Implementation in Advised Strategies: Welfare Guarantees from Posted-Price Mechanisms When Demand Queries Are NP-Hard

Linda Cai
Princeton University, Princeton, NJ, USA
tcai@cs.princeton.edu

Clay Thomas
Princeton University, Princeton, NJ, USA
claytont@cs.princeton.edu

S. Matthew Weinberg
Princeton University, Princeton, NJ, USA
smweinberg@princeton.edu

Abstract

State-of-the-art posted-price mechanisms for submodular bidders with m items achieve approximation guarantees of $O((\log\log m)^3)$ [1]. Their truthfulness, however, requires bidders to compute an NP-hard demand-query. Some computational complexity of this form is unavoidable, as it is NP-hard for truthful mechanisms to guarantee even an $m^{1/2-\varepsilon}$-approximation for any $\varepsilon > 0$ [21]. Together, these establish a stark distinction between computationally-efficient and communication-efficient truthful mechanisms.

We show that this distinction disappears with a mild relaxation of truthfulness, which we term implementation in advised strategies. Specifically, *advice* maps a tentative strategy either to that same strategy itself, or one that dominates it. We say that a player follows advice *as long as they never play actions which are dominated by advice*. A poly-time mechanism guarantees an α-approximation in implementation in advised strategies if there exists advice (which runs in poly-time) for each player such that an α-approximation is achieved whenever all players follow advice. Using an appropriate bicriterion notion of approximate demand queries (which can be computed in poly-time), we establish that (a slight modification of) the [1] mechanism achieves the same $O((\log\log m)^3)$-approximation in implementation in advised strategies.

2012 ACM Subject Classification Theory of computation → Algorithmic mechanism design; Theory of computation → Solution concepts in game theory

Keywords and phrases Combinatorial auctions, Posted-Price mechanisms, Submodular valuations, Incentive compatible

Digital Object Identifier 10.4230/LIPIcs.ITCS.2020.61

Funding *S. Matthew Weinberg*: Supported by NSF CCF-1717899.

Acknowledgements We thank Sepehr Assadi, Sahil Singla and the anonymous reviewers for helpful discussions.

1 Introduction

Combinatorial auctions have been at the forefront of Algorithmic Game Theory since its inception as a lens through which to study the relative power of *algorithms* for honest agents versus *mechanisms* for strategic agents. Specifically, there are n buyers with combinatorial valuations $v_1(\cdot), \ldots, v_n(\cdot)$ over subsets of m items, and the designer wishes to allocate the items so as to maximize the *welfare*, $\sum_i v_i(S_i)$ (where S_i is the set allocated to bidder i). Without concern for computation/communication/etc., the celebrated Vickrey-Clarke-Groves

mechanism [41, 9, 28] provides a black-box reduction from precisely optimal mechanisms to precisely optimal algorithms. Of course, precisely optimal algorithms are NP-hard and require exponential communication in most settings of interest (for example, when buyers have submodular valuations over the items, which we'll take as the running example for the rest of the introduction), rendering VCG inapplicable. On the algorithmic front, poly-time/poly-communication constant-factor approximation algorithms are known [38, 30, 33, 4, 42, 24] and a central direction in algorithmic mechanism design is understanding whether these guarantees are achievable by computationally/communication efficient truthful mechanisms as well.

From the communication complexity perspective, this problem is still wide open: state-of-the-art truthful mechanisms guarantee an $O((\log\log m)^3)$-approximation [1], yet no lower bounds separate achievable guarantees of mechanisms from algorithms (that is, it could very well be the case that truthful, poly-communication mechanisms can achieve the same guarantees as poly-communication algorithms). From the computational perspective, however, a landmark result of Dobzinski and Vondrak establishes that for all $\varepsilon > 0$, an $m^{1/2-\varepsilon}$-approximation is NP-hard for truthful mechanisms [21]. As poly-time algorithms guarantee a $e/(e-1)$-approximation [42], this establishes a strong separation between computationally-efficient algorithms and computationally-efficient truthful mechanisms.

So while the communication perspective has seen exciting progress in recent years [15, 3], the computational perspective is generally considered fully resolved. In this paper, we present a new dimension to the computational perspective, motivated by the following two examples. Consider first the truthful mechanism of [1]. The core of the mechanism is a posted-price mechanism: it visits each bidder one at a time, posts a price p_j on each remaining item j, and offers the option to purchase any set S of items at total price $\sum_{j\in S} p_j$ (see Section 4 and Appendix B for a full description of their mechanism, which also includes randomization, pre-procesesing, and learning). The auxiliary parts of the mechanism run in poly-time,[1] and the offered prices can also be computed in poly-time. While it might sound like this mechanism should be poly-time, the catch is that it's NP-hard for the buyer find their utility-maximizing set, called a *demand query*. Therefore, the mechanism is either not truthful (because the buyers do not select their utility-maximizing sets), or requires solving an NP-hard problem (because the buyers pick their favorite sets). Still, the analysis of [1] and related mechanisms [12, 18, 31, 11, 26, 14, 22, 14] seems fairly robust, suggesting that perhaps they should maintain their guarantees under reasonable strategic behavior. Indeed, the focus of this paper is a novel solution concept (described below) under which the [1] $O((\log\log m)^3)$-approximation is maintained in polynomial time.

A New Solution Concept: Implementation in Advised Strategies. To get intuition for our solution concept, consider the following example due to [39]: there is only a single buyer, but the buyer can receive only k of the m items (this is the one-buyer case of Combinatorial Public Projects). Since there is just a single buyer, the obvious mechanism for the designer simply allows the buyer to pick any set of size k for free (call this the "Set-For-Free" mechanism). The same catch is that it is NP-hard for the buyer to pick their favorite set, so Set-For-Free is again not truthful (because the buyer picks a suboptimal set) or solving an NP-hard problem (because they find their favorite set). In fact, [39] establishes that it is NP-hard for truthful mechanisms to achieve a $m^{1/2-\varepsilon}$-approximation for any $\varepsilon > 0$. Algorithmically, a poly-time $e/(e-1)$-approximation is known [35], providing again a strong separation.

[1] Rather, they can be slightly modified to run in poly-time – see Section 4 and Appendix B.

We ask instead: what should one reasonably expect to happen if a strategic buyer participated in Set-For-Free? Consider the set S output by the poly-time algorithm of [35]. It is certainly reasonable for the buyer to select some set $T \neq S$: perhaps a different heuristic finds a better set. But it seems irrational for the buyer to select some set T with $v(T) < v(S)$. We therefore pose that there should be *some* reasonable solution concept under which Set-For-Free guarantees an $e/(e-1)$-approximation. Indeed, Set-For-Free guarantees an $e/(e-1)$-approximation under our proposed "implementation in advised strategies."

Formally, we will think of Set-For-Free as simply asking the buyer to report a set of size at most k, and then awarding them that set for free. In addition, the designer provides *advice*: a Turing machine $A(\cdot,\cdot)$ which takes as input the buyer's valuation $v(\cdot)$ (possibly as a circuit/Turing machine itself, or accessing it via value queries) and a tentative set T, then recommends a set S to purchase that is at least as good as T. Specifically in Set-For-Free, we will think of the advice as running the [35] approximation algorithm to get a set S' and outputting $S := \arg\max\{v(T), v(S')\}$. We say that a bidder *follows advice* if they select a set S with $A(v, S) = S$. The idea is that it seems irrational for the buyer to select a set without this property, when the advice gives a poly-time algorithm to improve it.

For a general mechanism, we think of advice as a Turing machine which takes as input the current state of the mechanism, the buyer's valuation, and a tentative action, then advises an (maybe the same, maybe different) action to take. Importantly, we say that advice is *useful* if for all strategies s, either the advice maps s to itself, or to another strategy which dominates it (see Section 2 for full definition). Intuitively, this suggests that it is irrational for a buyer to use a strategy which advice does not map to itself. We postpone to Section 2 a formal definition of what it means to follow advice, but note here that our definition is a natural relaxation of dominant strategies: if a mechanism has a dominant strategy s, then the only strategy which follows advice that recommends s is s itself. We say that mechanism guarantees a poly-time α-approximation in implementation in advised strategies whenever the mechanism itself concludes in poly-time, and there exists poly-time advice A such that an α-approximation is guaranteed whenever all bidders follow advice A. Again, note that the assumption on bidder behavior is quite permissive: they need not play a dominant, or even undominated strategy. We just assume they do not play a strategy which the advice itself dominates.

Advice via Approximate Demand Queries. We now revisit posted-price mechanisms, which achieve approximation guarantees of $O((\log\log m)^3)$, but whose truthfulness requires buyers to compute NP-hard demand queries. Instead, we pursue guarantees in implementation in advised strategies. For a posted-price mechanism with price vector $\mathbf{p}$, our proposed advice will take as input a tentative set T for purchase, and the buyer's valuation $v(\cdot)$, and recommend a set S guaranteeing $v(S) - \mathbf{p}(S) \geq v(T) - \mathbf{p}(T)$.[2] More specifically, our advice will compute a tentative recommendation S' independently of T, then simply recommend $\arg\max\{v(S') - \mathbf{p}(S), v(T) - \mathbf{p}(T)\}$. Again, our behavioral assumption does not assume that the buyer will purchase the set S' tentatively recommended, just that they will not irrationally ignore the advice in favor of a lower-utility set.

The remaining challenge is now to find concrete advice under which the [1] approximation guarantees are maintained. A first natural attempt is simply an approximate demand oracle: have a tentative recommendation S' with $v(S') - \mathbf{p}(S') \geq c \cdot \max_T\{v(T) - \sum_{j\in T} p_j\}$. Unfortunately, even this is NP-hard for any $c = \Omega(1/m^{1-\varepsilon})$ (for any $\varepsilon > 0$) [25]. Instead,

[2] Throughout the paper we will use notation $\mathbf{p}(S) := \sum_{i\in S} \mathbf{p}_i$.

we design bicriterion approximate demand oracles. Specifically, for some $c, d < 1$, a (c, d)-approximate demand oracle produces a set S' satisfying $v(S') - \mathbf{p}(S') \geq c \cdot \max_T\{v(T) - \mathbf{p}(T)/d\}$. That is, the guaranteed utility is at least an c-fraction of the optimum *if all prices were increased by a factor of* $1/d$. We design a simple greedy $(1/2, 1/2)$-approximation in poly-time (based on [33]), and further establish that the [1] mechanism maintains its approximation guarantee up to an additional $\min\{c, d\}$ factor when bidders follow advice provided in this manner by a (c, d)-approximate demand oracle. This allows us to conclude the main result of this paper:

▶ **Theorem 1.** *There exists a poly-time mechanism which achieves an* $O((\log\log m)^3)$*-approximation to the optimal welfare for any number of submodular buyers in implementation in advised strategies.*

1.1 Roadmap

Combinatorial auctions have a long history within AGT, along with related problems like Combinatorial Public Projects. The most related work is overviewed in Section 1, but we provide additional context in Section 1.2. Section 2 contains a formal definition of implementation in advised strategies, repeating our motivating examples and providing additional discussion.

In Section 3, we design our poly-time $(1/2, 1/2)$-approximate demand oracles for submodular valuations. The proof is fairly simple, but we include the complete proof in the body for readers unfamiliar with [33] (readers familiar with [33] will find the outline simliar).

In Section 4, we establish that existing posted-price mechanisms maintain their approximation guarantees as long as buyers follow advice given by $(\Omega(1), \Omega(1))$-approximate demand oracles. We include a complete analysis of the main lemma of [26] concerning "fixed price auctions" for readers unfamiliar with this aspect (readers familiar with [26] will find the outline similar). We defer all aspects of the analysis of [1] to Appendix B.

Finally, in Section 5, we design simple poly-time $(\frac{1}{\sqrt{m}}, \frac{1}{1+\sqrt{m}})$-approximate demand oracles for subadditive valuations. This is essentially the best possible even for XOS valuations,[3] due to known lower bounds on welfare-maximization with value queries [17] and the results of Section 4.

1.2 Discussion and Related Work

There is a vast literature studying combinatorial auctions, which we will not attempt to overview in its entirety here. We summarize the lines of work most relevant to ours below.

VCG-based Mechanisms. The Vickrey-Clarke-Groves mechanism provides a poly-time/poly-communication black-box reduction from precise welfare maximization with a truthful mechanism to precise welfare maximization with an algorithm for any class of valuation functions [41, 9, 28]. The same reduction applies for "maximal-in-range" approximation algorithms, but this approach provably cannot achieve sub-polynomial approximations in poly-time (unless P = NP) or subexponential communication [8, 7, 10]. Still, in some regimes (e.g. arbitrary monotone valuations with poly-time/poly-communication, or XOS valuations with poly-time), no better than a $\Theta(\sqrt{m})$-approximation is achievable even with honest players, and a $\Theta(\sqrt{m})$-approximation is achievable via truthful VCG-based mechanisms [32, 17].

[3] More precisely, it is unconditionally hard to obtain a $(1/m^{1/2-\varepsilon}, 1/m^{1/2-\varepsilon})$-approximate demand query for XOS valuations using polynomially many black-box value queries.

Combinatorial Auctions in the Computational Model. Taking the above discussion into account, valuation function classes above XOS (including subadditive, or arbitrary monotone) are "too hard", in the sense that $\Theta(\sqrt{m})$ is the best approximation achievable in poly-time even without concern for incentives, and this guarantee can be matched by VCG-based truthful mechanisms. Other valuation function classes like Gross Substitutes are "easy", in the sense that precise welfare maximization is achievable in poly-time, so the VCG mechanism is poly-time as well. Submodular valuations are a fascinating middle ground. Here, an algorithmic $e/(e-1)$-approximations is possible in poly-time (and it is NP-hard to do better) [42, 34, 19], but a long series of works establishes that it is NP-hard for a truthful mechanism to even achieve an $m^{1/2-\varepsilon}$-approximation (for any $\varepsilon > 0$) [36, 7, 8, 10, 13, 20, 19, 21].

On this front, our work establishes that significantly better ($O((\log\log m)^3)$) guarantees are achievable with a slightly relaxed solution concept, matching the state-of-the-art in the communication model. In addition to the standalone motivation for the communication model discussed below, our work establishes that resolving key open questions (e.g. is there a constant-factor approximation in the communication model for submodular valuations?) may have strong implications in the computational model as well (via implementation in advised strategies).

Combinatorial Auctions in the Communication Model. In the communication model, only arbitrary monotone valuations are "too hard" per the above discussion: a 2-approximation is possible for subadditive valuations, and a $\frac{1}{1-(1-1/n)^n}$-approximation is possible for XOS valuations in poly-communication, both of which are tight [24, 17, 23]. Yet, no truthful constant-factor approximations are known (the state-of-the-art is $O((\log\log m)^3)$ for submodular/XOS [1] or $O(\log m \log\log m)$ for subadditive [12]). On the lower bounds side, *no* separations are known between the approximation ratios of truthful mechanisms and non-truthful algorithms using $\text{poly}(n, m)$ communication, even for *deterministic* truthful mechanisms (where the $O(\sqrt{m})$-approximation of [17] remains the state-of-the-art). Determining whether such a separation exists is the central open problem of this agenda (e.g. [15, 3]).

On this front, our work in some sense unifies the state-of-the-art for submodular valuations in the communication and computational models via implementation in advised strategies. So in addition to the standalone interest in establishing (or disproving) a separation in the communication model, such a result will now likely have implications in the computational model as well.

