Improved Truthful Mechanisms for Combinatorial Auctions with Submodular Bidders

Sepehr Assadi*
Rutgers University
sepehr.assadi@rutgers.edu

Sahil Singla[†]
Princeton University & Institute for Advanced Study
singla@cs.princeton.edu

Abstract—A longstanding open problem in Algorithmic Mechanism Design is to design computationally-efficient truthful mechanisms for (approximately) maximizing welfare in combinatorial auctions with submodular bidders. The first such mechanism was obtained by Dobzinski, Nisan, and Schapira [STOC'06] who gave an $O(\log^2 m)$-approximation where m is number of items. This problem has been studied extensively since, culminating in an $O(\sqrt{\log m})$-approximation mechanism by Dobzinski [STOC'16].

We present a computationally-efficient truthful mechanism with approximation ratio that improves upon the state-of-the-art by an exponential factor. In particular, our mechanism achieves an $O((\log \log m)^3)$-approximation in expectation, uses only $O(n)$ demand queries, and has universal truthfulness guarantee. This settles an open question of Dobzinski on whether $\Theta(\sqrt{\log m})$ is the best approximation ratio in this setting in negative.

Keywords-Combinatorial Auctions, Truthful Mechanisms, Submodular Bidders

I. INTRODUCTION

In a combinatorial auction, m items are to be allocated between n bidders. Each bidder i has a valuation function v_i that describes their value $v_i(S)$ for every bundle S of items. The goal is to design a mechanism that finds an allocation A of items that maximizes the *social welfare*, which is defined as $\mathsf{val}(A) := \sum_i v_i(A_i)$ where A_i is the bundle allocated to bidder i. For a mechanism to be feasible, it needs to be *computationally-efficient*, i.e., run in $\text{poly}(m, n)$ time given access to certain queries to valuation functions, namely value queries and demand queries (see Section II for definitions). Mechanisms should also take into account the strategic behavior of the bidders. A mechanism in which the dominant strategy of each bidder is to reveal their true valuation in response to given queries is called *truthful*. For randomized mechanisms, we consider *universally truthful* mechanisms which are distributions over truthful mechanisms (this is a stronger guarantee than truthful-in-expectation considered also in the literature, e.g. [1], [2]; see Appendix A).

A "paradigmatic" [3]–[5], "central" [6], [7], and "arguably the most important" [8] problem in Algorithmic Mechanism Design is to design mechanisms for combinatorial auctions that are both computationally-efficient and truthful. At the root of this problem is the question of whether there is an inherent clash between computational-efficiency and truthfulness. On one hand, the celebrated VCG mechanism of Vickrey-Clarke-Groves [9]–[11] is a truthful mechanism for this problem that returns the welfare maximizing allocation. Alas, this mechanism requires finding the welfare maximizing allocation exactly, which is not possible in $\text{poly}(m, n)$ time for most classes of valuations. On the other hand, from a purely algorithmic point of view, constant factor approximation algorithms exist for many interesting classes of valuations, but they are no longer truthful.

A particular case of this problem that has received significant attention is when the valuation functions of all the bidders are *submodular*[1] (see, e.g. [3], [8], [12]–[19] and references therein). There is no poly-time algorithm for finding the optimal allocation of submodular bidders [15], [18], [20] and thus VCG mechanism is not computationally-efficient here. On the other hand, by using only value queries, a simple greedy algorithm can achieve a 2-approximation [12] and this can be further improved to $(\frac{e}{e-1})$-approximation [21], and even slightly better [22] by using demand queries. This leads to one of the earliest and the most basic questions in Algorithmic Mechanism Design:

> *How closely can the approximation ratio of truthful mechanisms for submodular bidders match what is possible from an algorithmic point of view that ignore strategic behavior?*

Already more than a decade ago, Dobzinski, Nisan, and Schapira [23] gave the first non-trivial answer to this question by designing an $O(\sqrt{m})$-approximation mechanism. This approximation ratio was soon after exponentially improved by the same authors [3] to $O(\log^2 m)$, which in turn was improved to $O(\log m \log \log m)$ by Dobzinski [8], and then to $O(\log m)$ by Krysta and Vöcking [17]. Breaking

[1] A valuation function v is submodular iff $v(S \cup T) + v(S \cap T) \le v(S) + v(T)$ for all S and T; see also Section II-A.

this logarithmic barrier remained elusive until a recent breakthrough of Dobzinski [19] that achieved an $O(\sqrt{\log m})$ approximation. Whether $\Theta(\sqrt{\log m})$ is the "correct" answer to this question or further improvements are possible was widely open thereafter [19].

Our Result.: We show that $\Theta(\sqrt{\log m})$ is *not* the correct answer to this question and in fact one can improve upon this ratio by an *exponential* factor.

Theorem 1. *There exists a universally truthful mechanism for combinatorial auctions with submodular valuations that achieves an approximation ratio of $O((\log \log m)^3)$ to the social welfare in expectation using polynomial number of value and demand queries.*

We shall note that our mechanism (as well as all previous ones in [3], [8], [17], [19]) actually works for the much broader class of *XOS* valuations (see Section II for definition). Our result reduces the gap between the approximation ratio of truthful mechanisms vs algorithms for submodular and XOS bidders significantly, namely, from $\text{poly}(\log(m))$ in prior work to $\text{poly}(\log \log(m))$.

Similar to [19], our result implies a $\text{poly}(m, n)$ time algorithm with *explicit access* to valuations, when valuations are *budget additive*, i.e., for every S, $v(S) = \min(b, \sum_{j \in S} v(\{j\}))$ for some fixed b. These valuations have been studied extensively in the past (see, e.g. [19], [24], [25]) and a simple reduction from Knapsack shows that it is NP-hard to compute a demand query for these valuations. Yet, similar to [19], our mechanism uses demand queries of a very specific form, and these can be computed in polytime. We omit the details here and instead refer the reader to [19, Section 6].

Our Techniques.: All previous work on this problem [3], [8], [17], [19], at their core, relied on the following key observation: to design truthful mechanisms for submodular or XOS bidders, "all" we need is to find "good" estimates of the *item prices* in an optimal allocation; the rest can be handled by a simple *fixed-price auction* using these prices. We also use this observation but depart from prior work in the following key conceptual way. Previous work mainly aimed to learn coarse-grained "statistics" about the prices, say, the range they should belong to [3], [8], and used these statistics to "guess" a small number of good prices (e.g., $O(1)$ prices in [3], [8], and $O(\sqrt{\log m})$ in [19]), whereas we instead strive to "learn" the entire price vector of items in a fine-grained way (at least for a large fraction of items). This fine-grained view is the key factor that allows us to get much more accurate prices and ultimately leads to the exponentially improved performance of our mechanism.

A cornerstone of our approach is a "learning process" which starts with a simple guess of item prices and *iteratively* refine this guess until it converges to suitable

prices for different items. Each iteration of this process involves running *several* fixed-price auctions with the prices learned so far and use the resulting allocations to refine our learned prices further. The key to the analysis of this mechanism is the "Learnable-Or-Allocatable Lemma" (Lemma 7): Roughly speaking, we prove that in each iteration of this process, we can either refine our learned prices significantly (Learnable), or the fixed-price auction with the currently learned prices already gets a high-welfare allocation (Allocatable). Thus, after a *few* iterations, the resulting prices have been refined enough to allow for a high-welfare allocation. One ingredient in the proof of this lemma is an interesting property of fixed-price auctions that stems from their greedy nature: if we run a fixed-price auction with a *random ordering* of bidders, either we obtain a high-welfare allocation or we sell almost all items (most likely to wrong bidders). Such a property was first proved (in a similar but not identical form) by Dobzinski [19] and is closely related to other similar results about greedy algorithms for maximum matching [26], matroid intersection [27], and constrained submodular maximization [28].

Further related work.: The gap between the approximation ratio of truthful mechanisms and general algorithms has been studied from numerous angles in the literature. It is known that algorithms that use only $\text{poly}(m, n)$ many value queries, or are poly-time in the input representation (for succinctly representable valuations) can achieve only $m^{\Omega(1)}$-approximation [7], [14], [14], [16], [29], [30] (the latter assuming RP $\neq$ NP). However, these results no longer apply for mechanisms that are allowed other natural types of queries, e.g., demand queries[2]. This has led the researchers to study the communication complexity of this problem that can capture arbitrary queries to valuations [18], [23], [31]–[38]. Although a clear path for proving a separation between the communication complexity of truthful mechanisms and general algorithms was shown recently in [35] (see also [37], [38]), no such separation is still known.

II. PRELIMINARIES

Notation.: We denote by N the set of bidders and by M the set of items. We use bold-face letters to denote vectors of prices and capital letters for allocations. For a price vector $\boldsymbol{p}$ and a set of items $M' \subseteq M$, we define $\boldsymbol{p}(M') := \sum_{j \in M'} p_j$. For an allocation $A = (A_1, \ldots, A_n)$, we sometimes abuse the notation and use A to denote the set of allocated items. A restriction of allocation A to bidders in $N' \subseteq N$ and items $M' \subseteq M$ is an allocation A' consisting of $A_i \cap M'$ for every $i \in N'$.

---

[2]Demand queries are quite natural from an economic point of view as they simply return the most valued bundle for the bidder at the given item prices; see Section II.

A. Submodular and XOS Valuation Functions

We make the standard assumption that valuation v_i of each bidder i is normalized, i.e., $v_i(\emptyset) = 0$, and monotone, i.e., $v_i(S) \leq v_i(T)$ for every $S \subseteq T \subseteq M$. We are interested in the case when bidders valuations are *submodular* and hence capture the notion of "diminishing marginal utility" of items for bidders. A valuation v is submodular iff $v(S \cup T) + v(S \cap T) \leq v(S) + v(T)$ for any $S, T \subseteq M$.

Submodular functions are a strict subset of *XOS* valuations also known as *fractionally additive* valuations (see, e.g. [12], [39]) defined as follows. A valuation a is additive iff $a(S) = \sum_{j \in S} a(\{j\})$ for every bundle S. A valuation function v is XOS iff there exists t additive valuations $\{a_1, \ldots, a_t\}$ such that $v(S) = \max_{r \in [t]} a_r(S)$ for every $S \subseteq M$. Each a_r is referred to as a *clause* of v. If $a \in \arg \max_{r \in [t]} a_r(S)$, then a is called a *maximizing clause* for S and $a(\{j\})$ is a *supporting price* of item j in this maximizing clause. We say that an allocation $A = (A_1, \ldots, A_n)$ of items to n bidders with XOS valuation is *supported* by prices $q = (q_1, \ldots, q_m)$ iff each q_j is a supporting price for item j in the maximizing clause of the bidder i to whom j is allocated, i.e., $j \in A_i$.

Query access to valuations.: Since valuations have size exponential in m, a common assumption is that valuations are specified via certain queries instead, in particular, value queries and demand queries. A value query to valuation v on bundle S reveals the value of $v(S)$. A demand query specifies a price vector p on items and the answer is the "most demanded" bundle under this pricing, i.e., a bundle $S \in \arg \max_{S'} \{v(S') - p(S')\}$.

B. A Fixed-Price Auction

We use a standard fixed-price auction as a subroutine in our mechanism. For an *ordered* set N of bidders, M of items, and a price vector p, FixedPriceAuction(N, M, p) is defined as follows.

FixedPriceAuction(N, M, p)

1) Iterate over the bidders i of the ordered set N in the given order:
   a) Allocate $A_i \in \arg \max_{S \subseteq M} \{v_i(S) - p(S)\}$ to bidder i and update $M \leftarrow M \setminus A_i$.
2) Return the allocation $A = (A_1, \ldots, A_n)$.

It is easy to see that FixedPriceAuction can be implemented using one demand query per bidder. Its truthfulness is also easy to check as bidders have no influence on the pricing mechanism.

