex $U$ with two leaf children $L$ and $R$, and let $O_U = \text{weight}_{\epsilon'}(l, r) \cdot (O_L \oplus O_R) \in \mathcal{R}_U^O$. The expected welfare of $O_U$ is by induction weight$_{\epsilon'}(l, r) \cdot (v_L(a) + v_R(b))$, i.e., $O_U$ is the welfare maximizing weighted allocation in a structured range with bidders $L$ and $R$ that allocates at most $o_U$ items, which is by definition the value of $v_U(o_U)$. \end{proof}

Fix $\delta = \frac{\ln \frac{1}{1-\epsilon}}{\log m}$. The important property of distributions in the range is that, similar to structured ranges, the bundle $k$ that is associated with a vertex is in the neighborhood of a breakpoint of the corresponding (meta-) bidder.

**Lemma 3.16** Let $O$ be a welfare maximizing distribution in $\mathcal{R}^n_T$. Let $L$ and $R$ be two sibling vertices that are descendants of $O$. Let $o_L$ and $o_R$ be the bundles associated with $L$ and $R$ in $O$, respectively. Let $t$ be the maximal non-negative integer such that $o_L$ and $o_R$ are multiples of $2^t$. Then, $o_L$ and $o_R$ are in the neighborhood of a $2^t$-breakpoint of $v_L$ and $v_R$, respectively.
Proof: Let \( P \) be the parent of \( L \) and \( R \), let its associated bundle in \( O \) be \( o_P \). By Lemma 3.15 and the definition of meta bidders, \( v_P(o_P) \) is equal to the expected welfare of a distribution of the form \( \text{weight}((L, o_{LR}),(O, o_{OR})) \) – the welfare maximizing distribution that allocates at most \( o_P \) items to bidders \( L \) and \( R \) in a structured range. By Lemma 3.6, \( o_L \) and \( o_R \) are in neighborhoods of \( 2^l \)-breakpoints of \( v_L \) and \( v_R \), respectively.

\[
\delta v = 0
\]

3.3.2 The FPTAS

The FPTAS we construct is the obvious one: find the optimal allocation to meta-bidders \( \{1, \ldots n/2\} \) and \( \{n/2 + 1, n\} \) in a structured range, then proceed recursively to find the best allocation between bidders \( \{1, \ldots \frac{n}{2}\} \) and \( \{\frac{n}{2} + 1, \ldots \frac{n}{2}\} \), and so on. In a naive implementation of this algorithm every value query to the valuations of the two meta-bidders is calculated recursively “on the fly”. However, a short calculation shows that in this implementation the number of value queries is polynomial in \((\log m)^{\log n}, n, \frac{1}{\epsilon}\). To improve the running time, we use the fact that to calculate \( v_T(s) \), for any bundle \( s \), we only need to know the value of a relatively small number of bundles in the neighborhood of breakpoints of the two (meta-) bidders that are \( T \)'s children.

We are also interested in a “strongly polynomial” algorithm. I.e., the running time should be independent of \( \log \frac{\max_i v_i(m)}{\min_i v_i(1)} \). Towards this end, define:

**Definition 3.17** A bundle \( s \) is an \( r \)-significant \( b \)-breakpoint of \( v \) if it is a \( b \)-breakpoint of \( v \) and if either \( s = 0 \) or \( v(s) \geq r \).

By the next lemma, we can “ignore” insignificant breakpoints:

**Lemma 3.18** Let \( O \) be a welfare maximizing distribution in \( R_T^+ \). Denote its expected welfare by \( |O| \). Let \( L, R \) be two sibling descendants of \( T \) in level \( l \). Let \( o_L, o_R \) be the bundles associated with \( L, R \) in \( O \), respectively. Let \( t > 0 \) be the largest integer s.t. both \( o_L \) and \( o_R \) are multiples of \( 2^t \). Then, \( o_L \) and \( o_R \) are in the neighborhood of an \( r \)-significant \( 2^t \)-breakpoint of \( v_L \) and \( v_R \), respectively, for \( r = \delta^t |O| \).

**Proof:** From Lemma 3.16 we know that \( o_L \) and \( o_R \) are in the neighborhood of \( 2^l \)-breakpoints of \( v_L \) and \( v_R \). It remains to show that they are \( r \)-significant. We make use of the following claim:

**Claim 3.19** Let \( \text{weight}_c(\vec{\delta}) \cdot \vec{\delta} \) be a welfare maximizing distribution of \( s \) items in an \( \epsilon \)-structured range for \( 2 \) bidders. Denote its expected welfare by \( |\cdot| \). For each \( i \in T \) with \( o_i \neq 0 \) we have that \( v_i(o_i) \geq \delta |\cdot| \).

**Proof:** The proof is almost identical to the proof of Lemma 3.6, and it is only sketched. Suppose for contradiction that there is some bidder \( k \) with \( v_k(o_k) < \delta |\cdot| \) and \( o_k > 0 \). Let \( t \) be the maximal integer s.t. each \( o_i \) is a multiple of \( 2^t \). Define the following allocation \( \vec{a} \): \( a_k = 0 \), and for the other bidder \( i \neq k \), let \( a_i = o_i \) (notice that \( \sum_i a_i \leq s \)). All bidders but one receive no items, thus \( \text{weight}_c(\vec{a}) = (1 + 2\delta) \text{weight}_c(\vec{\delta}) \), by definition of \( \text{weight}_c \). We have that \( \sum_i v_i(o_i) \geq (1 - 2\delta) \sum_i v_i(o_i) \). However, now \( \text{weight}_c(\vec{a}) \sum_i v_i(o_i) > \text{weight}_c(\vec{\delta}) \sum_i v_i(o_i) \), a contradiction.

We prove the lemma by induction on the level of the tree, starting with children of \( T \) to the leaves. By Lemma 3.15, \( v_T(s) = |O| \). \( v_T \) is a meta bidder, thus \( v_T(s) \) is also equal to the value of a welfare maximizing allocation in a structured range: \( w \cdot (o_L, o_R) \). By the claim \( v_L(o_L) \geq \delta |O| \). Similarly, \( v_R(o_R) \geq \delta |O| \).

Assume correctness for level \( l \) and prove for \( l + 1 \). For \( L \) and \( R \) in level \( l + 1 \) with parent \( P \), from arguments similar to those above it follows that \( v_L(o_L, v_R(o_R) \geq \delta v_P(o_P) \), where \( o_L, o_R, \) and \( o_P \) are the bundles associated with \( L, R \) and \( P \) in \( O \), respectively. By induction we get that: \( v_R(o_R), v_L(o_L) \geq \delta v_P(o_P) \geq \delta^{l+1} v_T(s) \), as needed.

**Theorem 3.20** The algorithm in Figure 3 is a truthful-in-expectation FPTAS for multi-unit auctions.
1. For each vertex $T$ in level $l$ evaluate (recursively, from bottom to top) all bundles that are in the neighborhood of an $r_l$-significant 2$^l$-breakpoint of $v_T$ (for $t = 1, 2, \ldots, \lfloor \log m \rfloor$, and $r_l = \delta^l \max_{i \in N} v_i(m)/2$)

2. Evaluate (recursively, from top to bottom starting from the top vertex $N$ and $o_N = m$) for each vertex $T$ with children $L$ and $R$ a welfare maximizing weighted distribution $\text{weight}^\epsilon((o_L, o_R)) \cdot \text{weight}^\epsilon((o_L, o_R))$ of (meta-) bidders $L$ and $R$, where $o_L + o_R \leq o_T$. Let $w_T = \text{weight}^\epsilon((o_L, o_R))$.

