



Management Science

Publication details, including instructions for authors and subscription information:
<http://pubsonline.informs.org>

Adaptive Pricing in Combinatorial Auctions

Sébastien Lahaie, Benjamin Lubin

To cite this article:

Sébastien Lahaie, Benjamin Lubin (2025) Adaptive Pricing in Combinatorial Auctions. *Management Science* 71(10):8967–8993. <https://doi.org/10.1287/mnsc.2024.4993>

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License. You are free to download this work and share with others for any purpose, even commercially if you distribute your contributions under the same license as the original, and you must attribute this work as “*Management Science*.” Copyright © 2025 The Author(s). <https://doi.org/10.1287/mnsc.2024.4993>, used under a Creative Commons Attribution License: <https://creativecommons.org/licenses/by-sa/4.0/>.

Copyright © 2025 The Author(s)

Please scroll down for article—it is on subsequent pages



With 12,500 members from nearly 90 countries, INFORMS is the largest international association of operations research (O.R.) and analytics professionals and students. INFORMS provides unique networking and learning opportunities for individual professionals, and organizations of all types and sizes, to better understand and use O.R. and analytics tools and methods to transform strategic visions and achieve better outcomes. For more information on INFORMS, its publications, membership, or meetings visit <http://www.informs.org>

Adaptive Pricing in Combinatorial Auctions

Sébastien Lahaie,^a Benjamin Lubin^{b,*}
^aGoogle Research, New York, New York 10011; ^bInformation Systems Department, Boston University, Boston, Massachusetts 02215

*Corresponding author

Contact: slahaie@google.com,  <https://orcid.org/0000-0002-7828-7289> (SL); blubin@bu.edu,  <https://orcid.org/0000-0002-2786-0997> (BL)

Received: August 1, 2019

Revised: December 31, 2021;
December 20, 2022

Accepted: February 2, 2023

Published Online in Articles in Advance:
February 26, 2025

<https://doi.org/10.1287/mnsc.2024.4993>

Copyright: © 2025 The Author(s)

Abstract. We introduce the first adaptively priced iterative combinatorial auction design, which gradually extends price expressiveness as the rounds progress. This mechanism achieves both high efficiency and fast convergence across a wide range of valuation domains. We implement our auction design using polynomial prices, show how to detect when the current price structure is insufficient to clear the market, and show how to correctly expand the polynomial structure to guarantee progress. An experimental evaluation confirms that our auction is competitive with bundle-price auctions in domains where these excel, namely multiminded valuations, but also performs well in domains favorable to linear prices, such as valuations with pairwise synergy.

History: Accepted by Axel Ockenfels, behavioral economics and decision analysis.



Open Access Statement: This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License. You are free to download this work and share with others for any purpose, even commercially if you distribute your contributions under the same license as the original, and you must attribute this work as “*Management Science*. Copyright © 2025 The Author(s). <https://doi.org/10.1287/mnsc.2024.4993>, used under a Creative Commons Attribution License: <https://creativecommons.org/licenses/by-sa/4.0/>.”

Supplemental Material: The data files are available at <https://doi.org/10.1287/mnsc.2024.4993>.

Keywords: combinatorial auctions • adaptive pricing • polynomial prices • market clearing

1. Introduction

A combinatorial auction (CA) allows bidders to bid on bundles of items to enable them to express complement or substitute preferences, leading to an increase in allocative efficiency. CAs have found widespread application in domains such as spectrum license allocation (Cramton 2013), TV advertising (Goetzendorf et al. 2015), and industrial procurement (Sandholm 2013). Many fielded CAs include an iterative elicitation phase to guide bidders in selecting packages to bid on, to encourage price discovery, and to bring transparency to the allocation process (Cramton 2013). A leading example is the combinatorial clock auction (CCA), which has been used by government entities around the world to conduct spectrum auctions and has generated more than \$20 billion in revenue (Ausubel et al. 2006, Ausubel and Baranov 2017).

A key point of differentiation among the various iterative CA designs proposed to date is their choice of pricing scheme. Proponents of item-price auctions argue that this simple price structure reduces the cognitive burden of package bidding and promotes effective price discovery (Kwasnica et al. 2005). On the other hand, bundle-price auctions such as *iBundle* (Parkes 1999) and the ascending-proxy auction (Ausubel and

Milgrom 2002) are supported by incentive analyses and provable efficiency guarantees. In practice, the CCA attempts to resolve this tension by using an ascending item-price clock phase together with a final round of sealed package bidding (Ausubel and Baranov 2017). Nevertheless, even in the CCA the efficiency of the final allocation in the clock phase is crucial for price discovery, and item prices can limit this efficiency. Moreover, CAs are appropriate precisely for allocation problems where bidders have complex valuations that do not admit linear clearing prices (Bikhchandani and Ostroy 2002).

Given that there are several pricing schemes to choose from, sellers are faced with the problem of selecting this crucial design element a priori without detailed knowledge on bidder valuations that would be needed to guide this choice. In this paper, we address this problem through the introduction of *adaptive* pricing. We design the first iterative auction that employs prices at a middle ground between linear and bundle prices and that augments the price structure as the auction progresses. We implement our auction using *polynomial* prices (Abernethy et al. 2016). Polynomial prices are a natural extension of item prices where combinations of items may be priced as well; for instance, quadratic terms add

surcharges to pairs of items. This added expressiveness allows polynomial prices to clear the market for a wider class of valuations than linear prices and with enough price terms they are guaranteed to clear the market like bundle prices. Our auction begins with linear prices and can provably detect when extra price terms are needed to guarantee progress toward market clearing.