Posted Price Mechanisms. Posted-price mechanisms are ubiquitous in mechanism design, owing to their simplicity and surprising ability to guarantee good approximations through a variety of lenses [31, 11, 26, 14, 22, 1]. Very recent work also establishes posted-price mechanisms as the unique class of mechanisms which is "strongly obviously strategy-proof" [37]. One minor downside of these mechanisms is that they require buyers to compute NP-hard demand queries. Our work formally mitigates this downside under implementation in advised strategies. For example, our work immediately extends the price of anarchy bounds of [11] to hold in equilibria which are poly-time learnable for submodular buyers (previously the equilibria required computation of demand queries).

Combinatorial Public Projects. Combinatorial Public Projects is a related problem, which has also received substantial attention. Here, the designer may select any set of k items, but *every* bidder receives all k items (instead of the bidders each receiving disjoint sets of

items, as in auctions). We used the single submodular bidder Combinatorial Public Projects problem as a motivating example due to the hardness results established in [39]: no poly-time truthful mechanism can achieve an approximation ratio better than $O(\sqrt{m})$ (unless P = NP). Contrast this with the general communication model, where Set-For-Free is truthful and precisely optimizes welfare. Follow up work of [6] establishes that while the single-bidder CPPP is inapproximable only because demand queries are NP-hard, the strong multi-bidder inapproximability results of [36, 8] hold even when bidders have access to demand oracles.

[31, 14, 1] already establish that the aforementioned computational separations no longer hold with demand oracles (so the story for combinatorial auctions differs greatly from combinatorial public projects). Our work further establishes that the same guarantees are achievable in truly polynomial time, under a relaxed solution concept.

(c, d)-Approximate Demand Oracles. To the best of our knowledge, bicriterion approximate demand oracles have not previously been considered. However, prior work regarding approximation algorithms for nonnegative submodular functions with bounded curvature subject to a matroid constraint designs a randomized $(1 - 1/e, 1 - 1/e)$-approximate demand oracle for submodular functions (under a different name) [40, 27, 29]. We include our $(1/2, 1/2)$-approximate demand oracles for submodular functions as they are deterministic and significantly simpler (note also that determinism makes our related solution concepts significantly cleaner).

Related Solution Concepts. The most related existing solution concept to ours is Algorithmic Implementation in Undominated Strategies [2]. Here, a mechanism achieves an α-approximation if whenever players play *any undominated strategy*, the resulting allocation is an α-approximation (and for all dominated strategies, there is a poly-time algorithm to find an undominated one which dominates it). For posted-price mechanisms, this solution concept has no bite as the only undominated strategy is to pick the utility-maximizing set, and it is not possible to find this set in poly-time. We establish in Observation 10 that implementation in advised strategies is a relaxation of algorithmic implementation in undominated strategies. The purpose of our novel solution concept is to have bite even when it is NP-hard to find an undominated strategy.

2 Implementation in Advised Strategies

Motivated by the example of [39], we first relax the requirement that a mechanism be truthful, instead requiring that a mechanism achieve its approximation guarantee whenever players behave in a manner which is not clearly irrational. Before proposing our formal definition, let's examine it applied to two motivating examples.

Example One: [39]. There is a single buyer with submodular valuation function $v(\cdot)$. The seller's mechanism (Set-For-Free) allows the buyer to state any set of size k, and receive that set for free. Recall that it is NP-hard for the buyer to find their favorite set of size k – so if the mechanism is to be truthful, it is not poly-time (unless P = NP). The buyer can indeed find an $e/(e-1)$-approximation in poly-time [35], but assuming the buyer will run this particular algorithm (or any specific approximation algorithm) is perhaps too strong an assumption.

Instead, we assume simply that the buyer picks a set yielding *at least as much utility* as this $e/(e-1)$-approximation. Specifically, we will think of the designer as providing a poly-time mechanism (Set-For-Free – the buyer states a set of size k and receives that set

for free), and a poly-time *advice algorithm* (takes as input the buyer's valuation function $v(\cdot)$, and a tentative set S, then runs the $e/(e-1)$-approximation to get a set T and outputs $\arg\max\{v(S), v(T)\}$, tie-breaking for S). Intuitively, we are claiming that it is certainly rational for the buyer to purchase a set other than T, but that it is irrational to purchase a set with $v(S) < v(T)$.

Example Two: Posted-Price Mechanisms. Consider now any posted-price mechanism. Again, we think of the designer as providing a poly-time mechanism (for all i, computes in poly-time a price vector to offer bidder i, based on interactions with bidders $< i$), along with a poly-time advice algorithm (takes as input the buyer's valuation function $v(\cdot)$, a tentative set S, computes in poly-time a set T and outputs $\arg\max\{v(S) - \mathbf{p}(S), v(T) - \mathbf{p}(T)\}$, again tie-breaking for S). Again note that we are claiming that it may be rational for the buyer to purchase a set other than T, but that it is irrational to purchase a set yielding lower utility than T.

Importantly, we emphasize that we assume the buyer achieves at least as much *utility* as recommended (a well-justified behavioral assumption, although not particularly convenient for welfare guarantees), and *not* that the buyer picks a set guaranteeing them at least as much welfare (more convenient for analyzing welfare guarantees, but an unmotivated assumption). With these instantiations in mind, we now build up language to present our formal definition.

▶ **Definition 2** (Mechanism as an Extended Form Game)**.** *Formally, a* mechanism *is just an extended form game: at every state, it solicits actions from one or more players and (possibly randomly) updates its state. With probability one, the mechanism eventually reaches a terminal state, and (possibly randomly) outputs an allocation of items and payments charged.*

A mechanism is poly-time *if every state update is poly-time computable, and the mechanism reaches a terminal state with probability one after* $\text{poly}(n, m)$ *updates.*

▶ **Definition 3** (Strategies, Utility, and Dominance)**.** *A* strategy $s(\cdot)$ *for player i is simply a mapping from the current state x of the mechanism to an action $s(x)$.*

We denote by $u_i(v_i, \vec{s})$ the expected utility of player i when their valuation function is $v_i(\cdot)$ and the players use strategy profile $\vec{s}$.

Strategy $s(\cdot)$ dominates *strategy $s'(\cdot)$ for player i with valuation $v_i(\cdot)$ if for all s_{-i}, $u_i(v_i, s_i; \vec{s}_{-i}) \geq u_i(v_i, s'_i; \vec{s}_{-i})$, and there exists an $\vec{s}_{-i}$ such that $u_i(v_i, s_i; \vec{s}_{-i}) > u_i(v_i, s'_i; \vec{s}_{-i})$.*

▶ **Definition 4** (Advice)**.** *Advice is a function $A(\cdot, \cdot, \cdot)$ which takes as input the valuation $v_i(\cdot)$ of a player, a state x of a mechanism, and a tentative action a, then (possibly randomly) outputs an advised action $A(v_i, x, a)$. We say that advice is poly-time if it is poly-time computable.*

Observe that every advice $A(\cdot, \cdot, \cdot)$, valuation function $v_i(\cdot)$, and tentative strategy $s(\cdot)$ induces a strategy $A^{v_i,s}(\cdot)$ with $A^{v_i,s}(x) := A(v_i, x, s(x))$.

▶ **Definition 5** (Useful Advice)**.** *We say that advice A is* useful *if:*

1. *For all $s(\cdot)$, and all $v_i(\cdot)$, either $A^{v_i,s}(\cdot) = s(\cdot)$, or $A^{v_i,s}(\cdot)$ dominates $s(\cdot)$ (for valuation $v_i(\cdot)$).*
2. *For all $s(\cdot)$ and all $v_i(\cdot)$, $A^{v_i,A^{v_i,s}}(\cdot) = A^{v_i,s}(\cdot)$ (Advice is idempotent – applying advice to $s(\cdot)$ twice is the same as applying it once).*

Intuitively, Property 1 guarantees that the bidder should indeed follow advice given by A instead of whatever strategy they had originally planned. Property 2 guarantees essentially that the bidder does not get "stuck" in an exponentially-long loop trying to repeatedly improve their strategy via advice (because the loop terminates after one iteration). Let us briefly observe the following implication of our definition (which we explore further when revisiting our two main examples):

▶ **Observation 6.** *Let $A(\cdot,\cdot,\cdot)$ be useful, and let $A(v_i, x, a) \neq a$. Then for all s such that $s(x) = a$, $A^{v_i,s}(\cdot)$ dominates $s(\cdot)$.*

Intuitively, useful advice A separates strategies into *advised* strategies (where $A^{v_i,s}(\cdot) = s(\cdot)$), and *ill-advised* strategies (where $A^{v_i,s}(\cdot)$ dominates $s(\cdot)$). We say that a bidder follows advice if they use an advised strategy.

▶ **Definition 7** (Follows Advice)**.** *We say that $s(\cdot)$ is advised for $v_i(\cdot)$ under A if $A^{v_i,s}(\cdot) = s(\cdot)$. A bidder with valuation $v_i(\cdot)$* follows advice *A if they use a strategy which is advised under A.*

Intuitively, we are claiming that it is irrational for a player to use an ill-advised strategy $s(\cdot)$ (because they could instead use the strategy $A^{v_i,s}(\cdot)$, which dominates it).

▶ **Definition 8** (Implementation in Advised Strategies)**.** *We say that a mechanism M guarantees an α-approximation in* implementation in advised strategies *with advice A if A is useful and for all $v_1(\cdot), \ldots, v_n(\cdot)$, if all players follow advice A, the resulting allocation in M achieves (expected) welfare at least $\alpha \cdot \mathrm{OPT}(v_1, \ldots, v_n)$. If both M and A are poly-time, we say that M guarantees a poly-time α-approximation in implementation in advised strategies (without referencing A).*

Let's now briefly revisit our two examples through the formal definitions. First, recall the single-bidder mechanism Set-For-Free. Set-For-Free is poly-time: it takes as input a set and simply outputs that set, and terminates after one iteration. Consider the advice algorithm which takes as input $v(\cdot)$, and a set S, then runs the $e/(e-1)$-approximation algorithm of [35] to get a set T and outputs $\arg\max\{v(S), v(T)\}$, tie-breaking for S (the mechanism has only a single non-terminal state, so this completely specifies the advice). Then the advice indeed recommends a dominating strategy whenever it recommends $T \neq S$, and is idempotent (tie-breaking in favor of S is subtly necessary for this claim – if instead the advice tie-broke for T, then it might recommend an action distinct from S which does not dominate it). Moreover, observe that the bidder follows advice if and only if they choose a higher value set than produced by the algorithm, and therefore we're guaranteed a $e/(e-1)$-approximation whenever the bidder follows advice.

Posted-price mechanisms with poly-time computable prices are poly-time: they iteratively take as input a set from bidder i, assign that set to bidder i, then run a poly-time computation to determine the prices for bidder $i+1$. They terminate after n iterations. We will later design poly-time *approximate demand oracles*, which take as input $v_i(\cdot)$ and the price vector $\mathbf{p}$, and output a recommended set $D(v_i, \mathbf{p})$ in poly-time. For a posted-price mechanism, there are multiple non-terminal states, each corresponding to a different price vector $\mathbf{p}$. Our advice, on input $v_i, S, \mathbf{p}$, will advise the set $\arg\max\{v(S) - \mathbf{p}(S), v(D(v_i, \mathbf{p})) - \mathbf{p}(D(v_i, \mathbf{p}))\}$, again tie-breaking in favor of S.[4] If a strategy follows advice, it must, for all $\mathbf{p}$, select a set S satisfying $v_i(S) - \mathbf{p}(S) \geq v_i(D(v_i, \mathbf{p})) - \mathbf{p}(D(v_i, \mathbf{p}))$. It's not immediately clear why this property should provide meaningful welfare guarantees, but we will later argue that the right pairing of mechanism and notion of approximate demand oracle achieves polynomial-time welfare guarantees which match state-of-the-art guarantees for computationally-unbounded bidders.

[4] Subtly, note that tie-breaking in favor of T would violate the definition of usefulness, via Observation 6. Indeed, consider the strategy $s(\cdot)$ which purchases a utility-maximizing set on all prices $\neq \mathbf{p}$, and set S on $\mathbf{p}$. Then any advice which tie-breaks in favor of T on prices $\mathbf{p}$ does not map $s(\cdot)$ to itself, and does not dominate $s(\cdot)$ (because it generates the same utility on prices $\mathbf{p}$, and cannot generate strictly higher utility on any other prices, because $s(\cdot)$ is optimal).

Brief Discussion of Definitions. We chose our definitions with the goal of (a) providing a strict relaxation of truthfulness, and (b) doing so in a way that permits all rational behavior while (c) still eliminating enough irrational behavior to guarantee good welfare. We include in Appendix A a brief example motivating our decision to think of advice as *improving a given strategy* as opposed to *outright proposing a replacement strategy.* We conclude by establishing that implementation in advised strategies is a strict relaxation of truthfulness and algorithmic implementation in undominated strategies.

▶ **Observation 9.** *If player i with valuation v_i has a dominant strategy $s^*(\cdot)$ in mechanism M, and $A^{v_i,s}(\cdot) := s^*(\cdot)$ for all $s(\cdot)$, then the only strategy which follows advice is $s^*(\cdot)$ itself.*

▶ **Observation 10.** *Let M achieve an α-approximation in algorithmic implementation in undominated strategies. Then M achieves an α-approximation in implementation in advised strategies.*

Proof. Recall that algorithmic implementation in undominated strategies implies for all $v_i(\cdot)$ the existence of a poly-time function $f_{v_i}(\cdot)$ which takes as input a strategy $s(\cdot)$ and outputs a new strategy $f_{v_i}(s)$ such that if $s(\cdot)$ is dominated, $f_{v_i}(s)$ dominates s. Extend this so that $f_{v_i}(s) := s$ when $s(\cdot)$ is undominated. Now simply define $A^{v_i,s}(\cdot) := f_{v_i}(s)$, and then A is useful and poly-time. Moreover, the advised strategies are exactly the undominated strategies, so because M achieves an α-approximation whenever all players use undominated strategies, it also achieves an α-approximation whenever all players follow advice. ◀

Of course, the whole point of our definitions is that the designer may not be able to find a dominant (or even undominated) strategy in poly-time, and implementation in advised strategies has bite even under these circumstances while the previous solution concepts do not.

3 Approximate demand oracles

In this section, we develop our poly-time advice for posted-price mechanisms in the form of an *approximate demand oracle.* Recall that a demand oracle for valuation function $v(\cdot)$ takes as input a price vector $\mathbf{p}$ and outputs a set in $\arg\max_{S\subseteq M}\{v(S) - \mathbf{p}(S)\}$. Recall also that implementing a demand oracle is NP-hard when $v(\cdot)$ is submodular. In fact, it is NP-hard to even guarantee better than a m-approximation when $v(\cdot)$ is submodular (more precisely, for any $\varepsilon > 0$ it is NP-hard to guarantee a set T satisfying $v(T) - \mathbf{p}(T) \geq \frac{1}{O(m^{1-\varepsilon})} \cdot \max_S\{v(S) - \mathbf{p}(S)\}$ [25]). Motivated by this, we pursue instead a bicriterion approximation. Specifically:

▶ **Definition 11.** *For any $c, d \leq 1$, a (c, d)-*approximate demand oracle *takes as input a valuation function $v(\cdot)$ and a price vector $\mathbf{p}$ and outputs a set of items S such that*

$$v(S) - \mathbf{p}(S) \geq c \cdot \max_T\{v(T) - \mathbf{p}(T)/d\}.$$

That is, a (c, d)-demand oracle outputs a set guaranteeing at least a c-fraction of the optimal utility *if all prices were blown up by a factor of* $1/d$. We refer to the utility of the optimal bundle with these higher prices (i.e. $\max_T\{v(T) - \mathbf{p}(T)/d\}$) as the *benchmark* (so our goal is to be c-competitive with the benchmark). In this section, we establish that poly-time $(1/2, 1/2)$-approximate demand oracles exist for submodular functions, based on the simple greedy algorithm of [33].