3. Each bidder $i$ receives $o_i$ items with probability $\Pi_{T \in P} w_T$ where $P$ is the set that includes all the proper ancestors of $i$ in the hierarchal tree.

Before proving the theorem, we comment on the definition of $r_l$ in the algorithm. Let $|O|$ be the expected welfare of $O$. On one hand, observe that $|O| \geq \max_i v_i(m)/2$ (allocating all items to bidder $i$ that maximizes $v_i(m)$ is an allocation that is induced by some $D \in R$. $D$ has an expected welfare of at least $(1-\epsilon)v_i(m)$), and on the other hand $|O| \leq n \max_i v_i(m)$. In other words, $r_l$ is on one hand small enough so we can calculate the significant breakpoints of the valuations according to Lemma 3.18, and on the other hand is not too small, so we do not have too many significant points to evaluate. The proofs below make this discussion formal.

**Lemma 3.21** The algorithm finds a welfare maximizing distribution $O \in R$.

**Proof:** We prove the following claim first:

**Claim 3.22** Let $T$ be a vertex in level $l$. The $\delta^l|O|$-significant 2$^l$-breakpoint are evaluated correctly (for $t = 1, 2, \ldots, \lfloor \log n \rfloor$).

**Proof:** The proof is by induction on the level $l$, starting from the leaves to the top. For $T$ with two leaf children $L$ and $R$, the claim is trivially true, since $|O| \geq \max_i v_i(m)/2$ and by Lemma 3.18.

We assume correctness for $l+1$ and prove for $l$. Consider vertex $T$ in level $l$ with meta-bidder children $L$ and $R$. By induction the values of bundles in the neighborhood of $r_{l+1}$-significant breakpoints of $v_L$ and $v_R$ are computed correctly. Consider some bundle $a_T$ in the neighborhood of an $r_l$-significant breakpoint of $v_T$. Denote by $\text{weight}^\epsilon((a_L, a_R)) \cdot \text{weight}^\epsilon((a_L, a_R))$ the welfare maximizing weighted allocation to $v_L$ and $v_R$ that allocates at most $a_T$ items. By induction, the values of $v_L(a_L)$, $v_R(a_R)$ are calculated correctly, since they are $\delta^l$-significant breakpoints of $v_L$ and $v_R$ (Lemma 3.18). Hence $v_T(a_T)$ is calculated correctly, as needed.

Now we can prove the lemma by induction from the top vertex to the leaves. The welfare of the distribution induced by $O$ on the top vertex $N$ is, by Lemma 3.15, equal to a welfare maximizing weighted allocation to bidders $\{1, \ldots, \frac{n}{2}\}$ and $\{\frac{n}{2} + 1, \ldots, n\}$ in a structured range weight $\left(\left(o_{1, \ldots, \frac{n}{2}}, o_{\left(\frac{n}{2} + 1, \ldots, n\right)}\right) \cdot \left(o_{1, \ldots, \frac{n}{2}}, o_{\left(\frac{n}{2} + 1, \ldots, n\right)}\right)\right)$. By the same arguments as above, the algorithm finds a welfare maximizing weighted allocation to bidders $\{1, \ldots, \frac{n}{2}\}$ and $\{\frac{n}{2} + 1, \ldots, n\}$ of at most $o_{1, \ldots, \frac{n}{2}}$ items. Proceeding similarly until we reach the leaves, we end up with a distribution in $R$ with an expected welfare equal to that of $O$.

**Lemma 3.23** The algorithm runs in time $\text{poly}(n, \log m, \frac{1}{\epsilon})$.

**Proof:** Fix some vertex $T$ with children $R$ and $L$. Observe that $v_T(s)$, for any $s$, can be computed using only bundles in the neighborhood of $r$-significant $b$-breakpoints of $v_L$ and $v_R$ (for $b = 1, 2, 4, \ldots, m$).
To see that these values can be obtained efficiently, notice that each valuation \( v \) in level \( l \) has only \( \text{poly}(n, \frac{1}{\delta})r_l \)-significant \( b \)-breakpoints (by our choice of \( r \) – since \( 2^n \cdot \max_i v_i(m) \geq |O| \)). Recall that via binary search a \( b \)-breakpoint of \( v \) can be can be found in \( \text{poly}(\log m) \) value queries to \( v \). For each breakpoint we also need to query all bundles in its neighborhood. Hence we need to query each valuation \( v \) (of a bidder or a meta bidder) only \( \text{poly}(\log m) \) times. The number of valuations of bidders and meta-bidders in the tree is \( \text{O}(n) \). and thus the total number of queries is still \( \text{poly}(n, \log m, \frac{1}{\delta}) \). Notice that the computational overhead is polynomial in the number of value queries: for each value query we compute a welfare maximizing allocation of the previous level in a structured range of 2 bidders, which can be done in time \( \text{poly}(\log m, \frac{1}{\delta}) \). In total, the running time of the algorithm is \( \text{poly}(n, \log m, \frac{1}{\delta}) = \text{poly}(n, \log m, \frac{1}{\epsilon}) \).

This completes the proof of Theorem 3.20

4 Truthful in Expectation Mechanisms Have More Power

This section shows that polynomial-time truthful-in-expectation mechanisms are strictly more powerful than universally-truthful polynomial-time mechanisms. Ideally, we would like to prove this for multi-unit auctions. However, finding a non-trivial lower bound on the power of polynomial time mechanisms for multi-unit auctions, even deterministic ones, is a big open question. Hence we study a variant of multi-unit auctions that is artificially restricted, with the sole purpose of proving such a separation for the first time.

We prove that universally truthful polynomial time mechanisms cannot provide an approximation ratio better than 2 for this variant, while we provide a truthful in expectation FPTAS. As in multi-unit auctions, the (deterministic) VCG mechanism solves this problem optimally, but in exponential time.

In a restricted multi-unit auction, a set of \( m \) items, where \( m \) is a power of 2, is to be allocated to two bidders. The set of feasible allocations is restricted as follows: either no items are allocated, or all items are allocated with at least one item per bidder. Each bidder \( i \in \{1, 2\} \) has a valuation function \( v_i \), given as a black box, specifying the bidder’s value for each number of items. We restrict \( v_i \) to be normalized \((v_i(0) = 0)\) and strictly increasing. The objective is to maximize social welfare, the sum of the values of the bidders. Our algorithms should run in time polynomial in \( \log m \).

**Theorem 4.1** The following statements are true:

1. For every constant \( \epsilon > 0 \), every universally truthful \((2 - \epsilon)\)-approximation mechanism for restricted multi-unit auction requires \( \Omega(m) \) communication.

2. There exists a \((1 + \epsilon)\)-approximation algorithm for restricted multi-unit auctions that is truthful in expectation and runs in time \( \text{poly}(\log m, \frac{1}{\delta}) \) (an FPTAS).

Before proceeding with the proof, we discuss the differences between restricted multi-unit auctions and the standard multi-unit auctions discussed in previous sections, and the role these differences play in the proof.

