Black-Box Reductions in Mechanism Design

Zhiyi Huang [*] Lei Wang [†] Yuan Zhou [‡]

April 15, 2011

Abstract

A central question in algorithmic mechanism design is to understand the additional difficulty introduced by truthfulness requirements in the design of approximation algorithms for social welfare maximization. In this paper, by studying the problem of single-parameter combinatorial auctions, we obtain the first black-box reduction that converts *any* approximation algorithm to a truthful mechanism with essentially the *same* approximation factor in a prior-free setting. In fact, our reduction works for the more general class of *symmetric single-parameter* problems. Here, a problem is symmetric if its allocation space is closed under permutations.

As extensions, we also take an initial step towards exploring the power of black-box reductions for general single-parameter and multi-parameter problems by showing several positive and negative results. We believe that the algorithmic and game theoretic insights gained from our approach will help better understand the tradeoff between approximability and the incentive compatibility.

[*] University of Pennsylvania. Email: hzhiyi@cis.upenn.edu.

[†] Georgia Institute of Technology. Email: lwang@cc.gatech.edu.

[‡] Carnegie Mellon University. Email: yuanzhou@cs.cmu.edu.

1 Introduction

In an *algorithmic mechanism design* problem, we face an optimization problem where the necessary inputs are private valuations held by self-interested agents. The high-level goal of *truthful* mechanisms is to reveal these valuations via the bids of the agents and to optimize the objective simultaneously. In this paper, we will focus on the objective of social welfare maximization.

It is well known that the *VCG* mechanism ([27, 7, 14]) which maximizes the social welfare exactly is truthful. As usual in computer science, computational tractability is a necessary requirement. However, VCG is not computationally efficient in general. And unfortunately, the simple combination of approximation algorithms and VCG usually fails to preserve truthfulness. This raises the important open question (see [22]) of whether the design of truthful mechanisms is fundamentally harder than the design of approximation algorithms for social welfare maximization.

Recently, several positive results indicated that one can always convert an approximation algorithm to a truthful mechanism with the same approximation factor in the *Bayesian* setting where the distributions of the agents are public knowledge (see [16, 3, 15]). However, not much is known in the *prior-free* setting where no distribution is known.

In this paper, by studying the problem of *single-parameter combinatorial auctions*, we show the first black-box reduction that converts any approximation algorithm to a universal truthful mechanism with the same approximation factor in the prior-free setting.

In the single-parameter combinatorial auction problem, we are given a set $\mathcal{J}$ of m items and a *public* valuation function $f : 2^{\mathcal{J}} \to \mathbf{R}$. Assume that f is given via an oracle which takes a set S as input and returns $f(S)$. In addition, we have n agents each of whom has a *private* multiplier v_i^* such that the item set S provides $v_i^* f(S)$ amount of utility to agent i. The goal is to design a truthful mechanism which maximizes $\sum_i v_i f(S_i)$, where $S_1 \cdots S_n$ is a partition of $\mathcal{J}$.

This problem has its motivation in the TV ad auctions where the items are time slots and each agent is an advertiser whose private multiplier is her *value-per-viewer*. In [13], the authors provided a logarithmic approximate truthful mechanism for this problem under the assumption that f is submodular. However, the optimal approximation algorithm for the underlying social welfare maximization has a ratio of $1-1/e$ given by Vondrak ([28]). By our result, applying Vondrak's algorithm as a black-box, we immediately obtain a truthful mechanism with the optimal constant approximation ratio.

Main result. In fact, our black-box reduction not only works for this particular problem but for a broad class of *symmetric single parameter* problems. Formally, a mechanism design problem (with n agents) is *single-parameter* if each feasible allocation is represented as an n-dimensional real vector $\boldsymbol{x}$, and each agent i has a private value v_i such that her valuation of allocation $\boldsymbol{x}$ is given by $v_i x_i$. We further define that a problem is *symmetric* if the set of feasible allocations is closed under permutations: if $\boldsymbol{x}$ is feasible, so is $\pi \circ \boldsymbol{x}$ for any permutation π. Here $\pi \circ \boldsymbol{x}$ is defined as the vector $(x_{\pi(1)}, ..., x_{\pi(n)})$.

Theorem 1.1. *For a symmetric single-parameter mechanism design problem* Π*, suppose we are given an* α*-approximate* $(\alpha > 1)$ *algorithm* $\mathcal{A}$ *as a black-box, then for any constant* $\epsilon > 0$*, we can obtain a polynomial time truthful mechanism with approximation factor* $\alpha(1+\epsilon)$*.*

Many interesting mechanism design problems such as position auctions in sponsored search are in the class of symmetric single-parameter problems. In particular, it contains the problem of single-parameter combinatorial auctions that we are interested in.

Corollary 1.2. *For the single-parameter submodular combinatorial auction problem, there is an optimal 1-1/e approximate truthful mechanism.*

Our construction is based on the technique of *maximum-in-range*. Here, a maximum-in-range mechanism outputs the allocation maximizing the social welfare over a fixed range of allocations. Using the algorithm $\mathcal{A}$ as a black-box, we construct a range $\mathcal{R}$ such that social welfare maximization over $\mathcal{R}$ is efficient. And we will prove that the approximation factor obtained is essentially α.

In our reduction, we make no assumption on the black-box algorithm $\mathcal{A}$. In addition, while the black-box algorithm may be randomized, our reduction does not introduce any further randomization. If the algorithm is deterministic, then our mechanism is deterministically truthful.

Extensions. A natural extension of our result is to consider the general (possibly asymmetric) single-parameter mechanism design problems. By a novel relation between mechanism design and constraint satisfaction problems, we derive a significant lower bound of the approximability of maximum-in-range mechanisms for general single-parameter problems, which to some extent suggests the difficulty in designing factor-preserving black-box reductions.

However, the problem considered in our lower bound construction is not *downward-closed*. Here, a single-parameter problem is downward-closed if the following holds: let $\boldsymbol{x}$ be a feasible allocation, then any allocation $\boldsymbol{x}'$ obtained by decreasing one of the components of $\boldsymbol{x}$ to 0 is feasible as well. Many problems that satisfy the *free disposal* property, such as combinatorial auctions, are downward-closed. For the class of downward-closed single-parameter problems, we provide a black-box reduction that converts any *α-approximate rounding* algorithm to a truthful in expectation mechanism with essentially the same approximation factor. Our idea is to apply the technique in [19] to the single-parameter case.

Finally, we generalize our technique to the *multi-parameter* setting. We study the *constant-dimension* and *symmetric* mechanism design problem. We generalize our construction in the symmetric single-parameter case to this problem and obtain a black-box reduction that converts any algorithm into a truthful and quasi-poly-time mechanism with essentially the same approximation guarantee. Alternatively, we can obtain a black-box reduction that converts any algorithm into a truthful and polynomial time mechanism with logarithmic degradation in the approximation factor.

Related work. There has been a significant amount of work related to black-box reductions in mechanism design. In the single-parameter setting, the first black-box reduction was given by Briest et al. [4]. The authors studied the single-parameter binary optimization problem and they showed that any algorithm which is an FPTAS can be converted to a truthful mechanism that is also an FPTAS. Secondly, Babaioff et al. [1] studied the single-value combinatorial auction problem and they constructed a black-box reduction that converts an algorithm to a truthful mechanism with the approximation factor degraded by a logarithmic factor. Finally, the recent work by Goel et al. [13] provided a black-box reduction with a super constant degrade in approximation factor for partially public combinatorial auction.

For multi-parameter problems, there is no factor-preserving black-box reduction in general (e.g. [23]). This motivates the study of *truthfulness in expectation*, which is a weaker notion of incentive compatibility. Here, a randomized mechanism is truthful in expectation, if truth telling maximizes an agent's expected payoff. The initial effort in black-box reduction for multi-parameter problems is due to Lavi and Swamy [19], they showed a method to convert a certain type of algorithms called integrality-gap-verifiers to truthful in expectation mechanisms with the same approximation factors. Recently, Dughmi and Roughgarden [9] studied the class of packing problems. Via an elegant black-box reduction and smooth analysis, they showed that if a packing problem admits an FPTAS, then it admits a truthful in expectation mechanism that is an FPTAS as well. Balcan et al.[2] considered black-box reductions from the revenue maximization aspect. By the technique of sample complexity in machine learning, they gave revenue-preserving reductions from truthful mechanism design to the algorithmic pricing problems. At last, Dughmi et al. [10] introduce a method to convert convex rounding scheme into truthful in expectation mechanism and achieve an optimal $(1 - \frac{1}{e})$-approximation for the combinatorial auction problem when the valuations are a special type of submodular functions.

The previous discussion is about *prior-free* mechanism design. Another important area in algorithmic game theory is the *Bayesian* mechanism design where each agent's valuation is drawn from some publicly known prior distribution. Hartline and Lucier [16] studied this problem in the single-parameter setting. They constructed a clever black-box reduction that converts any non-monotone algorithm into a monotone one without compromising its social welfare. Following this work, Bei and Huang [3] and Hartline et al. [15] independently showed such black-box reductions in the multi-parameter setting as well.

Organization. In Section 3, we provide the detailed description and proof of our black-box reduction for the class of symmetric single-parameter problems. As a corollary, we obtain a black-box reduction for the single-parameter combinatorial auctions. In Section 4.1, we first show the limitation of maximum-in-range mechanism for general single-parameter problems, then in Section 4.2 we study the downward-closed single-parameter problems. Finally, Section 5 provides our result for the constant dimension symmetric mechanism design problem.