Algorithm 1 SimpleGreedy$(v, \mathbf{p}, M)$.

$S \leftarrow \emptyset$
for $j = 1, \ldots, m$ **do** ▷ For items in an arbitrary order
  if $v(S \cup \{j\}) - v(S) \geq 2\mathbf{p}(j)$ **then** ▷ If the marginal gain is at least twice the price
    $S \leftarrow S \cup \{j\}$ ▷ Then allocate that item
return S

▶ **Proposition 12.** *When $v(\cdot)$ is submodular, SimpleGreedy is a $(1/2, 1/2)$-approximate demand oracle.*

Proof. Our proof follows by induction on the number of items m. Importantly, observe that SimpleGreedy is recursive. Specifically, if we do not allocate item 1, then the remainder of the for loop is simply SimpleGreedy$(v, \mathbf{p}, M \setminus \{1\})$. If we do allocate item 1, then the remainder of the for loop is simply SimpleGreedy$(v_{\{1\}}, \mathbf{p}, M \setminus \{1\})$, where $v_S(T) := v(S \cup T) - v(S)$. Also importantly, observe that $v_S(\cdot)$ is submodular whenever $v(\cdot)$ is submodular (like [33], this is the only part of the proof which requires submodularity instead of subadditivity).

Now we begin with the base case. Observe that when $m = 1$, SimpleGreedy purchases the item if and only if the value exceeds twice the price. So when SimpleGreedy purchases the item, it is optimal. When SimpleGreedy doesn't purchase the item, the benchmark is 0 (because we compete with the optimal utility when the prices are doubled, which is zero). So in both cases, it guarantees a the required $(1/2, 1/2)$-approximation. This proves the base case.

Now assume that the proposition holds for a fixed $m \geq 1$, and consider the case with $m + 1$ items. First, observe that if SimpleGreedy does not allocate item 1, it is because $v(1) < 2\mathbf{p}(1)$. By submodularity of $v(\cdot)$ (in fact, subadditivity suffices), this implies that $v(S) - 2\mathbf{p}(S) > v(S \cup \{1\}) - 2\mathbf{p}(S \cup \{1\})$ for all $S \ni 1$ (and in particular, that the optimum when prices are doubled does not contain item 1). By the inductive hypothesis, SimpleGreedy finds a $(1/2, 1/2)$-approximation for $v(\cdot)$ on $M \setminus \{1\}$, which by the previous sentence is also a $(1/2, 1/2)$-approximation for $v(\cdot)$ on M, completing the inductive step in this case.

It remains to consider the case where SimpleGreedy allocates item 1. Let $S_2 := S \setminus \{1\}$ denote the set output by SimpleGreedy$(v_{\{1\}}, \mathbf{p}, M \setminus \{1\})$, and let $O^* := \arg\max\{v(Y) - 2\mathbf{p}(Y)\}$ be the optimum bundle if prices were doubled. Then the inductive hypothesis guarantees:

$$v_{\{1\}}(S_2) - \mathbf{p}(S_2) \geq v_{\{1\}}(O^*)/2 - \mathbf{p}(O^* \setminus \{1\}).$$

Suppose first $1 \in O^*$. The inductive hypothesis then implies:

$$\begin{aligned} v(O^*)/2 - \mathbf{p}(O^*) &= v_{\{1\}}(O^*)/2 - \mathbf{p}(O^* \setminus \{1\}) + v(\{1\})/2 - \mathbf{p}(\{1\}) \\ &\leq v_{\{1\}}(S_2) - \mathbf{p}(S_2) + v(\{1\})/2 - \mathbf{p}(\{1\}) \\ &\leq v(S) - \mathbf{p}(S). \end{aligned}$$

Above, the first and third lines are simply expanding the definition of $v_{\{1\}}(\cdot)$, and the second line follows by inductive hypothesis. Observe that this concludes a $(1/2, 1/2)$-approximation in the case that $1 \in O^*$. Now, suppose instead that $1 \notin O^*$. Then we have:

$$\begin{aligned} v(O^*)/2 - \mathbf{p}(O^*) &\leq v(O^* \cup \{1\})/2 - \mathbf{p}(O^*) = v_{\{1\}}(O^*)/2 + v(\{1\})/2 - \mathbf{p}(O^*) \\ &\leq v_{\{1\}}(S_2) - \mathbf{p}(S_2) + v(\{1\})/2 \\ &\leq v_{\{1\}}(S_2) - \mathbf{p}(S_2) + v(\{1\}) - \mathbf{p}(1) \\ &= v(S) - \mathbf{p}(S). \end{aligned}$$

Above, the first line follows by monotonicity and expanding the definition of $v_{\{1\}}(\cdot)$. The second line follows by inductive hypothesis. The third line follows as $v(\{1\}) \geq 2\mathbf{p}(1)$ by assumption that SimpleGreedy allocates item 1. The final line follows again by expanding $v_{\{1\}}(\cdot)$. This concludes both cases of the inductive step, and the proof of the proposition. ◀

This concludes our development of bicriterion approximate demand oracles. The following section establishes that a wide class of posted-price mechanisms that achieve good guarantees when buyers use precise demand queries maintain their guarantees when buyers follow advice given by bicriterion approximate demand oracles.

4 Welfare Guarantees with Approximate Demand Oracles

In this section, we demonstrate that a slight modification of the $O((\log\log m)^3)$ approximation of [1] (which is truthful when buyers implement precise demand oracles) maintains its approximation guarantee when buyers follow advice recommended by a $(1/2, 1/2)$-approximate demand oracle. We begin with the main insight below, followed by a precise statement of our main result.

4.1 Fixed Price Auctions with Approximate Demand Oracles

A key component of the [1] (and related) auctions is the notion of a *Fixed Price Auction*. A fixed price auction simply sets a price $\mathbf{p}(j)$ on item j, visits the buyers one at a time, and offers the buyer the option to purchase any set S of remaining items for price $\mathbf{p}(S)$ (so it is a posted-price mechanism which sets the same prices for all bidders).

A key lemma used by these works establishes that there *exists* a fixed price auction generating good welfare (when bidders implement exact demand oracles) for any instance with submodular bidders (or even XOS bidders).[5] One can view the [14, 1] auctions as attempting to *learn* such a "good" fixed price auction. The key intuition behind our extension is that good fixed price auctions still exist when bidders only implement approximate demand oracles. This is captured formally by Lemma 15 below, which first requires the notion of *supporting prices*.

▶ **Definition 13.** $\mathbf{q}$ *are* supporting prices *for* $v_1(\cdot), \dots, v_n(\cdot)$ *and allocation* $S_1, \dots, S_n$ *if:*

- *For all* i, T, $v_i(T) \geq \mathbf{q}(S_i \cap T)$.
- *For all* i, $v_i(S_i) = \mathbf{q}(S_i)$.

▶ **Fact 14.** *When all* $v_i(\cdot)$ *are XOS, supporting prices exist for any allocation* $S_1, \dots, S_n$.[6]

Much prior work leverages the fact that with precise demand queries, the fixed-price auction with prices $\mathbf{q}/2$ achieves half the optimal welfare. The intuition for our main result is that this key lemma extends to (c, d)-approximate demand queries by losing an additional $\min\{c, d\}$ factor. In the statement below, we will slightly abuse notation and say that a bidder "follows advice given by a (c, d)-approximate demand oracle" if they follow advice given by an algorithm which on input $v(\cdot), S$ computs a (c, d)-approximate demand query T, then advises $\arg\max\{v(S) - \mathbf{p}(S), v(T) - \mathbf{p}(T)\}$.

[5] This lemma appears at least as early as [16].

[6] Recall that submodular functions are XOS, and a function is XOS if it can be written as the maximum of additive functions. Supporting prices for items in S_i are defined by simply taking the additive function which defines $v_i(S_i)$.

▶ **Lemma 15.** *Let* $\mathbf{q}$ *be supporting prices for* $v_1(\cdot), \ldots, v_n(\cdot)$ *and* $S_1, \ldots, S_n$*. Then the fixed-price auction with prices* $d\mathbf{q}/2$ *guarantees welfare at least* $\min\{c, d\} \cdot \sum_i v_i(S_i)/2$ *when all bidders follow advice given by a* (c, d)*-approximate demand oracle.*

Proof. Let $S := \cup_i S_i$. Let also T_i denote the set purchased by bidder i (following advice given by a (c, d)-approximate demand oracle), and denote by $\textsc{Sold} = \bigcup_{i \in N} T_i$. Define $A_i = S_i \setminus \textsc{Sold}$. Because items in A_i are never allocated when bidder i is chosen to act (meaning that bidder i could choose to purchase the set A_i), and bidder i will choose a set guaranteeing at least as much utility as a (c, d)-approximate demand oracle, we have:

$$v_i(T_i) - d\mathbf{q}(T_i)/2 \geq c\left(v_i(A_i) - \frac{d\mathbf{q}(A_i)/2}{d}\right) = c\left(v_i(A_i) - \frac{1}{2}\mathbf{q}(A_i)\right).$$

The welfare achieved ($\sum_{i \in N} v_i(T_i)$) is exactly the sum of the utilities of each bidder ($v_i(T_i) - d\mathbf{q}(T_i)/2$) and the total revenue of the auction ($\sum_{i \in N} d\mathbf{q}(T_i)/2 = d\mathbf{q}(\textsc{Sold})/2$). By the definition of supporting prices (and the fact that $A_i \subseteq S_i$), we know that $v_i(A_i) \geq \mathbf{q}(A_i)$. Thus:

$$\begin{aligned}
\frac{d}{2}\mathbf{q}(\textsc{Sold}) + \sum_{i=1}^{n} v_i(T_i) - \frac{d}{2}\mathbf{q}(T_i) &\geq \frac{d}{2}\mathbf{q}(\textsc{Sold}) + c\left(\sum_{i=1}^{n} v_i(A_i) - \frac{1}{2}\mathbf{q}(A_i)\right) \\
&\geq \frac{d}{2}\mathbf{q}(\textsc{Sold}) + \frac{c}{2}\sum_{i=1}^{n}\mathbf{q}(A_i) \\
&\geq \frac{\min\{c, d\}}{2}\big(\mathbf{q}(\textsc{Sold}) + \mathbf{q}(S \setminus \textsc{Sold})\big) \\
&\geq \frac{\min\{c, d\}}{2}\mathbf{q}(S) = \frac{\min\{c, d\}}{2}\sum_i v_i(S_i).
\end{aligned}$$

The first inequality follows as each bidder follows advice of a (c, d)-approximate demand oracle. The second follows as $v_i(A_i) \geq \mathbf{q}(A_i)$ for all i (by definition of supporting prices). The third follows as $\cup_i A_i = S \setminus \textsc{Sold}$. The final inequality follows by basic arithmetic, and the final equality follows as $\mathbf{q}(S) = \sum_i v_i(S_i)$ by definition of supporting prices. ◀

Lemma 15 captures the main intuition for why existing posted-price guarantees can be extended to accommodate bicriterion approximate demand queries. Of course, the [1] mechanism is not just a single posted-price mechanism, and Lemma 15 is just one technical lemma used along the way (to be more precise, a generalization of Lemma 15 is used along the way, but the overly technical statement hides the intuition). But an outline similar to the proof of Lemma 15 establishes the more general claim. Section 4.2 formally states our main result, and all details of the proof aside from the above intuition can be found in Appendix B.

4.2 Formal Statement of Main Result

▶ **Theorem 16.** *Let* $\mathcal{V}$ *be a subclass of XOS valuations and let* D *be a poly-time* (c, d)*-approximate demand oracle for valuation class* $\mathcal{V}$*. Then there exists a poly-time mechanism for welfare maximization when all valuations are in* $\mathcal{V}$ *with approximation guarantee* $O\left(\max\left\{\frac{1}{c}, \frac{1}{d}\right\} \cdot (\log\log m)^3\right)$ *in implementation in advised strategies with polynomial time computable advice.*

Theorem 1 now follows from Theorem 16 as submodular valuations are a subclass of XOS which admits poly-time $(1/2, 1/2)$-approximate demand oracles. The poly-time mechanism witnessing Theorem 16 is a slight modification of [1]. The high-level approach of their

mechanism is the following: because $\mathcal{V}$ is XOS, Lemma 15 establishes that there *exists* a fixed-price mechanism which achieves an $1/O(\min\{c,d\}) = O(\max\{1/c, 1/d\})$ approximation in implementation in advised strategies. Of course, implementing this fixed-price auction requires complete knowledge of $v_1(\cdot), \ldots, v_n(\cdot)$, which the seller lacks. The mechanism of [1] essentially tries to iteratively guess a better and better set of fixed prices, and then pick one uniformly at random.

Intuitively, our adapted [1] mechanism works with (c,d)-approximated demand oracles for the same reason that Lemma 15 works with approximate demand oracles. Formally establishing this requires a bit of work, but much of the analysis of [1] treats the $(c,d) = (1,1)$ case of Lemma 15 as a black box, and therefore we can leverage most of their analysis as a black box as well. The generalized Lemma 15 (Lemma 20) provides all the properties of demand queries which their proof requires (and all the properties of approximate demand queries which our adaptation requires). A complete proof appears in Appendix B.

5 Approximate Demand Queries beyond Submodular

In this section, we explore approximate demand queries beyond submodular valuation functions. As the approximation guarantees of [1] hold for XOS valuations with precise demand queries, a poly-time $(\Omega(1), \Omega(1))$-approximate demand query would immediately extend their guarantees to XOS valuations under implementation in advised strategies. Interestingly, this very fact establishes that for all $\varepsilon > 0$, no poly-time $(\Omega(m^{-1/2+\varepsilon}), \Omega(m^{-1/2+\varepsilon}))$-approximate demand oracle exists for XOS valuations using subexponentially-many value queries.

▶ **Proposition 17.** *For all $\varepsilon > 0$, there is no $(\Omega(m^{-1/2+\varepsilon}, \Omega(m^{-1/2+\varepsilon}))$-approximate demand oracle for XOS valuations using $\mathrm{poly}(m)$ value queries.*

Proof. Assume for contradiction that the proposition were false. Then by Theorem 16, there exists an algorithm using $\mathrm{poly}(n,m)$ value queries that approximates the optimal welfare within $O\left(m^{1/2-\varepsilon} \cdot (\log\log m)^3\right) \in O(m^{1/2-\varepsilon/2})$ for XOS valuations. However, Theorem 6.1 of [17] proves that no such algorithm exists. ◀

To complete the picture, we also design poly-time $(\Omega(1/\sqrt{m}), \Omega(1/\sqrt{m}))$-approximate demand oracles for subadditive valuations (defined immediately below, based on the $\Omega(1/\sqrt{m})$-approximation of [17]), which is the best possible using subexponentially-many value queries.

Algorithm 2 SingleOrBundle$(v, \mathbf{p}, M)$.