- **Strictly Increasing Valuations, \( m \) is a power of 2:** These restrictions are only there to simplify the proof. They can be removed without changing the statement of the theorem.

- **Allocate all Items with at least One Item per Bidder or Allocate Nothing:** The all-items-are-allocated constraint fulfills the conditions of the characterization of deterministic truthful mechanisms for multi-unit auctions [10]. The restriction that each bidder must receive at least one item, and the relaxation allowing the empty allocation, will prove useful in arguing about universally truthful randomized mechanisms which were not considered in [10].

To prove Theorem 4.1, first we bound the power of universally truthful polynomial time mechanisms for restricted multi-unit auctions. Then, we show that there exists a truthful in expectation FPTAS for restricted multi-unit auctions.
4.1 A Lower Bound on Universally Truthful Mechanisms

4.1.1 From Universally Truthful to Deterministic Mechanisms

It is inconvenient to study randomized mechanisms directly. Therefore, we start by showing that the existence of a universally truthful mechanism with a good approximation ratio implies the existence of a deterministic mechanism that provides a good approximation on “many” instances. The following definition and propositions are adapted from [7]. We repeat the proof here for completeness.

Definition 4.2 Fix $\alpha \geq 1$, $\beta \in [0, 1]$, and a finite set $U$ of instances of restricted multi-unit auctions. A deterministic algorithm $B$ for restricted multi-unit auctions is $(\alpha, \beta)$-good on $U$ if $B$ returns an $\alpha$-approximate solution for at least a $\beta$-fraction of the instances in $U$.

Proposition 4.3 (essentially [7]) Let $U$ be some finite set of instances of restricted multi-unit auctions, and let $\alpha, \gamma \geq 1$. Let $A$ be a universally truthful mechanism for restricted multi-unit auctions that provides an expected welfare of $\frac{\text{OPT}(U)}{\alpha}$ for every instance $I \in U$ with expected communication complexity $cc(A)$. Then, there is a $\gamma \cdot cc(A)$-time (deterministic) algorithm in the support of $A$ that is $\left(\frac{1}{\alpha}, \frac{1}{\alpha} - \frac{1}{\gamma}\right)$-good on $U$ and has a communication complexity of $\gamma \cdot cc(A)$.

Proof: For each mechanism $D$ in the support of $A$, let $cc(D)$ denote its expected communication complexity. Let $A'$ be the mechanism obtained from $A$ by replacing in the support of $A$ each mechanism $D$ for which $cc(D) \geq \gamma cc(A)$ by a mechanism that never allocates any items. Notice that the expected approximation ratio of $A'$ is at least $\alpha' = \frac{1}{\alpha - \frac{1}{\gamma}}$.

Fix some instance $I \in U$. The expected welfare of $A'(I)$ is at least $\frac{\text{OPT}(I)}{\alpha'}$. The probability that $A(I)$ has welfare at least $\frac{\text{OPT}(I)}{\alpha'}$ is lower-bounded by $\frac{1}{\alpha'}$: this is achieved if $A(I)$ is optimal with probability $\frac{1}{\alpha'}$ and has welfare 0 otherwise. Hence, there exists a deterministic algorithm $B$ in the support of $A$ that returns an $\alpha'$-approximate solution for at least a $\frac{1}{\alpha'}$-fraction of the instances in $U$. Notice that the communication complexity of $B$ is $\gamma \cdot cc(A)$. \hfill $\square$

Let $A$ be a universally truthful mechanism for restricted multi-unit auctions with communication complexity $\text{poly}(\log m)$ and an approximation ratio of $2 - \epsilon$, for a constant $\epsilon > 0$. By the claim, there must exist a $\left(2 - \epsilon', \frac{1}{\gamma}\right)$-good deterministic mechanism in its support with communication complexity of $\log m \cdot cc(A)$ (for some constant $\epsilon' > 0$ and some set of instances $U$). We will show that the communication complexity of this “good” mechanism for that $U$ is $\Omega(m)$, thus the communication complexity of $A$ is $\Omega\left(\frac{m}{\log m}\right)$.

4.1.2 A Lower Bound on “Good” Deterministic Mechanisms

To start, we recall the following characterization result for deterministic mechanisms for multi-unit auctions:

Theorem 4.4 ([10]) Let $f$ be a truthful deterministic mechanism for multi-unit auctions with strictly monotone valuations that always allocates all items with range at least 3. Then, $f$ is an affine maximizer.

Fix some universally truthful mechanism $A$ for restricted multi-unit auctions. The support of $A$ contains three possible (non-disjoint) types of deterministic mechanisms:

1. Imperfect Mechanisms: Mechanisms $B$ where there exist valuations $u, v$ such $B(u, v) = (0, 0)$.
2. Tiny-Range Mechanisms: Mechanisms that have a range of size at most 2.
3. Affine Maximizers: Mechanisms that are affine maximizers.
Notice that there are no other mechanisms in the support of A: a mechanism for restricted multi-unit auctions that always allocates some items must always allocate all items. If its range is of size at least 3, then by Theorem 4.4 it must be an affine maximizer.

Claim 4.5 Let B be a deterministic imperfect mechanism for restricted multi-unit auctions. Then, there is a constant $C_B > 0$ such that $B(u', v') = (0, 0)$ whenever $u'(m), v'(m) < C_B$.

Proof: Let $u, v$ be such that $B(u, v) = (0, 0)$. Let $C_B = \min_{k>0} \min(u(k), v(k))$. Since valuations are normalized and strictly increasing, $C_B > 0$. Consider some $u', v'$ as in the statement of the claim. Observe that $B(u, v') = (0, 0)$: by weak monotonicity (see [18]) bidder 1 should be allocated no items, hence bidder 2 is allocated no items (recall that every non-empty feasible allocation assigns at least one item to each bidder). Similarly, $B(u', v') = (0, 0)$.

Let $A$ be a universally truthful mechanism for restricted multi-unit auctions that provides an approximation ratio of $2 - \epsilon$. Define $C_A = \min_B C_B$, where the minimum is taken over all imperfect mechanisms in A’s support. Fix a constant $\sigma$ such that $\sigma << \epsilon$. For each integer $k$ such that $1 \leq k \leq m - 1$, define the instance $I_k = (u_k, v_k)$ as follows: $u_k(t) = (\sigma C_A / 2m)t$ for all $t < k$, and $u_k(t) = C_A / 2$ for all $t \geq k$; $v_k(t) = \sigma C_A / 2m$t for all $t < m - k$, and $v_k(t) = C_A / 2$ for all $t \geq m - k$. Let $U = \{I_k\}_k$. Notice that the optimal welfare for $I_k$ is $C_A$ and is achieved by the allocation $(k, m - k)$. Any other allocation provides a welfare of at most $C_A / 2 + \sigma C_A / 2 < C_A / (2 - \epsilon)$.

Claim 4.6 Let A be a universally truthful mechanism for restricted multi-unit auctions that provides an approximation ratio of $2 - \epsilon$, for some constant $\epsilon$. Then, there exists an affine maximizer in the support of A with a range of size $\Omega(m)$ that is $(2 - \epsilon', \frac{1}{2})$-good on $U$ where both bidders have positive weights, for some constant $\epsilon' > 0$.