The following lemma gives a key property of this auction used in our proofs. Variants of this lemma have already appeared in the literature, e.g., in [3], [8], [19], [40], [41] (although we are not aware of this particular statement). For completeness, we prove this lemma in Appendix B.

Lemma 2. *Let* $\delta < 1/2$ *and define* $A :=$ FixedPriceAuction(N, M, p). *Suppose* O *is an allocation with supporting prices* q *and* M^* *is the set of items j with* $\delta \cdot q_j \leq p_j < \frac{1}{2} \cdot q_j$. *Then,* $\mathsf{val}(A) \geq \delta \cdot q(M^*)$.

III. THE HIGH-LEVEL OVERVIEW

We describe our mechanism using three parameters $\alpha := \Theta(1)$, $\beta := O(\log \log m)$, and $\gamma := \Theta(\alpha\beta)$. Let $O = (O_1, \ldots, O_n)$ be an optimal allocation with welfare OPT and $q = (q_1, \ldots, q_m)$ be its supporting prices (obviously, O and q are unknown). For now, let us assume that every q_j belongs to $\{1, \gamma, \gamma^2, \ldots, \gamma^K\}$, for some $K = O(\log m)$ (and hence prices are roughly poly(m) large).

The crux of our mechanism is to "learn" q, namely, find another price vector p such that for some subset $C \subseteq M$ with $q(C) \approx \mathsf{val}(O)$, p *point-wise* γ-approximates q for items in C (i.e., within a multiplicative factor of γ). Having learned such prices, we can run a fixed-price auction with prices p, and by Lemma 2, obtain an allocation with welfare $\approx \gamma \cdot \mathsf{val}(O)$.

In order to obtain the price vector p, we start with a rough guess $p^{(1)}$ for what prices should be (say, all ones), and update our guess over (at most) β *iterations*. In each iteration $i \in [\beta]$, we use the prices $p^{(i)}$ learned so far to find α new price vectors $p_1^{(i)}, \ldots, p_\alpha^{(i)}$, and "explore" for *each* item $j \in M$ which of these α vectors best represents its price in q, and then assign that price to item j in $p^{(i+1)}$. We continue this for β iterations until we converge to the desired price vector $p := p^{(\beta+1)}$, or we decide along the way that the prices learned so far are already "good enough". There are three main questions to answer here: (i) how to choose which prices to explore in each iteration, (ii) how to explore a new price for each item, and finally (iii) how to implement all this in a truthful (and computationally-efficient) manner. We elaborate on each part below.

Part (i) – which prices to explore.: This question can be best answered from the perspective of a single item $j \in M$. Originally, we set $p_j^{(1)} \in p^{(1)}$ to be 1, and so with our assumption that $q_j \in \{1, \gamma, \ldots, \gamma^K\}$, price $p_j^{(i)}$ will (γ^K)-approximate $q_j \in q$. We want $p_j^{(2)}$ to $(\gamma^{K/\alpha})$-approximate q_j in the next iteration. Thus, we simply need to check for every $\ell \in \{0, \ldots, \alpha - 1\}$, whether $q_j \geq \gamma^{\ell \cdot K/\alpha}$ or not (using part (ii) below). By picking the largest ℓ^* for which this is true, we can get a $(\gamma^{K/\alpha})$-approximation to q_j. As such, for each item, there are only α choices of prices that we need to explore next, which allows us to devise price vectors $p_1^{(1)}, \ldots, p_\alpha^{(1)}$ accordingly. We repeat the same idea for later iterations as well, maintaining that in iteration i, price $p_j^{(i)} \in p^{(i)}$ will $(\gamma^{K/\alpha^{i-1}})$-approximate q_j,

and use α prices as before in $\boldsymbol{p}_1^{(i)}, \ldots, \boldsymbol{p}_\alpha^{(i)}$ to update this to a (γ^{K/α^i})-approximation for the next iteration. This way, after $\beta = O(\log \log m)$ iterations, we obtain $p_j^{(\beta+1)}$ that γ-approximates q_j as desired. See Figure 1 for an illustration.

In the above discussion, we talked about an item j as if its price is learned correctly throughout (i.e., $p_j^{(i)}$ is $(\gamma^{K/\alpha^{i-1}})$-approximating q_j for all $i \in [\beta+1]$). Our mechanism cannot guarantee this property for every item (but rather for most of them). Moreover, we are also not able to decide which items have been correctly priced, so we simply treat all items as being priced correctly in the mechanism and perform the above process for them. This means that for some items, their price may have been learned incorrectly in some iteration; so we conservatively ignore their contribution from now on in the analysis. A key part of our analysis is to show that this does not hurt the performance of the mechanism by much, namely, $\boldsymbol{q}(C)$ is still a good approximation to $\mathsf{val}(O)$, where C is the set of items for which we learn their prices correctly.

Part (ii) – how to explore a new price.: For this part, we build on a key idea from [19] in using fixed-price auctions themselves as a "proxy" for determining correctness of a guess for item prices. The idea is as follows: suppose we run a fixed-price auction with prices $\boldsymbol{p}_\ell^{(i)}$ for $\ell \in [\alpha]$ that we want to explore in an iteration i. As these prices may be very far from $\boldsymbol{q}$ yet, there is no guarantee that this auction returns a high-welfare allocation. However, if we choose the ordering of bidders *randomly*, then the *only way* this auction does not succeed in outputting a high-welfare allocation is because it sold *almost all* the items at the current prices (most likely to wrong bidders). Hence, an item getting sold in a certain fixed-price auction is a "good indicator" that its price in $\boldsymbol{q}$ is at least as high as the price used in this fixed-price auction. Such an idea was used in [19] to narrow down the range of item prices from $O(\log m)$ values to $O(\sqrt{\log m})$, which in turn allows the mechanism to simply guess a correct price for each item and achieves an $O(\sqrt{\log m})$-approximation.

We take this idea to the next step to obtain our Learnable-Or-Allocatable Lemma (Lemma 7). Roughly speaking, we show that in each iteration i, starting from the set $C^{(i)}$ of correctly priced items, either one of the α auctions for exploring prices will lead to an $O(\beta^2)$-approximate allocation, or after this iteration we will manage to further refine the prices of almost all items in $C^{(i)}$. I.e., we obtain a set $C^{(i+1)}$ with $\boldsymbol{q}(C^{(i+1)}) \approx \boldsymbol{q}(C^{(i)})$ and with $\boldsymbol{p}^{(i+1)}$ approximating prices $\boldsymbol{q}$ for $C^{(i+1)}$ much more accurately than $\boldsymbol{p}^{(i)}$ (as described in part (i)). Hence, either during one of the iterations there is an auction that gives us an $O(\beta^2)$-approximation, or we eventually end up with $\boldsymbol{p}^{(\beta+1)}$ that point-wise γ-approximates $\boldsymbol{q}$ for a large set of items $C^{(\beta+1)}$. Therefore, by ensuring $\boldsymbol{q}(C^{(\beta+1)}) = \Omega(\mathsf{OPT})$, a fixed-price auction with prices $\boldsymbol{p}^{(\beta+1)}$ gives a $\gamma = O(\alpha\beta)$-

approximation by Lemma 2.

This outline oversimplifies many details. Let us briefly mention two here. Firstly, running fixed-price auctions only help us in not *underpricing* items for the next iteration; we also need to take care of *overpricing*. This is handled by making sure there is a *gap* of γ between different prices explored so that not many overpriced items can be sold in an auction. Also while for the purpose of this discussion we simply assumed the existence of this gap, in the actual mechanism we need to *create* this gap using a basic randomization idea. Secondly, our mechanism has no way of determining (in a truthful way) which case of the Learnable-Or-Allocatable Lemma we are in. This means that there are $\alpha \cdot \beta$ auctions in the mechanism and any one of them may give an $O(\beta^2)$-approximation welfare. (If not, then we can learn the prices accurately and the final auction would be an $O(\alpha\beta)$-approximation.) The solution here is then to simply pick one of the $(\alpha\beta+1)$ auctions *uniformly at random* and allocate according to that. This way we succeed in finding a good auction with probability at least $1/\alpha\beta$ and hence, in expectation, we obtain an $O(\alpha\beta^3)$-approximation.

Part (iii) – how to ensure truthfulness.: Recall that a fixed-price auction is truthful primarily because the responses of the bidders has no effect on the price of their allocated bundle. However, our mechanism consists of multiple fixed-price auctions and the outcomes of these auctions do influence the prices for *later* iterations. As such, to ensure truthfulness, each bidder should only participate in the auctions of a single iteration. Hence, at the beginning of the mechanism, we randomly partition the bidders into $\beta + 1$ groups $N_1, \ldots, N_{\beta+1}$. Then, in each iteration i, we use the bidders in group N_i for fixed-price auctions with prices $\boldsymbol{p}_1^{(i)}, \ldots, \boldsymbol{p}_\alpha^{(i)}$ to learn prices $\boldsymbol{p}^{(i+1)}$, and in the final iteration we run one fixed-price auction with bidders $N_{\beta+1}$ and prices $\boldsymbol{p} = \boldsymbol{p}^{(\beta+1)}$.

This partitioning of bidders results in a key challenge: Our goal in learning the prices should actually be different from what was stated earlier. In particular, the auctions in each iteration i with bidders N_i should reveal the $\boldsymbol{q}$ prices of items allocated in O to bidders in $N_{>i} := N_{i+1}, \ldots, N_{\beta+1}$, *as opposed to* bidders in N_i. This is because we are no longer able to allocate any item to bidders in $N_1, \ldots, N_i$. We handle this also by our Learnable-Or-Allocatable Lemma. Instead of learning the set $C^{(i+1)}$ with $\boldsymbol{q}(C^{(i+1)}) \approx \boldsymbol{q}(C^{(i)})$ in the Learnable case, we have a more refined statement in which the LHS is replaced with $\boldsymbol{q}$ of items allocated *only* to bidders in $N_{>i}$. This in turn requires a delicate choice of parameters and analysis to balance out two opposing forces: on one hand, we need N_i to be large enough so that we can "extrapolate" the learned prices in auctions with N_i to $N_{>i}$ (in the Learnable case); on the other hand, each N_i should be small enough so that by the time we end up learning the prices, the contribution of the remaining bidders is still large
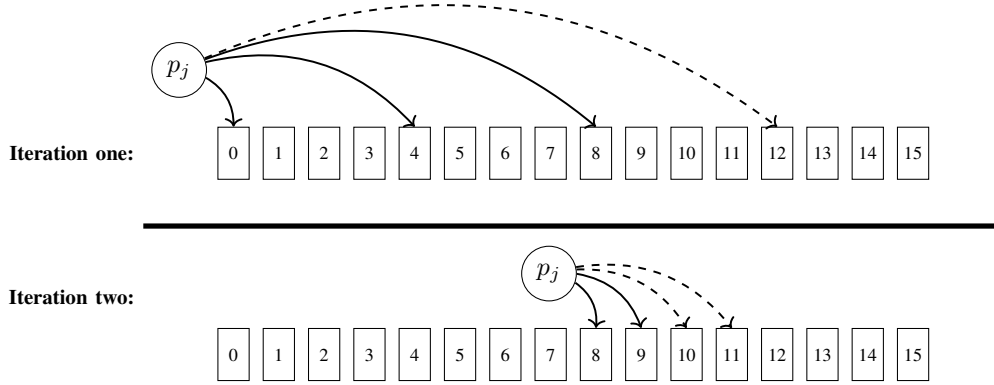
Figure 1: An illustration of the trajectory of the prices of a single item throughout the mechanism. Here, $\alpha = 4$ and $\beta = 2$. Each block i corresponds to price γ^i. Arrows correspond to the price of this item in the corresponding fixed-price auction; a solid arrow means the item was sold, while a dashed arrow means it was not. The learned price of this item in this example is γ^9.

enough.