$j \leftarrow \arg\max_{j \in M} v(j) - \mathbf{p}(j)$
$M^* \leftarrow \{j \in M : v(j) - (1+\sqrt{m})\mathbf{p}(j) > 0\}$
if $v(M^*) - \mathbf{p}(M^*) > v(j) - \mathbf{p}(j)$ **then**
  return M^*
else
  return $\{j\}$

▶ **Proposition 18.** SingleOrBundle$(v, \mathbf{p}, M)$ *is a $(\frac{1}{\sqrt{m}}, \frac{1}{1+\sqrt{m}})$-approximate demand oracle for subadditive valuation functions.*

Proof. Let S be the set that maximizes utility $(v(S) - (1+\sqrt{m}) \cdot \mathbf{p}(S))$. If $S = \emptyset$, then the benchmark is 0, and SingleOrBundle achieves non-negative utility. It remains to consider the case $v(S) - \mathbf{p}(S) > 0$.

In this case, let T be the set returned by SingleOrBundle$(v, \mathbf{p}, M)$. Call an item j *special* if:

$$v(j) - \mathbf{p}(j) \geq \frac{1}{\sqrt{m}}(v(S) - (1+\sqrt{m}) \cdot \mathbf{p}(S)).$$

Observe that if any item j is special, then we conclude:

$$v(T) - \mathbf{p}(T) \geq v(j) - \mathbf{p}(j) \geq \frac{1}{\sqrt{m}}(v(S) - (1+\sqrt{m}) \cdot \mathbf{p}(S)),$$

which is a $(\frac{1}{\sqrt{m}}, \frac{1}{1+\sqrt{m}})$-approximate demand oracle. If no item is special, then $\forall j \in M^*$, $v(j) - \mathbf{p}(j) < \frac{1}{\sqrt{m}}(v(S) - (1+\sqrt{m})\mathbf{p}(S))$. Summing this for all $j \in M^*$ yields:

$$\left(\sum_{j \in M^*} v(j)\right) - \mathbf{p}(M^*) < \frac{|M^*|}{\sqrt{m}}(v(S) - (1+\sqrt{m}) \cdot \mathbf{p}(S)) \leq \sqrt{m} \cdot v(S) \leq \sqrt{m} \cdot v(M^*). \tag{1}$$

The final inequality follows as S cannot contain items for which $v(j) < (1+\sqrt{m})p(j)$, as $v(\cdot)$ is subadditive. We can then conclude that:

$$\begin{aligned} v(M^*) - \mathbf{p}(M^*) &> \frac{1}{\sqrt{m}}\left(\left(\sum_{j \in M^*} v(j)\right) - \mathbf{p}(M^*)\right) - \mathbf{p}(M^*) \\ &= \frac{1}{\sqrt{m}}\left(\sum_{j \in M^*} v(j) - (1+\sqrt{m})\mathbf{p}(j)\right) \\ &\geq \frac{1}{\sqrt{m}}\left(\sum_{j \in S} v(j) - (1+\sqrt{m})\mathbf{p}(j)\right) \\ &\geq \frac{1}{\sqrt{m}}\left(v(S) - (1+\sqrt{m})\mathbf{p}(S)\right). \end{aligned}$$

The first inequality follows directly from (1). The second follows as $v(j) > (1+\sqrt{m})\mathbf{p}(j)$ for all $j \in M^*$, and $S \subseteq M^*$ (because $v(\cdot)$ is subadditive, and S is the utility-maximizing set at prices $(1+\sqrt{m})\mathbf{p}$). The third follows from subadditivity of $v(\cdot)$. We conclude that when there are no special items, the proposition is satisfied as well, completing the proof. ◀

References

1. Sepehr Assadi and Sahil Singla. Exponentially Improved Truthful Combinatorial Auctions with Submodular Bidders. In *Proceedings of the Sixtieth Annual IEEE Foundations of Computer Science (FOCS)*, 2019.
2. Moshe Babaioff, Ron Lavi, and Elan Pavlov. Single-value combinatorial auctions and algorithmic implementation in undominated strategies. *J. ACM*, 56(1):4:1–4:32, 2009. doi:10.1145/1462153.1462157.
3. Mark Braverman, Jieming Mao, and S. Matthew Weinberg. On Simultaneous Two-player Combinatorial Auctions. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 2256–2273, 2018. doi:10.1137/1.9781611975031.146.
4. Patrick Briest, Piotr Krysta, and Berthold Vöcking. Approximation techniques for utilitarian mechanism design. In *the 37th Annual ACM Symposium on Theory of Computing (STOC)*, 2005.

5 Niv Buchbinder, Moran Feldman, Joseph Naor, and Roy Schwartz. A Tight Linear Time (1/2)-Approximation for Unconstrained Submodular Maximization. *SIAM J. Comput.*, 44(5):1384–1402, 2015. doi:10.1137/130929205.

6 Dave Buchfuhrer. A Theory of Robust Hardness for Truthful Mechanism Design. Manuscript, 2011. URL: http://users.cms.caltech.edu/~dave/papers/oracles.pdf.

7 David Buchfuhrer, Shaddin Dughmi, Hu Fu, Robert Kleinberg, Elchanan Mossel, Christos H. Papadimitriou, Michael Schapira, Yaron Singer, and Christopher Umans. Inapproximability for VCG-Based Combinatorial Auctions. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2010.

8 David Buchfuhrer, Michael Schapira, and Yaron Singer. Computation and incentives in combinatorial public projects. In *Proceedings 11th ACM Conference on Electronic Commerce (EC-2010), Cambridge, Massachusetts, USA, June 7-11, 2010*, pages 33–42, 2010. doi:10.1145/1807342.1807348.

9 Edward H. Clarke. Multipart Pricing of Public Goods. *Public Choice*, 11(1):17–33, 1971.

10 Amit Daniely, Michael Schapira, and Gal 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*, pages 401–408, 2015. doi:10.1145/2746539.2746597.

11 Nikhil R. Devanur, Jamie Morgenstern, Vasilis Syrgkanis, and S. Matthew Weinberg. Simple Auctions with Simple Strategies. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation, EC '15, Portland, OR, USA, June 15-19, 2015*, pages 305–322, 2015. doi:10.1145/2764468.2764484.

12 Shahar Dobzinski. Two Randomized Mechanisms for Combinatorial Auctions. In *Proceedings of the 10th International Workshop on Approximation and the 11th International Workshop on Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 89–103, 2007.

13 Shahar Dobzinski. An Impossibility Result for Truthful Combinatorial Auctions with Submodular Valuations. In *Proceedings of the 43rd ACM Symposium on Theory of Computing (STOC)*, 2011.

14 Shahar 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, pages 940–948, New York, NY, USA, 2016. ACM. doi:10.1145/2897518.2897569.

15 Shahar Dobzinski. Computational Efficiency Requires Simple Taxation. In *FOCS*, 2016.

16 Shahar Dobzinski, Noam Nisan, and Michael Schapira. Truthful randomized mechanisms for combinatorial auctions. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 644–652. ACM, 2006.

17 Shahar Dobzinski, Noam Nisan, and Michael Schapira. Approximation Algorithms for Combinatorial Auctions with Complement-Free Bidders. *Math. Oper. Res.*, 35(1):1–13, 2010. doi:10.1287/moor.1090.0436.

18 Shahar Dobzinski, Noam Nisan, and Michael Schapira. Truthful randomized mechanisms for combinatorial auctions. *J. Comput. Syst. Sci.*, 78(1):15–25, 2012. doi:10.1016/j.jcss.2011.02.010.

19 Shahar Dobzinski and Jan Vondrák. From query complexity to computational complexity. In *Proceedings of the 44th Symposium on Theory of Computing (STOC)*, 2012.

20 Shahar Dobzinski and Jan Vondrak. The Computational Complexity of Truthfulness in Combinatorial Auctions. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, 2012.

21 Shahar Dobzinski and Jan Vondrák. Impossibility Results for Truthful Combinatorial Auctions with Submodular Valuations. *J. ACM*, 63(1):5:1–5:19, 2016. doi:10.1145/2786754.

22 Paul Duetting, Michal Feldman, Thomas Kesselheim, and Brendan Lucier. Prophet Inequalities Made Easy: Stochastic Optimization by Pricing Non-Stochastic Inputs. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 540–551, 2017. doi:10.1109/FOCS.2017.56.

23 Tomer Ezra, Michal Feldman, Eric Neyman, Inbal Talgam-Cohen, and S. Matthew Weinberg. Settling the Communication Complexity of Combinatorial Auctions with Two Subadditive Buyers. In *the 60th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2019.

24 Uriel Feige. On Maximizing Welfare When Utility Functions Are Subadditive. *SIAM J. Comput.*, 39(1):122–142, 2009. doi:10.1137/070680977.

25 Uriel Feige and Shlomo Jozeph. Demand Queries with Preprocessing. In *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I*, pages 477–488, 2014. doi:10.1007/978-3-662-43948-7_40.

26 Michal Feldman, Nick Gravin, and Brendan Lucier. Combinatorial Auctions via Posted Prices. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '15, pages 123–135, Philadelphia, PA, USA, 2015. Society for Industrial and Applied Mathematics. URL: http://dl.acm.org/citation.cfm?id=2722129.2722139.

27 Moran Feldman. Guess Free Maximization of Submodular and Linear Sums. In *Algorithms and Data Structures - 16th International Symposium, WADS 2019, Edmonton, AB, Canada, August 5-7, 2019, Proceedings*, pages 380–394, 2019. doi:10.1007/978-3-030-24766-9_28.

28 Theodore Groves. Incentives in Teams. *Econometrica*, 41(4):617–631, 1973.

29 Chris Harshaw, Moran Feldman, Justin Ward, and Amin Karbasi. Submodular Maximization beyond Non-negativity: Guarantees, Fast Algorithms, and Applications. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 2634–2643, 2019. URL: http://proceedings.mlr.press/v97/harshaw19a.html.

30 Stavros G. Kolliopoulos and Clifford Stein. *Approximating Disjoint-Path Problems Using Greedy Algorithms and Packing Integer Programs*, pages 153–168. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998. doi:10.1007/3-540-69346-7_12.

31 Piotr Krysta and Berthold Vöcking. Online Mechanism Design (Randomized Rounding on the Fly). In *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part II*, pages 636–647, 2012. doi:10.1007/978-3-642-31585-5_56.

32 Ron Lavi and Chaitanya Swamy. Truthful and Near-Optimal Mechanism Design via Linear Programming. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2005.

33 Benny Lehmann, Daniel Lehmann, and Noam Nisan. Combinatorial Auctions with Decreasing Marginal Utilities. In *the 3rd Annual ACM Conference on Electronic Commerce (EC)*, 2001.

34 Vahab S. Mirrokni, Michael Schapira, and Jan 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*, pages 70–77, 2008. doi:10.1145/1386790.1386805.

35 George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14(1):265–294, 1978.

36 Christos H. Papadimitriou, Michael Schapira, and Yaron Singer. On the Hardness of Being Truthful. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2008.

37 Marek Pycia and Peter Troyan. Obvious Dominance and Random Priority. In *Proceedings of the 2019 ACM Conference on Economics and Computation, EC 2019, Phoenix, AZ, USA, June 24-28, 2019.*, page 1, 2019. doi:10.1145/3328526.3329613.

38 Prabhakar Raghavan. Probabilistic Construction of Deterministic Algorithms: Approximating Packing Integer Programs. *J. Comput. Syst. Sci.*, 37(2):130–143, October 1988. doi:10.1016/0022-0000(88)90003-7.

39 Michael Schapira and Yaron Singer. Inapproximability of Combinatorial Public Projects. In *Internet and Network Economics, 4th International Workshop, WINE 2008, Shanghai, China, December 17-20, 2008. Proceedings*, pages 351–361, 2008. doi:10.1007/978-3-540-92185-1_41.

40 Maxim Sviridenko, Jan Vondrák, and Justin Ward. Optimal approximation for submodular and supermodular optimization with bounded curvature. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 1134–1148, 2015. doi:10.1137/1.9781611973730.76.

41 William Vickrey. Counterspeculations, Auctions, and Competitive Sealed Tenders. *Journal of Finance*, 16(1):8–37, 1961.

42 Jan 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*, pages 67–74, 2008. doi:10.1145/1374376.1374389.

A Brief Discussion of Definitions

The following example will motivate our decision to think of advice as *improving a given strategy* as opposed to *outright proposing a replacement strategy*.

Consider, for example, a single-bidder mechanism where the bidder faces one of k posted-price vectors $\mathbf{p}_1, \dots, \mathbf{p}_k$ chosen uniformly at random, and is asked to submit their desired sets $S_1, \dots, S_k$ before knowing which price is "real." Then the strategy which submits $S_i := \arg\max\{v(S) - \mathbf{p}_i(S)\}$ is dominant. In this case, advice could indeed simply propose this strategy to replace whatever else the bidder might try.

Things get more interesting, however, if the designer cannot recommend a dominant strategy. Consider instead a recommended strategy $T_1, \dots, T_k$ where $T_i \notin \arg\max\{v(S) - \mathbf{p}_i(S)\} \cup \arg\min\{v(S) - \mathbf{p}_i(S)\}$ for any i (call this strategy $\vec{T}$). If the designer shares the sets $T_1, \dots, T_k$ with the buyer, a reasonable buyer should certainly submit sets S_i satisfying $v(S_i) - \mathbf{p}_i(S_i) \geq v(T_i) - \mathbf{p}_i(T_i)$ for all i (because they could just swap any set violating this for T_i and strictly improve their utility). Consider then the strategy which sets $S_j \in \arg\max\{v(S) - \mathbf{p}_j(S)\}$ (for a single $j \in [k]$) and $S_i \in \arg\min\{v(S) - \mathbf{p}_i(S)\}$ for all $i \neq j$ (picks the optimal set for $\mathbf{p}_j$, and worst possible sets for all other $\mathbf{p}_i$, call this strategy $\vec{S}$). We don't want to say that a bidder originally planning to use $\vec{S}$ should instead use $\vec{T}$ (indeed, $\vec{T}$ does not dominate $\vec{S}$, and it's not a priori clear which strategy yields higher expected utility). But we do want to say that a bidder originally planning to use $\vec{S}$ should stick with S_j, and update S_i to T_i for all $i \neq j$. But in order to recommend such a strategy without knowing j in advance, $\vec{T}$ would need to be the dominant strategy itself. So in order for the solution concept to meaningfully apply to posted-price mechanisms without advising the dominant strategy itself, advice should really take the form of improving a tentative strategy rather than outright recommending a replacement. Lemma 29 below provides a representative example of how this solution concept can be harnessed for existing state-of-the-art mechanisms.

B Proof of Theorem 16

The full definition of "implementation in advised strategies" is very powerful, but a bit awkward to carry around. Throughout this appendix, we use the following definition of (c, d)-competitive sets, which simply says that a set of items will give the bidder utility at least as high as a (c, d)-approximate demand oracle.

▶ **Definition 19.** *A set S is a* (c, d)-competitive subset of M for v_i with prices $\mathbf{p}$ *if*

$$v_i(S) - \mathbf{p}(S) \geq c \cdot \max_{T \subseteq M} \{v_i(T) - \mathbf{p}(T)/d\}.$$

We say a bidder i picks (c, d) competitive sets *in a fixed price auction if, when the fixed price auction visits i, they pick a set which is a (c, d)-competitive subset of the collection of remaining items.*

The full proof of theorem 16 is fairly involved. We start off this section by providing the more technical version of lemma 15 in section B.1, which captures most of the properties of fixed price auctions with approximate demand queries which we need. Next in section B.2, we describe the "core algorithm" PriceLearningMechanism of [1]. Then in section B.3, using fairly elementary properties of fixed-price auctions, we prove the correctness of the PriceLearningMechanism, as long as 1) bidders pick (c, d)-competitive sets in every fixed price auction they participate in, and 2) we make the simplifying assumption 21. In section B.4, we remove the simplifying assumption, and prove that there exists poly time computable advice such that, when bidders are following the advice, they always pick (c, d)-competitive sets.