We conduct an extensive simulation study and find that our adaptive polynomial-price CA can simultaneously achieve fast convergence and high efficiency on diverse classes of valuations. By contrast, fixed item-price or bundle-price auctions necessarily fail on some of the classes, either by requiring a very large numbers of rounds or by failing to converge altogether. We use *iBundle* for our baseline bundle-price auction, which is closely related to the ascending-proxy auction (Parkes 1999, Ausubel and Milgrom 2002). For our baseline linear-price auction we use the clock phase of the widely used CCA as described in Ausubel and Baranov (2017). This baseline is similar to the auction design developed in Porter et al. (2003), who propose a more complex version where a final integer program is used to raise efficiency. We also compare our auction to an alternative with prices fixed to linear prices, isolating the benefit of adaptive pricing. We find that even in domains where linear prices typically clear the market, adaptive pricing can substantially speed up convergence with virtually no impact on efficiency.

By using adaptive prices, we add complexity only when it is needed. By using polynomial prices, we are using prices that bidders can think of as “linear prices plus a set of package discounts or surcharges.” Our mechanism is stand alone, and the allocation and prices to which it converges can be directly implemented. Innovations such as adaptive pricing have the potential for considerable real-world impact, given that combinatorial auctions are used to allocate resources worth billions of dollars, including spectrum (Bichler and Goeree 2017, Leyton-Brown et al. 2017), natural resources (Bichler et al. 2019), logistics (Ledyard et al. 2002), procurement (Hohner et al. 2003), and energy (Ausubel and Cramton 2010). However, fielding new mechanisms in high-stakes environments can be challenging. As an alternative, our mechanism could be deployed as a modification to the widely used CCA mechanism where the first (linear clock) phase is replaced with price discovery through adaptive pricing, followed by the CCA’s standard sealed bid phase.

1.1. Our Techniques and Results

The design of our auction follows the linear programming (LP) approach to iterative auction design pioneered by de Vries et al. (2007), building on the work of Bikhchandani et al. (2001) and Bikhchandani and Ostroy

(2002). Under this approach, the efficient allocation problem is formulated as a linear program (without any integer constraints on variables), and an auction corresponds to an algorithm for solving this LP. de Vries et al. (2007) observed that all the iterative CAs in the literature (at least, those with provable efficiency guarantees) could be categorized as either subgradient or primal-dual algorithms to solve the allocation LP: The steps of the algorithms have natural auction interpretations as bidding, provisional allocation, and price update steps. To ensure that the auction converges to an efficient allocation, the crucial property is that the LP formulation should have an integer optimal solution.

Our auction is an instance of the subgradient class, based on its price update method, but our test for price expansion uses ideas from primal-dual algorithms. Testing for price expansion is equivalent to testing whether the associated LP formulation of the allocation problem has an integer optimal solution. The key difficulty is that, during the auction, the bidder valuations that form the objective of the LP are not available, only the current bidder demands. Our main result (Theorem 1) shows that it is enough to test whether a *restricted primal* LP based purely on the demands has an integer solution. This leads to a price expansion policy (Corollary 1) that tests whether further progress is possible with the current price structure or whether price expansion (and possibly price personalization) is needed to clear the market. Although we prove these results in the context of polynomial prices for concreteness, they would straightforwardly extend to more general families of price structures at the cost of more abstraction. For polynomial prices in particular, our final result (Theorem 2) shows how to generate a new price term to guarantee progress or confirm that personalized prices are needed to clear the market. We also explain how the same general approach can handle a dynamic transition to personalized pricing, and even how a mix is possible by making some price terms anonymous and others personalized.

The next main contribution of this work is an extensive empirical evaluation of the adaptive-price auction against benchmarks including *iBundle* and the clock phase of the CCA. We consider valuations generated from the standard CATS test suite (Leyton-Brown et al. 2000), as well as valuations with very different structure (but still motivated by real-world instances) generated according to the quadratic model proposed by de Vries and Vohra (2003). Based on the assumption of straightforward bidding, our key finding is that the adaptive-price CA is the only auction able to perform at high levels of efficiency and fast convergence rates across all domains.

The lowest average efficiency of the adaptive-price CA across domains is 93.2%, whereas for the benchmarks it

drops to 46.8%–81.0%. Our auction is also able to clear nearly all instances of every distribution; its lowest rate of clearing across domains is 97% of the instances. In terms of rounds to convergence, adaptive pricing again leads to the best average performance across domains. Even in the absence of detailed a priori knowledge on the structure of bidder valuations, our experiments demonstrate that adaptive pricing achieves high rates of clearing, efficiency, and convergence.

1.2. Practical Considerations and Incentives

The focus of this work is on price structure and price updating in iterative CAs. Of course, many other aspects of design are also important, including reserve prices and payment rules that discourage gaming behavior (Ausubel and Baranov 2017). As our contribution is computational in nature, we consider incentive problems to be an orthogonal issue and assume throughout that bidders follow *straightforward* bidding, meaning that at each round they place a bid on a bundle that maximizes their utility at the given prices. This follows prior computational work on iterative auctions (Parkes 1999, Pikhovskiy et al. 2006, Bichler et al. 2009).