Comparison to Dobzinski [19].: We conclude this section by comparing our work with the previous best mechanism of Dobzinski [19] that achieved an $O(\sqrt{\log m})$ approximation. As described in Part (ii), our mechanism builds on a key idea from [19] in using fixed price auctions as a proxy for obtaining "good" prices for items. On a high level, the main difference between the two works is that Dobzinski [19] uses fixed price auctions to "learn" item prices in a *single-shot*, but with a relatively poor accuracy. Instead, we use fixed price auctions *iteratively* in order to learn the prices of (most) items quite accurately.

Concretely, assuming that all prices $q_j \in \{1, \gamma, \ldots, \gamma^K\}$ for $K = O(\log m)$, Dobzinski's mechanism can be viewed as a special case of our mechanism by setting $\beta = 1$ and $\alpha = \sqrt{\log m}$: Dobzinski first uses a set N_1 of bidders to run $\alpha = \sqrt{\log m}$ auctions to learn the prices of items (to within an $O(\sqrt{\log m})$ factor), and then runs one more fixed price auction with these learned prices on bidders $N_{\beta+1} = N_2$. The final auction is then chosen randomly from these $\sqrt{\log m}+1$ auctions to ensure truthfulness. Considering both the prices are learned only to within an $O(\sqrt{\log m})$ factor and the final auction is chosen from $\sqrt{\log m} + 1$ auctions, the approximation ratio of this mechanism is $O(\sqrt{\log m})$. Our mechanism on the other hand learns prices of items in multiple iterations ($\beta = O(\log \log m)$ iterations) via the Learnable-Or-Allocatable Lemma. This allows us to both use a much smaller number of auctions (poly($\log \log m$) many), and at the same time learn prices much more accurately (again to within a poly($\log \log m$) factor), which ultimately leads to our improved approximation ratio of $O((\log \log m)^3)$.

IV. THE MAIN MECHANISM

We give our main mechanism for combinatorial auctions with XOS valuations in this section. In the following, we present our mechanism with an additional simplifying assumption (Assumption 1). This assumption is made primarily to simplify the presentation of the mechanism and its analysis and as we show in Section VI is not necessary.

Assumption 1. *We assume there exists two non-negative numbers $\psi_{\min} \leq \psi_{\max}$ such that:*

(i) for every valuation, supporting price of any item for any clause belongs to $\{0\} \cup [\psi_{\min} : \psi_{\max}]$;

(ii) the ratio of these numbers, denoted by $\Psi := \psi_{\max}/\psi_{\min}$, is bounded by some fixed poly(m).

We further assume that the mechanism is given $\psi_{\min}$ and $\psi_{\max}$ as input.

In the following, we first present a simple tree-structure, named the *price tree*, that we use in our mechanism for discretizing prices at different scales. We then describe the method with which we assign different bidders to different auctions run by our mechanism. Finally, we present our mechanism and prove its computational efficiency and universal truthfulness guarantees. The analysis of the approximation ratio of our mechanism—the main technical contribution of the paper—appears in the subsequent section.

Parameters:: We define and use the following parameters in our mechanism.

- $\alpha := \Theta(1)$ – number of different auctions run in each iteration of our mechanism;
- $\beta := O(\log \log \Psi)$ – number of iterations in our mechanism;
- $\gamma := \Theta(\alpha\beta)$ – the accuracy to which we aim to learn

the true prices.

Moreover, the above parameters satisfy the following equations:

$$\alpha^{\beta+1} \geq \log_\gamma \Psi, \qquad 20\alpha\beta \leq \gamma \leq 30\alpha\beta. \qquad (1)$$

It is immediate to verify that one can choose α, β, γ satisfying all the above equations.

A. Price Trees and Their Properties

We define a simple tree-structure used for discretizing the range of prices in $[\psi_{\min} : \psi_{\max}]$ by our mechanism. The first part is a geometric partition of set of available prices as follows.

Definition 3. *We partition $[\psi_{\min} : \psi_{\max}]$ into $t := \lceil \log_\gamma \Psi \rceil$* **bins** $B_1, \ldots, B_t$ *where values inside each B_i are within a factor γ of each other. We use* price(B_i) *to denote the min value in B_i.*

We now use the concepts of bins to define a multi-level partitioning of $[\psi_{\min} : \psi_{\max}]$.

Definition 4 (Price Tree). *A* **price tree** $\mathcal{T}$ *is a rooted tree in which each node z is assigned two attributes: (i)* bins(z) *which is a subset of bins $B_1, \ldots, B_t$ with consecutive indices, and (ii)* price(z) *which is the value of* price(B_i) *where B_i is the smallest indexed bin inside* bins(z)*. The tree $\mathcal{T}$ satisfies the following properties:*

- *For the root z_r of $\mathcal{T}$,* bins(z_r) $:= (B_1, \ldots, B_t)$.
- $\mathcal{T}$ *has t leaf-nodes where the i-th left most leaf-node z_i of $\mathcal{T}$ has* bins(z_i) $= B_i$.
- *Every non-leaf node z of $\mathcal{T}$ has α children $z_1, \ldots, z_\alpha$ such that* bins(z_1) *contains the first α fractions of* bins(z), bins(z_2) *contains the second α fraction, and so on.*

By the choice of $\alpha^{\beta+1} \geq t$, the number of levels in $\mathcal{T}$ is $\beta + 1$ (see Figure 2 for an illustration).

A price tree $\mathcal{T}$ gives a nested partitioning of the range $[\psi_{\min} : \psi_{\max}]$ into $\beta + 1$ levels with different granularities. We say that a price p *belongs* to a node z of $\mathcal{T}$ iff p appears in one of the bins in bins(z); moreover, if $p = $ price(z), then we say p *strongly belongs* to z.

Definition 5. *We say a price vector $\boldsymbol{p} = (p_1, \ldots, p_m)$ is a* **level-i price vector** *iff every p_j strongly belongs to some node z_j in level i of $\mathcal{T}$.*

We assign α **canonical level-$(i+1)$ price vectors** $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_\alpha$ *to a level-i price vector $\boldsymbol{p}$, where in $\boldsymbol{p}_k = (p_1', \ldots, p_m')$ each p_j' strongly belongs to the k-th child of z_j to which p_j strongly belongs.*

Modified price trees.: Using price trees in our mechanism directly is problematic primarily because it is possible

that a price $p \in B_i$ for some bin B_i is actually closer to price(B_{i+1}) than price(B_i), hence making the learning of the correct bin for p not feasible.

To fix this issue, we consider the following two modified price trees $\mathcal{T}^o$ and $\mathcal{T}^e$ instead: $\mathcal{T}^o$ is a subtree of $\mathcal{T}$ obtained by retaining only the *odd* indexed bins $B_1, B_3, \ldots$ in bins of $\mathcal{T}$; $\mathcal{T}^e$ is defined analogously by retaining all *even* indexed bins. In our mechanism, we pick one of $\mathcal{T}^o$ or $\mathcal{T}^e$ at random and from there on, only consider the prices that belong to the bins that appear in the corresponding modified price tree. This way, for any two price p, p' that belong to two different nodes of the modified tree, there is at least a factor γ gap between p and p'.

B. Partitioning Bidders

Our main mechanism involves partitioning the set of bidders into $\beta+1$ different *groups* $N_1, \ldots, N_{\beta+1}$ and assigning them to different auctions throughout the mechanism. The partitioning of the bidders is done using the following procedure:

- `Partition`(N): Let $N' \leftarrow N$ and for $i = 1$ to β iterations: pick a random permutation of N' and insert the first $|N'|/(10\beta)$ bidders into N_i; update $N' \leftarrow N' \setminus N_i$. At the end, return $N_1, \ldots, N_\beta$ and $N_{\beta+1} := N'$.

C. Formal Description of the Mechanism

We are now ready to give our main mechanism under Assumption 1. The procedure `PriceUpdate` used in the mechanism is defined immediately after.

`PriceLearningMechanism`(N, M)

1) Let $(N_1, N_2, \ldots, N_{\beta+1}) := $ `Partition`(N).
2) Pick one of the modified trees $\mathcal{T}^o$ or $\mathcal{T}^e$ uniformly at random and denote it by $\mathcal{T}^\star$.
3) Let $\boldsymbol{p}^{(1)}$ be the (unique) level-1 (root) price of $\mathcal{T}^\star$. For $i = 1$ to β iterations:
   a) Let $\boldsymbol{p}_1^{(i)}, \ldots, \boldsymbol{p}_\alpha^{(i)}$ be the level-$(i + 1)$ canonical price vectors of $\boldsymbol{p}^{(i)}$ in $\mathcal{T}^\star$ (Definition 5).
   b) For $j = 1$ to α: run `FixedPriceAuction`$(N_i, M, \frac{\boldsymbol{p}_j^{(i)}}{2})$ and let $A_j^{(i)}$ be the allocation.
   c) W.p. $(1/\beta)$, pick $j^\star \in [\alpha]$ uniformly at random and return $A_{j^\star}^{(i)}$ as the final allocation; otherwise, let $\boldsymbol{p}^{(i+1)} := $ `PriceUpdate`$(A_1^{(i)}, \ldots, A_\alpha^{(i)}, \boldsymbol{p}_1^{(i)}, \ldots, \boldsymbol{p}_\alpha^{(i)})$, and continue.
4) Run `FixedPriceAuction`$(N_{\beta+1}, M, \frac{\boldsymbol{p}^{(\beta+1)}}{2})$ and return the allocation A^*.

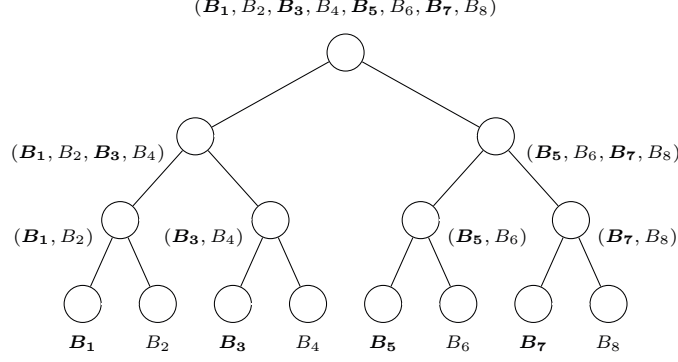The `PriceUpdate` procedure is done as follows:

Figure 2: An illustration of a price tree $\mathcal{T}$ with $\alpha = 2$, $\beta = 2$, and $t = 8$. By considering only the bold-face bins (with odd indices), we obtain the modified price tree $\mathcal{T}^o$.

- PriceUpdate($A_1^{(i)}, \ldots, A_\alpha^{(i)}, \boldsymbol{p}_1^{(i)}, \ldots, \boldsymbol{p}_\alpha^{(i)}$): For any item $j \in M$, we let p'_j be equal to $p_j \in \boldsymbol{p}_k^{(i)}$ where k is the *largest* index such that item j is allocated in $A_k^{(i)}$ (if j is never allocated, we set $k = 1$). Return $\boldsymbol{p}^{(i+1)} = (p'_1, \ldots, p'_m)$.

We shall right away remark that in PriceLearningMechanism, every price vector $\boldsymbol{p}^{(i)}$ computed in iteration i is a level-i price vector and hence the canonical price vectors defined in each iteration indeed do exist. We have the following theorem which is the main technical result of this paper.

Theorem 6. *For a combinatorial auction with n submodular (even XOS) bidders and m items, under Assumption 1, PriceLearningMechanism is universally truthful, uses $O(n)$ demand queries and polynomial time, and achieves an approximation ratio of $O((\log \log m)^3)$ in expectation.*

We remark that our mechanism in Theorem 6 only makes $O(1)$ queries to the valuation of each bidder, which is clearly optimal, and results in a highly efficient mechanism (computationally).