B.1 Generalization of Lemma 15

In this subsection we state and prove the generalization of Lemma 15, which will be used in the analysis of the PriceLearningMechanism.

The first term in the maximum below (and the "moreover" part of the lemma) relates the achieved welfare with the value of the unsold items, and will be used to handle "learning" phases in the mechanism. The second term of the maximum shows that once we have learned the prices well, we definitely get good welfare.

▶ **Lemma 20.** *Suppose $\{T_i\}_{i \in N} \leftarrow \text{FixedPriceAuction}(M, N, d\mathbf{p})$, where each bidder i picks a subset of the remaining items which is (c, d)-competitive set for v_i with prices $\mathbf{p}$. Let $\{O_i\}_{i \in N}$ be any allocation with supporting prices $\mathbf{q}$. Let S_i be the set of items j where $\delta\mathbf{q}(j) \leq \mathbf{p}(j) \leq \frac{1}{2}\mathbf{q}(j)$ and $j \in O_i$. Denote $S = \bigcup_{i \in N} S_i$ and $\textsc{Sold} = \bigcup_i T_i$. Then*

$$\sum_i v_i(T_i) \geq \max \begin{cases} \frac{c}{2} \cdot \mathbf{q}(S \setminus \textsc{Sold}), \\ \min\left(\frac{c}{2}, \quad \delta d\right) \cdot \mathbf{q}(S). \end{cases}$$

Moreover, suppose k is the last bidder in N. We also have $v_i(T_k) \geq \frac{c}{2}\mathbf{q}(S_k \setminus \textsc{Sold}_{<k})$, where $\textsc{Sold}_{<k} = \bigcup_{i<k} T_i$.

Proof. Let $A_i = S_i \setminus \textsc{Sold}$. Because items in A_i are never allocated when bidder i is chosen to act (and because each bidder picks a (c, d)-competitive set with prices $\mathbf{p}$), the utility of each bidder i satisfies

$$v_i(T_i) - d \cdot \mathbf{p}(T_i) \geq c \cdot (v_i(A_i) - \mathbf{p}(A_i)).$$

As $A_i \subseteq S_i$, we know $\delta \mathbf{q}(A_i) \leq \mathbf{p}(A_i) \leq \frac{1}{2}\mathbf{q}(A_i)$, and by the definition of supporting prices, we know that $\mathbf{q}(A_i) \leq v_i(A_i)$ Thus, $\{T_i\}_{i \in N}$ achieves welfare

$$\begin{aligned}\sum_{i \in N} v_i(T_i) &= \sum_{i \in N} \big(v_i(T_i) - d \cdot \mathbf{p}(T_i)\big) + d \sum_{i \in N} \mathbf{p}(T_i) \\ &\geq c \sum_{i \in N} \big(v_i(A_i) - \mathbf{p}(A_i)\big) + d \cdot \mathbf{p}(\textsc{Sold}) \\ &\geq \sum_i c(\mathbf{q}(A_i) - \frac{1}{2}\mathbf{q}(A_i)) + d\delta \cdot \mathbf{q}(\textsc{Sold}). \qquad (*)\end{aligned}$$

Observe that $S \setminus \textsc{Sold} = \bigcup_{i \in N} A_i$. Thus, ignoring the term $d\delta\mathbf{q}(\textsc{Sold})$ from (*), we can conclude the auction gets welfare at least $\frac{c}{2}\mathbf{q}(S \setminus \textsc{Sold})$. Moreover, (*) tells us we get welfare at least

$$\min\left(\frac{c}{2}, \delta d\right) \cdot \big(\mathbf{q}(S \setminus \textsc{Sold}) + \mathbf{q}(\textsc{Sold})\big) \geq \min\left(\frac{c}{2}, \delta d\right) \cdot \mathbf{q}(S),$$

from which we can conclude main statement of the lemma.

For the "moreover" component, simply observe that when bidder k was picked by the mechanism, the items in $A' := S_k \setminus \textsc{Sold}_{<k}$ were still available, and that $S_k \subseteq O_k$, so

$$v_i(T_k) \geq v_i(T_k) - d\mathbf{p}(T_k) \geq c(v_i(A') - \mathbf{p}(A')) \geq c(\mathbf{q}(A') - \mathbf{q}(A')/2) = \frac{c}{2}\mathbf{q}(A'). \qquad \blacktriangleleft$$

B.2 The Mechanism

For the reader's convenience, we first briefly describe the mechanism in [1] and quote the mechanism verbatim (the only change we need to make is that every price used by the mechanism is "discounted" by an extra factor of d, plus some slight simplifications in the "removing extra assumptions" step). Then we present a slightly condensed version of the analysis.

High-level overview. Posted price mechanisms for combinatorial auctions typically use the following high-level strategy: attempt to (approximately) *learn* the supporting prices (definition 13) of an optimal allocation, then sell the items at those prices. The key innovation of [1] is to "explore" prices for each item individually using a *price tree* in which each successive layer of the tree corresponds to a finer "granularity" of prices. Initially, each item is set at a price corresponding to the root of the tree, and in each successive round of the mechanism, the price of each item moves one layer down in the tree to a "more precise" price which corresponds to some child node of the old price.

The mechanism of [1] runs several fixed-price auctions for each round (i.e. each layer of the price tree). In each of these successive auctions, each item is priced higher and higher in a way corresponding to the children of the "old" price node of the item. The price in the next round of the mechanism is then the highest price in the next layer where the item was still sold. The idea here is that, in the next layer of prices, we need to make the prices as high as possible such that the items will still sell. Intuitively, this serves to refine our estimate for the supporting prices as we move a layer down in the tree.

In fact, the story is more subtle than this. The mechanism may not actually achieve a better approximation to the prices in each layer, but [1] prove that if you *do not* get a better estimate for the prices, then you can *already* get a good approximation to the optimal welfare at the current prices. These two cases exactly correspond to the "learnable" or "allocatable" cases in lemma 22 below. For this reason, for every layer of the price tree, the mechanism

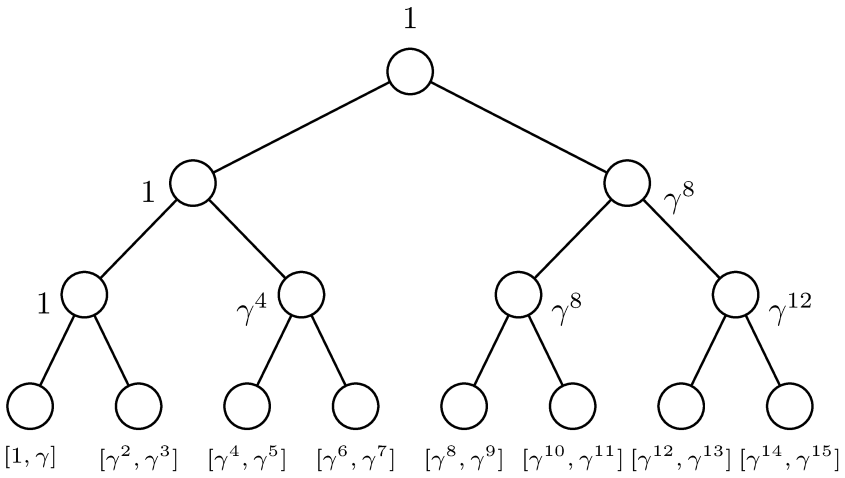

Figure 1 [1, Figure 2] An illustration of a price tree T^e with $\alpha = 2$, $\beta = 3$, and $\psi_{min} = 1$.

has some chance (proportional to the number of layers) of stopping early and allocating the items according to some fixed price auction in that layer. Thus, regardless of whether we always learn prices or if we hit the "allocatable" in some step, we will use a good auction with some probability.

Simplifying Assumption. It's useful for posted price mechanisms to know ahead of time the *range* of possible supporting prices of an optimal allocation. This assumption can be removed in a fairly "modular" way, as done in [1] (though we make some modifications in order to more easily fit our solution concept).

Let $\mathbf{q}$ be the supporting prices of an optimal allocation. Formally, our simplifying assumption is the following:

▶ **Assumption 21.** *There are known numbers ψ_{min}, ψ_{max} such that the supporting prices of any item in $\mathbf{q}$ are either 0 or in $[\psi_{min}, \psi_{max}]$, and ψ_{max}/ψ_{min} is polynomial in m.*

The price tree and mechanism parameters. We now formally describe the price tree in terms of three parameters:

- $\alpha = \Theta(1)$ is the branching factor of the tree (and the number of auctions in each iteration of the mechanism).
- $\beta = \Theta(\log\log(\psi_{max}/\psi_{min})) = \Theta(\log\log m)$ is the number of layers in the price tree (and the number of iterations of the mechanism).
- $\gamma = \Theta(\alpha\beta) = \Theta(\log\log m)$ is the "accuracy factor" of the prices.

We would like the leaves of the price tree correspond to "price buckets"

$$P = \{[\psi_{min}\gamma^i, \psi_{min}\gamma^{i+1}] \mid i = 0, 1, \ldots, k\}$$

for some k large enough that all prices in $[\psi_{min}, \psi_{max}]$ are considered. Informally, we take "learning a price $\mathbf{q}$ correctly" to mean that we find the bucket in P to which $\mathbf{q}$ belongs. We will assign "actual" prices in $\mathbf{p}$ according to the smallest price $\psi_{min}\gamma^i$ in the corresponding bucket. Our goal is that, if we "learn the price of $\mathbf{q}$ correctly", then the price used in $\mathbf{p}$ is within a γ factor of the true price in $\mathbf{q}$.

However, for technical reasons, we need a *gap* of at least γ between the prices of consecutive nodes in each layer of the tree (not just the leaves), so that prices will be guided to the closest leaf node *below* the price in $\mathbf{q}$ (this is desirable because lemma 20 requires prices in $\mathbf{p}$ to be less than in $\mathbf{q}$). To ensure this, the mechanism creates a price gap of factor γ between nodes by splitting P into

$$P^o = \{[\psi_{min}\gamma^{2k-1}, \psi_{min}\gamma^{2k}] \mid k = 1, \dots, \alpha^\beta\}$$
$$P^e = \{[\psi_{min}\gamma^{2k}, \psi_{min}\gamma^{2k+1}] \mid k = 0, \dots, \alpha^\beta - 1\}$$

Trees T^o and T^e are constructed with leaf nodes P^o and P^e respectively. We use T^* to denote either of T^o or T^e, and P^* will denote the corresponding P^o or P^e.

The price tree T^* is an α-branching tree with depth β (i.e. with $\beta + 1$ layers). The leaf nodes correspond, in left-to-right (depth-first search) order, to the price buckets in P^* (in increasing order). Furthermore, any non-leaf node x corresponding to a single price, which is the minimum value in any bucket of any leaf node which is a descendant of x. Thus, the prices corresponding to consecutive level-i nodes differ by a factor of $\gamma^{2\alpha^{\beta+1-i}}$.

Let $\mathbf{p}$ be a "level-i price vector", i.e. a vector in which the price of each item is a price which corresponds to some level-i node. We let $\text{next}_j^{(i)}(\mathbf{p}) = \mathbf{p}'$ denote the price vector constructed as follows: for each item $\ell \in M$, let x be the level-i node whose corresponding price is $\mathbf{p}(\ell)$. Then set $\mathbf{p}'(\ell)$ to the price corresponding to the jth child node of x. Thus, a precise formula is given by $\mathbf{p}'(\ell) = \gamma^{2\alpha^{\beta-i}(j-1)}\mathbf{p}(\ell)$. In words, $\text{next}_j^{(i)}(\mathbf{p})$ sets the price of each item ℓ to be the jth largest "refined price" below the current price of item ℓ.

The mechanism. We start by randomly picking a price tree T^o or T^e. The mechanism then proceeds in β iterations (though it may terminate early) in which a price vector $\mathbf{p}^{(i)}$ is constructed in each iteration i. Initially, $\mathbf{p}^{(1)}$ is the (unique) level-1 price vector of T^*. In each iteration, a $\frac{1}{10\beta}$ fraction of the bidders are selected uniformly at random, and the α different price vectors $\mathbf{p}_j^{(i)} = \text{next}_j^{(i)}(\mathbf{p}^{(i)})$ for $j = 1, \dots, \alpha$ are considered. The mechanism runs fixed price auction with the current set of bidders on prices $d\mathbf{p}_j^{(i)}/2$ for $j = 1, \dots, \alpha$. The new (level-$(i+1)$) vector $\mathbf{p}^{(i+1)}$ is then constructed as follows: for each item ℓ, $\mathbf{p}^{(i+1)}(\ell) = \mathbf{p}_j^{(i)}(\ell)$, where j is the highest index such that item ℓ sold in the auction with prices $d\mathbf{p}_j^{(i)}/2$. (or $\mathbf{p}^{(i)}(\ell)$ if no such j exists). In words, the new price of ℓ is the price of ℓ in the highest auction in iteration i for which item ℓ was sold. For each fixed price auction described in this paragraph, there is a $1/\Omega(\alpha\beta)$ chance that the mechanism will terminate early and return the allocation determined by the auction. This serves to strictly incentivizes bidder to pick good sets, but also serves an important purpose for achieving the desired approximation grantee, as discussed in the overview.

If the mechanism does not terminate early in iteration $1, \dots, \beta$, then the final step of the mechanism is to run a fixed price auction with all of the remaining bidders on prices $d\mathbf{p}^{(\beta)}/2$. (The hope is that, for a large fraction (weighted by $\mathbf{q}$) of the items, the price of the items is in the level-β bin which is closest to the price in $\mathbf{q}$, and thus we can apply lemma 20.)

The exact mechanism in [1] is quoted in Algorithm 3.

B.3 The Modified Analysis

Notation

We follow [1] and depart somewhat from conventional notation for the analysis of the mechanism. We let i denote an iteration of the mechanism, j denote an auction inside some iterations, b denote a bidder, and ℓ denote an item.

