Improved Truthful Mechanisms for Combinatorial Auctions with Submodular Bidders

SEPEHR ASSADI
Rutgers University
and
SAHIL SINGLA
Princeton University and Institute for Advanced Study

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 the 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 almost 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.

Categories and Subject Descriptors: F.2 [**Theory of computation**]: Mechanism Design

General Terms: Algorithms, Economics, Theory

Additional Key Words and Phrases: Combinatorial Auctions, Truthful Mechanisms

1. INTRODUCTION

A fundamental problem at the intersection of Computer Science and Economics is to allocate resources among strategic bidders. In combinatorial auctions we want to allocate interrelated resources. They arise naturally in applications like spectrum auctions where a bidder's valuation for a bundle of items need not be the sum of item values, and instead be a set function such as a submodular function.

A "paradigmatic" [Dobzinski et al. 2006; Abraham et al. 2012; Fotakis et al. 2017], "central" [Mu'alem and Nisan 2008; Dughmi and Vondrák 2011], and "arguably the most important" [Dobzinski 2007] problem in Algorithmic Mechanism Design is to design mechanisms for combinatorial auctions that are both *truthful* and *computationally-efficient* (see §2 for formal definitions). At the root of this problem is an inherent clash between computational-efficiency and truthfulness. On one hand, the celebrated VCG mechanism of [Vickrey 1961; Clarke 1971; Groves et al. 1973] is a truthful mechanism that returns the welfare maximizing allocation. Alas, this mechanism requires finding the welfare maximizing allocation exactly, which is not possible in poly(m, n) time for most valuations. On the other hand, from an algorithmic point of view, constant factor approximation algorithms exist for many interesting classes of valuations, but they are no longer truthful.

Authors' addresses: `sepehr.assadi@rutgers.edu`, `singla@cs.princeton.edu`

A particular case of this problem that has received significant attention is when the valuation functions of all the bidders are *submodular* functions as they capture the notion of diminishing return. There is no poly-time algorithm for finding the optimal allocation of submodular bidders [Mirrokni et al. 2008; Feige and Vondrák 2010; Dobzinski and Vondrák 2013] 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 [Lehmann et al. 2006] and this can be further improved to $(\frac{e}{e-1})$-approximation [Vondrák 2008], and even slightly better using demand queries [Feige and Vondrák 2006]. 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 a* ***purely algorithmic*** *point of view that ignores the strategic behavior of the bidders?*

Already more than a decade ago, Dobzinski, Nisan, and Schapira [Dobzinski et al. 2005] 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 [Dobzinski et al. 2006] to $O(\log^2 m)$, which in turn was improved to $O(\log m \cdot \log\log m)$ by [Dobzinski 2007], and then to $O(\log m)$ by [Krysta and Vöcking 2012]. Breaking this logarithmic barrier remained elusive until the recent $O(\sqrt{\log m})$ approximation breakthrough of [Dobzinski 2016a].

Our Contribution. Our main contribution is an almost *exponential* factor improvement over the $\Theta(\sqrt{\log m})$ approximation mechanism of [Dobzinski 2016a].

THEOREM 1. *There exists a randomized and universally truthful mechanism for combinatorial auctions with submodular valuations that achieves an* $O((\log\log m)^3)$ *approximation 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 [Dobzinski et al. 2006; Dobzinski 2007; Krysta and Vöcking 2012; Dobzinski 2016a]) actually works for the much broader class of *XOS* valuations (see §2 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 $\text{poly}(\log\log(m))$.

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 only achieve a $m^{\Omega(1)}$-approximation [Papadimitriou et al. 2008; Dobzinski 2011; Dughmi and Vondrák 2011; Dobzinski and Vondrák 2012; 2012; Daniely et al. 2015] (the latter assuming $\text{RP} \neq \text{NP}$).

Nevertheless, these results do not apply to mechanisms that are allowed other natural types of queries, e.g., demand queries. This has led the researchers to study the communication complexity of this problem that can capture arbitrary queries to valuations [Nisan 2000; Blumrosen and Nisan 2002; Dobzinski et al. 2005; Nisan and

Segal 2006; Dobzinski and Vondrák 2013; Dobzinski et al. 2014; Dobzinski 2016b; Assadi 2017; Braverman et al. 2018; Ezra et al. 2018]. Only recently, [Assadi et al. 2020] proved the first separation between the communication complexity of truthful mechanisms and general algorithms. Another relevant recent paper is of [Cai et al. 2020] who show how to run the mechanisms in this paper using only value queries (i.e., no demand queries) under some mild relaxations of truthfulness.