To induce bidders to bid straightforwardly, the auction can ensure two conditions: (1) bidders place bids consistent with some valuation function, and (2) bidders are charged Vickrey-Clarke-Groves (VCG) payments at the end. The first condition can be enforced by using an appropriate activity rule, and we defer a discussion of this to Section 3.4 once the details of our auction have been described. For the second condition, one possible approach is to maintain multiple price paths, effectively running several auctions in parallel (one with each agent removed) to eventually elicit enough information to compute VCG payments (Ausubel 2006). If both conditions hold, then straightforward bidding becomes an ex post Nash equilibrium of the auction mechanism (de Vries et al. 2007).

Alternatively, rather than using multiple price paths, one could view our adaptive-price auction as a possible replacement for the clock phase of the CCA and apply a suitable payment rule (e.g., VCG or core selecting) in a supplementary sealed-bid phase. Because it achieves high rates of market clearing, the adaptive-price auction obviates the need for a second phase. The final clearing prices can be used for payment, supporting the objectives of simplicity and transparency. Nonetheless, the adaptive-price auction is entirely compatible with a supplementary sealed-bid phase if so desired—for example, for the sake of familiarity to bidders, as a transition mechanism to a single-phase standard. Current activity rules linking the first and second phase, such as the relative and final cap rules (Ausubel and

Baranov 2017), do not depend on price structure but only the historical prices of bundles in the clock phase.

Without either multiple price paths or a second phase, the assumption of straightforward bidding is a strong one. In Appendix E, we investigate the robustness of our mechanism to a form of heuristic bidding where, although not adversarial, bidders respond to the mechanism only with approximate myopic best responses. We observe that the mechanism is reasonably robust to such bidder behavior but that robustness is generally domain dependent.

Another noteworthy practical aspect of the adaptive-price auction is that it uses nonmonotone prices (i.e., the price of any given bundle may not be purely ascending or descending). Although nonmonotonicity introduces no technical difficulties, it can complicate bidding in practice, especially for bidders that are trying to manage budgets. In Appendix F, we analyze the price trajectories generated by our auction and find that typically they initially overshoot the clearing price and are thus largely *descending*, especially in the early rounds. In later rounds, price decreases are punctuated by increases at fairly regular intervals to ensure prices converge to market-clearing prices. Our robustness checks against deviations from straightforward bidding and on the extent of price nonmonotonicity are encouraging, but before such a new format is actually used in practice, a thorough strategic analysis and laboratory experiments would be advised.

1.3. Related Work

Combinatorial auction formats with both linear and bundle prices have been studied extensively through theory, simulation studies, and laboratory experiments. In this section, we first cover the arguments in favor of each format and the limited prior work on alternative price formats. We next discuss the related literature on existence of Walrasian equilibria with indivisibilities and connections to the literature on preference elicitation in the field of artificial intelligence.

1.3.1. Bundle-Price Auctions. The arguments in favor of bundle-price auctions are that they lead to high levels of allocative efficiency in both theory and practice. Parkes (1999) introduces *iBundle*, a combinatorial auction with personalized bundle prices, along with a dynamic variant called *iBundle(d)* that switches from anonymous to personalized prices when needed. Ausubel and Milgrom (2002) introduce the ascending proxy auction, closely related to *iBundle*, and focus on its incentive properties. Both auctions provably achieve high efficiency as long as agents follow straightforward bidding. Mishra and Parkes (2007) extend *iBundle* so

that it computes VCG payments upon termination and thereby incentivizes agents to bid straightforwardly.

1.3.2. Linear-Price Auctions. An important argument in favor of linear-price auctions is that item prices provide informative feedback for the bidders, alleviating the difficult cognitive problem of package bidding, and leading to fewer auction rounds. Kwasnica et al. (2005) propose an item-price auction called *resource allocation design* (RAD) and find via laboratory experiments that it performs better than the baseline simultaneous multiple round auction (Milgrom 2000). The innovation of RAD over prior designs is to compute item prices that approximately clear the market. In separate laboratory experiments, Scheffel et al. (2011) find that *iBundle*, VCG, and linear-price formats such as *approximate linear prices* (ALPS, a design inspired by RAD) all have similar allocative efficiency in environments with up to 18 items; they also observe that *iBundle* requires many more rounds than linear-price auctions. Schneider et al. (2010) perform extensive simulations with several linear- and bundle-price auctions including *iBundle*, ALPS, and the CCA. They find that linear-price auctions are more robust to deviations from straightforward bidding, especially in terms of the number of rounds.

1.3.3. Other Price Formats. The types of polynomial-price auctions implemented in this work have been analyzed theoretically by Abernethy et al. (2016)—without adaptive pricing—who provide convergence rate bounds that depend on the polynomial degree. With the exception of the *iBundle(d)* variant that switches from anonymous to personalized prices (Parkes 1999), all auction formats in the literature have a fixed price structure. Day (2018) studies a variety of alternative pricing schemes in CAs that can be obtained via the duality between allocation and pricing and discusses their economic interpretation. It may be possible to embed some of these structures into our framework to yield other adaptive-price CAs.