2 Preliminaries

In this section, we will outline the basic concepts in mechanism design relevant to our paper.

Truthfulness. Let $\mathcal{X}$ be the set of all feasible allocations, and $v_i(\boldsymbol{x})$ be the private valuation of agent i if allocation $\boldsymbol{x} \in \mathcal{X}$ is picked. A typical goal of a mechanism is to reveal agents' private valuation functions via their bids and optimize the obtained social welfare simultaneously. Formally, suppose we are given n agents and let $\boldsymbol{v} = (v_1, ..., v_n)$ be the valuation functions reported by the agents. Based on this, a (deterministic) mechanism M will specify an allocation $\boldsymbol{x}(\boldsymbol{v}) \in \mathcal{X}$ and a payment $\boldsymbol{p}(\boldsymbol{v})$. We say M is *deterministically truthful*(or truthful), if the following conditions hold: for any $i, \boldsymbol{v}_{-i}$ and any v_i, v_i', we have $v_i(\boldsymbol{x}(v_i, \boldsymbol{v}_{-i})) - p_i(v_i, \boldsymbol{v}_{-i}) \geq v_i(\boldsymbol{x}(v_i', \boldsymbol{v}_{-i})) - p_i(v_i', \boldsymbol{v}_{-i})$.

When a mechanism is randomized, there are two notions of truthfulness: (1) *Universal truthfulness*: A universally truthful mechanism is a probability distribution over deterministically truthful mechanisms; (2) *Truthfulness in expectation*: A mechanism is truthful in expectation if an agent maximizes her *expected utility* by being truthful. Here, an agent's utility is defined as her valuation minus payment. It is easy to see that every deterministically truthful mechanism is universally truthful and every universally truthful mechanism is truthful in expectation.

Single-parameter mechanism design. In a *single-parameter* mechanism design problem, each allocation is represented as an n-dimensional real vector $\boldsymbol{x}$ (where n is the number of agents), and each agent i has a private value v_i such that her valuation of allocation $\boldsymbol{x}$ is given by $v_i x_i$. It is known [21] that for a single-parameter problem, a mechanism is truthful if and only if (1) the allocation rule is *monotone*: suppose $v_i \leq v_i'$, then $x_i(v_i, \boldsymbol{v}_{-i}) \leq x_i(v_i', \boldsymbol{v}_{-i})$; (2) each agent i's payment is determined by $p_i(\boldsymbol{v}) = v_i x_i(v_i, \boldsymbol{v}_{-i}) - \int_0^{v_i} x_i(t, \boldsymbol{v}_{-i}) dt$.

Maximum-in-range mechanisms. The *maximum-in-range* technique is a general approach in the field of mechanism design. It works as follows: The mechanism fixes a range $\mathcal{R}$ of allocations *without* any knowledge of the agents' valuations. Given any $\boldsymbol{v}$, let $\boldsymbol{x}^* = \text{argmax}_{\boldsymbol{x} \in \mathcal{R}} \sum_j v_j(\boldsymbol{x})$ and $\boldsymbol{x}^*_{-i} = \text{argmax}_{\boldsymbol{x} \in \mathcal{R}} \sum_{j \neq i} v_j(\boldsymbol{x})$ respectively. Now define payment p_i of agent i to be $\sum_{j \neq i} v_j(x^*_{-i}) - \sum_{j \neq i} v_j(x^*)$. It is now not difficult to see that with this payment function, it is in best interest of every agent to report their true valuations, irrespective of what others report. The major challenge in designing maximum-in-range mechanism is to balance between the size of the range and the approximation factor obtained. A larger range can obtain better approximation but yield greater computational complexity.

3 Symmetric Single-Parameter Mechanism Design

Recall that a single-parameter mechanism design problem is symmetric if the allocation space $\mathcal{X}$ is closed under permutations: if $\boldsymbol{x} \in \mathcal{X}$, then $\pi \circ \boldsymbol{x} = (x_{\pi(1)}, ..., x_{\pi(n)}) \in \mathcal{X}$ for any permutation π. In this section, we will prove Theorem 1.1: For a symmetric single-parameter problem Π, given any constant $\epsilon > 0$ and any α-approximate algorithm $\mathcal{A}$ as a black-box, we design a polynomial time truthful mechanism with approximation factor $(1 + \epsilon)\alpha$.

Our construction is based on the *maximum-in-range* technique. Given an algorithm $\mathcal{A}$, we define a range $\mathcal{R}$ by applying $\mathcal{A}$ as a black-box on a carefully chosen collection of *typical bid vectors*. Our mechanism is then maximum-in-range over $\mathcal{R}$. We will show: (1) To maximize social welfare over $\mathcal{R}$ for a

```
Normalization. Let $T_0 = \{\boldsymbol{v} : 1 = v_1 \geq ... \geq v_n\}$;
Discretizing. Let $Q = \{\eta^k : 0 \leq k \leq \lceil \log_{1/\eta}(n^M) \rceil\}$ where $M \geq \log_2 \frac{8}{\epsilon}$ is a constant. For any real value $z$, define $\lfloor z \rfloor_\eta = \eta^{\lceil \log_{1/\eta} z \rceil} \in Q$. Then we define a function $D : T_0 \mapsto T_0$ as follows: for each $\boldsymbol{v} \in T_0$ and for each $i$, define
    $$D(\boldsymbol{v})_i = \begin{cases} \lfloor v_i \rfloor_\eta & v_i \geq u = \frac{1}{n^M} \\ 0 & \text{otherwise} \end{cases}$$
    Let $T_1 = D(T_0)$;
Equalization. Let $n_k = \lfloor \beta^k \rfloor$ where $\beta > 1$ is a fixed constant and $0 \leq k \leq \lfloor \log_\beta n \rfloor$. Define a function $E : T_1 \to T_1$ as follows: for each $\boldsymbol{v} \in T_1$ and $1 \leq i \leq n$, $E(\boldsymbol{v})_i$ is set to be $v_{n_k}$ when $n_k \leq i < n_{k+1}$. At last, let $T = E(T_1)$.
```

given bid vector, we only need to examine polynomially many allocations in $\mathcal{R}$, hence our mechanism is efficient; (2) Every bid vector can be mapped to a typical bid with approximately the same social welfare, hence our mechanism performs almost as well as the algorithm $\mathcal{A}$. This proves the approximation factor.

Now we describe our range construction in detail for a given symmetric single-parameter problem Π, black-box algorithm $\mathcal{A}$ and constant $\epsilon > 0$.

3.1 Construction of the range

Let $V = \mathbf{R}_+^n$ be the collection of all possible bid vectors. Next we will provide a three-step procedure that chooses a subset $T \subseteq V$ as our collection of *typical bids*.

The first step is *normalization*: By properly reordering the agents and scaling their bids, we only consider the set T_0 of bids where $\boldsymbol{v} \in T_0$ if and only if $1 = v_1 \geq ... \geq v_n$; The second step is *discretization*. In this step, our goal is to obtain a finite set of bid vectors that approximately represent the whole valuation space V. To do this, given any vector $\boldsymbol{v} \in T_0$, we first apply the operation of *tail cutting*: We choose a small value u (e.g. $1/n^M$ for some constant M) and round all the entries smaller than u to 0; then, we discretize the interval $[u, 1]$ by considering $Q = \{\eta^k : k \geq 0\} \cap [u, 1]$ where $\eta < 1$ is a fixed constant. We will round down each of the remaining entries of $\boldsymbol{v}$ after the tail cutting to the closest value in Q. If we do the above for each $\boldsymbol{v} \in T_0$, we obtain a finite set of vectors T_1; The final step is *equalization*. We fix a constant $\beta > 1$ and partition $[n]$ into $\log_\beta n$ groups. For each vector in T_1, we equalize its entries within each group by setting them to be the value of the largest entry in the group. We then obtain the set of vectors T, and each vector in T is called a *typical bid*.

Now we provide the detailed description. In the following, we fix constants $\beta > 1$ and $\eta < 1$ such that $\frac{\beta}{\eta} = 1 + \epsilon/2$.

For a bid vector $\boldsymbol{v}$, let $\boldsymbol{x}^{\mathcal{A}}(\boldsymbol{v})$ be the allocation obtained by applying algorithm $\mathcal{A}$ on $\boldsymbol{v}$. Since the allocation space is closed under permutations, we may assume $\boldsymbol{x}^{\mathcal{A}}(\boldsymbol{v})_1 \geq \boldsymbol{x}^{\mathcal{A}}(\boldsymbol{v})_2 \geq ... \geq \boldsymbol{x}^{\mathcal{A}}(\boldsymbol{v})_n$. At last, let $\mathcal{R}_0 = \{\boldsymbol{x}^{\mathcal{A}}(\boldsymbol{v}) : \boldsymbol{v} \in T\}$ and we finally define our range as $\mathcal{R} = \{\pi \circ \boldsymbol{x} : \boldsymbol{x} \in \mathcal{R}_0, \pi \in \Pi_n\}$ where Π_n consists of all permutations over n elements.

3.2 Analysis

Now we analyze the performance of our mechanism. Since the mechanism is maximum-in-range, it is truthful. We will show that it has polynomial running time and an approximation factor of $\alpha(1 + \epsilon)$.

Running time. We show that the social welfare maximization over $\mathcal{R}$ is solvable within polynomial time, hence our maximum-in-range mechanism is efficient.