Proof: By Proposition 4.3 there exists a deterministic mechanism $B$ in the support of $A$ that provides a $(2 - \epsilon')$-approximation on more than half of the instances in $U$, for some constant $\epsilon' > 0$ (choose $\gamma = \log m$). Notice that $u_k(m), v_k(m) < C_A$ for each instance $I_k = (u_k, v_k)$ in $U$, and as a result any imperfect mechanism in the support of $A$ does not allocate any items on any instance in $U$. Therefore, $B$ cannot be imperfect. $B$ provides an approximation ratio of $2 - \epsilon$ for more than half of the instances in $U$, that is, outputs the optimal allocation for more than half of the instances in $U$. Every instance in $U$ has a different optimal allocation, and thus the range of $B$ is of size at least $m / 2$. In particular, $B$ is not a tiny-range mechanism. Thus, $B$ allocates all items and has a range of size at least $\frac{3m}{2}$. By Theorem 4.4 it must be an affine maximizer. Observe that the weights of both bidders must be positive: otherwise the mechanism is a tiny range mechanism.

To finish the proof of the lower bound the following claim suffices:

Claim 4.7 (essentially [8]) An affine maximizer with a range of size $t$ and a positive weight for each bidder has communication complexity of at least $t$.

Proof: Let $B$ be an affine maximizer. Denote its range by $\mathcal{R}$. There exists non-negative constants $w_1, w_2$, and constants $\{c_i(s_1, s_2)\}_{(s_1, s_2) \in \mathcal{R}}$, such that $B(u, v) = \max_{(s_1, s_2) \in \mathcal{R}} (w_1 u(s_1) + w_2 v(s_2) + c_i(s_1, s_2))$ for all valuations $u, v$.

We reduce from the disjointness problem on $t$ bits. In this problem Alice holds a string $(a_1, \ldots, a_t) \in \{0, 1\}^t$ and Bob holds a string $(b_1, \ldots, b_t) \in \{0, 1\}^t$. The goal is to determine if there exists some $i$ such that $a_i = b_i = 1$. It is known that any deterministic algorithm for this problem has a communication complexity of $t$.

Let $p >> \max_{(s_1, s_2)} |c_i(s_1, s_2)|$. Define some one-to-one and onto correspondence $f : \mathcal{R} \rightarrow [t]$. Define the following valuations: $u(i) = (2i \cdot p + a_{f(i, m - i)} \cdot p) / w_1$ if $(i, m - i) \in \mathcal{R}$, otherwise $u(i) = 2i \cdot p / w_1$. Also define $v(i) = (2i \cdot p + b_{f(i, m - i)} \cdot p) / w_2$ if $(m - i, i) \in \mathcal{R}$, otherwise $v(i) = 2i \cdot p / w_2$. The (optimal) solution returned by $B$ on input $(u, v)$ has welfare $(2m + 2)p$ if and only if there is some $i$ with $a_i = b_i = 1$. Hence, the communication complexity of $B$ is at least $t$. 

\[ \]
4.2 An FPTAS for Restricted Multi-Unit Auctions

All that is left is to show a better-than-2 truthful in expectation mechanism for restricted multi-unit auctions:

**Lemma 4.8** There exists a truthful-in-expectation FPTAS for restricted multi-unit auctions.

A simple variation of the FPTAS for a fixed number of bidders yields an FPTAS for restricted multi-unit auctions. Specifically, we will show that there is a poly $(\log m, \frac{1}{\varepsilon})$ algorithm for multi-unit auctions that provides a $(1 + \varepsilon)$-approximation to the optimal solution in the range of allocations permitted in restricted multi-unit auctions. Furthermore, this algorithm optimizes over some range of distributions over allocations $\mathcal{R}$. The support of each distribution $D \in \mathcal{R}$ contains only allocations that permitted in restricted multi-unit auctions. This implies the existence of a truthful in expectation $(1+\varepsilon)$ approximation algorithm for restricted multi-unit auctions.

We start by defining a new weight function for restricted multi-unit auction:

$$r\text{-weight}_\varepsilon(s) = \begin{cases} 0, & s = (m, 0) \text{ or } s = (0, m); \\ \text{weight}_\varepsilon(s), & \text{otherwise.} \end{cases}$$

A restricted $\varepsilon$-structured weight is defined similarly to a structured range, but with respect to $r\text{-weight}_\varepsilon$:

$$\{r\text{-weight}_\varepsilon(s) \cdot s\}_{s \in S},$$

where $S$ is the set of all allocations of $m$ items to these $n$ bidders. The definition of a weighted allocation is also done now using $r\text{-weight}_\varepsilon$. The range of the algorithm $\mathcal{R}$ will be restricted $\varepsilon$-structured range of allocations of $m$ items to the two bidders. The algorithm is very similar to the FPTAS for a constant number of bidders, but with the following changes: the expected welfare of a weighted allocation calculated in step 2 is defined using $r\text{-weight}_\varepsilon$, and not using $\text{weight}_\varepsilon$; in the second bullet of step 1 we consider only allocations that are permitted for restricted multi-unit auctions.

The correctness of the FPTAS described above follows by straightforward modifications to the proofs of section 3 for a constant number of bidders.

Acknowledgements

We thank Bobby Kleinberg, Noam Nisan, Sigal Oren, and Chaitanya Swamy for helpful discussions and comments.

References


ABSTRACT

We present an incentive-compatible polynomial-time approximation scheme for multi-unit auctions with general k-minded player valuations. The mechanism fully optimizes over an appropriately chosen sub-range of possible allocations and then uses VCG payments over this sub-range. We show that obtaining a fully polynomial-time incentive-compatible approximation scheme, at least using VCG payments, is NP-hard. For the case of valuations given by black boxes, we give a polynomial-time incentive-compatible 2-approximation mechanism and show that no better is possible, at least using VCG payments.

Categories and Subject Descriptors
F.2.8 [Analysis of Algorithms and Problem complexity]: Miscellaneous

General Terms
Theory

Keywords
Combinatorial Auctions, Incentive Compatibility

1. INTRODUCTION

We consider multi-unit auctions of m identical items among n bidders. In our setting we view the number of items m as “large” and desire mechanisms whose computational complexity is polynomial in the number of bids needed to represent m. Every bidder i has a valuation function v_i : \{1..m\} \rightarrow \mathbb{R}, where v_i(q) denotes his value for obtaining q items. We assume that v_i is weakly monotone (free disposal), and normalized (v_i(0) = 0). Our goal is the usual one of maximizing the social welfare Σ_i v_i(q_i) where Σ_q_i ≤ m. In the general case, each v_i is represented by m real numbers, and in abstract settings may be accessed as a black box. In a concrete setting, we will assume that v_i is represented as a k-minded bid, i.e. given by a list: (q_1, p_1) ... (q_k, p_k), where v_i(q) = max\{j \mid q_j ≤ q\} p_j. This corresponds to a XOR bidding language [8]. In general, k can be as large as m, and the case k = 1 is the single-minded case.

This problem has received much attention, starting from Vickrey’s seminal paper [11] who described an incentive compatible mechanism for the case of “downward sloping valuations” in which the items can be optimally allocated greedily. The general case, however, is NP-hard, as the single-minded case is just a re-formulation of the weighted knapsack problem. Luckily, the knapsack problem has a fully-polynomial time approximation scheme (i.e. can be approximated to within a factor of 1 + ϵ in time polynomial in n, log m, ϵ^{-1}), and it is not hard to see that the algorithm directly extends to the general case of multi-unit auctions.