2. BACKGROUND AND COMPARISON TO PRIOR WORK

Notation and Preliminaries. We denote by N the set of n bidders and by M the set of m 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 $\boldsymbol{p}(M') := \sum_{j \in M'} p_j$. is an allocation A' consisting of $A_i \cap M'$ for every $i \in N'$.

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 [Feige 2006] 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 $S \subseteq M$. 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 $\boldsymbol{q} = (q_1, \ldots, q_m)$ 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., $j \in A_i$.

We are going to make the following simplifying assumption. This assumption is natural and can be lifted entirely; see [Assadi and Singla 2019, Section 6].

Assumption 1. *We assume there exists two non-negative numbers $\psi_{\min} \leq \psi_{\max}$ such that (i) $\Psi := \psi_{\max}/\psi_{\min}$ is bounded by some fixed* poly(m)*, and (ii) for every valuation, supporting price of any item for any clause belongs to $\{0\} \cup [\psi_{\min} : \psi_{\max}]$. We further assume that the mechanism is given $\psi_{\min}$ and $\psi_{\max}$ as input.*

Combinatorial Auctions. 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 poly(m, n) time given query access to valuation functions. The most standard queries considered in this context are (i) value queries: "Given a bundle S, what is the value of $v(S)$?", and (ii) demand queries: "Given a vector of prices on items $\boldsymbol{p}$, what is the "most demanded" bundle, i.e., a bundle $S \in \arg\max_{S'} \{v(S') - \boldsymbol{p}(S')\}$?". At the same time, 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 in, e.g., [Lavi and Swamy 2005] and [Dughmi et al. 2011]).

Fixed-Price Auctions (FPAs). All previous attempts for designing truthful and computationally-efficient mechanisms for this problem [Dobzinski et al. 2006; Dobzinski 2007; Krysta and Vöcking 2012; Dobzinski 2016a], 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 *supporting prices* in an optimal allocation; the rest can be handled by a simple *fixed-price auction* using these prices. In the following, we first define fixed-price auctions and then formalize this observation in Lemma 1.

For an *ordered* set N of bidders, M of items, and a price vector $\boldsymbol{p}$, we define $\texttt{FixedPriceAuction}(N, M, \boldsymbol{p})$ as follows: Iterate over the bidders i of the ordered set N in the given order and allocate $A_i \in \arg\max_{S \subseteq M}\{v_i(S) - \boldsymbol{p}(S)\}$ to bidder i and update $M \leftarrow M \setminus A_i$.

It is easy to see that FixedPriceAuction can be implemented using one demand query per bidder. It is truthful because bidders have no influence on the prices.

The following lemma gives a key property of this auction.Variants of this lemma have already appeared in the literature, e.g., in [Dobzinski et al. 2006; Dobzinski 2007; Feldman et al. 2015; Dobzinski 2016a; Ehsani et al. 2018].

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

Intuitively, Lemma 1 says that if we run a FPA with "approximately correct" prices for a subset of items (compared to their supporting prices in an optimal allocation), the resulting welfare would be almost as good as the optimal welfare obtained from this subset of items.

Let us now show how can one use FPAs and Lemma 1 to obtain a very simple $O(\log m)$ approximation truthful mechanism. Given Assumption 1, consider a FPA that draws a random integer $k \in \left[0, \lceil \log_2 \Psi \rceil\right]$ and sets a price $\boldsymbol{p}$ of $2^k \cdot \psi_{\min}$ on every item. By Lemma 1, we know that the welfare of this auction is at least a constant fraction of the sum of supporting prices around $2^k \cdot \psi_{\min}$. Since $\log_2(\frac{\psi_{\max}}{\psi_{\min}}) = O(\log m)$, all non-zero supporting prices have at least $1/\Omega(\log m)$ chance to appear, which gives an $O(\log m)$ approximation.

Our Techniques and Comparison to Prior Work. Our mechanism also uses FPAs but departs from prior work in the following key conceptual way. Instead of learning a coarse-grained "statistics" about the prices, say, the range they should belong to [Dobzinski et al. 2006; Dobzinski 2007], and using these statistics to "guess" a small number of good prices (e.g., $O(1)$ prices in [Dobzinski et al. 2006; Dobzinski 2007], and $O(\sqrt{\log m})$ in [Dobzinski 2016a]), we 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 allows us to get much more accurate prices and ultimately leads to the exponential improvement.