First of all, we show that the $|\mathcal{R}_0|$ is polynomial in n. We only need to prove the following lemma:

Lemma 3.1. $|\mathcal{R}_0| \leq |T| \leq n^{1/\log_2 \beta + M/\log_2(1/\eta)}$.

Proof. The first inequality follows from the definition of $\mathcal{R}_0$. Now we prove the second one. Observe that for each vector $\boldsymbol{v}$ in T_1, $E(\boldsymbol{v})$ is uniquely determined by the values $\{v_{n_k} : 0 \le k \le \lfloor \log_\beta n \rfloor\} \subseteq Q \bigcup \{0\}$. Moreover, we have that $v_{n_{k-1}} \ge v_{n_k}$ for all k. Therefore, let H be the class of non-increasing functions from $\{0, 1, ..., \lfloor \log_\beta n \rfloor\}$ to $Q \bigcup \{0\}$, thus $|T| \le |H|$. Since $|Q| = \lceil \log_{1/\eta}(n^M) \rceil$, It is not difficult to see,

$$|H| \le \binom{\lfloor \log_\beta n \rfloor + \lceil \log_{1/\eta}(n^M) \rceil}{\lceil \log_{1/\eta}(n^M) \rceil} \le 2^{\lfloor \log_\beta n \rfloor + \lceil \log_{1/\eta}(n^M) \rceil} \le n^{1/\log_2 \beta + M/\log_2(1/\eta)}. \qquad \square$$

Now we are ready to prove the running time guarantee. Let $\mathsf{opt}_{\mathcal{R}}(\boldsymbol{v})$ be the allocation maximizes the social welfare over $\mathcal{R}$ for the given bid vector $\boldsymbol{v}$. Let σ be the permutation such that $v_{\sigma(1)} \ge ... \ge v_{\sigma(n)}$. Obviously, for each $\boldsymbol{x} \in \mathcal{R}_0$, we have $\boldsymbol{v} \cdot (\sigma^{-1} \circ \boldsymbol{x}) \ge \boldsymbol{v} \cdot (\pi \circ \boldsymbol{x})$ for all permutation π. Therefore, $\mathsf{opt}_{\mathcal{R}}(\boldsymbol{v}) \in \{\sigma^{-1} \circ \boldsymbol{x} : \boldsymbol{x} \in \mathcal{R}_0\}$. By Lemma 3.1, $|\{\sigma^{-1} \circ \boldsymbol{x} : \boldsymbol{x} \in \mathcal{R}_0\}| = |\mathcal{R}_0| \le n^{1/\log_2 \beta + M/\log_2 \eta}$, this implies that $\mathsf{opt}_{\mathcal{R}}(\boldsymbol{v})$ can be found in polynomial time.

Approximation factor. We show that the approximation factor of our mechanism is $\alpha(1+\epsilon)$. Given any bid vector $\boldsymbol{v}$, by reordering and scaling properly, we may assume $\boldsymbol{v} \in T_0$, then we consider the typical bid $E(D(\boldsymbol{v}))$. We show that for any sorted allocation $\boldsymbol{x}$, the social welfare $\boldsymbol{v} \cdot \boldsymbol{x}$ is $(1+\epsilon)$-approximated by $E(D(\boldsymbol{v})) \cdot \boldsymbol{x}$, hence an α-approximate solution for social welfare maximization with respect to $E(D(\boldsymbol{v}))$ is an $\alpha(1+\epsilon)$-approximate solution for $\boldsymbol{v}$. This proves the desired approximation guarantee.

Now we provide the detail. We first show that by considering $D(\boldsymbol{v})$ instead of $\boldsymbol{v} \in T_0$, the social welfare is rounded down by at most a factor of $\eta(1-\epsilon/4)$.

Lemma 3.2. *For any $\boldsymbol{v} \in T_0$ and any allocation $\boldsymbol{x}$ s.t. $x_1 \ge ... \ge x_n$, we have $D(\boldsymbol{v}) \cdot \boldsymbol{x} \le \boldsymbol{v} \cdot \boldsymbol{x} \le \frac{1}{\eta(1-\epsilon/4)} D(\boldsymbol{v}) \cdot \boldsymbol{x}$.*

Proof. The first inequality holds by definition. Now we prove the second one. We first show that the social welfare affected by "tail cutting" is bounded by a fraction of $\epsilon/4$.

Claim 3.3. $\sum_{i:v_i \ge 1/n^M} v_i x_i \ge (1-\epsilon/4) \sum_{i=1}^n v_i x_i$.

Proof. For the ease of notation, we let $A = \sum_{i=1}^n v_i x_i$ and $B = \sum_{i:v_i \ge 1/n^M} v_i x_i$. Thus $A = B + \sum_{i:v_i < 1/n^M} v_i x_i \le B + \frac{1}{n^M} \sum_{i:v_i < 1/n^M} x_i$. Since $x_1 \ge ... \ge x_n$ and $1 = v_1 \ge ... \ge v_n$, we have $\frac{1}{n} \sum_{i=1}^n x_i \le \sum_{i=1}^n \frac{\sum_{j=1}^n v_j}{n} x_i \le \sum_{i=1}^n v_i x_i = A$. The last inequality holds because of rearrangement inequality. Therefore, we have $A \le B + \frac{1}{n^{M-1}} \big(\frac{1}{n} \sum_{i:v_i < 1/n^M} x_i \big) \le B + \frac{1}{n^{M-1}} \big(\frac{1}{n} \sum_{i=1}^n x_i \big) \le B + \frac{1}{n^{M-1}} A \le B + \frac{\epsilon}{4} A$. Hence we have $B \ge (1-\epsilon/4)A$. $\square$

Let $\boldsymbol{v}' = D(\boldsymbol{v})$, it is easy to see: $\boldsymbol{v}' \cdot \boldsymbol{x} = \sum_{i:v_i \ge 1/n^M} v'_i x_i \ge \eta \sum_{i:v_i \ge 1/n^M} v_i x_i \ge \eta(1-\frac{\epsilon}{4}) \boldsymbol{v} \cdot \boldsymbol{x}$. $\square$

Secondly, we show that the social welfare increases by at most a factor of β by considering $E(\boldsymbol{v})$ instead of $\boldsymbol{v}$ for any $\boldsymbol{v} \in T_1$.

Lemma 3.4. *For any $\boldsymbol{v} \in T_1$ and any allocation $\boldsymbol{x}$ s.t. $x_1 \ge \cdots \ge x_n$, we have $\boldsymbol{v} \cdot \boldsymbol{x} \le E(\boldsymbol{v}) \cdot \boldsymbol{x} \le \beta \boldsymbol{v} \cdot \boldsymbol{x}$.*

Proof. The first inequality is implied by the definition of E. We will prove the second one. Let $\boldsymbol{v}' = E(\boldsymbol{v})$ and $L = \lceil \log_{1/\eta} n \rceil$. Since $\boldsymbol{v}, \boldsymbol{v}' \in T_1$, we have that $v_i, v'_i \in Q$ for each i. Thus, if we let $l_i = \log_\eta v_i$ and $l'_i = \log_\eta v'_i$ respectively for each i, then l_i's and l'_i's are non-decreasing sequences. By our construction, it is easy to see: (1) for all $1 \le i \le n$, $l'_i \le l_i$; (2) for all $l \in [0, L]$, $|\{i : l'_i \le l\}| \le \beta |\{i : l_i \le l\}|$. Since $x_1 \ge ... \ge x_n$, we have the following:

Claim 3.5. *For any $l \in [0, L]$ and $1 \le i \le n$, $\sum_{i:l'_i \le l} x_i \le \beta \sum_{i:l_i \le l} x_i$.*

Observe that if we define $W_l = \eta^l$ for $0 \le l \le L$ and $W_{L+1} = 0$, then $\sum_{i=1}^n x_i v_i = \sum_{i=1}^n x_i W_{l_i} = \sum_{i=1}^n x_i \sum_{l=l_i}^L (W_l - W_{l+1}) = \sum_{l=0}^L (W_l - W_{l+1}) \sum_{i:l_i \le l} x_i$.

Similarly, we have $\sum_{i=1}^n x_i v'_i = \sum_{l=0}^L (W_l - W_{l+1}) \sum_{i:l'_i \le l} x_i$. By Claim 3.5, we have $\boldsymbol{v}' \cdot \boldsymbol{x} \le \beta \boldsymbol{v} \cdot \boldsymbol{x}$. $\square$

By Lemma 3.2 and Lemma 3.4, we have the following:

Corollary 3.6. *For any $\boldsymbol{v} \in T_0$ and any allocation $\boldsymbol{x}$ such that $x_1 \geq \cdots \geq x_n$, we have: $\eta(1-\epsilon/4)\boldsymbol{v}\cdot\boldsymbol{x} \leq E(D(\boldsymbol{v}))\cdot\boldsymbol{x} \leq \beta\boldsymbol{v}\cdot\boldsymbol{x}$.*

Now we prove the approximation guarantee of our mechanism. Given any bid vector $\boldsymbol{v}$, without loss of generality, we may assume $\boldsymbol{v} \in T_0$. Let $\boldsymbol{z}^*$ be the optimal solution of social welfare maximization for $\boldsymbol{v}$ and $\boldsymbol{x}^*$ be the solution output by our mechanism. In addition, let $\boldsymbol{y}^*$ be the optimal solution for the typical bid $E(D(\boldsymbol{v}))$. Then, by Corollary 3.6, we have $\boldsymbol{v}\cdot\boldsymbol{x}^* \geq \frac{1}{\beta}E(D(\boldsymbol{v}))\cdot\boldsymbol{x}^*$. Since our algorithm is maximum-in-range, allocation $\boldsymbol{x}^*$ is at least as good as the allocation by algorithm $\mathcal{A}$ with respect to typical bid vector $E(D(\boldsymbol{v}))$. Hence, we have $E(D(\boldsymbol{v}))\cdot\boldsymbol{x}^* \geq \frac{1}{\alpha}E(D(\boldsymbol{v}))\cdot\boldsymbol{y}^*$. Further, by optimality of $\boldsymbol{y}^*$ and Corollary 3.6, we have $E(D(\boldsymbol{v}))\cdot\boldsymbol{y}^* \geq E(D(\boldsymbol{v}))\cdot\boldsymbol{z}^* \geq \eta\left(1-\frac{\epsilon}{4}\right)\boldsymbol{v}\cdot\boldsymbol{z}^*$.