This is where the well-known difficulty of constructing approximation mechanisms kicks in: it is not possible, in general, to convert an arbitrary approximation algorithm into an incentive compatible mechanism [9, 6]. While a significant amount of success has been obtained to date in doing this for various “single-parameter” settings such as single-minded bidders in multi-unit auctions, hardly any success has been obtained for more general valuations like k-minded bidders described above.

For the case of multi-unit auctions with single-minded bidders, [1] presents a truthfully fully polynomial time approximation scheme (FPTAS), improving upon a previous result of [7]. The only result known for k-minded bidders is a randomized 2-approximation mechanism that is truthful in expectation [5]. This paper presents a polynomial time approximation scheme (PTAS) for the general case.

Theorem: For every fixed ϵ > 0, there exists an incentive-compatible (1 + ϵ)-approximation mechanism for multi-unit auctions with k-minded bidders whose running time is polynomial in n, log m, and k. The dependence of the running time on ϵ is exponential.

The mechanism fully optimizes the social welfare over some carefully constructed sub-range of allocations and then simply uses VCG payments to ensure incentive compatibility. The main challenge is that of identifying an appropriate sub-range over which exact optimization is computationally possible, and yet such that this optimization is a valid approximation over the optimal allocation. It is known that such exact optimization over a sub-range is the only way to get an incentive-compatible mechanism using VCG payments [9, 2]. We will call this type of algorithms maximal-in-range (MIR) algorithms, or VCG-based algorithms.
We also generalize this mechanism to stronger bidding languages such as the one used in [3]. However, we prove two ways in which the mechanism can not be improved upon. First, we show that the dependence on \( \epsilon \) cannot be made polynomial without destroying incentive compatibility. This is in contrast to the algorithmic possibility of fully polynomial time approximation, and to what is possible for single-minded bidders [1].

**Theorem:** No fully polynomial time incentive compatible approximation mechanism that uses VCG payments exists, unless \( \text{P=NP} \).

Then we show that the dependence on \( k \) is necessary, and that no approximation scheme is possible in a general black-box model. This is shown in a general communication model, and even for two bidders.

**Theorem:** Every approximation mechanism among two bidders with general valuations that uses VCG payments requires exponentially many queries to obtain an approximation factor that is smaller than 2.

We do present an incentive compatible approximation mechanism in the general black box model that does obtain a factor of 2. This improves upon the randomized one of [5] that is only truthful in expectation.

**Theorem:** There exists a truthful 2-approximation mechanism for multi-unit auctions among general valuations whose running time is polynomial in \( n \) and \( \log m \). The access to bidders' valuations is through value queries: “what is \( v_i(q) \)” for all \( q \). The mechanism uses VCG payments.

The main open problem remains to improve the approximation ratio obtained, in an incentive compatible way. The only general method known for constructing such mechanisms is VCG, and our lower bounds state that this will not work here. Partial negative results on this problem appear in [4], which may support a conjecture that VCG-based mechanisms may be the only possible truthful mechanisms that obtain a reasonable approximation ratio, thus proving that our results are optimal.

**Paper Organization**

In Section 2 we present the PTAS for \( k \)-minded bidders, and the 2-approximation in the black-box model. Section 3 considers lower bounds for MIR algorithms in both models. In Section 4 we describe a general construction, and its algorithmic applications: truthful mechanisms for other models and more powerful bidding languages.

2. **THE BASIC MECHANISMS**

2.1 **A Truthful PTAS for \( k \)-Minded Bidders**

We will design an MIR algorithm for this problem, which directly yields an incentive-compatible VCG-based mechanism. We will define a range \( \mathcal{R} \) of allocations, and prove that if all bidders are \( k \)-minded then the algorithm outputs in polynomial time the best allocation in \( \mathcal{R} \). Let us start with defining the subrange \( \mathcal{R} \).

---

1 This is yet another improvement upon the mechanism of [5] which requires the stronger demand queries.

---

**Definition 1.** We say that an allocation \( (s_1, \ldots, s_n) \) is \( t \)-round if there exists a set \( T \) of bidders, \( |T| \leq t \), such that both conditions hold:

- For each bidder \( i \not\in T \), \( s_i \) is a multiple of \( \max\left(\frac{m-l}{(n-t)^2}, 1\right) \), where \( l = \sum_{j \in T} s_j \).
- \( \sum_{i \not\in T} s_i \leq \max\left(\frac{m-l}{(n-t)^2}, 1\right) \cdot (n-t)^2 \)

We let \( \mathcal{R} \) be the set of all \( t \)-round allocations for some fixed \( t \) (that will depend only on the approximation guarantee). Next we prove that the value of the best allocation in \( \mathcal{R} \) is close to the optimum:

**Lemma 1.** Let \( (a_1, \ldots, a_n) \) be an optimal \( t \)-round allocation, and \( (o_1, \ldots, o_n) \) an optimal unrestricted allocation, then \( \sum_i v_i(a_i) \geq (1 - \frac{1}{t+1}) \sum_i v_i(o_i) \).

**Proof.** Let us start with an optimal unrestricted allocation \( (o_1, \ldots, o_n) \), and use it to construct a \( t \)-round allocation with high value. Assume that all items are allocated (without loss of generality due to the monotonicity of the valuations), and that \( v_1(a_1) \geq \ldots \geq v_n(o_n) \). Let \( T = \{1, \ldots, t\} \) be the set of \( t \) bidders from Definition 1, and assign each bidder \( i \not\in T \) a bundle of size \( o_i \). As in Definition 1, let \( l = \sum_{i \not\in T} o_i \).

Let \( j \not\in T \) be the bidder who got the largest number of items \( o_j \geq \frac{m-l}{n-t} \). For each \( i \not\in T \), \( i \not= j \), round up each \( o_i \) to the nearest multiple of \( b = \max\left(\frac{m-l}{(n-t)^2}, 1\right) \). Assign bidder \( j \) no items. This is a valid \( t \)-round allocation since if \( b \neq 1 \) we added at most \( (n-t)^2 \cdot \frac{m-l}{(n-t)^2} \leq \frac{m-l}{n-t} \) items by rounding up, but deleted at least \( \frac{m-l}{n-t} \) items by removing \( o_j \). Notice that the second condition also holds. If \( b = 1 \), observe that even the optimal allocation is \( t \)-round. As for the value of the solution, observe that each bidder \( i \not= j \) gets a bundle no smaller than \( o_i \). Also observe that \( v_j(o_j) \leq \frac{\sum_{i \not= j} v_i(o_i)}{t+1} \), which gives the required approximation. \( \square \)

Our MIR approximation algorithm will try each subset of at most \( t \) bidders to be the set \( T \) of bidders. For each possible selection of \( T \), the algorithm will consider all possible allocations to bidders in \( T \) according to the \( k \) bids each bidder submitted. That is, we will consider the allocation that assigns each bidder \( i \in T \) exactly \( s_i \) items, if and only if \( \sum_{i \in T} s_i \leq m \), and for each \( s_i \) there is a bid \( (s_i, p_i) \) in the \( k \) bids of bidder \( i \) (for some \( p_i > 0 \)).