Algorithm 3 PriceLearningMechanism(N, M).

procedure PARTITION(N)
  Permute N uniformly at random.
  for $i = 1, 2, ..\beta$ **do**
    Remove $\frac{|N|}{10\beta}$ bidders uniformly at random from N; assign them to the set N_i.
  Put the remaining items in N into $N_{\beta+1}$.
procedure PRICEUPDATE$(A_1,\ldots,A_\alpha,\ \mathbf{p}_1,\ldots,\mathbf{p}_\alpha)$
  For each $\ell \in M$, let $\mathbf{p}'(\ell) = \mathbf{p}_j(\ell)$ for the
    highest value of j such that ℓ is allocated in A_j (or $\mathbf{p}_1(\ell)$ if no j exists)
  Return $\mathbf{p}'$

Let $(N_1, N_2, ...N_{\beta+1}) \leftarrow$ Partition(N)
Pick one of the modified trees T^o or T^e uniformly at random and denote it by T^*
Let $\mathbf{p}^{(1)}$ be the (unqiue) level-1 (root) price of T^*
for $i = 1, \ldots, \beta$ **do**
  For $j = 1, \ldots, \alpha$, let $\mathbf{p}_j^{(i)} = \text{next}_j^{(i)}(\mathbf{p}^{(i)})$
  For $j = 1, \ldots, \alpha$: run FixedPriceAuction(N_i, M, $d\mathbf{p}_j^{(i)}/2$) and let $A_j^{(i)}$ be the allocation
  With probability $(1/\beta)$, pick $j^* \in [\alpha]$ u.a.r. and return $A_{j^*}^{(i)}$ as the final allocation
  Otherwise, let $\mathbf{p}^{(i+1)} \leftarrow$PriceUpdate$(A_1^{(i)}, ..., A_\alpha^{(i)}, \mathbf{p}_1^{(i)}, ..., \mathbf{p}_\alpha^{(i)})$, and continue
Run FixedPriceAuction$(N_{\beta+1}, M, d\mathbf{p}^{(\beta+1)}/2)$ and return the allocation A^*

Let O be an optimal allocation with supporting prices $\mathbf{q}$ and OPT be the optimal welfare resulting from allocation O. Let $\mathbf{q}^*$ be $\mathbf{q}$ restricted to items whose prices are in some bucket of P^*. Let O^* be the collection of those items. Let $N_1, \ldots, N_{\beta+1}$ denote the groups of bidders from the Partition function. Given $(N_i)_i$ and T^* as picked by the mechanism, define price vectors $\mathbf{q}^{(i)}$ as $\mathbf{q}^*$, restricted to items which are allocated in O^* to bidders from $N_i, N_{i+1}, \ldots, N_{\beta+1}$ (intuitively, we restricted attention to items which could still go to the same bidder in A as in O, and give price 0 to items that can no longer be allocated to the right bidder in O). Call item ℓ correctly priced at iteration i if $\mathbf{q}^{(i)}(\ell)$ is in the bin corresponding to some leaf node which is a child of the node corresponding to $\mathbf{p}^{(i)}(\ell)$. Let $C^{(i)}$ denote all items priced correctly before iteration i begins. Note that $C^{(1)} = O^*$ and that an item can only be in $C^{(i)}$ if it is also in $C^{(j)}$ for $j = 1, \ldots, i-1$, so $C^{(1)} \supseteq C^{(2)} \supseteq \ldots \supseteq C^{(\beta+1)}$. We separate $C^{(i)}$ into $C_1^{(i)}, C_2^{(i)}, ...C_\alpha^{(i)}$, where $C_j^{(i)}$ is the subset of items in $C^{(i)}$ that are priced correctly in $\text{next}_j^{(i)}(\mathbf{p}^{(i)})$. For any set of bidders N' and items D, let $O^D_{N'}$ be the restriction of O^* to items in D and bidders in N'.

Assumptions

Throughout the claims in this section, we assume all bidders pick (c, d)-competitive sets in every fixed price auction they participate in, though we may not restate this assumption in every claim statement[7]. We also assume that the optimal allocation O has supporting prices $\mathbf{q}$.

[7] It is somewhat easier to prove that PriceLearningMechanism is implementable in advised strategies compared to GeneralizedMechanism below. However, we hold off and only demonstrate that Generalized-Mechanism is implementable in advised strategies, both for completeness, and in order to demonstrate that our solution concept "composes well" to be useful for complicated mechanisms.

The following lemma is the heart of the proof of the approximation ratio of mechanism 3.

▶ **Lemma 22** (Learnable-Or-Allocable Lemma from [1]). *Assume 21, and suppose all bidders pick (c,d)-competitive sets in every fixed price auction they participate in. For any iteration $i \in [\beta]$, conditioned on any outcome of first $i-1$ iterations and choice of T^*,*

1. *either* $\mathbb{E}\left[\mathbf{q}^{(i+1)}(C^{(i+1)})\right] \geq \mathbf{q}^{(i)}(C^{(i)}) - \frac{OPT}{3\beta}$*, where the expectation is over N_i;*
2. *or* $\mathbb{E}\left[\text{val}(A_{j^*})^{(i)}\right] \geq \frac{c}{O(\alpha\beta^2)}OPT$.

First we prove a series of claims before proving the Learnable-Or-Allocable lemma, following the same outline as [1]. For claims 23 and 24, we fix some $j \in [\alpha]$ and let $D = C_j^{(i)}$. Note that $\mathbf{q}^{(i)}(C_j^{(i)}) = \mathbf{q}^{(i)}(O^D_{N_{\geq i}})$, as $\mathbf{q}^{(i)}$ zeros out items allocated in O to bidders from $N_{<i}$.

▷ Claim 23. (5.3 from [1]) Deterministically, $\text{val}(A_j^{(i)}) \geq \frac{c}{2} \cdot \mathbf{q}^{(i)}(O^D_{N_i} \setminus A_j^{(i)})$.

Proof. Recall that $O^D_{N_i}$ is the restriction of O^* to items in $C_j^{(i)}$ and bidders in N_i. The definition of item ℓ being "priced correctly" means that $\mathbf{p}^{(i)}(\ell) \leq \mathbf{q}^{(i)}(\ell)$. Thus, for any $\ell \in O^D_{N_i}$ we get that $0 \cdot \mathbf{q}^{(i)}(\ell) \leq d\mathbf{p}^{(i)}(\ell)/2 \leq \mathbf{q}^{(i)}(\ell)/2$. Thus, the claim follows from lemma 20. ◁

▷ Claim 24. (5.4 from [1]) By randomness of choice of N_i from $N_{\geq i}$, $\mathbb{E}\left[\text{val}(A_j^{(i)})\right] \geq (\frac{c}{20\beta}) \cdot \mathbb{E}\left[\mathbf{q}^{(i)}(O^D_{N>i} \setminus A_j^{(i)})\right]$.

Proof. Consider picking a bidder $k \in N_{>i}$ uniformly at random and running an imaginary fixed price auction on $N_i \cup \{k\}$, where k is the last bidder chosen to act. Then by Lemma 20 (parameters in the lemma take values $N = N_i \cup \{k\}$, $\textsc{Sold}_{<k} = A_j^{(i)}$, $S_k = O_k^D$), the value bidder k gets from the imaginary fixed price auction satisfy $v_k(T_k) \geq \frac{c}{2} \cdot \mathbf{q}^{(i)}(O_k^D \setminus A_j^{(i)})$. We now take the expectation over the randomness on bidders $N_i \cup k$,

$$\begin{aligned}\mathop{\mathbb{E}}_{N_i, k\in N_{>i}}[v_k(T_k)] \geq \frac{c}{2} \cdot \mathop{\mathbb{E}}_{N_i, k\in N_{>i}}\left[\mathbf{q}^{(i)}(O_k^D \setminus A_j^{(i)})\right] &= \frac{c}{2} \cdot \mathop{\mathbb{E}}_{N_i}\left[\frac{1}{|N_{>i}|} \cdot \sum_{k\in N_{>i}} \mathbf{q}^{(i)}(O_k^D \setminus A_j^{(i)})\right] \\ &= \frac{1}{|N_{>i}|} \cdot \frac{c}{2} \cdot \mathbb{E}\left[\mathbf{q}^{(i)}(O^D_{N_{>i}} \setminus A_j^{(i)})\right].\end{aligned}$$

Observe that the expectation of $\text{val}(A_j^{(i)})$ is the same as the expected welfare of bidders in N_i in the imaginary fixed price auction. Since the bidders in N_i arrive before bidder k, their expected welfare in the imaginary fixed price auction is larger equal to that of bidder k. Thus by linearity of expectation

$$\mathbb{E}\left[\text{val}(A_j^{(i)})\right] \geq \frac{|N_i|}{|N_{>i}|} \cdot \frac{c}{2} \cdot \mathbb{E}\left[\mathbf{q}^{(i)}(O^D_{N_{>i}} \setminus A_j^{(i)})\right] \geq \left(\frac{c}{20\beta}\right) \cdot \mathbb{E}\left[\mathbf{q}^{(i)}(O^D_{N_{>i}} \setminus A_j^{(i)})\right]. \quad \triangleleft$$

▷ Claim 25. (5.2 from [1]) For any j, we have

$$\frac{22\beta}{c} \cdot \mathbb{E}\left[\text{val}(A_j^{(i)})\right] + \mathbb{E}\left[\mathbf{q}^{(i)}(C_j^{(i)} \cap A_j^{(i)})\right] \geq \mathbb{E}\left[\mathbf{q}^{(i)}(C_j^{(i)})\right].$$

Proof. By combining Claim 23 and 24, we have

$$\begin{aligned}\left(\frac{20\beta}{c} + \frac{2}{c}\right)\mathbb{E}\left[\text{val}(A_j^{(i)})\right] &\geq \mathbb{E}\left[\mathbf{q}^{(i)}(O^D_{N>i} \setminus A_j^{(i)})\right] + \mathbb{E}\left[\mathbf{q}^{(i)}(O^D_{N_i} \setminus A_j^{(i)})\right] \\ &= \mathbb{E}\left[\mathbf{q}^{(i)}(C_j^{(i)} \setminus A_j^{(i)})\right].\end{aligned}$$

Because $O^D_{N_{\geq i}}$ is exactly $C^{(i)}_j$. Thus, we get

$$\frac{20\beta+2}{c}\cdot\mathbb{E}\left[\text{val}(A^{(i)}_j)\right]+\mathbb{E}\left[\mathbf{q}^{(i)}(C^{(i)}_j\cap A^{(i)}_j)\right]\geq\mathbb{E}\left[\mathbf{q}^{(i)}(C^{(i)}_j)\right].$$ ◁

The previous claim can be thought of as a preliminary version of the entire learnable-or-allocatable lemma. In expectation, we get something comparable to the items which are correctly priced in auction j of round i (i.e. $\mathbf{q}^{(i)}(C^{(i)}_j)$). The contribution come from either the items which sold in the round they were "supposed to" (i.e. $C^{(i)}_j\cap A^{(i)}_j$) or the welfare of the current allocation (i.e. $A^{(i)}_j$) (with an extra $O(\beta)$ factor). The previous claims dealt with individual auctions within an iteration – next we handle iterations as a whole.

We still have to account for two things: items which sell in auctions where the prices are *too high* and the loss in welfare from the fact that bidders in N_i will no longer be allocated items in later rounds. The proofs in [1] hold as written – only the properties of the price tree and the structure of the auctions are used.

▷ Claim 26. (5.5 from [1])

$$\mathbf{q}^{(i)}(C^{(i+1)})\geq\sum_{j=1}^{\alpha}\mathbf{q}^{(i)}(A^{(i)}_j\cap C^{(i)}_j)-\frac{OPT}{10\beta}.$$

Proof. The key observation here is that the set of "overpriced" items represent a small fraction of the optimal revenue. Let U be the set of items that are allocated in FixedPriceAuction with price above their correct price in round i. The set of items that are allocated in the correct round but not priced correctly is exactly $\left(\bigcup_j A^{(i)}_j\cap C^{(i)}_j\right)\setminus C^{(i+1)}$. This must be a subset of U. Thus, $\mathbf{q}^{(i)}(C^{(i+1)})\geq\sum_{j=1}^{\alpha}\mathbf{q}^{(i)}(A^{(i)}_j\cap C^{(i)}_j)-\mathbf{q}(U)$.

Consider an allocation that gives all items in U to the bidder in the highest priced auction where it is ever allocated. Such an allocation must give welfare $\leq OPT$, but $\geq\gamma\mathbf{q}^{(i)}(U)$ due to the price gap in the tree structure. Thus $\mathbf{q}^{(i)}(U)\leq\frac{1}{\gamma}OPT\leq\frac{OPT}{10\beta}$ (by choosing $\gamma=\theta(\log\log m)\geq 10\beta$) . ◁

▷ Claim 27. (5.6 from [1])

$$\mathbb{E}\left[\mathbf{q}^{(i+1)}(C^{(i+1)})\right]\geq\mathbb{E}\left[\mathbf{q}^{(i)}(C^{(i+1)})\right]-\frac{OPT}{10\beta}.$$

Proof. This follows simply from the fact that $\mathbf{q}^{(i+1)}$ is exactly $\mathbf{q}^{(i)}$ with items corresponding (under O) to bidders in N_i set to zero, and that bidders join N_i with probability $1/(10\beta)$. ◁

Proof of Learnable-Or-Allocable Lemma, Lemma 22. By Claim 26 and 27,

$$\mathbb{E}\left[\mathbf{q}^{(i+1)}(C^{(i+1)})\right]\geq\sum_{j=1}^{\alpha}\mathbb{E}\left[\mathbf{q}^{(i)}(A^{(i)}_j\cap C^{(i)}_j)\right]-\frac{OPT}{5\beta},\tag{2}$$

We now have two cases. First, assume

$$\sum_{j=1}^{\alpha}\mathbb{E}\left[\mathbf{q}^{(i)}(A^{(i)}_j\cap C^{(i)}_j)\right]\geq\mathbb{E}\left[\mathbf{q}^{(i)}(C^{(i)})\right]-\frac{2}{15\beta}OPT.\tag{3}$$

Together with (2) this immediately implies that

$$\mathbb{E}\left[\mathbf{q}^{(i+1)}(C^{(i+1)})\right] \geq \mathbb{E}\left[\mathbf{q}^{(i)}(C^{(i)})\right] - \frac{OPT}{3\beta}.$$

and we are in the "learnable case".

On the other hand, if equation (3) is false, then we can sum the inequality in claim 25 for each $j = 1, \ldots, \alpha$ to get

$$\begin{aligned}\mathbb{E}\left[\mathbf{q}^{(i)}(C^{(i)})\right] &\leq \frac{22\beta}{c}\sum_{j=1}^{\alpha}\mathbb{E}\left[\mathrm{val}(A_j^{(i)})\right] + \sum_{j=1}^{\alpha}\mathbb{E}\left[\mathbf{q}^{(i)}(C_j^{(i)} \cap A_j^{(i)})\right] \\ &< \frac{22\beta}{c}\sum_{j=1}^{\alpha}\mathbb{E}\left[\mathrm{val}(A_j^{(i)})\right] + \mathbb{E}\left[\mathbf{q}^{(i)}(C^{(i)})\right] - \frac{2}{15\beta}OPT.\end{aligned}$$

Thus

$$\begin{aligned}&\frac{22\beta}{c}\sum_{j=1}^{\alpha}\mathbb{E}\left[\mathrm{val}(A_j^{(i)})\right] \geq \frac{2}{15\beta}\cdot OPT \\ &\Rightarrow \mathbb{E}\left[\mathrm{val}(A_{j^*}^{(i)})\right] = \frac{1}{\alpha}\sum_{j=1}^{\alpha}\mathbb{E}\left[\mathrm{val}(A_j^{(i)})\right] \geq \frac{2c}{22 * 15\alpha\beta^2}\cdot OPT = \frac{c}{O(\alpha\beta^2)}\cdot OPT.\end{aligned}$$

and we are in the "allocatable" case. ◀

Theorem 16 now follows readily follow from the Learnable or Allocable Lemma.