In all, we have $\boldsymbol{v}\cdot\boldsymbol{x}^* \geq \frac{\eta}{\alpha\beta}\left(1-\frac{\epsilon}{4}\right)\boldsymbol{v}\cdot\boldsymbol{z}^*$. Since we choose β and η such that $\beta/\eta = 1+\epsilon/2$, we have $\boldsymbol{v}\cdot\boldsymbol{z}^* \leq \alpha(1+\epsilon/2)\boldsymbol{v}\cdot\boldsymbol{x}^*/(1-\epsilon/4) \leq \alpha(1+\epsilon)\boldsymbol{v}\cdot\boldsymbol{x}^*$. This completes our analysis.

4 General Single-Parameter Mechanism Design

In the previous section, we study the black-box reductions in the symmetric single-parameter setting. However, many interesting mechanism design problems do not satisfy the symmetric property, hence to study black-box reductions in a general single-parameter setting is an important extension of our result.

In Section 4.1, we derive a significant approximability gap between maximum-in-range mechanisms and approximation algorithms in the most general single-parameter setting. To some extent, this suggests the difficulty in designing factor-preserving black-box reductions. However, we show in Section 4.2 that for the important subclass of *downward-closed* problems, we will still be able to establish a factor-preserving black-box reduction that converts a certain type of algorithms to truthful in expectation mechanisms.

4.1 General case: limitation of MIR mechanisms

In this section, we establish a novel relation between single-parameter mechanism design (SPMD) and maximum constraint satisfaction problems (MaxCSP). This relation shows that the computational ability of MIR mechanisms is much more restricted than the ability of general approximation algorithms – in the sense that, approximation ratio of a MIR mechanism for some SPMD problem can be arbitrarily worse than that of the best approximation algorithm for the "corresponding" MaxCSP problem.

Specifically, for every MaxCSP problem Γ that is NP-Hard, we set up a corresponding SPMD problem Γ', mapping (which can be done in polynomial time) each instance $\mathcal{I} \in \Gamma$ to a profile of agent valuation $v_{\mathcal{I}}$, while $\mathsf{opt}_{\Gamma}(\mathcal{I}) = \mathsf{opt}_{\Gamma'}(v_{\mathcal{I}})$. For every (efficient) MIR mechanism, we show that unless NP $\subsetneq$ P/poly, the approximation guarantee of the mechanism (on Γ') can be no better than that of a random assignment for the corresponding MaxCSP problem Γ, and therefore is arbitrarily worse than the guarantee of the best approximation algorithms, for some carefully chosen Γ.

For the sake of exposition, we choose Γ to be Max k-AllEqual (which can be any MaxCSP problem, although the gap between performance of MIR mechanisms and that of approximation algorithms might be different). The Max k-AllEqual problem is defined as follows.

Definition 1 (Max k-AllEqual)**.** *Given a set C of clauses of the form $l_1 \equiv l_2 \equiv \cdots \equiv l_k$ (k constant), where each literal l_i is either a Boolean variable x_j or its negation $\bar{x}_j$. The goal is to find an assignment to the variables x_i so as to maximize the number of satisfied clauses.*

The Max k-AllEqual problem is NP-Hard. In fact, it is NP-Hard to approximate Max k-AllEqual problem within a factor of $2^{c\sqrt{k}}/2^k$ for some constant $c > 0$, according to [24, 17, 11]. On the algorithmic side, there is an $\Omega(k/2^k)$-approximation algorithm for Max k-AllEqual shown in [6]. The algorithm based on SDP-relaxation and randomized rounding, but it can be efficiently derandomized by embedding the SDP solution to a low dimensional space, via derandomized Johnson-Lindenstrauss transformation [18, 8, 20].

CSP-based hard instance for MIR mechanisms. We describe the corresponding SPMD problem for Max k-AllEqual as follows. For a Max k-AllEqual problem with n variables, we set up $M = (2n)^k$ agents in $M_{\textsc{Max}\ k\text{-}\textsc{AllEqual}}$, each corresponding to a clause $c : l_1 \equiv l_2 \equiv \cdots \equiv l_k$. Recall that in a SPMD problem, each agent's valuation is a single real number that specifies its utility for being served. We conclude the description of our hard instance by defining the winner set, i.e. the set of feasible subset of agents being served. For any Boolean assignment $x : [n] \to \{\mathsf{true}, \mathsf{false}\}$, let $C(x) \subseteq [M]$ be the set of clauses that are satisfied by x. We define the set of feasible allocation functions $Y \subseteq \{\boldsymbol{y} : [M] \to \{0,1\}\}$ to be $Y = \{\mathbf{1}_{C(x)} | x : [n] \to \{\mathsf{true}, \mathsf{false}\}\}$.

Given a Max k-AllEqual instance $\mathcal{I}$ with set C of clauses, define the valuation function for the agents, $\boldsymbol{v}_{\mathcal{I}} = \boldsymbol{v} : [M] \to \{0,1\}$, to be the indicator function of C, i.e. $\boldsymbol{v}(c) = \mathbf{1}_C(c) = \begin{cases} 1 & c \in C \\ 0 & \text{otherwise} \end{cases}$.

Note that here we assume that every clause appears at most once in C. But the hard instance can be easily generalized to weighted case, by letting $\boldsymbol{v}(c)$ be the weight of clause c.

Analysis. It's easy to check the following fact.

Fact 4.1. $\mathsf{opt}(\mathcal{I}) = \max_{x:[n]\to\{\mathsf{true},\mathsf{false}\}} \left\{\boldsymbol{v} \cdot \mathbf{1}_{C(x)}\right\} = \max_{y \in Y} \{\boldsymbol{v} \cdot \boldsymbol{y}\} = \mathsf{opt}(v_{\mathcal{I}})$.

Now, we prove that there is a significant gap between the approximation guarantee of any MIR mechanism and that of the approximation algorithms. The following theorem shows that MIR mechanism performs $\Omega(k)$ times worse than the approximation algorithm for the corresponding algorithmic task, for any constant $k > 0$.

Theorem 4.2. *Assuming* NP $\not\subseteq$ P/poly*, there is no polynomial time MIR mechanism with approximation ratio better than* $2(1+\epsilon)/2^k$*, for any constant* $\epsilon > 0$*.*

The proof of Theorem 4.2 consists of two steps. Assuming there is an MIR mechanism with range $\mathcal{R}$ achieving $2(1+\epsilon)/2^k$ approximation guarantee, we firstly show that $\mathcal{R}$ needs to be exponentially large. Then we use Sauer-Shelar Lemma to argue that when $\mathcal{R}$ is sufficiently large, it must cover all possible assignments for a constant fraction of the n variables in Max k-AllEqual, and we can use this mechanism exactly solve Max k-AllEqual problem on this fraction of variables, which is NP-Hard.

The above technique was first introduced in [5] to show the inapproximability result in combinatorial auctions. However, their construction relies on the complicated private structures of agents' valuations, hence does not apply in our problem. Our approach can be viewed as a novel generalization of their technique in single-parameter mechanism design.

Now we proceed to the first step by proving the following lemma.

Lemma 4.3. $|\mathcal{R}| > \exp\left(\dfrac{\epsilon^2 n}{4k2^k}\right)$.

Proof. Consider a random Max k-AllEqual instance $\mathcal{I}$ generated as follows. Partition the n variables into $\lceil n/k \rceil$ groups, each of which contains exactly k variable. For every group with variables $x_{i_1}, x_{i_2}, \cdots, x_{i_k}$, generate a clause $l_1 \equiv l_2 \equiv \cdots \equiv l_k$ where l_j is chosen from x_{i_j} and $\bar{x_{i_j}}$ uniformly and independently from other choices. Note that $\mathcal{I}$ has $\lceil n/k \rceil$ clauses, and is always satisfiable, i.e. $\mathsf{opt}(\mathcal{I}) = \lceil n/k \rceil$.

For every allocation $\boldsymbol{y} \in \mathcal{R}$,$\mathbf{E}\left[\boldsymbol{v}_{\mathcal{I}} \cdot \boldsymbol{y}\right] = \frac{2\lceil n/k \rceil}{2^k}$, since each clause is satisfied with probability $2/2^k$. Since the clauses are generated independently, by Chernoff bound, for $\epsilon < 1$, we have $\mathbf{Pr}\left[\boldsymbol{v}_{\mathcal{I}} \cdot \boldsymbol{y} > (1+\epsilon) \cdot \frac{2\lceil n/k \rceil}{2^k}\right] < \exp\left(-\frac{\epsilon^2}{3} \cdot \frac{2\lceil n/k \rceil}{2^k}\right) < \exp\left(-\frac{\epsilon^2 n}{4k2^k}\right)$. By the approximation guarantee of the mechanism and a union bound, we know that $|\mathcal{R}| > \exp(\epsilon^2 n/(4k2^k))$. □

In the second step, we make use of the following lemma.