1.3.4. Walrasian Equilibria. The study of iterative auctions takes its roots in the literature on the existence of Walrasian equilibria (i.e., linear-price competitive equilibria) and the classical *tâtonnement* procedure (Walras 1874). Kelso and Crawford (1982) introduced the notion of *gross substitutes* to guarantee the convergence of a natural salary adjustment procedure between workers and firms. Gul and Stacchetti (1999, 2000) later used this notion to analyze the existence of Walrasian equilibrium in the presence of indivisible items and to define an ascending-price auction that converges to VCG payments. Lehmann et al. (2006) study valuation classes that exhibit decreasing marginal values more generally,

which includes gross substitutes and submodular valuations among others. For a detailed coverage of this literature and its key results, we refer to the excellent survey by Paes Leme (2017). Baldwin and Klemperer (2019) have identified other demand types besides gross substitutes that admit Walrasian equilibria in the presence of indivisibilities. Whenever these elegant characterizations are not known to hold in specific applications, our auction offers a way to extend beyond linear prices to restore the existence of competitive equilibrium with indivisibilities.

1.3.5. Preference Elicitation. Our work connects with a growing literature in the field of artificial intelligence, beginning with Blum et al. (2004) and Lahaie and Parkes (2004), that views iterative CAs as mechanisms for *preference elicitation* with the goal of computing an efficient (or approximately efficient) allocation. This research views bidders as black boxes that can respond to different types of queries: demand queries, where bidders best-respond to prices as in a typical auction, but also value queries, where bidders state their values for given bundles. Blumrosen and Nisan (2010) study the relative power of demand and value queries to learn valuation functions. In particular, they show that there are valuations classes for which finding an efficient allocation requires an exponential number of bundle-price demand queries, but only a polynomial number of linear-price demand queries. Brero et al. (2017, 2018) introduce iterative CAs that embed machine learning algorithms based on value queries to learn preferences; these can accommodate a wide range of models including deep neural networks (Weissteiner and Seuken 2020).

2. Background

We consider a model where a single seller holds a set of m distinct items that is auctioned among n agents (the buyers). The items are indivisible, and there is unit supply of each item. Let M denote the set of items and N denote the set of agents. A *bundle* is a subset of items $Z \subseteq M$; we denote the set of bundles by $\mathcal{Z} (= 2^M)$, which includes the empty bundle. The preferences of agent $i \in N$ are captured by its *valuation* function $v_i: \mathcal{Z} \rightarrow \mathbb{R}$. We assume that valuations are normalized and monotone: $v_i(\emptyset) = 0$ and $v_i(Z) \leq v_i(Z')$ for $Z \subseteq Z'$. We refer to the set of valuations that satisfy these conditions as the class of *general valuations*. We will not make any further assumptions on valuations as we develop our auction, although we will naturally impose some further structure in our examples and empirical evaluation. An *allocation* is a vector of bundles $\mathbf{Y} = (Y_1, \dots, Y_n)$ assigned to the agents, where agent i obtains bundle Y_i . An allocation $\mathbf{Y}$ is *feasible* if $Y_i \cap Y_j = \emptyset$ for $i \neq j$; we denote the set

of feasible allocations by $\mathcal{F}$. An allocation $\mathbf{Y}$ is *efficient* if

$$\mathbf{Y} \in \arg \max_{\mathbf{Y}' \in \mathcal{F}} \sum_{i \in N} v_i(Y'_i), \quad (1)$$

or in words, if it maximizes the total value to the agents among all feasible allocations. As we have formulated it thus far, the efficient allocation problem is computationally intractable because the inputs themselves (the valuations) are of exponential size. Even when restricting the class of valuations so that each agent only values a single bundle (i.e., single-minded agents), computing the efficient allocation is NP-hard by reduction from weighted set packing (Nisan 2000). Combinatorial auction implementations, both single-shot and iterative, therefore make use of integer programming techniques to solve these problems in practice (Sandholm 2006). Iterative auctions typically solve a revenue-maximization problem at each round, which is just the efficient allocation problem where valuations are replaced with prices. Depending on price structure, this may be simpler or just as computationally difficult as efficient allocation with respect to valuations. We return to this issue when we discuss the problem of computing a provisional allocation in Section 3.3.

2.1. Market Clearing

When running an iterative auction, the seller maintains *prices* for each agent $i \in N$. Like valuations, the prices are mappings $p_i : \mathcal{Z} \rightarrow \mathbb{R}$. As written, the prices may be *personalized*, meaning that different agents may face different prices; the vector of personalized prices is denoted $\mathbf{p} = (p_1, \dots, p_n)$. Several bundle-price auctions proposed in the literature use personalized prices, including *iBundle* (Parkes 1999), which will serve as a baseline for our experiments. Prices are *anonymous* if $p_i = p_j$ for all $i, j \in N$; in this case, we usually drop the agent index and simply write p to represent the prices.

We assume that agents have quasi-linear utility, so that agent i 's utility for obtaining bundle Z at prices p_i is $u_i(Z; p_i) = v_i(Z) - p_i(Z)$. Agent i 's ϵ -demand set at prices p_i is

$$D_i^\epsilon(p_i) = \{Z \in \mathcal{Z} : u_i(Z; p_i) \geq u_i(Z'; p_i) - \epsilon, Z' \in \mathcal{Z}\}. \quad (2)$$

In words, this is the set of bundles that maximize the agent's utility at the given prices to within an additive error of ϵ . We write $D_i = D_i^0$ to denote the agent's exact demand mapping. Given prices $\mathbf{p}$, the *revenue* of allocation $\mathbf{Y}$ is simply $\sum_{i \in N} p_i(Y_i)$. We assume that the seller does not value the items, so the utility it derives from an auction is the revenue collected. The seller's *supply*

set at prices $\mathbf{p}$ is

$$S(\mathbf{p}) = \arg \max_{\mathbf{Y} \in \mathcal{F}} \sum_{i \in N} p_i(Y_i), \quad (3)$$

namely the set of feasible allocations that maximize the seller's revenue. We stress that (2) and (3) define set-valued mappings, not just selections of a single bundle or allocation.