For each selection of \( T \) and allocation to the bidders in \( T \) according to their bids, the algorithm splits the remaining \( m - l \) items into at most \( (n-t)^2 \) equi-sized bundles of size \( \max\left(\frac{m-l}{(n-t)^2}, 1\right) \), where \( l \) is the total number of items that bidders in \( T \) get. The maximum-in-range algorithm will optimally allocate these equi-size bundles among the bidders that are not in \( T \). Finally, the algorithm outputs the best allocation among all allocations considered. All that is left is to show the following two lemmas:

**Lemma 2.** For every fixed \( t \) the above algorithm runs in time polynomial in \( n \) and \( \log m \).

**Proof.** There are at most \( t \cdot n^t \) possible selections of sets \( T \). For each selection of \( T \) there are at most \( k^t \) allocations to bidders in \( T \) that are considered. Finding the optimal allocation to bidders not in \( T \) is by dynamic programming. Let \( b \) be the size of the equi-size bundles. Without loss
of generality, we assume that \( T = \{ n - t + 1, \ldots, n \} \). We calculate the following information for every \( 1 \leq i \leq n - t \) and \( 1 \leq q \leq (n - t)^2 \): \( M(i, q) \) is the maximum value that can be obtained by allocating at most \( q \) equi-size bundles among bidders \( 1 \ldots i \). Each entry can be filled in polynomial time using the relations: \( M(i, q) = \max_{b \leq q} v_i(q'b) + M(i - 1, q - q') \). In particular notice that if \( b = 1 \) then the number of equi-size bundles is polynomial in the number of bidders, thus the number of entries in the table is polynomial also in this case. Overall we get that the algorithm runs in time polynomial in \( n \) and \( \log m \), for every fixed \( t \).

**Lemma 3.** The above algorithm finds an optimal \( t \)-round allocation.

**Proof.** First, notice that the algorithm outputs a \( t \)-round allocation. Let us prove that it outputs an optimal one. Let \( O = \{ o_1, \ldots, o_n \} \) be an optimal \( t \)-round allocation, let \( T \) be the set of bidders from Definition 1, and let \( i = \Sigma_1 \leq r o_i \). For each bidder \( i \in T \) remove the maximal number \( q_i \) of items from \( o_i \) such that \( v_i(o_i) = v_i(o_i - q_i) \). Observe that it is possible that \( q_i = 0 \) and that there exists a pair \( (q_i, p_i) \) in \( i \)'s XOR bids such that \( q_i = o_i - q_i \). We now handle the bidders that are not in \( T \). Each bidder \( i \in T \) holds a bundle that is a multiple of \( b = \max(\left\lfloor \frac{m - i}{n - t} \right\rfloor , 1) \) in \( O \), while in order to the allocation that we construct to be \( t \)-round we need the bidders not in \( T \) to receive multiples of \( b' = \max(\left\lfloor \frac{m - i}{n - t} \right\rfloor , 1) \), for \( i = \Sigma_1 \leq r o_i \). However, notice that \( b' \geq b \), and that the number of equi-size bundles is at least the same. Hence, by assigning each bidder \( i \notin T \) the same number of equi-size bundles as in \( O \), bidder \( i \) holds at least the same value as in \( O \). The lemma follows since the algorithm considers the newly constructed allocation.

We therefore have the following theorem:

**Theorem 1.** There exists a truthful computationally-efficient VCG-based mechanism that provides a \( \left( 1 - \frac{1}{n - t} \right) \)-approximation for multi-unit auctions with \( k \)-minded bidders in time polynomial in \( n \), \( \log m \), \( k \), for every constant \( t \).

### 2.2 A \( 2 \)-Approximation for Multi-Unit Auctions with Black-Box Access

Let us consider the multi-unit auction problem with general valuations given by black boxes. We will assume in our algorithm an “oracle access” to it that may be queried for \( v_i(q) \), where \( q \) is the given bundle size\(^2\).

We will design a \( 2 \)-approximation MIR algorithm for this problem, which again yields an incentive-compatible VCG-based mechanism. Our MIR approximation algorithm will first split the items into \( n^2 \) equi-sized bundles of size \( b = \left\lfloor \frac{m}{n} \right\rfloor \) as well as a single extra bundle of size \( r \) that holds the remaining elements (thus \( n^2b + r = m \)). The maximum in range algorithm will optimally allocate these whole bundles among the \( n \) bidders. What we need to show is the following two simple facts:

**Lemma 4.** An optimal allocation of the bundles can be found in time polynomial in \( n \) and \( \log m \).

\(^2\)This is analogous to the weakest “value query” in a combinatorial auction setting. Our lower bounds presented later will apply to all other query types as well.

**Theorem 2.** There exists a truthful computationally-efficient VCG-based mechanism that provides a \( 2 \)-approximation for multi-unit auctions with general valuations.

### 3. LOWER BOUNDS FOR VCG-BASED MECHANISMS

We now move on to show that both mechanisms essentially achieve the best approximation ratios possible. We say that an allocation \( (s_1, \ldots, s_n) \) is complete if all items are allocated: \( \Sigma_i s_i = m \). Consider an MIR algorithm for \( n \) bidders that does not have full range of complete allocations. I.e., for some \( 0 \leq s_1, \ldots, s_{n-1} \leq m, \Sigma_i s_i \leq m \) it never outputs the allocation \( (s_1, \ldots, s_{n-1}, m - \Sigma_i s_i) \). Now consider the set of valuations where for every bidder \( i \) \( v_i(q) = 1 \) if and only if \( q \geq s_i \) (and \( 0 \) otherwise). The only allocation with value \( n \) is \( (s_1, \ldots, s_{n-1}, m - \Sigma_i s_i) \) which is not in the range, while all other allocations have a value of at most \( n - 1 \).

From this we can easily get a lower bound for any computationally efficient MIR algorithm in the models considered in this paper. We start with a lower bound in the black-box model. The lower bound is on the number of queries that the bidders must be queried, and holds for any type of query — i.e., in a general communication setting.

**Proposition 1.** An MIR algorithm for multi-unit auctions that achieves an approximation ratio better than \( 2 \) requires exponential communication. The result also applies for randomized settings.
Lemma 6. [10] Finding the optimal allocation in multi-unit auctions requires exponential communication, even if there are only two bidders and even for just finding the value of the allocation. This lower bound also applies for both randomized and nondeterministic settings.

Thus, any MIR algorithm for 2 bidders that uses sub-exponential communication will be non-optimal and thus, as argued above, gives no better than a $2$-approximation. The case of more than 2 bidders follows by setting all valuations to 1.

The second result rules out the existence of FPTAS for $k$-minded bidders. In other words, the dependence of the running time in $\frac{1}{k}$ cannot be made polynomial. The result essentially applies to all models that allow single-minded bidders, e.g., XOR bids, and the bidding language used in [3].

Proposition 2. Every polynomial-time MIR algorithm cannot achieve an approximation ratio better than $(1 - \frac{1}{k})$, unless $P = NP$.

Proof. Similarly to the previous proposition, and by the standard reduction from knapsack to multi-unit auctions, it follows that for every polynomial-time MIR algorithm there exist large enough $n$ and $m$ for which the range of complete allocations is not full, unless $P = NP$. The lemma follows by the discussion above.