A cornerstone of our approach is a "learning process" which starts with a simple guess of item prices and *iteratively* refine this guess. Each iteration of this process involves running *several* FPAs with the prices learned so far and use the resulting allocations to further refine our learned prices. The key to the analysis of this mechanism is the "Learnable-Or-Allocatable Lemma": Roughly speaking, we prove that in each iteration, we can either refine our learned prices significantly (Learnable), or the FPA 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 FPAs that stems from their greedy nature: if we run a FPA 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 2016a] and is closely related to other similar results about greedy algorithms for maximum matching [Konrad et al. 2012], matroid intersection [Guruganesh and Singla 2017], and constrained submodular maximization [Norouzi-Fard et al. 2018].

3. THE HIGH-LEVEL OVERVIEW

We now give a streamlined overview of our main mechanism in Theorem 1. 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 $\boldsymbol{q} = (q_1, \ldots, q_m)$ be its supporting prices (obviously, O and $\boldsymbol{q}$ are unknown). For now, let us assume that every q_j belongs to $\left\{1, \gamma, \gamma^2, \ldots, \gamma^K\right\}$, for some $K = O(\log m)$ by Assumption 1 (and hence prices are roughly poly(m) large).

The crux of our mechanism is to "learn" $\boldsymbol{q}$, namely, find another price vector $\boldsymbol{p}$ such that for some subset $C \subseteq M$ with $\boldsymbol{q}(C) \approx \mathsf{val}(O)$, $\boldsymbol{p}$ *point-wise* γ-approximates $\boldsymbol{q}$ for items in C (i.e., within a multiplicative factor of γ). Having learned such prices, we can run a FPA with prices $\boldsymbol{p}$, and by Lemma 1, obtain an allocation with welfare $\approx \gamma \cdot \mathsf{val}(O)$.

In order to obtain the price vector $\boldsymbol{p}$, we start with a rough guess $\boldsymbol{p}^{(1)}$ 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 $\boldsymbol{p}^{(i)}$ learned so far to find α new price vectors $\boldsymbol{p}^{(i)}_1, \ldots, \boldsymbol{p}^{(i)}_\alpha$, and "explore" for *each* item $j \in M$ which of these α vectors best represents its price in $\boldsymbol{q}$, and then assign that price to item j in $\boldsymbol{p}^{(i+1)}$. We continue this for β iterations until we converge to the desired price vector $\boldsymbol{p} := \boldsymbol{p}^{(\beta+1)}$, 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.

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 $p^{(1)}_j \in \boldsymbol{p}^{(1)}$ to be 1, and so with our assumption that $q_j \in \left\{1, \gamma, \ldots, \gamma^K\right\}$, price $p^{(i)}_j$ will (γ^K)-approximate $q_j \in \boldsymbol{q}$. We want $p^{(2)}_j$ 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 $\boldsymbol{p}_1^{(1)}, \ldots, \boldsymbol{p}_\alpha^{(1)}$ accordingly. We repeat the same idea for later iterations as well, maintaining that in iteration i, price $p_j^{(i)} \in \boldsymbol{p}^{(i)}$ 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 $(\gamma^{K/\alpha^{i}})$-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.

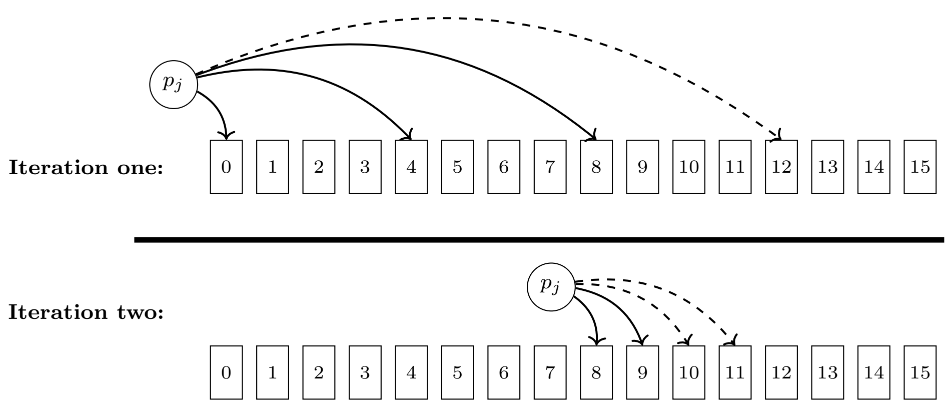

Fig. 1. A trajectory of the prices of a single item in the mechanism. Here, $\alpha = 4$ (number of auctions per iteration) and $\beta = 2$ (number of iterations). Each block i corresponds to price γ^i. Arrows correspond to the price of this item in the FPA; a solid arrow means the item was sold at that price, while a dashed arrow means it was not. The learned price of this item is γ^9.