An iterative auction proceeds over rounds, updating a provisional allocation along with prices at each round until there is a balance between demand and supply. To formalize this balancing notion, we say that prices $\mathbf{p}$ are *clearing prices* if there exists a feasible allocation $\mathbf{Y} \in S(\mathbf{p})$ such that $Y_i \in D_i(p_i)$ for all $i \in N$. In words, the allocation simultaneously maximizes the seller's revenue and every agent's utility at the given prices; in this sense, there is a balance between supply and demand.¹ We say that the clearing prices *support* such an allocation $\mathbf{Y}$ (which need not be unique). We define ϵ -clearing prices analogously by requiring $Y_i \in D_i^\epsilon(p_i)$ instead for each agent. Given ϵ -clearing prices $\mathbf{p}$ supporting an allocation $\mathbf{Y}$, we call the pair $(\mathbf{Y}, \mathbf{p})$ an ϵ -competitive equilibrium.² The following standard result, which is used throughout the literature on combinatorial auctions (Parkes 1999, de Vries et al. 2007, Mishra and Parkes 2007), gives the key property of competitive equilibrium. We defer all proofs to Appendix A.

Proposition 1 (Bikhchandani and Ostroy 2002). *If $(\mathbf{Y}, \mathbf{p})$ is an ϵ -competitive equilibrium, then $\mathbf{Y}$ is efficient to within an additive error of $n\epsilon$.*

This result motivates a natural termination criterion: the auction halts if the provisional allocation and current prices form an ϵ -competitive equilibrium, where ϵ is a discount set by the auctioneer. The auction first maintains the invariant that the provisional allocation always maximize revenue at the current prices, which is the supply-side condition of competitive equilibrium. Next, the auction offers an ϵ discount on the price of any bundle in the provisional allocation. If the buyers follow straightforward bidding and accept their allocated bundles, this verifies the demand-side condition of competitive equilibrium. Proposition 1 establishes that this termination rule comes with an efficiency guarantee, which improves with smaller discount parameter ϵ . Consequently, the discount parameter is used in practice (and in our evaluation) to control a tradeoff between the auction's speed of convergence and the efficiency of its final allocation.

Remark 1. A valuation is a function $v_i : \mathcal{Z} \rightarrow \mathbb{R}$ that can in principle take on negative values, but the assumptions of normalization and monotonicity imply that general valuations are always nonnegative. Similarly, a

price function $p_i : \mathcal{Z} \rightarrow \mathbb{R}$ could in principle assign negative prices for some bundles. However, the price of an allocated bundle in a competitive equilibrium cannot be negative because this would contradict the condition that the allocation maximize revenue (discarding a negative-price bundle would increase the revenue of the allocation). Furthermore, if we introduce a “dummy agent” into the auction with a value of zero for all bundles, then no bundle can have a negative price in a competitive equilibrium: The bundle with the most negative price would maximize the dummy agent’s utility but could not be part of a revenue-maximizing allocation. Therefore, by using this device of a dummy agent, it is straightforward for the auction to automatically ensure that prices are nonnegative if this is considered important in practice.

2.2. Price Structure

Because the domain of bundles is exponential in size, an iterative auction cannot explicitly maintain a price for every possible bundle even with a moderate number of items. We must therefore consider sparse encodings for prices. The classic approach is to specify item prices (also called linear prices), and the price of a bundle is then defined as the sum of its constituent item prices.

In this work we will implement our adaptive-price auction using *polynomial* prices, a generalization of item pricing. These prices are formally specified as a pair $(\mathcal{M}, w)$ where $\mathcal{M} \subseteq \mathcal{Z}$ is a collection of bundles and $w : \mathcal{M} \rightarrow \mathbb{R}$ is a function mapping the bundles to *coefficients*. The price of a bundle Z is then defined as

$$p(Z) = \sum_{Z' \in \mathcal{M} : Z' \subseteq Z} w(Z'). \quad (4)$$

We require $\emptyset \notin \mathcal{M}$ so that the prices are normalized to $p(\emptyset) = 0$. The set $\mathcal{M}$ captures the “monomials” in the polynomial prices. In practice, $\mathcal{M}$ should ideally remain of low cardinality. If we impose the constraint that $\mathcal{M}$ only consist of single-item bundles, then we recover linear prices. To understand why the structure in (4) corresponds to a polynomial, suppose that $\mathcal{M}$ only contains bundles of size at most two, which leads to *quadratic* prices. In this case, polynomial prices over three items $\{a, b, c\}$ can alternatively be expressed in the form

$$\begin{aligned} &w(\{a\})x_a + w(\{b\})x_b + w(\{c\})x_c + w(\{a, b\})x_ax_b \\ &+ w(\{a, c\})x_ax_c + w(\{b, c\})x_bx_c, \end{aligned}$$

where $x_a, x_b, x_c \in \{0, 1\}$ are indicator variables for the items contained in the bundle being priced. This is a quadratic polynomial, with the coefficients given by w . The *degree* of polynomial prices is $\max_{Z \in \mathcal{M}} |Z|$ and their *cardinality* is $|\mathcal{M}|$. Note that individual coefficients in

the polynomial may be negative. However, as explained in the Remark 1, the price function represented by such a polynomial should nonetheless take on nonnegative values if it corresponds to clearing prices.