Lemma 4.4 (Sauer-Shelah Lemma [25, 26]). *For any family S of subsets of a universe U, there is a subset $E \subseteq U$ of size $\Theta\left(\frac{\log|S|}{\log|U|}\right)$, such that for each subset $E' \subseteq E$, there is a $T \in S$ satisfying $E' = T \cap E$.*

Let $U = [n]$ be the universe, thus any subset T of U can be (one-to-one) corresponded to a Boolean assignment $x : [n] \to \{\mathsf{true}, \mathsf{false}\}$ with $x^{-1}(\mathsf{true}) = T$. Let S be the set of Boolean assignments whose $C(\cdot)$ image is in $\mathcal{R}$ (recall $C(x)$ is the set of all clauses satisfied by x), i.e. let $S = \{x^{-1}(\mathsf{true}) | \forall x : [n] \to \{\mathsf{true}, \mathsf{false}\} : \mathbf{1}_{C(x)} \in \mathcal{R}\}$. Since $\mathcal{R} \subseteq Y$ and by definition of Y (which is the set of feasible allocation functions), for every $\boldsymbol{y} \in R$, there is at least one x such that $\mathbf{1}_{C(x)} = \boldsymbol{y}$. Therefore $|S| \geq |\mathcal{R}| > \exp\left(\frac{\epsilon^2 n}{4k2^k}\right)$ (by Lemma 4.3). Now we apply Lemma 4.4. We get a subset $E \subseteq U = [n]$ of size $\Omega_k(\epsilon^2 n / \log n)$, such that for each $E' \subseteq E$, there is a $T \in S$ satisfying $E' = T \cap E$. Then we prove the following lemma.

Lemma 4.5. *For every* Max k-AllEqual *problem instance $\mathcal{I}'$ with $|E| = \Omega_k(\epsilon^2 n / \log n)$ variables, there is a polynomial time algorithm, outputting a valuation function v for $M = (2n)^k$ agents, such that social welfare achieved by the MIR mechanism with range $\mathcal{R}$ is exactly* $\mathsf{opt}(\mathcal{I}')$.

Proof. Let C' be the set of clauses given in $\mathcal{I}'$, rename the variables in $\mathcal{I}'$ so that their indices appear in E. Then let $\boldsymbol{v} = \mathbf{1}_{C'}$. We see that the social welfare achieved by MIR mechanism with range $\mathcal{R}$ is

$$\begin{aligned}
\max_{y \in \mathcal{R}} \{\boldsymbol{v} \cdot \boldsymbol{y}\} &= \max_{x:[n]\to\{\mathsf{true},\mathsf{false}\}:\mathbf{1}_{C(x)} \in R} \left\{\boldsymbol{v} \cdot \mathbf{1}_{C(x)}\right\} && \text{(definition of } Y\text{, and } \mathcal{R} \subseteq Y) \\
&= \max_{x:[n]\to\{\mathsf{true},\mathsf{false}\}:x^{-1}(\mathsf{true}) \in S} \left\{\boldsymbol{v} \cdot \mathbf{1}_{C(x)}\right\} && \text{(definition of } S) \\
&= \max_{x:E\to\{\mathsf{true},\mathsf{false}\}:\exists T \in S, x^{-1}(\mathsf{true}) = T \cap E} \left\{\boldsymbol{v} \cdot \mathbf{1}_{C(x)}\right\} && (v(c) = 0 \text{ if } c \text{ contains variables outside } E) \\
&= \max_{x:E\to\{\mathsf{true},\mathsf{false}\}} \left\{\boldsymbol{v} \cdot \mathbf{1}_{C(x)}\right\} && \text{(property of } E) \\
&= \mathsf{opt}(\mathcal{I}').
\end{aligned}$$

Together with the hardness of Max k-AllEqual, Lemma 4.5 implies Theorem 4.2. □

4.2 Downward-closed case: black-box reduction

In the previous approximability lower bound construction, we observe that the Max k-AllEqual problem studied does not satisfy the *downward-closed* property. Formally, a single-parameter problem is downward-closed if the following holds: let $\boldsymbol{x}$ be a feasible allocation, then any allocation $\boldsymbol{x}'$ obtained by decreasing one of the components of $\boldsymbol{x}$ to 0 is feasible as well. Problems that satisfy the *free disposal* property fall into this class. For example, the partially public combinatorial auction problem [13] is a downward-closed single-parameter mechanism design problem.

4.2.1 Relaxation-rounding algorithm

We will consider the single-parameter social welfare maximization problem with n agents as a discrete maximization problem with linear objective: $\max\{\boldsymbol{v} \cdot \boldsymbol{x} : \boldsymbol{x} \in C\}$, where C is a finite and downward-closed subsets of $[0,1]^n$. Let $\mathsf{opt}(\boldsymbol{v})$ denote the optimal value of this problem with respect to valuations $\boldsymbol{v}$. Since solving this problem optimally is usually computationally intractable, approximate algorithms are desirable. An important technique for designing approximate algorithms is the following *relaxation-rounding* approach (e.g. the LP-based algorithms for social welfare maximization [4, 12]).

A relaxation-rounding algorithm $\mathcal{A} = (C^*, \mathcal{A}^{\text{round}})$ consists of two parts.

- $C^* \supseteq C$ is a relaxed set of feasible allocations so that we can solve the relaxed constrained maximization problem $\max\{\boldsymbol{v} \cdot \boldsymbol{x} : \boldsymbol{x} \in C^*\}$ optimally in polynomial time (e.g. LP relaxation, SDP relaxation etc.). We will let $\mathsf{opt}^*(\boldsymbol{v})$ denote the optimal of this relaxed problem. A simple fact is that $\mathsf{opt}^*(\boldsymbol{v}) \geq \mathsf{opt}(\boldsymbol{v})$.
- $\mathcal{A}^{\text{round}}$ is a poly-time rounding algorithm that converts any feasible allocation $\boldsymbol{x}^* \in C^*$ of the relaxed problem into a feasible allocation $\boldsymbol{x} \in C$ of the original problem.

The relaxation-rounding algorithm $\mathcal{A} = (C^*, \mathcal{A}^{\text{round}})$ will first solve the relaxed constrained maximization problem optimally, then use the rounding algorithm to convert the optimal relaxed allocation into a feasible allocation of the original problem.

The rounding algorithm $\mathcal{A}^{\text{round}}$ is an α-rounding if for any agent valuations $\boldsymbol{v} \in \mathbb{R}_+^n$ and any relaxed allocation $\boldsymbol{x}^* \in C^*$, we have that $\alpha \mathcal{A}^{\text{round}}(\boldsymbol{x}^*) \cdot \boldsymbol{v} \geq \boldsymbol{x}^* \cdot \boldsymbol{v}$. It is easy to see that if the rounding algorithm is an α-rounding then the relaxation-rounding algorithm is α-approximate. Unfortunately, such α-approximate allocation algorithm does not always admit truthful mechanism.

4.2.2 MIR mechanism based on exact rounding

It turns out that if we can further impose the stronger α-exact-rounding property on the rounding algorithm, then we can construct an MIR mechanism on the range $\mathcal{R} = C^*/\alpha = \{\boldsymbol{x}^*/\alpha : \boldsymbol{x}^* \in C^*\}$. More concisely, a rounding algorithm is an α-exact-rounding if for any relaxed allocation $\boldsymbol{x}^* \in C^*$, we have $\alpha \, \mathbf{E}\left[\mathcal{A}^{\text{round}}(\boldsymbol{x}^*)_i\right] = \boldsymbol{x}_i^*$, that is, the expected allocation for each agent is rounded down by exactly an α factor. In this case, the allocation given by the algorithm is the social welfare maximizing allocation in range $\mathcal{R}$. And hence this allocation rule admits an MIR mechanism on this range.

At the first glance, the α-exact-rounding property seems unreasonably strong: It requires the rounding algorithm to round down each coordinate by exactly the same factor α. Surprisingly, this property is without loss of generality in the downward-closed case.

Lavi and Swamy [19] studied the case when C is a packing integral polytope and proved that any α-rounding algorithm can be converted into an α-exact-rounding algorithm and hence a α-approximate MIR mechanism.

We notice that the constraint of C being a packing integral polytope can be relaxed to arbitrary finite and downward closed-domain[1]. More concisely, the following strengthened theorem holds.

Theorem 4.6 (Implicit in [19])**.** *Suppose C is a downward-closed and finite set of feasible allocations, $(C^*, \mathcal{A}^{round})$ is a relaxation-rounding algorithm with an α-rounding. Then, we con convert $\mathcal{A}^{round}$ into a poly-time α-exact-rounding algorithm for the same relaxed constrained set C^*.*

We prove this theorem in Appendix A for completeness.

As a simple corollary, we get that there is a computationally efficient black-box reduction from mechanism design to relaxation-rounding based algorithm design in the downward-closed case.

Corollary 4.7. *Suppose C is a downward-closed constrained set, then we can convert any relaxation-rounding algorithm into an MIR mechanism with the same approximation ratio for social welfare.*

4.2.3 Further discussion on Lavi and Swamy's approach

We will conclude this section with a further discussion on the approach of exact rounding. Lavi and Swamy demonstrated in their original paper [19] the application of this approach and yielded truthful mechanisms with very good approximation ratio for a number of problems, including the combinatorial auction problems with general valuations, single-minded valuations, additive valuations. However, the technique does not extends to, for example, sub-modular valuations.