Part (ii) **– how to explore a new price.** For this part, we build on a key idea from Dobzinski [2016a] in using FPAs themselves as a "proxy" for determining correctness of a guess for item prices. The idea is as follows: suppose we run a FPA 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 FPA is a "good indicator" that its price in $\boldsymbol{q}$ is at least as high as the price used in this FPA. Such an idea was used in Dobzinski [2016a] 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. 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 FPA with prices $\boldsymbol{p}^{(\beta+1)}$ gives a $\gamma = O(\alpha\beta)$-approximation by Lemma 1.

This outline oversimplifies many details. Let us briefly mention two here. Firstly, running FPAs 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 (for the purpose of this discussion we simply assumed the existence of this gap, while in the actual mechanism we need to *create* this gap). 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 FPA 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 FPAs 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 FPAs 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 FPA 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 $\boldsymbol{q}(C^{(i+1)}) \approx \boldsymbol{q}(C^{(i)})$, we have a more refined statement in which the LHS is replaced with $\boldsymbol{q}$ of items allocated *only* to bidders in $N_{>i}$.

4. CONCLUDING REMARKS AND OPEN PROBLEMS

In this letter, we reported on our work in [Assadi and Singla 2019] in designing a computationally-efficient and universally truthful mechanism for combinatorial auctions with submodular (even XOS) bidders with $O((\log\log m)^3)$-approximation.

The obvious question left open at this point is whether this gap can be improved further. We do not believe that our $O((\log\log m)^3)$ approximation is the best possible (in fact, our bounds can be slightly improved already; see [Assadi and Singla 2019]). On the other hand, the limit of our approach seems to be an $\Omega(\log\log m)$ approximation. Can one improve the approximation ratio all the way down to a constant? We shall note that even improving the approximation ratio of our mechanism down to $O(\log\log m)$ seems challenging.

REFERENCES

Abraham, I., Babaioff, M., Dughmi, S., and Roughgarden, T. 2012. Combinatorial auctions with restricted complements. In *Proceedings of EC*.

Assadi, S. 2017. Combinatorial auctions do need modest interaction. In *Proceedings of EC*.

Assadi, S., Khandeparkar, H., Saxena, R., and Weinberg, M. 2020. Separating the communication complexity of truthful and non-truthful combinatorial auctions. In *Proceedings of STOC*.

Assadi, S. and Singla, S. 2019. Improved truthful mechanisms for combinatorial auctions with submodular bidders. In *Proceedings of FOCS*.

Blumrosen, L. and Nisan, N. 2002. Auctions with severely bounded communication. In *Proceedings of FOCS*.

Braverman, M., Mao, J., and Weinberg, S. M. 2018. On simultaneous two-player combinatorial auctions. In *Proceedings of SODA*.

Cai, L., Thomas, C., and Weinberg, S. M. 2020. Implementation in advised strategies: Welfare guarantees from posted-price mechanisms when demand queries are np-hard. In *Proceedings of ITCS*.

Clarke, E. H. 1971. Multipart pricing of public goods. *Public choice 11,* 1, 17–33.

Daniely, A., Schapira, M., and Shahaf, G. 2015. Inapproximability of truthful mechanisms via generalizations of the VC dimension. In *Proceedings of STOC*.

Dobzinski, S. 2007. Two randomized mechanisms for combinatorial auctions. In *Proceedings of APPROX and RANDOM*. 89–103.

Dobzinski, S. 2011. An impossibility result for truthful combinatorial auctions with submodular valuations. In *Proceedings of STOC*.

Dobzinski, S. 2016a. Breaking the logarithmic barrier for truthful combinatorial auctions with submodular bidders. In *Proceedings of STOC*.

Dobzinski, S. 2016b. Computational efficiency requires simple taxation. In *Proceedings of FOCS*.