Besides item prices, the other most common structure is *bundle* prices. These are represented by a collection of bundle-value pairs captured in an expression of the form $(X_1, \alpha_1) \oplus \dots \oplus (X_k, \alpha_k)$, which represents the following price function over bundles $Z \in \mathcal{Z}$:

$$p(Z) = \max\{\alpha_\ell : X_\ell \subseteq Z, \ell = 1, \dots, k\}.$$

The $\oplus$ operator refers to the fact that only one of the X_ℓ bundles can be chosen.³ For personalized prices, a separate expression is listed for each agent. We define the *cardinality* of bundle prices as k , the number of bundles in the price expression. In bundle-price auctions like *iBundle*, an (X_ℓ, α_ℓ) term is added each time a new bundle X_ℓ is bid on by an agent. To ensure that the corresponding price function is nonnegative, the term $(\emptyset, 0)$ can be added to the price expression.

The crucial property of a price structure is its cardinality (i.e., representation size), which determines how computationally efficient it is for an auction to work with the price structure and for bidders to best respond to quoted prices. It is straightforward to construct examples of price functions where the polynomial representation has linear cardinality whereas bundle prices have exponential cardinality, and vice versa. However, the main purpose of prices in an auction is to generate inequalities of the form

$$v_i(X_i) - p_i(X_i) \geq v_i(Z) - p_i(Z)$$

for all $Z \in \mathcal{Z}$, where X_i is a bundle assigned to agent i . In technical terms, we want the price structure to be able to represent a *supergradient* to the agent’s valuation function at any bundle we might assign to the agent, as this is the property needed to support an efficient allocation. Even when the agents have complex, general valuations, the supergradients to these valuations can often be represented using much simpler, low-cardinality polynomials, and we exploit this fact in the design of our auction.

Although it is intuitive to view the polynomial coefficients as package surcharges or discounts, for large bundles these coefficients can cancel each other, which complicates interpretation. We make no claim that polynomial prices are somehow more meaningful or interpretable than bundle prices in general, as this is necessarily application specific.

2.3. Existence of Clearing Prices

The existence of clearing prices with specific structure (e.g., linear, anonymous) depends on the class of agent valuations at hand. We say that a price structure is *fully*

expressive if it can represent any price function over the domain of bundles $\mathcal{Z}$ (subject only to normalization). With general valuations, fully expressive personalized prices are needed to ensure that clearing prices can be represented in the worst case (Bikhchandani and Ostroy 2002); bundle prices and degree- m polynomials are fully expressive, but in either case, the cardinality of the prices may be exponential in the number of items. If the valuations are super-additive, then anonymous prices suffice to clear the market (Parkes 2006, Lahaie and Parkes 2009).⁴ At the other extreme, anonymous linear prices are known to clear the market when agents have gross substitutes valuations (Kelso and Crawford 1982, Gul and Stacchetti 1999), which includes additive and unit-demand valuations as special cases. Candogan et al. (2018) study the relative clearing power of linear versus quadratic prices, both anonymous and personalized. They provide instances that quadratic prices can clear but that linear prices cannot clear and identify valuation classes besides gross substitutes where the two are equally effective at supporting market clearing.

The following examples illustrate how it can be difficult a priori to know what structure of prices is needed to clear the underlying valuations. With just a slight change to valuations, one can go from an instance with linear clearing prices to an instance where prices must essentially be fully expressive. In the examples, a bundle-value pair (X, α) represents a *single-minded* agent with valuation

$$v_i(Z) = \begin{cases} \alpha & \text{if } Z \supseteq X \\ 0 & \text{otherwise.} \end{cases}$$

A single-minded valuation expresses complementarity among the items in X .

Example 1. There are three items $\{a, b, c\}$ and four single-minded agents with valuations $(\{a, b\}, 3)$; $(\{b, c\}, 3)$; $(\{a, c\}, 3)$; $(\{a, b, c\}, 5)$. The efficient allocation is to give all items to the last agent. It is easy to verify that this allocation is supported by linear prices of 1.5 for each item. If we change the last agent slightly to $(\{a, b, c\}, 4)$, then the efficient allocation remains unchanged, but it is an easy exercise to show that now linear and even quadratic clearing prices no longer exist. Because there are just three items, degree-3 (i.e., cubic) polynomial prices do clear the example, and because the valuations are super-additive, there exist anonymous bundle clearing prices (Parkes 1999). A solution to this example is given in Section 5.

Example 2. A general counterexample that yields the inexistence of *anonymous* clearing prices is given by Nisan and Segal (2006). Their construction also yields examples where personalized clearing prices fail to exist whenever the prices are not fully expressive.

Given valuation v_1 , let the second agent's valuation be $v_2(Z) = v_1(M) - v_1(M \setminus Z)$. Nisan and Segal (2006) show that the unique clearing prices are $p_1 = v_1$ and $p_2 = v_2$ with such valuations. The prices must therefore be personalized, and if there are m items, then v_1 can be constructed such that it cannot be represented by any polynomial of degree strictly less than m .