[1] The authors would like to thank Deeparnab Chakrabarty and an anonymous reviewer for suggesting that Lavi and Swamy's approach might work in this setting. In an earlier draft of this paper, we prove a slightly weaker result via a different approach.

The main obstacle is that the technique of exact rounding requires the rounding algorithm to work on arbitrary valuations. More concisely, if we want to restrict our attention to valuations that satisfies certain properties, we need to be able to encode these properties with C, the set of feasible allocations. For instance, there are natural polytopes for combinatorial auctions with general valuations, single-minded valuations, and additive valuations. To our knowledge, however, there is no such polytope that encodes the sub-modularity of valuation functions.

In contrast, in the single-dimension setting, the agent valuations have very simple structures by definition: Each agent's value is just a single real number. All the messy structures are hidden in C, the set of feasible allocation. Therefore, we can naturally adopt Lavi and Swamy's approach into this setting for every C that is finite and downward-closed.

5 Symmetric Multi-Parameter Mechanism Design

We discuss the single-parameter mechanism design problems in the previous sections. As a natural generalization, we consider the multi-parameter problems in this section. We will demonstrate how to extend the discretization technique to the symmetric multi-parameter domain.

As before, a problem is *symmetric* if $\boldsymbol{S} = (S_1, ..., S_n)$ is a feasible allocation implies that $\pi \circ \boldsymbol{S} = (S_{\pi(1)}, ..., S_{\pi(n)})$ is also a feasible allocation for any permutation π. Moreover, a mechanism design problem is Δ-dimension if the valuation of each agent i can be naturally represented by a Δ-dimension vector $\boldsymbol{u}_i = (u_{i1}, \dots, u_{i\Delta}) \in \mathbb{R}_+^\Delta$. We let $v(S, \boldsymbol{u})$ denote an agent's value of an allocation S when its valuation function is given by a Δ-dimension vector $\boldsymbol{u}$. We will assume that the problem satisfies the following properties:

- **Monotonicity.** For any $1 \le i \le n$, $\boldsymbol{S}$, $\boldsymbol{u}_i$ and $\boldsymbol{u}'_i$ such that $u_{ij} \ge u'_{ij}$ for any $1 \le j \le \Delta$, we have $v(S_i, \boldsymbol{u}_i) \ge v(S_i, \boldsymbol{u}'_i)$.
- **Sub-linear influence.** For any $1 \le k \le \Delta$, $\beta > 1$, $\boldsymbol{u}$ and $\boldsymbol{u}'$ such that for any $1 \le i \le n$, $u_{ij} = u'_{ij}$ for any $j \ne k$, and $u_{ik} \le \beta u'_{ik}$, we have $\mathsf{opt}(\boldsymbol{u}) \le \beta \mathsf{opt}(\boldsymbol{u}')$.
- **Negligible tail.** For any $\delta > 0$, let $\boldsymbol{u}_i^\delta$ be the tail-truncated values: $u_{ij}^\delta = u_{ij}$ if $u_{ij} \ge \delta \max_{s,t} u_{st}$ and $u_{ij}^\delta = 0$ otherwise. For any constant $\epsilon > 0$, there is a polynomially small $\delta > 0$, so that for any allocation $\boldsymbol{S}$ and any values $\boldsymbol{u}_i$'s, we have $(1+\epsilon)\sum_{i=1}^n v(S_i, \boldsymbol{u}_i^\delta) \ge \sum_{i=1}^n v(S_i, \boldsymbol{u}_i)$.

These assumptions are without loss of generality in many mechanism design problems. For example, consider the following:

- **Multi-item auction.** In multi-item auctions, we consider n agents and m different types of items, each of which has a finite supply. Each agent i has a private m-dimension vector of values $\boldsymbol{u}_i = (u_{i1}, \dots, u_{im})$. Agent i's value of a bundle S with x_j items of type j, $1 \le j \le m$, is $v(S, \boldsymbol{u}_i) = \sum_{j=1}^m x_j u_{ij}$. This is a m-dimensional problem that satisfies our assumptions.
- **Combinatorial auction.** In combinatorial auctions, we consider n agents and m different items. Each agent i has a private 2^m-dimension vector $\boldsymbol{u}_i$ so that for each subset of items $S \in 2^{[m]}$, agent i's value of bundle S is $v(S, \boldsymbol{u}_i) = u_{iS}$. This is a 2^m-dimensional problem that satisfies our assumptions.

In the following discussion, we will focus on mechanism design problems with small dimension. Via techniques similar to those in Section 3, we can show that there exist factor-preserving reductions for the constant dimension and symmetric mechanism design problems as well, except that the running time is now quasi-polynomial in the input size.

Theorem 5.1. *For any Δ-dimension symmetric mechanism design problem Π where Δ is a constant, suppose $\mathcal{A}$ is an α-approximate algorithm, then for any constant $\epsilon > 0$, we can get an truthful and $(1+\epsilon)\alpha$-approximate mechanism that runs in quasi-polynomial time given $\mathcal{A}$ as a black-box.*

Alternatively, we can alleviate the running time by having greater degrade in the approximation factor. More concretely, given any approximation algorithm as a black-box, we can obtain a polynomial time mechanism with poly-logarithmic degrade in the approximation factor.

Theorem 5.2. *For any Δ-dimension symmetric mechanism design problem Π where Δ is a constant, suppose $\mathcal{A}$ is an α-approximate algorithm, then for any constant $\epsilon > 0$, we can get a truthful and $\alpha\,\mathrm{polylog}$-approximate mechanism that runs in polynomial time given $\mathcal{A}$ as a black-box.*

The detail constructions of the above theorems are deferred to Appendix B.

Acknowledgments

The authors would like to thank Nina Balcan, Yang Cai, Deeparnab Chakrabarty, Florin Constantin, Gagan Goel, Venkatesan Guruswami, Vijay Vazirani and Jiajin Yu for many useful comments and helpful discussions.

References

[1] Moshe Babaioff, Ron Lavi, and Elan Pavlov. Single-value combinatorial auctions and implementation in undominated strategies. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithm (SODA)*, pages 1054–1063, New York, NY, USA, 2006. ACM. 2

[2] Maria-Florina Balcan, Avrim Blum, Jason D. Hartline, and Yishay Mansour. Mechanism design via machine learning. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 605–614, 2005. 2

[3] Xiaohui Bei and Zhiyi Huang. Bayesian incentive compatibility via fractional assignments. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), to appear*, 2011. 1, 2

[4] Patrick Briest, Piotr Krysta, and Berthold Vöcking. Approximation techniques for utilitarian mechanism design. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 39–48, New York, NY, USA, 2005. ACM. 2, 8

[5] Dave Buchfuhrer, Shaddin Dughmi, Hu Fu, Robert Kleinberg, Elchanan Mossel, Christos Papadimitriou, Michael Schapira, Yaron Singer, and Chris Umans. Inapproximability for vcg-based combinatorial auctions. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2010. 7

[6] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. Near-optimal algorithms for maximum constraint satisfaction problems. *ACM Trans. Algorithms*, 5:32:1–32:14, July 2009. 6

[7] Edward H. Clarke. Multipart pricing of public goods. *Public Choice*, 11(1), September 1971. 1

[8] Ilias Diakonikolas, Daniel M. Kane, and Jelani Nelson. Bounded independence fools degree-2 threshold functions. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2010. 7

[9] Shaddin Dughmi and Tim Roughgarden. Black-box randomized reductions in algorithmic mechanism design. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2010. 2

[10] Shaddin Dughmi, Tim Roughgarden, and Qiqi Yan. From convex optimization to randomized mechanisms: Toward optimal combinatotial auctions for submodular bidders. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC)*, 2011. 2

[11] Lars Engebretsen and Jonas Holmerin. More efficient queries in PCPs for NP and improved approximation hardness of maximum CSP. *Random Structures & Algorithms*, 33(4):497–514, 2008. 6

[12] Uriel Feige. On Maximizing Welfare When Utility Functions Are Subadditive. *SIAM Journal on Computing*, 39(1):122–142, 2009. 8

[13] Gagan Goel, Chinmay Karande, and Lei Wang. Single-parameter combinatorial auctions with partially public valuations. In *Proceedings of the 3rd International Symposium on Algorithmic Game Theory (SAGT)*, 2010. 1, 2, 8

[14] Theodore Groves. Incentives in teams. *Econometrica*, 41(4):617–631, July 1973. 1

[15] Jason Hartline, Robert Kleinberg, and Azarakhsh Malekian. Bayesian incentive compatibility via matchings. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), to appear*, 2011. 1, 2

[16] Jason D. Hartline and Brendan Lucier. Bayesian algorithmic mechanism design. In *Proceedings of the 42nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 301–310, New York, NY, USA, 2010. ACM. 1, 2

[17] Johan Håstad and Avi Wigderson. Simple analysis of graph tests for linearity and pcp. *Random Structures and Algorithms*, 22(2):139–160, 2003. 6

[18] Daniel M. Kane and Jelani Nelson. A derandomized sparse Johnson-Lindenstrauss transform. *Arxiv preprint arXiv:1006.3585*, 2010. 7

[19] Ron Lavi and Chaitanya Swamy. Truthful and near-optimal mechanism design via linear programming. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 595–604, Washington, DC, USA, 2005. IEEE Computer Society. 2, 9, 13

[20] Raghu Meka and David Zuckerman. Pseudorandom generators for polynomial threshold functions. In *Proceedings of the 42nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 427–436, 2010. 7