To see that PriceLearningMechanism is truthful, notice that every bidder b is participating in at most α fixed-price auctions of FixedPriceAuction for which the prices of items have already been fixed entirely independent of b's valuations (and responses). Moreover, for bidders in $N_1, \ldots, N_\beta$ that participate in more than one auction, the choice of which items (if any) they are being allocated across the auctions is entirely independent of the auction outcome and is determined by the random coin tosses in Line (3c). This still does not imply that truth telling is a dominant strategy as a bidder can "threat" another bidder by presenting wrong valuations in subsequent auctions they both participate in (see, e.g. [8], [19]). To fix this, we make each bidder b output the preferences in all fixed-price auctions b participates in *simultaneously* (or alternatively hide bidders responses from each other). As was observed in [8], [19] this ensures the truthfulness of the mechanism.

Computational efficiency of PriceLearningMechanism and the bound on number of demand queries follow immediately from the fact that each bidder is participating in at most $\alpha = \Theta(1)$ fixed-price auctions, each of which requires one demand query per bidder.

V. THE ANALYSIS OF MAIN MECHANISM

We now present the analysis of the approximation ratio of PriceLearningMechanism.

Notation.: To avoid confusion, throughout this section, we use "i" to index the iterations, "j" to index the auctions inside an iteration, "b" to index the bidders, and "ℓ" to index the items.

We pick an optimal allocation $O = (O_1, \ldots, O_n)$ of items with supporting prices $\boldsymbol{q} = (q_1, \ldots, q_m)$ and denote by OPT the welfare of this allocation. We further define $O^\star$ as the restriction of O to items with supporting prices in $\boldsymbol{q}$ that belong to the modified price tree $\mathcal{T}^\star$ chosen by the mechanism. Similarly, $\boldsymbol{q}^\star$ is defined by zeroing out the price of items in $\boldsymbol{q}$ that are not allocated by $O^\star$ and leaving the rest unchanged. We also define the following series of refinement of $\boldsymbol{q}$ based on the bidders in $N_1, \ldots, N_{\beta+1}$ and the choice of $\mathcal{T}^\star$. For every $i \in [\beta + 1]$, $\boldsymbol{q}^{(i)} = (q_1^{(i)}, \ldots, q_m^{(i)})$ is defined so that for every item $\ell \in M$, $q_\ell^{(i)} = 0$ iff ℓ is allocated in $O^\star$ to some bidder in $N_1, \ldots, N_{i-1}$ or is not allocated at all, and otherwise $q_\ell^{(i)} = q_\ell$ for $q_\ell \in \boldsymbol{q}^\star$.

Fix any iteration i and the price vector $\boldsymbol{p}^{(i)} = (p_1^{(i)}, \ldots, p_m^{(i)})$ obtained by the mechanism so far. We say an item $\ell \in M$ is *correctly priced* in iteration i iff $p_\ell^{(i)}$ belongs to the same level-i node in $\mathcal{T}^\star$ as $q_\ell^{(i)}$. Note that by construction, $p_\ell^{(i)}$ always strongly belongs to a node, and hence for any correctly priced item, we have $p_\ell^{(i)} \leq q_\ell^{(i)}$. We use $C^{(i)}$ to denote the set of all items that are correctly priced

throughout *all* iterations 1 to i. Hence, under this definition, $O^\star = C^{(1)} \supseteq C^{(2)} \supseteq \ldots \supseteq C^{(\beta+1)}$. The definition of the price tree ensures that by moving from $C^{(1)}$ towards $C^{(\beta+1)}$ we are learning the priced of correctly prices items more and more accurately.

Learnable-Or-Allocatable Lemma.: The goal of our mechanism is to learn a set $C^{(\beta+1)}$ such that $q^{(\beta+1)}(C^{(\beta+1)})$ is still sufficiently large compared to $q^\star(O^\star)$. Having reached such a state, we can run a fixed price auction with price vector $p^{(\beta+1)}/2$ with bidders $N_{\beta+1}$. Since for items in $C^{(\beta+1)}$, their price in $p^{(\beta+1)}$ and $q^{(\beta+1)}$ are within a γ factor of each other, we can invoke Lemma 2 and obtain an allocation with welfare at least γ fraction of $q^{(\beta+1)}(C^{(\beta+1)})$.

Of course, in general, it is too much to expect that our mechanism can converge to a particular price vector $q^\star$ (think of a case where there are many different optimal allocations with different prices; converging to one such price vector necessarily means not converging to the other ones). The following lemma, which is the heart of the proof, however states that in each iteration, we can either "learn" the prices of most items more accurately than before, or we can already "allocate" the items efficiently enough at the current prices.

Lemma 7 (Learnable-Or-Allocatable Lemma). *For any iteration $i \in [\beta]$, conditioned on any outcome of first $i-1$ iterations and choice of $\mathcal{T}^\star$:*

(i) *either* $\mathbb{E}\left[q^{(i+1)}(C^{(i+1)})\right] \geq q^{(i)}(C^{(i)}) - \frac{\mathsf{OPT}}{3\beta}$, *where the expectation is over N_i;*

(ii) *or* $\mathbb{E}\left[\mathsf{val}(A_{j^\star}^{(i)})\right] \geq \frac{\mathsf{OPT}}{200\alpha \cdot \beta^2}$, *where the expectation is over N_i and $j^\star \in [\alpha]$.*

*We refer to the first case as **Learnable** and to the second one as **Allocatable**.*

We prove Lemma 7 next and then use it to conclude the proof of Theorem 6.

A. Proof of Lemma 7 – Learnable-Or-Allocatable Lemma

We start with a high level overview. We prove this lemma in three steps:

(i) *No underestimating prices:* We first show (Lemma 8) that for any of the auctions in this iteration, either most of the correctly priced items (and with respect to this auction) are sold, or this auction itself can result in a high welfare. This step allows us to argue that for many of the items we can sell them in these auctions with a price *at least as high* as their true price, and hence we will not underestimate their prices in this iteration. The proof of this part crucially uses the fact that the bidders are coming in a random order and is along the

lines of a similar argument by Dobzinski [19].

(ii) *No overestimating prices:* We then show (Lemma 11) that in these auctions only a small fraction of items may continue to get sold even past their correct price. Roughly speaking, this is because if we could actually sell many items in auctions with higher prices, this implies that the true welfare of the auction is larger than OPT, a contradiction. This part relies on the "price gap" we introduced in price trees by picking $\mathcal{T}^e$ or $\mathcal{T}^o$ (instead of $\mathcal{T}$ itself).

(iii) *Handling removed bidders:* Finally, in Claim 12 we argue that even if we ignore the items for bidders in N_i (as the mechanism no longer considers these bidders), the *remaining* correctly priced items still have a substantial contribution. This part of the proof uses the fact that we only consider a small random subset N_i of the remaining bidders.

We now present the formal proof. Throughout the proof, we fix $i \in [\beta]$ and condition on the outcome of the first $i-1$ iterations and the choice of $\mathcal{T}^\star$. Conditioning on the outcome of the first $i-1$ iterations fixes the set of bidders $N_1, \ldots, N_{i-1}$ but bidders in N_i are chosen randomly from the remaining bidders. Fixing the bidders $N_1, \ldots, N_{i-1}$ also fixes the price vector $q^{(i)}$. This conditioning also fixes the level-i price vector $p^{(i)}$ and its canonical level-$(i+1)$ price vectors $p_1^{(i)}, \ldots, p_\alpha^{(i)}$. The set $C^{(i)}$ of items that have been correctly priced so far is also fixed.

We partition the correctly priced items $C^{(i)}$ into α sets $D_1^{(i)}, \ldots, D_\alpha^{(i)}$, defined as follows. For an item $\ell \in C^{(i)}$, let z_ℓ denote the node in level i of $\mathcal{T}$ that both $p_\ell^{(i)}$ and $q_\ell^{(i)}$ belong to. Suppose the child-node of z_ℓ to which $q_\ell^{(i)}$ belongs is $z_{\ell,j}$ for some $j \in [\alpha]$. We place item ℓ in $D_j^{(i)}$ in this case. Note that under this partitioning, the level $(i+1)$ node $z_{\ell,j}$ to which $q_\ell^{(i)}$ belongs is the same node that $p_\ell \in p_j^{(i)}$ (strongly) belongs to; thus, for items in $D_j^{(i)}$, $p_j^{(i)} \leq q^{(i)}$.

In the following lemma, we use the construction of `Partition` to argue that for any $j \in [\alpha]$, we either allocate most items in $D_j^{(i)}$ in the fixed price auction with price vector $p_j^{(i)}$ or otherwise this auction is obtaining a large welfare.

Lemma 8. *For any $j \in [\alpha]$, we have $20\beta \cdot \mathbb{E}\left[\mathsf{val}(A_j^{(i)})\right] + \mathbb{E}\left[q^{(i)}(A_j^{(i)} \cap D_j^{(i)})\right] \geq q^{(i)}(D_j^{(i)})$.*

Proof: We define $N_{\geq i} := N \setminus (N_1 \cup \ldots \cup N_{i-1})$. In the following, all expectations are taken over the choice of N_i from $N_{\geq i}$. Recall that in `Partition`, N_i is chosen from $N_{\geq i}$ by picking a random permutation and picking the first $|N_{\geq i}|/(10\beta)$ bidders in N_i.

Define $O_{N_i}^D$ as the restriction of $O^\star$ to items in $D_j^{(i)}$ and

bidders in N_i. Similarly, define $O^D_{N_{>i}}$ as the restriction of O to items in $D^{(i)}_j$ and bidders in $N_{>i} := N_{\geq i} \setminus N_i$. Note that $\boldsymbol{q}^{(i)}(D_j) = \boldsymbol{q}^{(i)}(O^D_{N_i}) + \boldsymbol{q}^{(i)}(O^D_{N_{>i}})$ (recall that $\boldsymbol{q}^{(i)}$ gives price 0 to items not allocated to bidders in $N_{\geq i}$).

Proof of this lemma is by a simple combination of the following two claims.

Claim 9. *Deterministically,* $\mathsf{val}(A^{(i)}_j) \geq \boldsymbol{q}^{(i)}(O^D_{N_i} \setminus A^{(i)}_j)/2$.

Proof: For any bidder $b \in N_i$, when it was bidder b's turn to pick a set in allocation $A^{(i)}_j$ of $\texttt{FixedPriceAuction}(N_i, M, \boldsymbol{p}^{(i)}_j/2)$, b could have picked $O^D_b \setminus A^{(i)}_j \subseteq O^D_{N_i}$ and obtain the profit of

$$v_b(O^D_b \setminus A^{(i)}_j) - \boldsymbol{p}^{(i)}_j(O^D_b \setminus A^{(i)}_j)/2$$
$$\geq \boldsymbol{q}^{(i)}(O^D_b \setminus A^{(i)}_j) - \boldsymbol{p}^{(i)}_j(O^D_b \setminus A^{(i)}_j)/2$$
$$\geq \boldsymbol{q}^{(i)}(O^D_b \setminus A^{(i)}_j)/2.$$

The first inequality is because $\boldsymbol{q}^{(i)}$ is a supporting price for $O^D_b \setminus A^{(i)}_j$ and the second one is because $\boldsymbol{p}^{(i)}_j \leq \boldsymbol{q}^{(i)}$ on the items in $D^{(i)}_j$. As bidder b maximizes the profit by picking $A^{(i)}_{j,b}$, we have

$$\mathsf{val}(A^{(i)}_j) = \sum_{b \in N_i} v_b(A^{(i)}_{j,b})$$
$$\geq \sum_{b \in N_i} \boldsymbol{q}^{(i)}(O^D_b \setminus A^{(i)}_j)/2$$
$$= \boldsymbol{q}^{(i)}(O^D_{N_i} \setminus A^{(i)}_j)/2. \quad \blacksquare \text{ Claim 9}$$

$\blacksquare$

Claim 10. *By randomness of choice of N_i from $N_{\geq i}$,* $\mathbb{E}\left[\mathsf{val}(A^{(i)}_j)\right] \geq (\frac{1}{10\beta}) \cdot \mathbb{E}\left[\boldsymbol{q}^{(i)}(O^D_{N_{>i}} \setminus A^{(i)}_j)/2\right]$.