With indivisible items, it is typically the case that clearing prices, when they exist, are not unique. As our focus is on reaching an efficient allocation, uniqueness is not a concern, but for some applications, selecting particular clearing prices may be relevant. For instance, in an auction for public resources, the auctioneer may wish to compute minimal clearing prices to extract as little surplus as possible. Abernethy et al. (2016) introduce a regularization term in the dual pricing objective to reach minimal clearing prices and work out the implications for the price increment policy, but we do not pursue such a refinement in this work.

3. Iterative Auction

In this section, we first describe a nonadaptive version of the polynomial price auction where the polynomial terms are *fixed*. For the sake of simplicity, we also describe the auction using *anonymous* prices. We introduce the main auction steps performed at each round and then explain the process of computing a provisional allocation. The overall auction scheme that actually interfaces with bidders, detailed in Algorithm 1, follows a standard format: A round consists of steps for bidding, allocation, and price updating. Rounds continue until the market clears. The key novel element is a price expansion step, which remains a black box to the bidders, like winner determination. We defer adaptive pricing (Algorithm 2), including how to switch to personalized prices if needed, to the next section.

Our auction is an instance of the subgradient class of iterative auctions (de Vries et al. 2007) because it computes an “excess demand” vector and increments prices in the direction of this vector. An auction from the primal-dual class, in contrast, formulates a “restricted primal” allocation problem as an LP, whose dual gives a price increment in the direction of clearing prices. An advantage of the subgradient approach is that it only requires a single bundle from each bidder's demand set to compute excess demand, whereas the primal-dual approach needs each bidder's entire demand set. A subgradient auction also does not need any information about valuation magnitudes to ensure convergence, as long as it uses a suitable diminishing price increment. In contrast, a primal-dual auction needs a priori information on valuation magnitudes to ensure that its price increment does not overshoot.⁵

Algorithm 1 (Iterative Auction)

Input: Initial price monomials $\mathcal{M}$; Discount ϵ ; Stepsize schedule $\{\eta^t\}$; Max rounds $t_{\max}$; Max time $\tau_{\max}$; Epoch length ϕ

Output: Allocation $\mathbf{Y}^t$; prices p^t ; Clearing status $\{\perp, \top\}$

```

1  $w^1(Z) \leftarrow 0 \quad \forall Z \in \mathcal{M}$                                 ▷ Initialize price coefficients
2  $\mathcal{B}_i^0 \leftarrow \{\emptyset\} \quad \forall i \in N$                             ▷ Initialize observed bundles
3  $t \leftarrow 0$                                                     ▷ Initialize round
4 do                                                            ▷ Loop over rounds
5    $t \leftarrow t + 1$                                             ▷ Increment round
6    $p^t \leftarrow (\mathcal{M}, w^t)$                                     ▷ Establish prices
7    $\mathbf{Y}^t \leftarrow \text{SupplyQuery}(p^t, \mathcal{B}^{t-1})$                 ▷ Compute revenue maximizing allocation; see Section 3.3
8    $X_i^t \leftarrow \text{DemandQuery}(i, p^t, Y_i^t, \epsilon) \quad \forall i \in N$     ▷ Query each bidder  $i$  with  $Y_i^t$  discounted by  $\epsilon$ 
9    $\mathcal{B}_i^t \leftarrow \mathcal{B}_i^{t-1} \cup \{X_i^t\} \quad \forall i \in N$             ▷ Update observed bundles
10  if  $X_i^t = Y_i^t \quad \forall i \in N$  then                            ▷ Provisional allocation accepted
11    return  $(\mathbf{Y}^t, p^t, \top)$                                     ▷ Auction cleared
12  end
13   $(\mathcal{M}, w^t) \leftarrow \text{ExpandPrices}(\mathbf{X}^t, \mathcal{B}^t, \mathcal{M}, w^t, t, \phi)$     ▷ Obtain new price terms; see Algorithm 2
14   $w^{t+1} \leftarrow \text{UpdatePrices}(\mathbf{X}^t, \mathbf{Y}^t, w^t, \eta^t)$         ▷ Update price coefficients; see Equation (5)
15 while  $t < t_{\max} \wedge \text{ElapsedTime}() < \tau_{\max}$                 ▷ Rounds remaining
16 return  $(\mathbf{Y}^t, p^t, \perp)$                                         ▷ Auction stopped

```

3.1. Auction Steps

The auction proceeds over rounds, indexed by t , which consist of *allocation*, *bidding*, and *price update* steps. There is also a *termination* criterion checked at each round. The auction maintains polynomial prices $(\mathcal{M}, w^t)$ where the set of monomials $\mathcal{M}$ is fixed for now, but the coefficients w^t are updated across rounds. The polynomial coefficients are initialized to zero, leading to initial prices at zero. As the auction prices are anonymous for now (see Section 4.4 for price personalization), we write p^t rather than $\mathbf{p}^t$ to refer to the prices.

Pseudo-code for the iterative auction is provided in Algorithm 1. The auction is parametrized by a discount $\epsilon > 0$ and a step size schedule $\{\eta^t\}_{t=1}^\infty$ that controls the magnitude of the price updates as the rounds progress. In our implementation, runs of the auction were bounded by a maximum number of rounds $t_{\max}$ and a maximum runtime $\tau_{\max}$. The epoch length ϕ listed as part of the input controls the frequency of price expansion (covered in the next section). The choice of initial monomials $\mathcal{M}$ is up to the auctioneer; in our implementation, we initialize with all singleton bundles, so that the auction begins with linear prices. We now describe the auction and pseudo-code steps in detail, referring to the line numbers in Algorithm 1.