[21] Roger Myerson. Optimal auction design. *Mathematics of operations research*, 6(1):58–73, Feb 1981. 3

[22] Noam Nisan and Amir Ronen. Algorithmic mechanism design. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing (STOC)*, pages 129–140, 1999. 1

[23] Christos Papadimitriou, Michael Schapira, and Yaron Singer. On the hardness of being truthful. In *In 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2008. 2

[24] Alex Samorodnitsky and Luca Trevisan. A PCP characterization of NP with optimal amortized query complexity. In *Proceedings of the 32nd Annual ACM symposium on Theory of computing (STOC)*, pages 191–199. ACM, 2000. 6

[25] Norbert Sauer. On the density of families of sets. *Journal of Combinatorial Theory, Series A*, 13(1):145–147, 1972. 8

[26] Saharon Shelah. A combinatorial problem; stability and order for models and theories in infinitary languages. *Pacific Journal of Mathematics*, 41(1):247–261, 1972. 8

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

[28] Jan Vondrak. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of the 40th Annual ACM Symposium on Theory of computing (STOC)*, pages 67–74, New York, NY, USA, 2008. ACM. 1

A Proof of Theorem 4.6

Proof of Theorem 4.6. Let $C = \{\boldsymbol{x}^l : 1 \le l \le K\}$ where K is a finite number. We first show that given any $\boldsymbol{x}^* \in C^*$, we can find a convex combination of vectors in C that dominates $\frac{\boldsymbol{x}^*}{\alpha}$ componentwise. To be precise, consider the following (LP1) and its dual (LP2):

$$\begin{aligned} \min \sum_{l\in[K]} \lambda_l & & \text{(LP1)} \\ s.t. \sum_{l\in[K]} \lambda_l x_i^l \ge \frac{x_i^*}{\alpha} & & \forall i \in [n], \\ \sum_{l\in[K]} \lambda_l \ge 1, & & \\ \lambda_l \ge 0 & & \forall l \in [K]. \end{aligned}$$

$$\begin{aligned} \max \frac{1}{\alpha} \sum_{i\in[n]} x_i^* v_i + z & & \text{(LP2)} \\ s.t. \sum_{i\in[n]} x_i^l v_i + z \le 1 & & \forall l \in [K], \\ v_i \ge 0 & & \forall i \in [n]. \end{aligned}$$

Similar to [19], we can prove the following:

Claim A.1. *If $\boldsymbol{v}, z$ is feasible, then we have $\frac{1}{\alpha}\sum_{i\in[n]} x_i^* v_i + z \le 1$. Furthermore, if this inequality is reversed, one can use $\mathcal{A}^{round}$ to find a violated constraint in (LP2) in polynomial time.*

As a result, we conclude that there exists an optimal solution $(\lambda_l)_{l\in[K]}$ of (LP1) with polynomial support $\lambda_1, ..., \lambda_p > 0$ where $p = \text{poly}(n)$ and $\sum_{1\le l\le p} \lambda_l = 1$.

For each $i \in [n]$, we define

$$q_i = \frac{\frac{x_i^*}{\alpha}}{\sum_{1\le l\le p} \lambda_l x_i^l} \le 1.$$

and we define a vector $\overline{\boldsymbol{x}}^l$ as follows: $\overline{x}_i^l = q_i x_i^l$. Since C is downward-closed and $\boldsymbol{x}^l \in C$ for all $1 \le l \le p$, we have $\overline{\boldsymbol{x}}^l \in C$ for all $1 \le l \le p$.

Now we construct a rounding algorithm $\mathcal{A}^{\text{exact}}$ as follows: We first choose a vector $\boldsymbol{x}^l$ with probability λ_l for $1 \le l \le p$; Then we obtain a random vector $\overline{\boldsymbol{x}}^l$ by

$$\overline{x}_i^l = \begin{cases} x_i^l & \text{with probability } q_i \\ 0 & \text{with probability } 1 - q_i \end{cases}$$

It is easy to see that $\mathcal{A}^{\text{exact}}$ is a polynomial time α-exact-rounding algorithm. This completes the proof. □

B Ommited Construction in Section 5

In this section, we will provide the detail construction of the claimed reductions in Section 5. In Section B.1, we show a factor-preserving black-box reduction that converts any approximation algorithm to a truthful mechanism with quasi-polynomial running time. Alternatively, in Section B.2, we alleviate the running time by degrading the approximation ratio and we provide a polynomial time reduction with poly-logarithmic degrade in approximation factors.

B.1 Quasi-polynomial time reduction

In this section, we prove the following theorem:

Theorem B.1 (Theorem 5.1 restated)**.** *For any Δ-dimension symmetric mechanism design problem Π where Δ is a constant, suppose $\mathcal{A}$ is an α-approximate algorithm, then for any constant $\epsilon > 0$, we can get an truthful and $(1+\epsilon)\alpha$-approximate mechanism that runs in quasi-polynomial time given $\mathcal{A}$ as a black-box.*

Let us outline the quasi-polynomial time reduction for Δ-dimension mechanism design problems. We will consider MIR mechanisms.

Suppose $1 > \delta > 0$ is polynomially small and satisfies that $(1+\epsilon/2\Delta)\sum_{i=1}^n v(S_i, \boldsymbol{u}_i^\delta) \geq \sum_{i=1}^n v(S_i, \boldsymbol{u}_i)$ for any allocation $\boldsymbol{S}$ and any valuation profile $\boldsymbol{u}_i$. Such δ always exists by the assumption of negligible tail. For any $\boldsymbol{\eta} = (\eta_1, \dots, \eta_\Delta) \in \{0, -1, \dots, (2\Delta/\epsilon)\ln\delta\}^\Delta$, let $\boldsymbol{u}^{\boldsymbol{\eta}}$ denote the valuation $(e^{\epsilon\eta_1/2\Delta}, \dots, e^{\epsilon\eta_\Delta/2\Delta})$. We will let $U = \{\boldsymbol{u}^{\boldsymbol{\eta}} : \boldsymbol{\eta} \in \{0, -1, \dots, (2\Delta/\epsilon)\ln\delta\}^\Delta\}$ denote the set of such valuations. We will assume that there is a canonical total order $(\preceq, U)$ for U[2]. We will consider an MIR mechanism with range $\mathcal{R} = \{\pi \circ \mathcal{A}(\boldsymbol{u}_1, \dots, \boldsymbol{u}_n) : \boldsymbol{u}_1 \preceq \cdots \preceq \boldsymbol{u}_n \in U\,,\, \pi \in \Pi_n\}$, where Π_n is the set of permutations over $[n]$.

Computational efficiency. Given an allocation $\boldsymbol{S} = (S_1, \dots, S_n)$ and a profile $(\boldsymbol{u}_1, \dots, \boldsymbol{u}_n)$, we can compute the permutation $\pi \in \Pi_n$ that maximizes $\sum_{i=1}^n v(S_{\pi(i)}, \boldsymbol{u}_i)$ by solving a bipartite max-weight matching problem. The total number of different valuation profiles $\boldsymbol{u}_1 \preceq \cdots \preceq \boldsymbol{u}_n \in U$ is at most $\left|[n]^U\right| = n^{O((2\Delta|\ln\delta|/\epsilon)^\Delta)} = 2^{O(\log n|\ln\delta|^\Delta)}$ since there is a compact description of such profile by counting the number of valuations of each type. Recall that δ is polynomially small, the MIR mechanism runs in quasi-polynomial time for any fixed constant Δ and ϵ. By further discretizing the number of valuation of each type, we can obtain an MIR mechanism that runs in $2^{O(\log\log n|\ln\delta|^\Delta)}$ time. Since this is only a minor improvement and the analysis is almost identical to that in Section 3, we omit the tedious details here.

Approximation ratio. For any valuation profile $\boldsymbol{u}$, we can without loss of generality assume that it is normalized such that $\max_{st} u_{st} = 1$. We will show how to map such a valuation profile $\boldsymbol{u}$ to a discretized profile $\boldsymbol{u}^* \in U$, such that any α-approximate allocation $\boldsymbol{S}^*$ for $\boldsymbol{u}^*$ is a $(1+\epsilon)\alpha$-approximate allocation for $\boldsymbol{u}$. By our choice of range, this will prove the desired approximation guarantee.

We will map one dimension at a time. For $0 \leq k \leq n$, we let

$$U_k = \{(\boldsymbol{u}_1, \dots, \boldsymbol{u}_n) : \forall 1 \leq i \leq n, 1 \leq j \leq k\,,\, u_{ij} \in \{1, e^{-\epsilon/2\Delta}, e^{-\epsilon/\Delta}, \dots, \delta\} \cup [0, \delta]\}\ .$$

In other word, for any profile of valuations $\boldsymbol{u}^k \in U_k$, each entry of $\boldsymbol{u}^k$ in the first k dimensions either equals a discretized value or is negligibly small.

For any profile of valuation $\boldsymbol{u}^k \in U_k$, we will map $\boldsymbol{u}^k$ to a profile $\boldsymbol{u}^{k+1} \in U^{k+1}$ defined as follows: For all $1 \leq i \leq n$, let $u_{ij}^{k+1} = u_{ij}^k$ for any $j \neq k+1$, and let $u_{i(k+1)}^{k+1} = \max\{e^{-\ell(\epsilon/2\Delta)} : e^{-\ell(\epsilon/2\Delta)} \leq u_{i(k+1)}^k\,,\, \ell \in \mathbf{Z}\}$.