▶ **Theorem 28.** *Suppose ψ_{min} and ψ_{max} are given and satisfy assumption 21. Suppose the optimal allocation O has supporting prices $\mathbf{q}$, and suppose bidders pick (c, d)-competitive sets in every fixed price auction they participate in. Then mechanism 3 achieves an $O\left(\max\left\{\frac{1}{c}, \frac{1}{d}\right\} \cdot (\log\log m)^3\right)$ approximation to the optimal welfare.*

Proof. Note that by the Learnable or Allocable Lemma, in the mechanism there are only two situation that can occur, 1) event E_1: "learnable" occurs in every iteration $i = 1, 2, ...\beta$, or 2) event E_2: "allocable" occurs in some iteration k. Denote the welfare from the mechanism as $Welf$. Then $\mathbb{E}[Welf]$ satisfy the equation

$$\mathbb{E}[Welf] \geq \min\left(\mathbb{E}[Welf \mid E_1], \mathbb{E}[Welf \mid E_2]\right).$$

Now we bound $\mathbb{E}[Welf \mid E_1]$ and $\mathbb{E}[Welf \mid E_2]$, respectively.

- Suppose that "learnable" occurs for each iteration $i = 1, 2, ...\beta$ in the mechanism. Because $C^{(1)}$ consist of items whose prices belong to the bins of P^*, we know that $\mathbb{E}\left[\mathbf{q}^{(1)}(C^{(1)})\right] = OPT/2$. Thus,

  $$\mathbb{E}\left[\mathbf{q}^{\beta+1}(C^{\beta+1})\right] \geq \mathbb{E}\left[\mathbf{q}^{(1)}(C^{(1)}) - \frac{OPT}{3}\right] = \frac{OPT}{2} - \frac{OPT}{3} = \frac{OPT}{6}.$$

  Let $W_{\beta+1}$ be the welfare achieved when the mechanism allocate in the last iteration of fixed price auction. Since for any correctly priced item $j \in C^{(\beta+1)}$, $\frac{1}{2}\mathbf{q}(j) \geq \mathbf{p}(j) \geq \frac{1}{\gamma}\mathbf{q}(j)$, by lemma 20, $W_{\beta+1} \geq \min\left(\frac{c}{2}, \frac{d}{\gamma}\right) \cdot \mathbb{E}\left[\mathbf{q}^{\beta+1}(C^{\beta+1})\right] = O\left(\min\left(c, \frac{d}{\beta}\right)\right) \cdot OPT$.
  It's easy to verify that the mechanism allocates in last iteration with constant probability. Thus, in this case we get $\mathbb{E}[Welf|E_1]$ at least $O\left(\min\left(c, \frac{d}{\beta}\right)\right) \cdot OPT$.

- In the case where "learnable" does not occur for some iteration i, "allocable" must occur at this iteration. Thus

$$\mathbb{E}\left[\text{val}(A_{j^*}^{(i)})\right] = \frac{c}{O(\alpha\beta^2)} \cdot OPT.$$

  The mechanism allocate in iteration i with probability $(1-1/\beta)^{i-1} \cdot 1/\beta = O(1/\beta)$, thus in this case $\mathbb{E}[Welf \mid E_2]$ is at least $\frac{c}{O(\alpha\beta^3)} \cdot OPT$.

Since $\beta = \Theta(\log\log m)$, we conclude that mechanism M achieves an approximation ratio of

$$O\left(\max\left(\frac{1}{c}, \frac{\beta}{d}, \frac{\alpha\beta^3}{c}\right)\right) = \max\left(\frac{1}{c}, \frac{1}{d}\right) \cdot O(\log\log m)^3. \quad \blacktriangleleft$$

B.4 Removing Assumptions

In this section we prove Theorem 16 in full generality by 1) removing the assumption that the supporting price lies in $\{0\} \cup [\psi_{min}, \psi_{max}]$, where $\psi_{max}/\psi_{min} = \text{poly}(m)$, and 2) showing that this generalized mechanism can be implemented in advised strategies. We use a similar (but slightly simplified) extension to PriceLearningMechanism following previous work on truthful mechanisms for XOS bidders [12, 18, 14, 1][8]. Our variation both simplifies the analysis and allows us to satisfy the formal definition of implementation in advised strategies more easily.

The final mechanism is as follows.

Algorithm 4 GeneralizedMechanism(N, M).

1: Pick a subset of bidders $N_{stat} \subseteq N$ by sampling each bidder in N independently and with probability $\frac{1}{2}$. Let $N_{mech} = N \setminus N_{stat}$.
2: Run a second price auction on the grand bundle M with bidders in N_{stat}. Let SPA be the welfare of the resulting allocation. With probability $\frac{1}{2}$, return the resulting allocation and terminate. With the remaining probability, continue.
3: Set $\psi_{min} = \frac{1}{4m^2} \cdot SPA$ and $\psi_{max} = 4m \cdot SPA$.
4: Run PriceLearningMechanism (Mechanism 3) on bidders in N_{mech} with ψ_{min} and ψ_{max} and return the allocation.

First, we show that implementation in advised strategies allows us to force bidders to play truthfully in the second-price auction of mechanism 4, and to pick (c,d)-competitive sets in the PriceLearningMechanism.

▶ **Lemma 29.** *Suppose we are given a (c,d)-approximate demand oracle D for valuations $\mathcal{V}$. Then there exists a useful poly-time computable advice A for mechanism 4 such that, if a strategy s is advised for v_i under A, then any bidder in N_{stat} will play truthfully in the second price auction, and any bidder in N_{mech} will pick (c,d)-competitive sets in every fixed price auction they participate in.*

Proof. As in prior works [1, 12, 14], to formally meet our solution concept we need all actions by a single bidder to happen simultaneously in order to preclude bidders from "threatening" each other (for example, if a different bidder will only let me have items in future auctions if

[8] Prior works have some probability of selling the grand bundle M in a second price auction (to handle "dominant bidders") or running a different algorithm to collect basic "statistics" on the bidders. We combine the two approaches by using the result of the second price auction to calculate the statistics (at the cost of some loss in the polynomial factor in assumption 21).

I lie in the current auction, then truthful play does not dominate lying). Thus, we formally implement GeneralizedMechanism as a game where each bidder can act in exactly one node. If the bidder is assigned in N_{stat}, the mechanisms simultaneously asks all bidders in N_{stat} for a single bid on the grand bundle. If the bidder is put in N_{mech}, and then into N_i for $i < \beta + 1$, then the bidder needs to participate in α fixed-price auctions simultaneously in a single game node. Thus, the bidder reports a list $(T_j)_{j=1,\dots,\alpha}$ of α subsets of M, where T_j is still available in auction j of the mechanism when it is bidder i's turn to pick a set. Bidders in $N_{\beta+1}$ report similarly, but participate in only one auction.

Recall that the advice function $A(v_i, x, a)$ takes as input the valuation function v_i of player i, a node x of the game, and a "tentative" action a which the player may play. The advice works as follows: for a node x which corresponds to a bidder in N_{stat}, A can ignore the tentative action a and recommend truthful play in the second price auction, i.e. $A(v_i, x, a) = v_i(M)$ in this case. If x corresponds to a bidder put in $N_i \subseteq N_{mech}$ for some $i < \beta$, then the tentative action a is some list of sets $(S_1, \dots, S_\alpha)$ which bidder i may choose in each auction. For each of the α auctions, A will run the (c, d)-approximate demand query D (with prices and remaining items known from the node x) to get a sets $T_1, \dots, T_\alpha$. Then, A will return $(S'_1, \dots, S'_\alpha)$, where S'_i is whichever of S_i or T_i that gives bidder i higher utility. The advice behaves similarly for bidders in $N_{\beta+1} \subseteq N_{mech}$.

It's clear that, if D is computable in poly-time, then $A(v_i, x, a)$ is computable in poly-time.

We now show that A is *useful* (definition 5). A satisfies the required idempotency property, because for bidders in N_{stat}, the result of A is a constant, and for bidders in N_{mech}, the result is given by taking the max of sets S_j with the result of D (which is fixed given bidder's valuation v_i and a node x of the game).

For any s_i and for any randomness in the mechanism, it's clear that $A^{v_i,s}$ gets i utility at least as high as s. For, if i is in N_{stat}, then $A^{v_i,s}$ recommends a dominant strategy, and if i is in N_{mech}, then the utility of i is completely determined by the unique node in which i is chosen to act, and $A^{v_i,s}$ will differ from s only in selecting sets with higher utility for i. Moreover, if $s \neq A^{v_i,s}$, then either s and $A^{v_i,s}$ differ for some node corresponding to a bidder in N_{stat}, or s and $A^{v_i,s}$ differ for some node corresponding to a bidder in N_{mech}. In the first case, because $A^{v_i,s}$ is dominant, there exists v_{-i} and random outcomes of the mechanism which get i strictly higher utility. In the second case, there must be some auction in which the advice $A^{v_i,s}$ selects strictly better sets than s, and because there is positive probability that each auction is the allocation returned by the mechanism, there are some random outcomes of the mechanism which get the bidder strictly more utility. Thus, if $A^{v_i,s} \neq s$, then $A^{v_i,s}$ dominates s.

Finally, it's clear that if a bidder plays according to strategy $A^{v_i,s}$for any s, then if the bidder is in N_{stat} then they play truthfully, and if the bidder is in N_{mech} then they select (c, d)-competitive sets. ◀

Now, we show that algorithm 4 successfully allows us to remove assumption 21. Let S be any set of bidders and let $OPT(S)$ denote the optimal welfare possible for bidders in S. We say (ψ_{min}, ψ_{max}) is *correct for* S if $\psi_{min} \leq OPT(S)/m^2$ and $\psi_{max} \geq OPT(S)$. We call a bidder i dominant for a set S if $v_i(O_i) > \frac{OPT(S)}{8}$.

▶ **Lemma 30.** *Let* $\mathbf{q}$ *be the supporting prices of an optimal allocation of items to bidders in some set* S*. If* (ψ_{min}, ψ_{max}) *is correct for* S*, then the supporting prices of a* $(1 - o(1))$ *fraction of the items (weighted by their supporting prices) are in the range* $I = [\psi_{min}, \psi_{max}]$*. More formally,* $\sum_{j \in M} \mathbb{1}[\mathbf{q}(j) \in I] \cdot \mathbf{q}(j) \geq (1 - \frac{1}{m}) \cdot OPT(S)$*.*

Proof. Since $\psi_{max} \geq OPT(S)$, we know that for all item j, $\mathbf{q}(j) \in [0, \psi_{max}]$. Now we count the sum over supporting prices of items whose supporting price is $\leq OPT(S)/m^2$.

$$\sum_{\mathbf{q}(j) \leq OPT(S)/m^2} \mathbf{q}(j) \leq m \cdot OPT(S)/m^2 = OPT(S)/m.$$

Thus

$$\sum_{j \in [m]} \mathbb{1}\left[\mathbf{q}(j) \in I\right] \cdot \mathbf{q}(j) \geq \sum_{j \in [m]} \mathbf{q}(j) - \sum_{\mathbf{q}(j) \leq OPT(S)/m^2} \mathbf{q}(j) \geq \left(1 - \frac{1}{m}\right) \cdot OPT(S). \quad \blacktriangleleft$$

▶ **Corollary 31.** *For bidders in S and items in M, if (ψ_{min}, ψ_{max}) is correct for S, and $\psi_{max}/\psi_{min} = \text{poly}(m)$, then* PriceLearningMechanism(S, M) *returns an allocation with expected welfare $\frac{1}{r} \cdot (1 - \frac{1}{m}) \cdot OPT(S)$, where $r = O\left(\max\left\{\frac{1}{c}, \frac{1}{d}\right\} (\log\log m)^3\right)$.*

Proof. Again let $\mathbf{q}$ be the supporting prices of an optimal allocation of items to bidders in some set S. Observe that although Theorem 28 assumes all supporting price to be in $0 \cup [\psi_{min}, \psi_{max}]$, the proof holds as is for approximating $\sum_{j \in [m]} \mathbb{1}\Big[\mathbf{q}(j) \in [\psi_{min}, \psi_{max}]\Big] \cdot \mathbf{q}(j)$ (i.e. the contribution to the optimal welfare of items whose supporting price is in $[\psi_{min}, \psi_{max}]$). If (ψ_{min}, ψ_{max}) is correct for S, then

$$\sum_{j \in [m]} \mathbb{1}\Big[\mathbf{q}(j) \in [\psi_{min}, \psi_{max}]\Big] \cdot \mathbf{q}(j) \geq (1 - \frac{1}{m}) \cdot OPT(S).$$

We conclude that PriceLearningMechanism returns an allocation with expected welfare

$$\frac{1}{r} \cdot \sum_{j \in [m]} \mathbb{1}\Big[\mathbf{q}(j) \in [\psi_{min}, \psi_{max}]\Big] \cdot \mathbf{q}(j) = \frac{1}{r} \cdot (1 - \frac{1}{m}) \cdot OPT(S),$$

where $r = O\left(\max\left\{\frac{1}{c}, \frac{1}{d}\right\} (\log\log m)^3\right)$. $\blacktriangleleft$

The following lemma follows from a standard application of chernoff bound and is quoted verbatim from [1]. It allows us to show that, with constant probability, a good fraction of the welfare is achievable by bidders in both N_{stat} and N_{mech}.

▶ **Lemma 32.** [12, 18, 14, 1] *Let $O = (O_1, ...O_n)$ 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 $v_i(O_i) \leq \epsilon \cdot OPT$, then $\sum_{i \in N'} v_i(O_i) \geq (\rho/2) \cdot OPT$ w.p. at least $1 - 2 \cdot \exp(-\frac{\rho}{2\epsilon})$.*

Finally, once the previous lemma has been applied, we will need this lemma to prove that we set the parameters correctly for PriceLearningMechanism.

▶ **Lemma 33.** *If N_{stat} satisfy $OPT(N_{stat}) \geq \frac{1}{4} \cdot OPT$ and $OPT(N_{mech}) \geq \frac{1}{4} \cdot OPT$, then (ψ_{min}, ψ_{max}) is correct for N_{mech}.*

Proof. Assume N_{stat} satisfy $OPT(N_{stat}) \geq \frac{1}{4} \cdot OPT$ and $OPT(N_{mech}) \geq \frac{1}{4} \cdot OPT$. We know that $SPA < OPT(N_{stat})$. Thus

$$4 \cdot OPT(N_{mech}) \geq OPT \geq OPT(N_{stat}) \geq SPA,$$
$$\Rightarrow \psi_{min} = \frac{1}{4m^2} \cdot SPA \leq \frac{1}{m^2} \cdot OPT(N_{mech}).$$

Moreover, since SPA is at least the value of M for any bidder in N_{stat}, we have $m \cdot SPA \geq OPT(N_{stat})$. Thus

$$\psi_{max} = 4m \cdot SPA \geq 4 \cdot OPT(N_{stat}) \geq OPT \geq OPT(N_{mech}).$$

We conclude that (ψ_{min}, ψ_{max}) is correct for N_{mech}. ◀

▶ **Theorem 34.** *For valuation functions $v_1, \ldots, v_n$, suppose the optimal allocation O has supporting prices $\mathbf{q}$. Let D be a (c,d)-approximate demand oracle for valuation in $\{v_1, \ldots, v_n\}$. Then mechanism 4 with advice A as in lemma 29 gets a $O\left(\max\left\{\frac{1}{c}, \frac{1}{d}\right\} \cdot (\log\log m)^3\right)$ fraction of the optimal welfare in implementation in advised strategies.*

Proof. Lemma 29 shows that there exists poly time computable advice such than, whenever a bidder in N_{stat} follows advice, they play truthfully, and whenever a bidder in N_{mech} follows advice, they pick (c,d)-competitive sets in every fixed price auction they participate in.