Proof: For the purpose of this proof, it helps to think of picking N_i alternatively by repeating the following for $n_i := |N_i|$ *steps*: sample a bidder uniformly at random from $N_{\geq i}$, include it in N_i, and remove it from consideration for sampling from now on. It is immediate that the distribution of N_i is the same under this and the original definition.

For every $k \in [n_i]$, define $N_{i,k} \subseteq N_i$ as the set N_i constructed *before* step k and $O^D_{\geq k}$ as the restriction of $O^\star$ to $D^{(i)}_j$ and $N_{\geq i} \setminus N_{i,k}$. Thus, $O^D_{\geq k} \supseteq O^D_{N_{>i}}$ and hence $\boldsymbol{q}^{(i)}(O^D_{\geq k}) \geq \boldsymbol{q}^{(i)}(O^D_{N_{>i}})$ for every k. Recall that $\texttt{FixedPriceAuction}$ operates in a greedy manner and hence allocation of bidders participating in the auction before step k are already determined by step k. Define $A_{<k}$ as the set of items allocated by auction *before* step k and let $u_k := v_b(A^{(i)}_{j,b})$ where b is the chosen bidder in step k and $A^{(i)}_{j,b}$ is the allocation b will get by participating in $\texttt{FixedPriceAuction}(N_i, M, \boldsymbol{p}^{(i)}_j/2)$.

We first prove that $u_k \geq \boldsymbol{q}(O^D_{\geq k,b} \setminus A_{<k})/2$. This is precisely because of the same reason as in Claim 9 that b could have chosen $O^D_{\geq k,b} \setminus A_{<k}$ but decided to pick another set. Define $n_{\geq i} := |N_{\geq i}|$. Recall that b is chosen uniformly at random from the $(n_{\geq i} - k + 1)$ bidders at step k and hence,

$$\mathbb{E}_b[u_k] \geq \frac{1}{n_{\geq i} - k + 1} \cdot \frac{\boldsymbol{q}^{(i)}(O^D_{\geq k} \setminus A_{<k})}{2}$$
$$\geq \frac{1}{n_{\geq i}} \cdot \frac{\boldsymbol{q}^{(i)}(O^D_{\geq k} \setminus A_{<k})}{2}. \tag{2}$$

We can thus write,

$$\mathbb{E}_{N_i}\left[\mathsf{val}(A^{(i)}_j)\right]$$
$$= \sum_{k=1}^{n_i} \mathbb{E}_{N_{i,k}} \mathbb{E}_b[u_k \mid N_{i,k}]$$
$$\geq \frac{1}{2n_{\geq i}} \cdot \sum_{k=1}^{n_i} \mathbb{E}_{N_{i,k}}\left[\boldsymbol{q}^{(i)}(O^D_{\geq k} \setminus A_{<k}) \mid N_{i,k}\right]$$
$$\text{(by Eq (2))}$$
$$\geq \frac{1}{2n_{\geq i}} \cdot \sum_{k=1}^{n_i} \mathbb{E}_{N_i}\left[\boldsymbol{q}^{(i)}(O^D_{N_{\geq i}} \setminus A^{(i)}_j)\right]$$
$$\text{(as } O^D_{N_{\geq i}} \subseteq O^D_{\geq k} \text{ and } A^{(i)}_j \supseteq A_{<k} \text{ always)}$$
$$= \frac{n_i}{n_{\geq i}} \cdot \mathbb{E}_{N_i}\left[\boldsymbol{q}^{(i)}(O^D_{N_{\geq i}} \setminus A^{(i)}_j)/2\right]$$
$$= \left(\frac{1}{10\beta}\right) \cdot \mathbb{E}\left[\boldsymbol{q}^{(i)}(O^D_{N_{\geq i}} \setminus A^{(i)}_j)/2\right]. \quad \blacksquare \text{ Claim 10}$$

$\blacksquare$

We can now conclude the proof of Lemma 8 as follows. By Claims 9 and 10,

$$20\beta \cdot \mathbb{E}\left[\mathsf{val}(A^{(i)}_j)\right]$$
$$\geq \mathbb{E}\left[\boldsymbol{q}^{(i)}(O^D_{N_i} \setminus A^{(i)}_j) + \boldsymbol{q}^{(i)}(O^D_{N_{\geq i}} \setminus A^{(i)}_j)\right]$$
$$= \mathbb{E}\left[\boldsymbol{q}^{(i)}(D^{(i)}_j \setminus A_j)\right]$$
$$= \boldsymbol{q}^{(i)}(D^{(i)}_j) - \mathbb{E}\left[\boldsymbol{q}^{(i)}(D^{(i)}_j \cap A^{(i)}_j)\right].$$

This concludes the proof. $\quad \blacksquare$ Lemma 8 $\quad \blacksquare$

The quantity $A^{(i)}_j \cap D^{(i)}_j$ bounded in Lemma 8 is closely related to the set of correctly priced items at iteration $i+1$, namely $C^{(i+1)}$. The only difference between the two sets is that some items in $A^{(i)}_j \cap D^{(i)}_j$ can be allocated even in $A^{(i)}_k$ for $k > j$ and hence in $\texttt{PriceUpdate}$, we assign a larger price to them. In the following, we prove that the contribution of such items cannot be too large.

Lemma 11. *We have* $\boldsymbol{q}^{(i)}(C^{(i+1)}) \geq \sum_{j=1}^{\alpha} \boldsymbol{q}^{(i)}(A^{(i)}_j \cap D^{(i)}_j) - \frac{\mathsf{OPT}}{10\beta}$.

Proof:

By definition of `PriceUpdate`, we know that items in $A_j^{(i)} \cap D_j^{(i)}$ will join $C_j^{(i+1)}$ iff they do not belong to some $A_k^{(i)}$ for $k > j$. For each $k > j$, let OE_k be the set of items in $D_j^{(i)} \cap A_j^{(i)}$ that are also allocated in $A_k^{(i)}$. Then, $OE_{j+1} \cup \ldots \cup OE_\alpha$ forms the set of all items in $D_j^{(i)}$ that the mechanism *overestimates* their price in iteration i. We bound the contribution of such items.

Fix some $k > j$. Consider the level $i + 1$ of the price tree $\mathcal{T}^\star$. There are α^i nodes in this level to which the $\boldsymbol{q}^{(i)}$-price of an item in $D_j^{(i)}$ can belong to. Let $oe_{k,1}, \ldots, oe_{k,\alpha^i}$ be the number of items corresponding to these nodes that were allocated in $A_k^{(i)}$ as well. Hence, $|OE_k| = \sum_{\ell=1}^{\alpha^i} oe_{k,\ell}$. Moreover, let $p_{k,1}, \ldots, p_{k,\alpha^i}$ be the *maximum* prices that belong to these nodes. Finally, let $p'_{k,1}, \ldots, p'_{k,\alpha^i}$ be the prices that these items were sold in $A_k^{(i)}$. Because $\mathcal{T}^\star$ is either $\mathcal{T}^o$ or $\mathcal{T}^e$, we have $p_{k,\ell} \leq \gamma^{k-j} \cdot p'_{k,\ell}$ (there is a factor γ gap between the maximum price of any bin B_x and minimum price of B_{x+2}).

Since all the items in OE_k are sold in a single application of `FixedPriceAuction`, we know that there exists an allocation with supporting prices $p'_{k,\ell}$ for $oe_{k,\ell}$ items for all $\ell \in [\alpha^i]$. As such,

$$
\begin{aligned}
\mathsf{OPT} &\geq \sum_{\ell=1}^{\alpha^i} p'_{k,\ell} \cdot oe_{k,\ell} \\
&\geq \gamma^{k-j} \cdot \sum_{\ell=1}^{\alpha^i} p_{k,\ell} \cdot oe_{k,\ell} \\
&\geq \gamma^{k-j} \cdot \boldsymbol{q}^{(i)}(OE_k),
\end{aligned}
\tag{3}
$$

by definition of $p_{k,\ell}$ as the maximum price inside the nodes of $\mathcal{T}^\star$ that prices of $\boldsymbol{q}^{(i)}(OE_k)$ belong to. Summing up Eq (3) for all choices of $k > j$, we have

$$
\sum_{k=j+1}^{\alpha} \boldsymbol{q}^{(i)}(OE_k) \leq \sum_{k=j+1}^{\alpha} \frac{1}{\gamma^{k-j}} \cdot \mathsf{OPT} \leq \frac{2}{\gamma} \cdot \mathsf{OPT}.
$$

Finally, as there are α choices for j, we have

$$
\begin{aligned}
&\left(\sum_{j=1}^{\alpha} \boldsymbol{q}^{(i)}(A_j^{(i)} \cap D_j^{(i)}) \right) - \boldsymbol{q}^{(i)}(C^{(i+1)}) \\
&\leq \quad \sum_{j=1}^{\alpha} \frac{2}{\gamma} \cdot \mathsf{OPT} \quad = \quad \frac{2\alpha}{\gamma} \cdot \mathsf{OPT} \quad \leq \quad \frac{\mathsf{OPT}}{10\beta},
\end{aligned}
$$

by the choice of $\gamma \geq 20\alpha\beta$ in Eq (1). ∎ Lemma 11 ∎

So far we only considered prices with respect to $\boldsymbol{q}^{(i)}$. We now extend the bounds to $\boldsymbol{q}^{(i+1)}$, for which we need to remove the correctly priced items corresponding to bidders in N_i.

Claim 12. *We have* $\mathbb{E}\left[\boldsymbol{q}^{(i+1)}(C^{(i+1)})\right] \geq \mathbb{E}\left[\boldsymbol{q}^{(i)}(C^{(i+1)})\right] - \frac{\mathsf{OPT}}{10\beta}$.

Proof: For a bidder b, we write $C_b^{(i)}$ as the set of items in $C^{(i)}$ that are allocated to b in $O^\star$ (i.e., take their price in $\boldsymbol{q}^\star$ because of bidder b); this is similarly defined for $C_b^{(i+1)}$. We can write,

$$
\begin{aligned}
&\mathbb{E}\left[\boldsymbol{q}^{(i+1)}(C^{(i+1)})\right] \\
&= \mathbb{E}\left[\boldsymbol{q}^{(i)}(C^{(i+1)}) - \sum_{b \in N_i} \boldsymbol{q}^{(i)}(C_b^{(i+1)})\right] \\
&\geq \mathbb{E}\left[\boldsymbol{q}^{(i)}(C^{(i+1)}) - \sum_{b \in N_i} \boldsymbol{q}^{(i)}(C_b^{(i)})\right],
\end{aligned}
$$

because $C^{(i+1)} \subseteq C^{(i)}$. Since each bidder joins N_i with probability $(1/10\beta)$, this implies

$$
\begin{aligned}
&\mathbb{E}\left[\boldsymbol{q}^{(i+1)}(C^{(i+1)})\right] \\
&\geq \mathbb{E}\left[\boldsymbol{q}^{(i)}(C^{(i+1)})\right] - \frac{1}{10\beta} \cdot \boldsymbol{q}^{(i)}(C_b^{(i)}) \\
&\geq \mathbb{E}\left[\boldsymbol{q}^{(i)}(C^{(i+1)})\right] - \frac{\mathsf{OPT}}{10\beta}. \quad \blacksquare \text{ Claim 12}
\end{aligned}
$$

∎

We now have all the ingredients needed to prove Lemma 7.