This concludes the proof of the lower bounds for MIR mechanisms, except for one technical detail that should be explicitly mentioned. Our lower bounds were for MIR algorithms, while VCG-based mechanisms are only proved to give algorithms that are equivalent to MIR algorithms. However, both proofs hold even for finding the value of the optimal allocation and thus directly apply also to algorithms that are equivalent to MIR algorithms.

4. A GENERAL CONSTRUCTION AND APPLICATIONS

We present a construction that generalizes the PTAS for $k$-minded bidders described in Section 2. The construction takes a maximal-in-range algorithm for a constant number of bidders in some bidding language or model\(^3\), and extends it to a truthful mechanism for an unbounded number of bidders. Yet, the extension loses only an arbitrarily small constant in the approximation ratio.

We describe three applications of the construction. First, we improve the PTAS for $k$-minded bidders of Section 2. Then, we consider the bidding language considered in [3]. In [3] an approximately truthful PTAS for this bidding language was described, while we present a truthful VCG-based PTAS (this is the best possible since Section 3 essentially rules out the possibility of a VCG-based PTAS). Finally, we present a truthful $(\frac{1}{k} + \epsilon)$-approximation mechanism for the case the valuations are sub-additive (a.k.a. complement free) and are accessed via a black box.

\(^3\)By a model we mean, e.g., some restriction on the valuations or on how they can be accessed.

4.1 The Setting

Fix some bidding language or a model for multi-unit auction in which the bidders can answer value queries. Let $A$ be a maximal-in-range algorithm for $t$ bidders and at most $m$ items in this model. Denote the complexity of $A$ by $A(t, m)$, its range by $R_{a,t,m}$, and its approximation guarantee by $\alpha$.

The Construction

1. Build the set $Q$ of allocations as follows:
   
   (a) Let $u = (1 + \frac{1}{t})$. Let $L = \{0, 1, [u], [u^2], \ldots, u^{\lfloor \log u m \rfloor}, m\}$.
   
   (b) For every set $T$ of bidders, $|T| \leq t$, and $l \in L$:
      
      i. Run $A$ with $m - l$ items and the set $T$ of bidders. Denote by $s_i$ the number of items $A$ allocates to each bidder $i \in T$.
      
      ii. Split the remaining $l$ items into at most $n^2$ bundles, each consists of $\max(\lfloor \frac{1}{t} \rfloor, 1)$ items.
      
      iii. Find the optimal allocation of the equal-size bundles among the bidders that are not in $T$.
      
      Denote by $s_i$ the allocation to each bidder $i \notin T$.
      
   iv. Add $(s_1, \ldots, s_n)$ to $Q$.

2. Output the allocation with the highest welfare in $Q$.

Theorem 3. The construction is maximal in range, and runs in time $\text{poly}(\log m, n, A(m, t))$, for every constant $t$. It outputs allocation with value of $(\alpha - \frac{1}{t})$ of the optimal allocation.

Proof. We will make use of the following definition:

Definition 2. An allocation is $(R, l, l)$-round if:

- $R$ is a range, and in each $R \in R$ at most $t$ bidders are allocated up to $m - l$ items.
- There exists a set $T$ of bidders, $|T| \leq t$, such that the bidders in $T$ are allocated according to some allocation in $R$.
- Each bidder $i \notin T$ receives an exact multiple of $\max(\lfloor \frac{1}{t} \rfloor, 1)$ units, and $\Sigma_{i \in T} s_i \leq \max(\lfloor \frac{m - l}{n} \rfloor, 1) \cdot n^2$.

It is not hard to verify that the construction always outputs allocations that are in the following range:

$$R = \{ S | S \text{ is a } (R, A, k, m - l, k, l) \text{-round allocation} \}$$

where $l \in L$ and $k \leq t$.

Lemma 7. Let $(a_1, \ldots, a_n)$ be an optimal unrestricted allocation. There exists an allocation $(s_1, \ldots, s_n) \in R$ for which $\Sigma_i v_i(s_i) \geq (\alpha - \frac{1}{t}) \cdot \Sigma_i v_i(a_i)$.

Proof. Denote the value of the optimal solution by OPT. Without loss of generality, assume that $v_1(a_1) \geq \ldots \geq v_n(a_n)$. Let $l \in L$ be the largest such that $m - l \geq \Sigma_i v_i(a_i)$. Let $(s_1, \ldots, s_l)$ be the allocation that $A$ outputs on bidders $1, \ldots, t$ bidders and $m - l$ items, and assign each bidder $1 \leq i \leq t$, $s_i$ items. Observe that $\Sigma_{i=1}^t v_i(s_i) \geq (\alpha - \frac{1}{t}) \cdot \Sigma_{i=1}^l v_i(a_i)$. To handle bidders $t + 1$ to $n$, we claim that reducing the number of items in a given instance does not hurt the quality of the optimal solution too much:
Claim 1. Fix an instance of a multi-unit auction with $n$ bidders and $m$ items. Let $(a_1, \ldots, a_n)$ be an optimal solution, and denote its value by $OPT_m$. Then, the value of the optimal solution on $m' \leq m$ is at least $(\frac{m}{m'} \cdot OPT_m - v_{\text{max}})$, where $v_{\text{max}} = \max_i v_i(a_i)$.

Proof. Order bidders 1 to $n$ by decreasing order of density $\frac{v_i(a_i)}{m'}$. Let $j$ be the largest integer such that $\sum_{i \leq j} a_i \leq m'$. Observe that $\sum_{i \leq j} v_i(a_i) \geq \frac{m'}{m} \cdot OPT_m$. Now observe that by removing bidder $j + 1$, the first $j$ bidders form an allocation that uses at most $m'$ items, and that $\sum_{i=1}^j v_i(a_i) \geq \frac{m'}{m} \cdot OPT_m - v_{\text{max}}$. \hfill $\square$

In the allocation we construct, bidders $t+1, \ldots, n$ will share $l$ items. Let $j \in \arg \max_{2 \leq t+1} v_j(a_i)$. Notice that $v_j(a_i) \leq \frac{m}{m'+1} \cdot OPT_m$. Denote the optimal allocation of these bidders when using only $l$ items by $(a'_1, \ldots, a'_n)$. By the claim, $\sum_{i=t+1}^n v_i(a'_i) \geq \frac{1}{4} \cdot \sum_{i=t+1}^n v_i(a_i) - v_j(a_i)$.

Let $y$ be the bidder who is allocated the bundle of the largest size in $(a'_1, \ldots, a'_n)$. Let $b = \max\{\frac{1}{4}, 1\}$. Round up the assignment of each bidder $i \geq t+1, i \neq y$ to the nearest multiple of $b$. Assign bidder $y$ no items. Notice that this allocation is in $R$. By monotonicity, the value of every other bidder $i, i \neq y$, is at least the same, and thus the value these bidders hold in the new allocation is at least

$$\sum_{i=t+1, i \neq y}^n v_i(a'_i) - \frac{OPT}{t+1} \geq \frac{1}{4} m \sum_{i=t+1}^n v_i(a_i) - 2 \cdot \frac{OPT}{t+1}$$

Taking into account the first $t$ bidders, the total welfare of the allocation is at least $\alpha \sum_{i=1}^n v_i(a_i) + \sum_{i=t+1}^n v_i(a_i) - \frac{3 OPT}{t+1}$.

Hence, this allocation holds a value of at least $(\alpha - \frac{3}{t+1})OPT$.