Recall that a bidder is *dominant* if they contribute more than a 1/8 fraction of the welfare of an optimal allocation. Next we show that whether there is a dominant bidder or not, the expected welfare from GeneralizedMechanism is an $O\left(\max\left\{\frac{1}{c}, \frac{1}{d}\right\}(\log\log m)^3\right)$ approximation to OPT in implementation in advised strategy with advice B.

- When there is a dominant bidder, then with $\frac{1}{2}$ probability the dominant bidder would be selected in the N_{stat} group. Conditioned on this, with $\frac{1}{2}$ probability the resulting allocation from running second price auction on the N_{stat} group would be realized. Since a dominant bidder is in N_{stat} group, the welfare from the second price auction is at least $\frac{OPT}{8}$. Thus the expected welfare of GeneralizedMechanism, conditioned on there being a dominant bidder, is at least $\frac{1}{2} \cdot \frac{1}{2} \cdot \frac{OPT}{8} = \frac{OPT}{32}$.
- When there is no dominant bidder, then by Lemma 32, $OPT(N_{stat}) \geq \frac{1}{4} \cdot OPT$ with probability at least $1 - 2e^{-2}$, which means $OPT(N_{stat}) < \frac{1}{4} \cdot OPT$ with probability $< 2e^{-2}$. Symmetrically, $O(N_{mech}) < \frac{1}{4} \cdot OPT$ with probability $< 2e^{-2}$. By union bound, both $OPT(N_{stat})$ and $OPT(N_{mech})$ is $\geq \frac{1}{4} \cdot OPT$ with probability at least $1 - 4e^{-2}$, which is still a positive, constant probability.
  Let's call the event where $OPT(N_{stat}) \geq \frac{1}{4} \cdot OPT$ and $OPT(N_{mech}) \geq \frac{1}{4} \cdot OPT$ the *good* event.
  By Lemma 33, if the good event occurs, then (ψ_{min}, ψ_{max}) is correct for N_{mech}. By construction in GeneralizedMechanism, $\psi_{max}/\psi_{min} = O(m^3)$. By Corollary 31, conditioned on ψ_{min} and ψ_{max} begin set correctly and $\psi_{max}/\psi_{min} = poly(m)$, priceLearningMechanism returns an allocation that achieves welfare $\frac{1}{r} \cdot (1 - \frac{1}{m}) \cdot OPT(N_{mech})$, where $r = O\left(\max\left\{\frac{1}{c}, \frac{1}{d}\right\}(\log\log m)^3\right)$. Since the good event occurs, $OPT(N_{mech}) \geq \frac{1}{4} \cdot OPT$. We conclude that conditioned on the good event, the expected welfare from PriceLearningMechanism $O\left(\max\left\{\frac{1}{c}, \frac{1}{d}\right\}(\log\log m)^3\right)$ approximates OPT. As the event "the good event happens and GeneralizedMechanism runs PriceLearningMechanism in setp 4" occurs with constant probability, we conclude that Generalized mechanism achieves expected welfare at least $OPT/O\left(\max\left\{\frac{1}{c}, \frac{1}{d}\right\}(\log\log m)^3\right)$ when there is no dominant bidder.

Together with the fact that every allocation for XOS valuation functions has supporting prices, we immediately get theorem 16.

▶ **Theorem 16.** *Let $\mathcal{V}$ be a subclass of XOS valuations and let D be a poly-time (c,d)-approximate demand oracle for valuation class $\mathcal{V}$. Then there exists a poly-time mechanism for welfare maximization when all valuations are in $\mathcal{V}$ with approximation guarantee $O\left(\max\left\{\frac{1}{c}, \frac{1}{d}\right\} \cdot (\log\log m)^3\right)$ in implementation in advised strategies with polynomial time computable advice.* ◀

C Approximate Demand Queries vs. Approximate Welfare Approximation

As it happens, both SimpleGreedy and SingleOrBundle were inspired by simple known algorithms for approximate welfare maximization, combined with the following simple observation:

▶ **Proposition 35.** *S is the return of a demand query on prices $\mathbf{p}$ if and only if $(S, M \setminus S)$ is a welfare maximizing bundle for the following two player auction: one bidder has valuation function v, and the other bidder has additive valuation function given by $\mathbf{p}$.*

Proof. The utility of v is $v(S) - \mathbf{p}(S)$, which differs from the welfare $v(S) + \mathbf{p}(M \setminus S)$ only by the constant $\mathbf{p}(M)$. So maximizing these two objectives is equivalent. ◀

In particular, SimpleGreedy is exactly the 2-approximation algorithm from [33] played by a regular bidder and a "price bidder". SingleOrBundle is similarly inspired by the $\sqrt{m}$ approximation of [17]. However, we show below that approximate demand queries do not, in general, reduce to approximate welfare maximization.

▶ **Example 36.** Consider a budget additive valuation v with value 2 for every item and budget of $2\sqrt{m}$. That is, $v(S) = \max\{2|S|, 2\sqrt{m}\}$. Let $\mathbf{p}$ have price 1 for each item, i.e. $\mathbf{p}(S) = |S|$. The result of a demand query on $(v, \mathbf{p})$ is any set of size $\sqrt{m}$, with utility $\sqrt{m}$.

However, consider running an approximate welfare maximization mechanism $\mathcal{A}$ with two bidders: one with valuation $v(S)$ for bundle S and one with valuation $\mathbf{p}(S)$ for bundle S. The optimal allocation is to give any S of size exactly $\sqrt{m}$ to v, and give the rest of the items to $\mathbf{p}$. This has welfare $m + \sqrt{m}$. However, the allocation giving every item to $\mathbf{p}$ has welfare m. Thus, any constant factor approximation algorithm (for which no other grantees hold) may return this allocation, as $m + \sqrt{m} = (1 + o(1))m$.

This corresponds to an approximate demand query giving the bidder the empty set. As this has zero utility, it will fail to be any factor approximation ration of the optimal.

Moreover, the above example would still go through if we consider a few simple variations on the reduction given by Proposition 35. For example, if we discount prices by a constant factor, say d, it's still the case that $d(m - \sqrt{m}) + 2\sqrt{m} = (1 + o(1))dm$, so a constant-factor approximation algorithm $\mathcal{A}$ might give all items to the "price player".

Thus, approximate demand queries do not reduce to approximate welfare maximization (at least not as outlined by Proposition 35).

D Other Algorithms for approximate demand oracles

Here we give another algorithm for computing a $(1/2, 1/2)$-approximate demand oracle. Instead of being inspired by known welfare maximization algorithms, this technique is inspired by known submodular maximization algorithms. Namely, the algorithm MeetInMiddle below is exactly the algorithm DeterministicUSM from [5], run on the submodular function f given by $f(S) = v(S) - \mathbf{p}(S)$. When f is a nonnegative (possibly decreasing) submodular function, [5] shows that it gives a $1/3$ approximation to the maximum value of f. Unfortunately, the submodular utility function we are interested in is possibly negative, so this result does not apply (indeed, it is NP hard to achieve *any* nontrivial approximation ration for possibly negative submodular maximization, as we discussed in section 3).

Algorithm 5 MeetInMiddle$(v, \mathbf{p}, M)$.

$X \leftarrow \emptyset$ and $Y \leftarrow M$
for $j = 1, \ldots, m$ **do** ▷ For items in an arbitrary order
  Set $a_j \leftarrow v(X \cup j) - v(X) - \mathbf{p}(j)$ ▷ Invariant: $Y = X \cup \{j, \ldots, m\}$
  Set $b_j \leftarrow v(Y \setminus j) - v(Y) + \mathbf{p}(j)$
  if $a_j \geq b_j$ **then**
    Set $X \leftarrow X \cup j$
  else
    Set $Y \leftarrow Y \setminus j$
return X ▷ or return Y (as $X = Y$ by now)

For SimpleGreedy and SingleOrBundle, we needed to run an existing algorithm with the "higher" prices $\mathbf{p}/d$ to attain a (c, d)-approximate demand oracle for (i.e. a set S for which $v(S) - \mathbf{p}(S) \geq c \max_T v(T) - \mathbf{p}(T)/d$). Interestingly, we show that MeetInMiddle need to take the lower ("discounted") prices as input in order to provide an approximation guarantee.

We show that

1. For any $\epsilon > 0$, $S =$MeetInMiddle$(v, \mathbf{p}, M)$ is not a (ϵ, ϵ) approximate demand oracle for prices $\epsilon\mathbf{p}$ (i.e. there exists a valuation function v such that $v(S) - \epsilon\mathbf{p}(S) < \max_T\{v(T) - \mathbf{p}(T)\}$).
2. MeetInMiddle$(v, \mathbf{p}/2, M)$ is an $(\frac{1}{2}, \frac{1}{2})$ approximate demand oracle for prices $\mathbf{p}/2$ (i.e. for any submodular v we have $v(S) - \mathbf{p}(S)/2 \geq \frac{1}{2}\max_T\{v(T) - \mathbf{p}(T)\}$).

▶ **Example 37.** For any $\epsilon > 0$, let $K = 4/\epsilon$ and $N = 2 + (K-1)/\epsilon$ and $M = \{1, 2, ...N\}$. Consider the price vector $\mathbf{p}(1) = \frac{K}{2} + 1$ and $\forall i > 1 : \mathbf{p}(i) = 1 - \epsilon$ and the bidder valuation function $v(S) = K$ for any $S \ni 1$ and $v(S) = 1 + (|S| - 1)\epsilon$ for any $S \ni 1$. One can check that the valuation function is submodular.

MeetInMiddle will remove the first item from X, since $v(M-1) - v(M) + \mathbf{p}(1) = \frac{K}{2} + 1 > \frac{K}{2} - 1 = v(1) - \mathbf{p}(1)$. Similarly, one can check that the algorithm will then remove all items except the last item N, which it will keep. Thus the algorithm returns set $T = \{N\}$, so $v(T) - \mathbf{p}(T) = \epsilon$.

However, the optimal set is $O = \{1\}$. We have $\epsilon(v(O) - \mathbf{p}(O)) = \epsilon(\frac{2}{\epsilon} - 1) = 2 - \epsilon$. Thus $v(T) - \epsilon(p(T)) < \epsilon(v(O) - \mathbf{p}(O))$, and MeedInMiddle$(v, \mathbf{p}, M)$ is not a (ϵ, ϵ) approximate demand oracle for all constant $1 > \epsilon > 0$.

▷ Claim 38. If v is submodular, $S =$MeetInMiddle$(v, \mathbf{p}/2, M)$ is an $(\frac{1}{2}, \frac{1}{2})$ approximate demand oracle for prices $\mathbf{p}/2$ (i.e. $v(S) - \mathbf{p}(S)/2 \geq \frac{1}{2}\max_T\{v(T) - \mathbf{p}(T)\}$.

Proof. Let $T \leftarrow$MeetInMiddle$(v, \mathbf{p}/2, M)$ and $O = \arg\max_{S \subseteq M} v(S) - \mathbf{p}(S)$. We use induction on $|M|$.

For the base case, let $|M| = 1$. Observe that $a_1 = v(1) - \mathbf{p}(1)/2 = -b_1$. Thus, $T = \emptyset$ only when $v(1) \geq \mathbf{p}(1)/2$, so T is exactly $\arg\max_S v(S) - \mathbf{p}(S)/2 \geq v(O) - \mathbf{p}(O)$.

Now, let $|M| > 1$, and assume by induction that the claim is true for all $m' < |M|$. Consider the following two cases:

- If $1 \notin T$, then

$$v(1) - \frac{\mathbf{p}(1)}{2} = a_1 < b_1 = v(M \setminus 1) - v(M) + \frac{\mathbf{p}(1)}{2}$$
$$\Rightarrow \mathbf{p}(1) > v(1) + v(M) - v(M \setminus 1) \geq v(1). \qquad (*)$$

  thus $1 \notin O$. If $M' = M \setminus 1$, then $T =$MeetInMiddle$(v, \mathbf{p}/2, M')$ and $O = \arg\max_{S \subseteq M'} v(S) - \mathbf{p}(S)$. By the inductive hypothesis, $v(T) - \frac{1}{2}\mathbf{p}(T) \geq \frac{1}{2}(v(O) - \mathbf{p}(O))$.

- Suppose $1 \in T$. Let $M' = M \setminus 1$ and let O_2 be the set that maximizes utility on M' for v at prices $\mathbf{p}$ (i.e. $O_2 = \arg\max_{S \subseteq M'} v(S) - \mathbf{p}(S)$). The negation of $(*)$ plus the submodularity of v tells us that

$$\mathbf{p}(1) \leq v(1) + v(M) - v(M \setminus 1) \leq v(1) + v(O_2 \cup 1) - v(O_2). \tag{$\dagger$}$$

  Define a new submodular function v' on M' such that $v'(S) = v(S \cup 1) - v(1)$ for all $S \subseteq M'$. One can check that an item > 1 is added to X in MeetInMiddle$(v, \mathbf{p}/2, M)$ if and only if it is added to X in MeetInMiddle$(v', \mathbf{p}/2, M')$. Thus, $T \setminus 1 =$MeetInMiddle$(v', \mathbf{p}/2, M')$, and the inductive hypothesis tells us that $v'(T \setminus 1) - \frac{1}{2}\mathbf{p}(T \setminus 1) \geq \frac{1}{2}(v'(O_2') - \mathbf{p}(O_2'))$, where O_2' is the set that maximizes utility on M' for v' on prices $\mathbf{p}$ (i.e. $O_2' = \arg\max_{S \subseteq M'} v'(S) - \mathbf{p}(S)$).
  We now analyze two subcases:
  - If $1 \in O$, then $O = 1 \cup O_2'$. Thus, applying the inductive hypothesis we know

$$\begin{aligned} v(T) - \frac{\mathbf{p}(T)}{2} &= v(1) - \frac{1}{2}\mathbf{p}(1) + (v'(T \setminus 1) - \frac{1}{2}\mathbf{p}(T \setminus 1)) \\ &\geq v(1) - \frac{1}{2}\mathbf{p}(1) + \frac{1}{2}(v'(O_2') - \mathbf{p}(O_2')) \\ &\geq \frac{1}{2}\big(v(1) + v'(O_2') - \mathbf{p}(1) - \mathbf{p}(O_2')\big) = \frac{1}{2}(v(O) - \mathbf{p}(O))). \end{aligned}$$

  - If $1 \notin O$, then $O = O_2$. By rearranging $(\dagger)$, we get

$$\begin{aligned} &\mathbf{p}(1) \leq 2v(1) + v'(O_2) - v(O_2) \\ &\Rightarrow v(O_2) - v'(O_2) \leq 2\left(v(1) - \frac{\mathbf{p}(1)}{2}\right). \end{aligned} \tag{$\S$}$$

    Thus

$$\begin{aligned} \frac{1}{2}(v(O) - \mathbf{p}(O)) = \frac{1}{2}\big(v(O_2) - \mathbf{p}(O_2)\big) &= \frac{1}{2}\big(v'(O_2) - \mathbf{p}(O_2) + v(O_2) - v'(O_2)\big) \\ &\leq \frac{1}{2}\big(v'(O_2') - \mathbf{p}(O_2') + v(O_2) - v'(O_2)\big) \\ &\leq v'(T \setminus 1) - \frac{\mathbf{p}(T \setminus 1)}{2} + v(1) - \frac{\mathbf{p}(1)}{2} \\ &= v(T) - \frac{\mathbf{p}(T)}{2}. \end{aligned}$$

    Where the first inequality follows from the definition of O_2', and the second follows from the inductive hypothesis combined with $(\S)$. ◁