Proof of Lemma 7: By applying Lemma 8 to every $j \in [\alpha]$, we have

$$
\begin{aligned}
&20\beta \cdot \sum_{j=1}^{\alpha} \mathbb{E}\left[\mathsf{val}(A_j^{(i)})\right] + \mathbb{E}\left[\sum_{j=1}^{\alpha} \boldsymbol{q}^{(i)}(A_j^{(i)} \cap D_j^{(i)})\right] \\
&\geq \sum_{j=1}^{\alpha} \boldsymbol{q}^{(i)}(D_j^{(i)}).
\end{aligned}
\tag{4}
$$

The RHS of above is clearly $\boldsymbol{q}^{(i)}(C^{(i)})$. The second term in the LHS can be upper bounded by Lemma 11 and Claim 12,

$$
\begin{aligned}
&\mathbb{E}\left[\sum_{j=1}^{\alpha} \boldsymbol{q}^{(i)}(A_j^{(i)} \cap D_j^{(i)})\right] \\
&\leq \mathbb{E}\left[\boldsymbol{q}^{(i)}(C^{(i+1)})\right] + \frac{\mathsf{OPT}}{10\beta} \\
&\leq \mathbb{E}\left[\boldsymbol{q}^{(i+1)}(C^{(i+1)})\right] + \frac{2 \cdot \mathsf{OPT}}{10\beta}.
\end{aligned}
$$

Plugging in these bounds in Eq (4), we obtain

$$
\begin{aligned}
&20\beta \cdot \sum_{j=1}^{\alpha} \mathbb{E}\left[\mathsf{val}(A_j^{(i)})\right] + \mathbb{E}\left[\boldsymbol{q}^{(i+1)}(C^{(i+1)})\right] \\
&\geq \boldsymbol{q}^{(i)}(C^{(i)}) - \frac{\mathsf{OPT}}{5\beta}.
\end{aligned}
\tag{5}
$$

Now let us consider two cases. First suppose,

$$
\sum_{j=1}^{\alpha} \mathbb{E}\left[\mathsf{val}(A_j^{(i)})\right] \geq \frac{\mathsf{OPT}}{200\beta^2}.
\tag{6}
$$

242

In this case, $\mathbb{E}\left[\mathsf{val}(A_{j^\star}^{(i)})\right]$ for $j^\star$ chosen uniformly at random from $[\alpha]$ is at least $\frac{\mathsf{OPT}}{100\alpha\beta^2}$, hence satisfying item ii of the lemma (Allocatable case). We now consider the other case where the LHS of Eq (6) is smaller than the RHS. Plugging in this bound in Eq (5) implies that

$$\mathbb{E}\left[\boldsymbol{q}^{(i+1)}(C^{(i+1)})\right]$$
$$\geq \quad \boldsymbol{q}^{(i)}(C^{(i)}) - \frac{\mathsf{OPT}}{5\beta} - 20\beta \cdot \frac{\mathsf{OPT}}{200\beta^2}$$
$$> \quad \boldsymbol{q}^{(i)}(C^{(i)}) - \frac{\mathsf{OPT}}{3\beta}.$$

This satisfies item i of the lemma (Learnable case), concluding the proof.　■ Lemma 7 ■

B. Proof of Theorem 6 – Approximation Ratio

We now prove the bound on expected approximation ratio of `PriceLearningMechanism`. We first need some definitions.

For an iteration $i \in [\beta + 1]$, we use $Alg^{(i)} = (Alg_1, \ldots, Alg_n)$ to denote the allocation returned by our mechanism, conditioned on the mechanism reaching iteration i and on the outcomes of $N_1, \ldots, N_{i-1}$ as well as the choice of $\mathcal{T}^\star$. We use $\mathsf{ALG}^{(i)}$ to denote the welfare of allocation $Alg^{(i)}$. We note that except for $i = \beta + 1$, $Alg^{(i)}$ is a random variable.

Our main tool in this section is the following inductive lemma.

Lemma 13. *For $i \in [\beta + 1]$,*

$$\mathbb{E}\left[\mathsf{ALG}^{(i)}\right] \geq \frac{1}{200\alpha\beta^3} \cdot \left(1 - \frac{1}{\beta}\right)^{\beta+1-i} \cdot$$
$$\left(\boldsymbol{q}^{(i)}(C^{(i)}) - \frac{\mathsf{OPT}}{3\beta}(\beta + 1 - i)\right),$$

where the expectation is taken over the choice of $N_i, N_{i+1}, \ldots, N_\beta$.

Before proving this lemma, we show how it immediately implies the proof of Theorem 6.

Proof of Theorem 6 – Approximation Ratio:
By Lemma 13 for $i = 1$,

$$\mathbb{E}\left[\mathsf{ALG}^{(1)}\right]$$
$$\geq \quad \frac{1}{200\alpha\beta^3} \cdot \left(1 - \frac{1}{\beta}\right)^{\beta} \cdot \left(\boldsymbol{q}^{(1)}(C^{(1)}) - \frac{\mathsf{OPT}}{3\beta} \cdot \beta\right)$$
$$= \quad \Omega\left(\frac{1}{\alpha\beta^3}\right) \cdot \left(\boldsymbol{q}^{(1)}(C^{(1)}) - \frac{\mathsf{OPT}}{3}\right).$$

The only random event that we have not conditioned on in $\boldsymbol{q}^{(1)}(C^{(1)})$ is the choice of $\mathcal{T}^\star$. Let ALG denote the welfare

of allocation returned by the mechanism. We have,

$$\mathbb{E}\left[\mathsf{ALG}\right] = \Omega\left(\frac{1}{\alpha\beta^3}\right) \cdot \mathbb{E}_{\mathcal{T}^\star}\left[\boldsymbol{q}^{(1)}(C^{(1)}) - \frac{\mathsf{OPT}}{3}\right]$$
$$= \Omega\left(\frac{1}{\alpha\beta^3}\right) \cdot \left(\frac{\mathsf{OPT}}{2} - \frac{\mathsf{OPT}}{3}\right)$$
$$= \Omega\left(\frac{1}{\alpha\beta^3}\right) \cdot \mathsf{OPT},$$

where the second equality is because $\mathcal{T}^\star$ is chosen uniformly at random to be $\mathcal{T}^o$ or $\mathcal{T}^e$ and the bins in these two price trees partition the prices in $\boldsymbol{q}(O)$ by Assumption 1. As $\alpha = \Theta(1)$, and $\beta = O(\log\log\Psi)$, which is $O(\log\log m)$ under Assumption 1, we obtain that `PriceLearningMechanism` achieves an $O((\log\log m)^3)$ approximation in expectation.　■

We prove Lemma 13 using backward induction. We first show that the lemma easily holds true for the base case, namely, for $i = \beta + 1$, because of the performance of `FixedPriceAuction` for correctly priced items (Lemma 2). The heart of the induction step lies in Learnable-Or-Allocatable Lemma (Lemma 7) that states $\mathbb{E}\left[\boldsymbol{q}^{(i+1)}(C^{(i+1)})\right]$ is close to $\boldsymbol{q}^{(i)}(C^{(i)})$ unless we already have a good allocation. So we first use the induction hypothesis to show that the expected welfare of the mechanism is close to $\mathbb{E}\left[\boldsymbol{q}^{(i+1)}(C^{(i+1)})\right]$ and then use Lemma 7 to show it is close to $\boldsymbol{q}^{(i)}(C^{(i)})$.

Proof of Lemma 13: We use backward induction on i. Consider the base case for $i = \beta + 1$, where we want to show the following (note that $\mathsf{ALG}^{(\beta+1)}$ is no longer a random variable)

$$\mathsf{ALG}^{(\beta+1)} \geq \frac{1}{200\alpha\beta^3} \cdot \left(\boldsymbol{q}^{(\beta+1)}(C^{(\beta+1)})\right).$$

Since our mechanism has already reached iteration $i = \beta+1$, this means that for every correctly priced item $j \in C^{(\beta+1)}$, $p_j \in \boldsymbol{p}^{(\beta+1)}$ and $q_j \in \boldsymbol{q}^{(\beta+1)}$ both belong to a leaf-node of $\mathcal{T}^\star$, and consequently the same price bin. As such, by construction of bins, $p_j \leq q_j \leq \gamma \cdot p_j$ and hence running `FixedPriceAuction`$(N_{\beta+1}, M, \boldsymbol{p}^{(i+1)}/2)$ in this step of `PriceLearningMechanism`, by Lemma 2, results in allocation with welfare,

$$\mathsf{ALG}^{(\beta+1)} \geq \frac{1}{\gamma} \cdot \boldsymbol{q}^{(\beta+1)}(C^{(\beta+1)})$$
$$> \frac{1}{200\alpha\beta^3} \cdot \boldsymbol{q}^{(\beta+1)}(C^{(\beta+1)}),$$

by the choice of $\gamma = \Theta(\alpha\beta)$ in Eq (1). This proves the induction base.

We now prove the induction step. Suppose the lemma is true for iterations $\geq i + 1$ and we prove the induction step for iteration i. Notice that w.p. $1/\beta$ the mechanism outputs an allocation $A_{j^\star}^{(i)}$ for $j^\star$ chosen randomly from $[\alpha]$, and

otherwise it continues to the next iteration. This implies:

$$\mathbb{E}\left[\mathsf{ALG}^{(i)}\right]$$

$$\geq \frac{1}{\beta} \mathop{\mathbb{E}}_{N_i,j^\star}\left[\mathsf{val}(A_{j^\star}^{(i)})\right] + \left(1 - \frac{1}{\beta}\right) \cdot \mathop{\mathbb{E}}_{N_i}\left[\mathsf{ALG}^{(i+1)}\right]$$

$$\geq \frac{1}{\beta} \cdot \mathop{\mathbb{E}}_{N_i,j^\star}\left[\mathsf{val}(A_{j^\star}^{(i)})\right] + \left(1 - \frac{1}{\beta}\right)^{\beta+1-i} \cdot$$
$$\mathop{\mathbb{E}}_{N_i}\left[\frac{1}{200\alpha\beta^3}\left(\boldsymbol{q}^{(i+1)}(C^{(i+1)}) - \frac{\mathsf{OPT}}{3\beta}(\beta - i)\right)\right], \quad (7)$$

where the second inequality uses induction hypothesis. Now to prove the induction step we consider the two cases corresponding to Lemma 7:

(i) **Learnable case**, i.e., $\mathbb{E}\left[\boldsymbol{q}^{(i+1)}(C^{(i+1)})\right] \geq \boldsymbol{q}^{(i)}(C^{(i)}) - \frac{\mathsf{OPT}}{3\beta}$: Combining this with Eq (7),

$$\mathbb{E}\left[\mathsf{ALG}^{(i)}\right] \geq \left(1 - \frac{1}{\beta}\right)^{\beta+1-i} \cdot$$
$$\mathop{\mathbb{E}}_{N_i}\left[\frac{1}{200\alpha\beta^3}\left(\boldsymbol{q}^{(i)}(C^{(i)}) - \frac{\mathsf{OPT}}{3\beta} - \frac{\mathsf{OPT}}{3\beta}(\beta - i)\right)\right],$$

which implies the induction step.

(ii) **Allocatable case**, i.e., $\mathbb{E}\left[\mathsf{val}(A_{j^\star}^{(i)})\right] \geq \frac{\mathsf{OPT}}{200\alpha\cdot\beta^2}$: Combining this with Eq. (7),

$$\mathbb{E}\left[\mathsf{ALG}^{(i)}\right] \geq \frac{\mathsf{OPT}}{200\alpha\beta^3}$$
$$\geq \frac{1}{200\alpha\beta^3}\left(1 - \frac{1}{\beta}\right)^{\beta+1-i} \cdot$$
$$\mathbb{E}\left[\boldsymbol{q}^{(i)}(C^{(i)}) - \frac{\mathsf{OPT}}{2\beta}(\beta + 1 - i)\right],$$

where the last inequality uses $\mathsf{OPT} \geq \boldsymbol{q}^{(i)}(C^{(i)})$ and implies the induction step.