All that is left is to show that the construction runs in polynomial time:

Lemma 8. The optimal allocation in $R$ can be found in time $\text{poly}(\log m, n, A(m, t))$, for every constant $t$.

Proof. Step 1(b)(iii) of the construction can be implemented using a dynamic programming similarly to Lemma 2; optimality of the allocation in $R$ is clear. As for running time, the algorithm runs in time $\text{poly}(\log \frac{1}{\epsilon}, m \cdot n^k, A(m, t))$, which is polynomial in the relevant parameters for every constant $t$. \hfill $\square$

4.2 Applications of the Construction

4.2.1 A PTAS for $k$-Minded Valuations

We reprove the PTAS for $k$-minded bidders of Section 2: a multi-unit auction problem with $m$ items and $n$ $k$-minded bidders can be optimally solved by exhaustive search in time $\text{poly}(\log m, n^k)$, which is polynomial in $\log m$ and $k$ for every constant $n$. By the construction (and since an optimal algorithm is in particular maximal in range), we get a PTAS for $k$-minded bidders: for every constant $t$, we get a $(1 - \frac{3}{t+1})$-approximation in time polynomial in $n$ and $\log m$.

4.2.2 A PTAS for Marginal Piecewise Valuations

The following bidding language was used by [3]: a valuation $v$ is determined by a list of at most $k$ tuples denoted by $(u_1, m_1), \ldots, (u_k, m_k)$. The tuples determine the marginal utility of the $j$th item. In other words, to determine the value of a set of $s$ items, we sum over all the marginal utilities. I.e., for each item $j$, $u_k \leq j \leq u_{k+1}$, let his marginal utility be $r_j = m_k$, and for every $s \leq m$ let $v(s) = \sum_{j \leq m} r_j$. (In fact, the above bidding language is more powerful than the one described in [3], which allows only marginal-decreasing piecewise valuations.)

We now show how to optimally solve a multi-unit auction problem in this setting with a constant number of bidders. A PTAS follows, just as in the $k$-minded case.

We say that bidder $i$ is precisely assigned if he is allocated $s_i$ items, and for some $u_s > 0$ there exists a tuple $(s, u_s)$ in his $k$ bids. The main observation here is that there is an optimal solution $(a_1, \ldots, a_n)$ in which at most one bidder is not precisely assigned: suppose there are two bidders $i$ and $j$ that are not precisely assigned. Then, move items to the bidder with the higher (or equal) marginal utility. The value of the allocation cannot decrease. Continue this process until all bidders but at most one are precisely assigned.

Now optimally solving a multi-unit auction problem with a constant number of bidders is obvious: select each of the $n$ bidders in his turn to be the bidder that is not precisely assigned. In each iteration, let $i$ the bidder that is not precisely assigned, and go over all allocations in which all other bidders are precisely assigned. Then, assign bidder $i$ the remaining items. Since there are at most $k$ possible sets that make a bidder precisely assigned, the algorithm runs in time $\text{poly}(n \cdot (s \cdot k)^{n-1}, \log m)$, which is polynomial in $\log m$ and $k$ for every constant $n$.

4.2.3 A $(\frac{4}{5} + \epsilon)$-Approximation for Complement-Free Valuations

In this model we assume that the valuations are given as black boxes (as in Section 2.2), and that for each valuation $v$ and bundles $0 \leq s, t \leq m$ we have that $v(s) + v(t) \geq v(s + t)$.

Let us now describe an algorithm that provides an approximation ratio of $\frac{4}{5}$ in this setting when the number of bidders is constant. By the construction, we get a $(\frac{4}{5} + \epsilon)$-approximation VCG-based mechanism for unbounded number of bidders, for every constant $\epsilon$. The algorithm is quite simple: Fix a small enough constant $\delta > 1$. All bidders but at most one, can only receive a bundle $s$ that is a power of $\delta$ (including the empty bundle). The bidder that does not get a bundle of size that is a power of $\delta$ receives the remaining items. We use exhaustive search to find the optimal allocation in this range.

To see that the algorithm indeed provides an approximation ratio of $\frac{4}{5}$, look at an optimal solution $(a_1, \ldots, a_n)$. Without loss of generality, assume that $a_1 \geq \ldots \geq a_n$ (notice that unlike before now the bidders are ordered by their bundle size). Let $O$ be the set of bidders with odd indices, and $E$ be the set of bidders with even indices.

The analysis is divided into two cases. First suppose that $\sum_{i \in O} v_i(a_i) \geq \sum_{i \in E} v_i(a_i)$. Look at the following allocation: the bidders in $E$ are rounded up to the power of $\delta$ that is near $\frac{m}{2}$, the bidders in $O \setminus \{1\}$ are rounded up to the nearest power of $\delta$, and bidder 1 gets the remaining items. Notice that the above allocation is valid since for a small enough choice of $\delta$ we assign bidders in $O$ no more items
than what we removed from bidders in $E$. Also notice that this allocation is in the range. As for the approximation ratio, observe that bidders in $O$ hold the at least the same value as in the optimal solution, since each bidder in $O$ is allocated at least the same number of items as in the optimal solution. In addition, each bidder in $E$ holds at least half of the items allocated to him in the optimal solution. Thus, by subadditivity, bidders in $E$ hold together at least half of the value they hold in the optimal solution. In total, the value of the allocation obtained by the algorithm is at least $\frac{3}{4}$ of the value of the optimal solution.

Let us now handle the case where $\sum_{i \in O} v_i(o_i) < \sum_{i \in E} v_i(o_i)$. Look at the allocation where all bidders in $O$ are rounded up to the power of $\delta$ that is near $v_i^2$, and all bidders in $E$ are rounded up to the nearest power of $\delta$ (except for one arbitrary bidder in $E$ who gets the remaining items). This allocation is in the range, and the analysis is similar to the previous case, leaving us with an approximation ratio of $\frac{4}{3}$ also in the current case.

The running time of the algorithm is $\text{poly}(n \cdot (\log \delta m)^{n-1})$, which is polynomial in $\log m$ for constant $n$ and $\delta$.

Notice that the approximation ratio achieved is almost the best possible, as every MIR approximation algorithm that guarantees a factor better than $\frac{4}{3}$ requires exponential communication: by Lemma 6 finding the optimum solution of a multi-unit auction with two bidders requires exponential communications. We make the valuations sub-additive by defining for each $v$ a new valuation: $v'(s) = v(s) + v(m)$. Thus, as in Section 2.2, the range of every polynomial-time MIR mechanism for two bidders with subadditive valuations cannot contain all complete allocations. Fix some MIR algorithm, and let $(s_1, m-s_1)$ be a complete allocation that is not in the range. Consider the following instance: bidder 1 values at least $s_1$ items with a value of 2, and smaller bundles with a value of 1, and bidder 2 values at least $m-s_2$ items with a value of 2, and smaller bundles with a value of 1 (and 0 is the value of the empty bundle). Notice that the valuations of the bidders are indeed complement free. Also observe that the optimal welfare is 4, but the mechanism can achieve welfare of at most 3.

Acknowledgements

We are grateful to Liad Blumrosen and Ahuva Mu’alem for helpful discussions. This research was supported by grants from the Israel Science Foundation and the USA-Israel Binational Science Foundation.

5. REFERENCES