Let $\boldsymbol{x}^k$ denote the optimal allocation for profile $\boldsymbol{u}^k$. Note that we only scale down each entry in dimension $k+1$ by at most $e^{-\epsilon/2\Delta}$. By the assumptions of monotonicity and sub-linear influence, for any α-approximate allocation $\boldsymbol{S}$ for $\boldsymbol{u}^{k+1}$, we have that

$$\sum_{i=1}^n v(S_i, \boldsymbol{u}_i^k) \geq \sum_{i=1}^n v(S_i, \boldsymbol{u}_i^{k+1}) \geq \frac{\mathsf{opt}(\boldsymbol{u}^{k+1})}{\alpha} \geq \frac{\boldsymbol{u}^{k+1} \cdot \boldsymbol{x}^k}{\alpha} \geq \frac{\boldsymbol{u}^k \cdot \boldsymbol{x}^k}{\alpha\, e^{\epsilon/2\Delta}} = \frac{\mathsf{opt}(\boldsymbol{u}^k)}{\alpha\, e^{\epsilon/2\Delta}}\ .$$

So any α-approximate allocation $\boldsymbol{S}$ for $\boldsymbol{u}^{k+1}$ is an $e^{\epsilon/2\Delta}\alpha$-approximate algorithm for $\boldsymbol{u}^k$. Note that $\boldsymbol{u} \in U = U_0$. By repeatedly applying the above procedure Δ times, we get a profile $\boldsymbol{u}^\Delta \in U^\Delta$ so that any α-approximate allocation for $\boldsymbol{u}^\Delta$ is an $e^{\epsilon/2}\alpha$-approximate allocation for $\boldsymbol{u}$.

[2] For example, we may sort the elements in U by the following canonical labeling: $\ell(\boldsymbol{u}^{\boldsymbol{\eta}}) = \eta_1 + \eta_2[1 + (2\Delta/\epsilon)\ln\delta] + \cdots + \eta_\Delta[1 + (2\Delta/\epsilon)\ln\delta]^{\Delta-1}$.

Let us consider the tail-truncated version $\boldsymbol{u}^*$ of $\boldsymbol{u}^\Delta$: For any $1 \leq i \leq n$ and $1 \leq j \leq \Delta$, $u^*_{ij} = u^\Delta_{ij}$ if $u^\Delta_{ij} \geq \delta$ and $u^*_{ij} = 0$ otherwise. Suppose $\boldsymbol{S}^\Delta$ is the optimal allocation for $\boldsymbol{u}^\Delta$. Consider any α-approximate allocation $\boldsymbol{S}$ for $\boldsymbol{u}^*$, by the assumption of negligible tail, we have

$$\sum_{i=1}^n v(S_i, \boldsymbol{u}^\Delta_i) \geq \sum_{i=1}^n v(S_i, \boldsymbol{u}^*) \geq \frac{\mathsf{opt}(\boldsymbol{u}^*)}{\alpha} \geq \frac{\sum_{i=1}^n v(S^\Delta_i, \boldsymbol{u}^*)}{\alpha} \geq \frac{\sum_{i=1}^n v(S^\Delta_i, \boldsymbol{u}^\Delta)}{(1+\epsilon/2\Delta)\alpha} \enspace .$$

So $\boldsymbol{S}$ is a $(1+\epsilon/2\Delta)\alpha$-approximate allocation for $\boldsymbol{u}^\Delta$. Hence, it is a $(1+\epsilon)\alpha$-approximate allocation for $\boldsymbol{u}$ since $e^{\epsilon/2}(1+\epsilon/2\Delta) < (1+\epsilon)$. Note that $\boldsymbol{u}^* \in U$. We prove the claimed approximation guarantee.

B.2 Polynomial time reduction with poly-logarithmic degrade

Finally, let us turn to the poly-time reduction with poly-logarithmic degrade in the approximation ratio. We will show the following theorem.

Theorem B.2 (Theorem 5.2 restated)**.** *For any Δ-dimension symmetric mechanism design problem Π where Δ is a constant, suppose $\mathcal{A}$ is an α-approximate algorithm, then for any constant $\epsilon > 0$, we can get a truthful and $\alpha\,\mathrm{polylog}$-approximate mechanism that runs in polynomial time given $\mathcal{A}$ as a black-box.*

Suppose $1 > \delta > 0$ is polynomially small and satisfies that $2\sum_{i=1}^n v(S_i, \boldsymbol{u}^\delta_i) \geq \sum_{i=1}^n v(S_i, \boldsymbol{u}_i)$ for any allocation $\boldsymbol{S}$ and any profile of valuations $\boldsymbol{u}$. Consider an MIR mechanism with range

$$\mathcal{R} = \left\{ \pi \circ \mathcal{A}(\overbrace{\boldsymbol{u}, \ldots, \boldsymbol{u}}^{k}, \boldsymbol{0}, \ldots, \boldsymbol{0}) : \boldsymbol{u} \in \{1, e^{-1}, e^{-2}, \ldots, \delta\}^\Delta \,,\, 1 \leq k \leq n \,,\, \pi \in \Pi_n \right\} \enspace .$$

Computational efficiency. Similar to the previous case, once we pick $\boldsymbol{u}$ and k, we can efficiently find $\pi \in \Pi_n$ that maximizes $\sum_{i=1}^k v(\mathcal{A}(\boldsymbol{u}, \ldots, \boldsymbol{u}, \boldsymbol{0}, \ldots, \boldsymbol{0})_{\pi(i)}, \boldsymbol{u}_i)$. The number of different choices of $\boldsymbol{u}$ and k is only $O(n\,|\ln\delta|^\Delta)$. So this MIR mechanism runs in polynomial time.

Approximation ratio. Similar to the previous case, for any profile of valuations $\boldsymbol{u}$, suppose we assume without loss of generality that $\max_{st} u_{st} = 1$, we will find a discretized profile $\boldsymbol{u}^*$ so that there is a $\boldsymbol{u}' \in \{1, e^{-1}, e^{-2}, \ldots, \delta\}^\Delta$, for any $1 \leq i \leq n$, either $\boldsymbol{u}^*_i = \boldsymbol{u}'$ or $\boldsymbol{u}^*_i = \boldsymbol{0}$. We will show that any α-approximate allocation for $\boldsymbol{u}^*$ is an $O(\alpha\,|\ln\delta|^\Delta)$-approximate allocation for $\boldsymbol{u}$. Hence, by our choice of range, we prove the claimed approximation guarantee.

Let us explain how to find such a discretized profile $\boldsymbol{u}^*$. We will first obtain a scaled profile $\hat{\boldsymbol{u}}$ by scaling down each entry of $\boldsymbol{u}$ to the closest inverse of powers of e. By our assumptions of monotonicity and sub-linear influence, we have that any α-approximate allocation for $\hat{\boldsymbol{u}}$ is an $e^\Delta\alpha$-approximate allocation for $\boldsymbol{u}$.

Next, we consider the tail-truncated version $\tilde{\boldsymbol{u}}$ of $\hat{\boldsymbol{u}}$ by dropping the insignificant entries that are smaller than δ. By our choice of δ, we have that any α-approximate allocation for $\tilde{\boldsymbol{u}}$ is a 2α-approximate allocation for $\hat{\boldsymbol{u}}$ and hence is a $2e^\Delta\alpha$-approximate allocation for $\boldsymbol{u}$.

Finally, for any $\boldsymbol{w} \in \{1, 1/e, \ldots, \delta\}^\Delta$, we let $\boldsymbol{u}^{\boldsymbol{w}}$ denote the profile obtained by dropping all entries in $\tilde{\boldsymbol{u}}$ that do not equal $\boldsymbol{w}$, that is, $\boldsymbol{u}^{\boldsymbol{w}}_i = \boldsymbol{w}$ if $\tilde{\boldsymbol{u}}_i = \boldsymbol{w}$ and $\boldsymbol{u}^{\boldsymbol{w}} = 0$ otherwise. It is easy to verify that $\mathsf{opt}(\tilde{\boldsymbol{u}}) \geq \sum_{\boldsymbol{w}} \mathsf{opt}(\boldsymbol{u}^{\boldsymbol{w}})$. Hence, at least one of the $\mathsf{opt}(\boldsymbol{u}^{\boldsymbol{w}})$'s contributes at least an $\Omega(1/\,|\ln\delta|^\Delta)$ fraction of $\mathsf{opt}(\tilde{\boldsymbol{u}})$. So by monotonicity, we get that any α-approximate allocation $\boldsymbol{S}$ for this $\boldsymbol{u}^{\boldsymbol{w}}$ is an $O(\alpha\,|\ln\delta|^\Delta)$-approximate allocation for $\tilde{\boldsymbol{u}}$:

$$\sum_{i=1}^n v(S_i, \tilde{\boldsymbol{u}}_i) \geq \sum_{i=1}^n v(S_i, \boldsymbol{u}^{\boldsymbol{w}}_i) \geq \frac{\mathsf{opt}(\boldsymbol{u}^{\boldsymbol{w}})}{\alpha} \geq \Omega\left(\frac{\mathsf{opt}(\tilde{\boldsymbol{u}})}{\alpha\,|\ln\delta|^\Delta}\right)$$

Hence, $\boldsymbol{S}$ is a $O(\alpha\,|\ln\delta|^\Delta)$-approximate allocation for $\boldsymbol{u}$. Note that $\boldsymbol{u}^{\boldsymbol{w}}$ is a valid discretized profile. This completes our proof.