Dobzinski, S., Nisan, N., and Oren, S. 2014. Economic efficiency requires interaction. In *Proceedings of STOC*.

Dobzinski, S., Nisan, N., and Schapira, M. 2005. Approximation algorithms for combinatorial auctions with complement-free bidders. In *Proceedings of STOC*.

Dobzinski, S., Nisan, N., and Schapira, M. 2006. Truthful randomized mechanisms for combinatorial auctions. In *Proceedings of STOC*.

Dobzinski, S. and Vondrák, J. 2012. The computational complexity of truthfulness in combinatorial auctions. In *Proceedings of EC*.

Dobzinski, S. and Vondrák, J. 2013. Communication complexity of combinatorial auctions with submodular valuations. In *Proceedings of SODA*.

Dughmi, S., Roughgarden, T., and Yan, Q. 2011. From convex optimization to randomized mechanisms: toward optimal combinatorial auctions. In *Proceedings of STOC*. ACM.

DUGHMI, S. AND VONDRÁK, J. 2011. Limitations of randomized mechanisms for combinatorial auctions. In *Proceedings of FOCS*.

EHSANI, S., HAJIAGHAYI, M., KESSELHEIM, T., AND SINGLA, S. 2018. Prophet secretary for combinatorial auctions and matroids. In *Proceedings of SODA*.

EZRA, T., FELDMAN, M., NEYMAN, E., TALGAM-COHEN, I., AND WEINBERG, S. M. 2018. Complement-free couples must communicate: A hardness result for two-player combinatorial auctions. *CoRR abs/1811.09871*.

FEIGE, U. 2006. On maximizing welfare when utility functions are subadditive. In *Proceedings of STOC*.

FEIGE, U. AND VONDRÁK, J. 2006. Approximation algorithms for allocation problems: Improving the factor of 1 - 1/e. In *Proceedings of FOCS*.

FEIGE, U. AND VONDRÁK, J. 2010. The submodular welfare problem with demand queries. *Theory of Computing 6,* 1, 247–290.

FELDMAN, M., GRAVIN, N., AND LUCIER, B. 2015. Combinatorial auctions via posted prices. In *Proceedings of SODA*.

FOTAKIS, D., KRYSTA, P., AND VENTRE, C. 2017. Combinatorial auctions without money. *Algorithmica 77,* 3, 756–785.

GROVES, T. ET AL. 1973. Incentives in teams. *Econometrica 41,* 4, 617–631.

GURUGANESH, G. P. AND SINGLA, S. 2017. Online matroid intersection: Beating half for random arrival. In *Proceedings of IPCO*. Springer.

KONRAD, C., MAGNIEZ, F., AND MATHIEU, C. 2012. Maximum matching in semi-streaming with few passes. In *Proceedings of APPROX and RANDOM*.

KRYSTA, P. AND VÖCKING, B. 2012. Online mechanism design (randomized rounding on the fly). In *Proceedings of ICALP*. Springer.

LAVI, R. AND SWAMY, C. 2005. Truthful and near-optimal mechanism design via linear programming. In *Proceedings of FOCS*.

LEHMANN, B., LEHMANN, D. J., AND NISAN, N. 2006. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior 55,* 2, 270–296.

MIRROKNI, V. S., SCHAPIRA, M., AND VONDRÁK, J. 2008. Tight information-theoretic lower bounds for welfare maximization in combinatorial auctions. In *Proceedings of EC*.

MU'ALEM, A. AND NISAN, N. 2008. Truthful approximation mechanisms for restricted combinatorial auctions. *Games and Economic Behavior 64,* 2, 612–631.

NISAN, N. 2000. Bidding and allocation in combinatorial auctions. In *Proceedings of EC*.

NISAN, N. AND SEGAL, I. 2006. The communication requirements of efficient allocations and supporting prices. *J. Economic Theory 129,* 1, 192–224.

NOROUZI-FARD, A., TARNAWSKI, J., MITROVIC, S., ZANDIEH, A., MOUSAVIFAR, A., AND SVENSSON, O. 2018. Beyond 1/2-approximation for submodular maximization on massive data streams. In *Proceedings of ICML*.

PAPADIMITRIOU, C. H., SCHAPIRA, M., AND SINGER, Y. 2008. On the hardness of being truthful. In *Proceedings of FOCS*.

VICKREY, W. 1961. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance 16,* 1, 8–37.

VONDRÁK, J. 2008. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of STOC*.