This concludes the proof of the lemma. ∎

VI. Removing the Extra Assumptions

We now show how to remove Assumption 1 and prove our main result in its full generality. We shall emphasize that the main contribution of our work is in establishing Theorem 6; the remaining ideas here are standard for the most part and appear in similar forms in previous work on truthful mechanisms, e.g. in [3], [8], [19]. We present them for completeness.

Let $O = (O_1, \ldots, O_n)$ be an optimal allocation with welfare OPT and supporting prices $\boldsymbol{q}$. In order to remove Assumption 1, we find prices $\psi_{\min}$ and $\psi_{\max}$ such that $\psi_{\max}/\psi_{\min} = O(m^2)$, and for most items allocated by O, their price in $\boldsymbol{q}$ belongs to the range $[\psi_{\min} : \psi_{\max}]$; here, "most items" should be interpreted as items with prices in $\boldsymbol{q}$ that is a constant fraction of OPT. Having found such prices, we can then run `PriceLearningMechanism` from Section IV and apply Theorem 6 to finalize the proof

(strictly speaking, Assumption 1 stated that *all* prices in all valuations of bidders need to be in range $[\psi_{\min} : \psi_{\max}]$; however, as is evident from the proof of Theorem 6, we only applied this assumption to prices in $\boldsymbol{q}$).

To find $\psi_{\min}$ and $\psi_{\max}$, we partition N into two (almost) equal-size groups N_{stat} and N_{mech} randomly. We run any constant-factor approximation algorithm (and not a truthful mechanism) for welfare maximization with bidders in N_{stat} and items M, say, the algorithm of [12], to compute a value $\mathsf{ALG}_{\mathsf{stat}}$ which is an $O(1)$-approximation to $\mathsf{OPT}_{\mathsf{stat}}$ namely, the value of welfare maximizing allocation for N_{stat} and M. We completely ignore the allocation of these bidders and instead only set $\psi_{\min} := \mathsf{ALG}_{\mathsf{stat}}/m^2$ and $\psi_{\max} := \mathsf{ALG}_{\mathsf{stat}} \cdot \Theta(1)$. We then run `PriceLearningMechanism`$(N_{\mathsf{mech}}, M)$ with $\psi_{\min}$ and $\psi_{\max}$, and return the resulting allocation to bidders in N_{mech}.

The intuition behind the approach is that because we partitioned N into two *random* groups, $\mathsf{OPT}_{\mathsf{stat}}$ and consequently $\mathsf{ALG}_{\mathsf{stat}}$ should be an $O(1)$-approximation to $\mathsf{OPT}_{\mathsf{mech}}$, namely, the value of welfare optimizing allocation for bidders in N_{mech} (this intuition is not quite correct but for the moment let us ignore this fact). Thus, in an optimal allocation of M to N_{mech}, no item have price more than $\psi_{\max}$ and also the total contribution of items with price smaller than $\psi_{\min}$ is negligible, hence we can safely ignore them. This in turn implies that Assumption 1 holds and by Theorem 6, `PriceLearningMechanism`(N_{mech}, M) outputs an allocation with welfare $\mathsf{ALG}_{\mathsf{mech}}$ such that $\mathbb{E}[\mathsf{ALG}_{\mathsf{mech}}] \geq \mathsf{OPT}_{\mathsf{mech}} \cdot \Omega\left(\frac{1}{(\log\log m)^3}\right)$. Moreover, by the choice of N_{mech}, we have $\mathbb{E}[\mathsf{OPT}_{\mathsf{mech}}] = \mathsf{OPT}/2$. Thus, this should gives us an $O((\log\log m)^3)$ approximation in expectation.

As stated earlier, there is a slight problem with the above intuition. One cannot in general guarantee that by partitioning the bidders into two parts randomly, each part will have roughly the same contribution to the value of OPT. In particular, if there exists a bidder with a much higher contribution to OPT than the rest, the above approach is bound to fail. So we take care of this case separately as follows: With probability half, we simply run a second-price auction on the grand bundle M and sell it to the highest bidder entirely. With the remaining half probability, we run the above procedure. This ensures that if such a bidder exists, we get her contribution with probability half. Otherwise, with probability half, we can run the previous analysis.

A. The Final Mechanism

Our final mechanism is as follows.

FinalMechanism(N, M)

1) With probability $1/2$, run a second-price auction on

---

grand bundle M with all bidders, return the resulting allocation, and terminate. With the remaining probability, continue.
2) Pick N_{stat} by sampling each bidder in N independently and w.p. $1/2$. Let $N_{\text{mech}} := N \setminus N_{\text{stat}}$.
3) Run the 2-approximation algorithm of [12] on items M and bidders N_{stat}. Let ALG_{stat} be the welfare of the returned allocation. Let $\psi_{\min} := \text{ALG}_{\text{stat}}/m^2$ and $\psi_{\max} := 8 \cdot \text{ALG}_{\text{stat}}$.
4) Run `PriceLearningMechanism(`N_{mech}, M`)` with $\psi_{\min}$ and $\psi_{\max}$, and return the allocation.

We have the following theorem that formalizes our main result from Section I.

Theorem 14. *For a combinatorial auction with n submodular (even XOS) bidders and m items,* FinalMechanism *is universally truthful, uses* $\text{poly}(m, n)$ *demand and value queries, and achieves an approximation ratio of* $O((\log \log m)^3)$ *in expectation.*

The proof of truthfulness of Theorem 14 is quite easy. The case where we run the second-price auction is clearly truthful. For the other case, note that we never allocate any item to bidders in N_{stat} and so they might as well reveal their true valuations in response to the algorithm in Line (3). Finally, `PriceLearningMechanism` with bidders N_{mech} is truthful by Theorem 6. The bound on the number of queries also follows from [12] for Line (3) and Theorem 6 for Line (4), and since the second-price auction can be implemented with n value queries for the grand bundle. It thus only remains to analyze the approximation ratio of FinalMechanism, which we do in the next section.

B. Approximation Ratio of Final Mechanism

We use the following standard result that follows directly from Chernoff-Hoeffding bound.

Lemma 15 (cf. [3], [8], [19])**.** *Let $O = (O_1, \ldots, O_n)$ be an optimal allocation of items M to bidders N with welfare* OPT. *Suppose we sample each $i \in N$ w.p. ρ independently to obtain N'. If for every $i \in N$, we have $v_i(O_i) \leq \epsilon \cdot \text{OPT}$, then $\sum_{i \in N'} v_i(O_i) \geq (\rho/2) \cdot \text{OPT}$ w.p. at least $1 - 2 \cdot \exp\left(-\frac{\rho}{2 \cdot \epsilon}\right)$.*

Fix an optimal allocation $O = (O_1, \ldots, O_n)$ of items to bidders in N with welfare OPT. We say that a bidder $i \in N$ is *dominant* iff $v_i(O_i) \geq \text{OPT}/8$. For the analysis, we consider two cases: either (i) there exists at least one dominant bidder, or (ii) no bidder is dominant.

Case (i): A dominant bidder exists.: W.p. half, we decide to run the second-price auction. Let i be the bidder that gets the grand bundle M in the auction. Clearly i has to

be a dominant bidder in this case and thus $v_i(M) \geq \text{OPT}/8$ already. As such, in this case, the expected welfare of the allocation is at least $\text{OPT}/16$, concluding the proof.

Case (i): No dominant bidder exists.: W.p. half, we decide not to run the second-price auction. Let OPT_{stat} and OPT_{mech} be the welfare of the optimal allocation of M to N_{stat} and N_{mech}, respectively. By Lemma 15, applied to choice of N_{stat} and $N \setminus N_{\text{stat}}$ (both sets have the same distribution) with $\rho = 1/2$ and $\varepsilon = 1/8$, and a union bound, w.p. at least $1/2$, we have

$$\frac{1}{4} \cdot \text{OPT} \leq \text{OPT}_{\text{stat}} \leq \text{OPT} \qquad \text{and}$$
$$\frac{1}{4} \cdot \text{OPT} \leq \text{OPT}_{\text{mech}} \leq \text{OPT}. \tag{8}$$

In the following, we condition on the (independent) events that we do not run the second-price auction, and that Eq (8) holds, which happens w.p. $1/4$.

Fix a welfare maximizing allocation of M to N_{mech} with welfare OPT_{mech} and supporting prices $\boldsymbol{q} = (q_1, \ldots, q_m)$. Since we run a 2-approximation algorithm in Line (3), we know that $\frac{1}{8} \cdot \text{OPT} \leq \text{ALG}_{\text{stat}} \leq \text{OPT}$ by Eq (8). Hence, setting $\psi_{\max} = 8 \cdot \text{ALG}_{\text{stat}}$ ensures that $q_j \leq \psi_{\max}$ for every item $j \in M$. Moreover, let $M' \subseteq M$ be the set of items j such that $q_j \leq \psi_{\min}$. By definition of $\psi_{\min} = \text{ALG}_{\text{stat}}/m^2$ and since $\text{ALG}_{\text{stat}} > \text{OPT}_{\text{mech}}/8$, we get $\boldsymbol{q}(M') \leq \text{OPT}_{\text{mech}}/2$ (as $m \gg 8$). As such, we can simply ignore the contribution of all items in M' and still have a set of items $M \setminus M'$ that can be allocated to bidders in N_{mech} with welfare at least $\text{OPT}_{\text{mech}}/2 \geq \text{OPT}/8$. Moreover, the supporting prices of these items now belong to $[\psi_{\min} : \psi_{\max}]$. Hence, we can apply Theorem 6 under Assumption 1 and obtain that in this case, the expected welfare of the allocation is at most $O((\log \log m)^3)$ times smaller than OPT, finishing the proof of Theorem 14.

VII. CONCLUDING REMARKS AND OPEN PROBLEMS

We gave a randomized, computationally-efficient, and universally truthful mechanism for combinatorial auctions with submodular (even XOS) bidders that achieves an $O((\log \log m)^3)$-approximation. This reduces the gap between the approximation ratio achievable by truthful mechanisms vs arbitrary algorithms for this problem by an exponential factor from $\text{poly}(\log(m))$ to $\text{poly}(\log \log(m))$. The obvious question left open by our work is whether this gap can be improved further. We do not believe in any way that our $O((\log \log m)^3)$ approximation is the best possible[3]. On the other hand, the limit of our approach seems to be an $\Omega(\log \log m)$ approximation. It is a fascinating open

---

[3]Indeed, using a slightly more nuanced argument, our bounds can be improved to $O\left(\frac{(\log \log m)^3}{\log \log \log m}\right)$; however as this $\Theta(\log \log \log m)$ improvement is minor and for the sake of clarity, we used the slightly weaker analysis in the paper.

question whether one can improve the approximation factor all the way down to a constant. However, even improving the approximation ratio of our mechanism down to $O(\log \log m)$ already seems challenging, and is an interesting open question. On the lower bound front, proving any separation between the power of truthful mechanisms and algorithms when the access to input is via arbitrary queries, namely, the communication complexity setting, is also very interesting.

APPENDIX

A. Formal Definitions of Mechanisms and Truthfulness

Let $\mathcal{V}$ be a class of valuation functions defined over M, say, all submodular functions $2^M \to \mathbb{R}^+$, and $\mathcal{A}$ be the set of all possible allocations of M to n bidders. A deterministic mechanism for combinatorial auctions is a pair $(f, \boldsymbol{p})$ where $f : \mathcal{V}^n \to \mathcal{A}$ (representing the allocation to bidders) and $\boldsymbol{p} = (p_1, \ldots, p_n)$ where $p_i : \mathcal{V}^n \to \mathbb{R}^+$ (representing the price charged for item i). A randomized mechanism is simply a probability distribution over deterministic mechanisms.

Definition 16 (Truthfulness and Universal Truthfulness). *A deterministic mechanism (f, p) is* truthful *iff for all $i \in N$, $v_i, v_i' \in \mathcal{V}$ and $v_{-i} \in \mathcal{V}^{n-1}$, we have,*

$$v_i(f(v_i, v_{-i})_i) - p_i(v_i, v_{-i})$$
$$\geq v_i(f(v_i', v_{-i})_i) - p_i(v_i', v_{-i}).$$

A randomized mechanism is universally truthful *iff it is a distribution over truthful mechanisms.*

We note that beside universal truthfulness, the notion of truthful-in-expectation is also considered for randomized mechanisms that guarantee that bidding truthfully maximizes the *expected* profit; see, e.g. [1], [2], [7] and references therein. This is a much weaker guarantee than universal truthfulness we consider in this paper. In particular such mechanisms are only applicable when bidders are risk neutral and have no information about the outcomes of the random coin flips before they need to act; see [3, Section 1.2] for more details.

B. Proof of Lemma 2 – Fixed-Price Auctions

Lemma 17 (Restatement of Lemma 2). *Let $A :=$ `FixedPriceAuction`$(N, M, \boldsymbol{p})$ and $\delta < 1/2$ be a parameter. Suppose O is any allocation with supporting prices $\boldsymbol{q}$ and $M^\star \subseteq M$ is the set of items j with $\delta \cdot q_j \leq p_j < \frac{1}{2} \cdot q_j$. Then,* $\mathsf{val}(A) \geq \delta \cdot \boldsymbol{q}(M^\star)$.

Proof: Define the allocation $O^\star = (O_1^\star, \ldots, O_n^\star)$ as the restriction of O to $M^\star$. Define $\overline{A}_i = O_i^\star \setminus A$ for every $i \in N$ and $\overline{A} := \overline{A}_1 \cup \ldots \cup \overline{A}_n$. Bidder i could have chosen $\overline{A}_i$ in `FixedPriceAuction` but decided to pick another bundle A_i instead. This implies that:

$$v_i(A_i) - \boldsymbol{p}(A_i) \geq v_i(\overline{A}_i) - \boldsymbol{p}(\overline{A}_i). \tag{9}$$

We now use Eq (9) to prove the lemma. We have,

$$\mathsf{val}(A) = \sum_{i=1}^{n} v_i(A_i) = \boldsymbol{p}(A) + \sum_{i=1}^{n}(v_i(A_i) - \boldsymbol{p}(A_i))$$
$$\geq \boldsymbol{p}(A) + \sum_{i=1}^{n}(v_i(\overline{A}_i) - \boldsymbol{p}(\overline{A}_i))$$
$$\geq \boldsymbol{p}(A) + \sum_{i=1}^{n}(\boldsymbol{q}(\overline{A}_i) - \boldsymbol{p}(\overline{A}_i))$$

by Eq (9) and since $\overline{A}_i \subseteq O_i^\star \subseteq O_i$. Now using as $p_j \leq q_j/2$ for all $j \in \overline{A} \subseteq O^\star = M^\star$, we get

$$\mathsf{val}(A) \geq \boldsymbol{p}(A) + \sum_{i=1}^{n} \boldsymbol{p}(\overline{A}_i)$$
$$\geq \boldsymbol{p}(O^\star) \geq \delta \cdot \boldsymbol{q}(O^\star),$$

where the last two inequalities use $O^\star \subseteq \overline{A} \cup A$ and $\overline{A} \cap A = \emptyset$, and that $p_j \geq \delta \cdot q_j$ for all $j \in O^\star$. This concludes the proof as $\boldsymbol{q}(O^\star) = \boldsymbol{q}(M^\star)$ by definition. ∎

REFERENCES

[1] R. Lavi and C. Swamy, "Truthful and near-optimal mechanism design via linear programming," in *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005), 23-25 October 2005, Pittsburgh, PA, USA, Proceedings*, 2005, pp. 595–604.

[2] S. Dughmi, T. Roughgarden, and Q. Yan, "From convex optimization to randomized mechanisms: toward optimal combinatorial auctions," in *Proceedings of the forty-third annual ACM symposium on Theory of computing*. ACM, 2011, pp. 149–158.

[3] S. Dobzinski, N. Nisan, and M. Schapira, "Truthful randomized mechanisms for combinatorial auctions," in *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21-23, 2006*, 2006, pp. 644–652.

[4] I. Abraham, M. Babaioff, S. Dughmi, and T. Roughgarden, "Combinatorial auctions with restricted complements," in *Proceedings of the 13th ACM Conference on Electronic Commerce, EC 2012, Valencia, Spain, June 4-8, 2012*, 2012, pp. 3–16.

[5] D. Fotakis, P. Krysta, and C. Ventre, "Combinatorial auctions without money," *Algorithmica*, vol. 77, no. 3, pp. 756–785, 2017.

[6] A. Mu'alem and N. Nisan, "Truthful approximation mechanisms for restricted combinatorial auctions," *Games and Economic Behavior*, vol. 64, no. 2, pp. 612–631, 2008.

[7] S. Dughmi and J. Vondrák, "Limitations of randomized mechanisms for combinatorial auctions," in *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, 2011, pp. 502–511.

[8] S. Dobzinski, "Two randomized mechanisms for combinatorial auctions," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 10th International Workshop, APPROX 2007, and 11th International Workshop, RANDOM 2007, Princeton, NJ, USA, August 20-22, 2007, Proceedings*, 2007, pp. 89–103.

[9] W. Vickrey, "Counterspeculation, auctions, and competitive sealed tenders," *The Journal of finance*, vol. 16, no. 1, pp. 8–37, 1961.

[10] E. H. Clarke, "Multipart pricing of public goods," *Public choice*, vol. 11, no. 1, pp. 17–33, 1971.

[11] T. Groves *et al.*, "Incentives in teams," *Econometrica*, vol. 41, no. 4, pp. 617–631, 1973.

[12] B. Lehmann, D. J. Lehmann, and N. Nisan, "Combinatorial auctions with decreasing marginal utilities," *Games and Economic Behavior*, vol. 55, no. 2, pp. 270–296, 2006.

[13] S. Dobzinski and M. Schapira, "An improved approximation algorithm for combinatorial auctions with submodular bidders," in *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2006, Miami, Florida, USA, January 22-26, 2006*, 2006, pp. 1064–1073.

[14] S. Dobzinski and J. Vondrák, "The computational complexity of truthfulness in combinatorial auctions," in *Proceedings of the 13th ACM Conference on Electronic Commerce, EC 2012, Valencia, Spain, June 4-8, 2012*, 2012, pp. 405–422.

[15] U. Feige and J. Vondrák, "The submodular welfare problem with demand queries," *Theory of Computing*, vol. 6, no. 1, pp. 247–290, 2010.

[16] S. Dobzinski, "An impossibility result for truthful combinatorial auctions with submodular valuations," in *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, 2011, pp. 139–148.

[17] P. Krysta and B. Vöcking, "Online mechanism design (randomized rounding on the fly)," in *International Colloquium on Automata, Languages, and Programming*. Springer, 2012, pp. 636–647.

[18] S. Dobzinski and J. Vondrák, "Communication complexity of combinatorial auctions with submodular valuations," in *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, 2013, pp. 1205–1215.

[19] S. Dobzinski, "Breaking the logarithmic barrier for truthful combinatorial auctions with submodular bidders," in *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, 2016, pp. 940–948.

[20] V. S. Mirrokni, M. Schapira, and J. Vondrák, "Tight information-theoretic lower bounds for welfare maximization in combinatorial auctions," in *Proceedings 9th ACM Conference on Electronic Commerce (EC-2008), Chicago, IL, USA, June 8-12, 2008*, 2008, pp. 70–77.

[21] J. Vondrák, "Optimal approximation for the submodular welfare problem in the value oracle model," in *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, 2008, pp. 67–74.

[22] U. Feige and J. Vondrák, "Approximation algorithms for allocation problems: Improving the factor of 1 - 1/e," in *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*, 2006, pp. 667–676.

[23] S. Dobzinski, N. Nisan, and M. Schapira, "Approximation algorithms for combinatorial auctions with complement-free bidders," in *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, 2005, pp. 610–618.

[24] N. Andelman and Y. Mansour, "Auctions with budget constraints," in *Algorithm Theory - SWAT 2004, 9th Scandinavian Workshop on Algorithm Theory, Humlebaek, Denmark, July 8-10, 2004, Proceedings*, 2004, pp. 26–38.

[25] D. Chakrabarty and G. Goel, "On the approximability of budgeted allocations and improved lower bounds for submodular welfare maximization and GAP," in *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, 2008, pp. 687–696.

[26] C. Konrad, F. Magniez, and C. Mathieu, "Maximum matching in semi-streaming with few passes," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 15th International Workshop, APPROX 2012, and 16th International Workshop, RANDOM 2012, Cambridge, MA, USA, August 15-17, 2012. Proceedings*, 2012, pp. 231–242.

[27] G. P. Guruganesh and S. Singla, "Online matroid intersection: Beating half for random arrival," in *International Conference on Integer Programming and Combinatorial Optimization*. Springer, 2017, pp. 241–253.

[28] A. Norouzi-Fard, J. Tarnawski, S. Mitrovic, A. Zandieh, A. Mousavifar, and O. Svensson, "Beyond 1/2-approximation for submodular maximization on massive data streams," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, 2018, pp. 3826–3835.

[29] C. H. Papadimitriou, M. Schapira, and Y. Singer, "On the hardness of being truthful," in *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, 2008, pp. 250–259.

[30] A. Daniely, M. Schapira, and G. Shahaf, "Inapproximability of truthful mechanisms via generalizations of the VC dimension," in *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, 2015, pp. 401–408.

[31] N. Nisan, "Bidding and allocation in combinatorial auctions," in *Proceedings of the 2nd ACM Conference on Electronic Commerce (EC-00), Minneapolis, MN, USA, October 17-20, 2000*, 2000, pp. 1–12.

[32] L. Blumrosen and N. Nisan, "Auctions with severely bounded communication," in *43rd Symposium on Foundations of Computer Science (FOCS 2002), 16-19 November 2002, Vancouver, BC, Canada, Proceedings*, 2002, pp. 406–415.

[33] N. Nisan and I. Segal, "The communication requirements of efficient allocations and supporting prices," *J. Economic Theory*, vol. 129, no. 1, pp. 192–224, 2006.

[34] S. Dobzinski, N. Nisan, and S. Oren, "Economic efficiency requires interaction," in *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, 2014, pp. 233–242.

[35] S. Dobzinski, "Computational efficiency requires simple taxation," in *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, 2016, pp. 209–218.

[36] S. Assadi, "Combinatorial auctions do need modest interaction," in *Proceedings of the 2017 ACM Conference on Economics and Computation, EC '17, Cambridge, MA, USA, June 26-30, 2017*, 2017, pp. 145–162.

[37] M. Braverman, J. Mao, and S. M. Weinberg, "On simultaneous two-player combinatorial auctions," in *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, January 7-10, 2018*, 2018, pp. 2256–2273.

[38] T. Ezra, M. Feldman, E. Neyman, I. Talgam-Cohen, and S. M. Weinberg, "Complement-free couples must communicate: A hardness result for two-player combinatorial auctions," *CoRR*, vol. abs/1811.09871, 2018.

[39] U. Feige, "On maximizing welfare when utility functions are subadditive," in *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21-23, 2006*, 2006, pp. 41–50.

[40] M. Feldman, N. Gravin, and B. Lucier, "Combinatorial auctions via posted prices," in *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, 2015, pp. 123–135.

[41] S. Ehsani, M. Hajiaghayi, T. Kesselheim, and S. Singla, "Prophet secretary for combinatorial auctions and matroids," in *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, 2018, pp. 700–714.