3.1.1. Allocation. At the beginning of each round t , the seller computes a provisional allocation $\mathbf{Y}^t$ that maximizes its revenue at current prices p^t , so that $\mathbf{Y}^t \in S(p^t)$. In Line 7, this subroutine is referred to as `SupplyQuery`, taking in the current prices p^t as input. In the first round the prices are zero and ties are broken so that the first provisional allocation is empty. Under polynomial

prices, computing the provisional allocation can be straightforwardly formulated as an integer linear program, using binary variables to encode which items each agent receives and to relate the items to the polynomial terms. However, the program consists of $\Theta(mnd)$ constraints and $\Theta(nm + nd)$ variables, where d is the cardinality of the polynomial prices. The program size scales too poorly with the cardinality to be of practical use. In Section 3.3, we will describe a scalable approach to computing the provisional allocation which performs well in practice, at the expense of weaker theoretical efficiency guarantees. This approach makes use of the history of bundles bid on by each agent, hence the inclusion of this history as an argument to `SupplyQuery` in Line 7.

3.1.2. Bidding. Next comes the bidding step. Given the current prices p^t and provisional allocation $\mathbf{Y}^t$, each agent i may choose to accept the bundle Y_i^t or bid by communicating a different bundle X_i^t . In the former case, the agent is essentially bidding on its allocated bundle, so we notate this as $X_i^t = Y_i^t$. When reasoning about its bid, we stipulate that the agent may take a discount of ϵ on the price of Y_i^t . (This discount serves a practical purpose as explained in Section 3.2.) Note that, following the usual clock auction format, an agent only communicates which bundle it wishes to bid on, without explicit reference to price. The process of each agent i placing a bid, given current prices p^t , the proposed bundle Y_i^t and a discount on this bundle ϵ , is denoted as subroutine `DemandQuery` in Line 8.

3.1.3. Termination. If every agent $i \in N$ accepts the bundle Y_i^t it is allocated in the current round, or in other

words if $X_i^t = Y_i^t$ for all $i \in N$, then the auction terminates. This termination check corresponds to Lines 10–12.

3.1.4. Price Expansion. A key feature of our mechanism is its use of adaptive price expansion, which may alter the structure of the prices and thereby increase their ability to clear the market. This step is denoted by the subroutine `ExpandPrices` in Line 13. In this section we assume `ExpandPrices` is a no-op, yielding a non-adaptive version of the mechanism. We devote Section 4 to carefully constructing a definition of `ExpandPrices` that will yield the full *adaptive* version of our mechanism.

3.1.5. Price Update. If the termination condition does not hold, then prices are updated for the next round. Following Abernethy et al. (2016), the update rule for the polynomial prices takes the form

$$w^{t+1}(Z) = w^t(Z) + \eta^t |\{i \in N : Z \subseteq X_i^t\}| - \eta^t |\{i \in N : Z \subseteq Y_i^t\}|, \quad (5)$$

for all nonempty $Z \in \mathcal{M}$. The rule can be interpreted as an update to coefficient $w(Z)$ in the direction of *excess demand* for the combination of items Z : We count the number of agent bids that contain Z and subtract away the number of allocated bundles that contain Z (there can be at most one of the latter because the provisional allocation is feasible). We then update the coefficient on Z by this quantity, weighted by the step size η^t . Note that when $X_i^t = Y_i^t$ for all $i \in N$ (i.e., the termination condition holds), the price update evaluates to zero. The price coefficient can be updated upward or downward, and this translates into nonmonotonic prices. However, the coefficient is updated downward only when it is not demanded at all at the current prices. The price update procedure specified by (5) is invoked as subroutine `UpdatePrices` in Line 14.

3.2. Auction Convergence

If each agent follows a *straightforward bidding* strategy, then by definition at each round t each agent i 's bid satisfies $X_i^t \in D_i(p^t)$. However, if $Y_i^t \in D_i^e(p^t)$, then the agent chooses to accept its provisional allocation specifically because of the price discount. Recall that the provisional allocation is chosen such that $\mathbf{Y}^t \in S(p^t)$. Under straightforward bidding, the termination condition therefore amounts to verifying that $(\mathbf{Y}^t, p^t)$ constitutes an ϵ -competitive equilibrium, and therefore that $\mathbf{Y}^t$ is approximately efficient by Proposition 1.

Under straightforward bidding, the auction formally corresponds to a subgradient algorithm to solve the dual linear program to the efficient allocation problem—We refer to proposition 4.1 of Abernethy et al. (2016) for a formal proof. The fact that utility-maximizing bundles (2) combine with a revenue-maximizing allocation (3) to yield a subgradient is closely related to classic microeconomic

results (Roy's identity and Shephard's lemma) relating demand and supply functions to the derivatives of indirect utility and cost functions (Varian 1992). Subgradient algorithms for linear programs, and convex programs more generally, are only guaranteed to converge to an optimal solution in the limit, rather than a finite number of steps. This is the reason for introducing the ϵ discount in practice. Other iterative auctions of the subgradient class, such as *iBundle*, also use this device. A larger ϵ discount induces earlier convergence, at the expense of a weaker efficiency guarantee upon termination.

3.3. Provisional Allocation

We noted in the previous section that the problem of finding a revenue-maximizing allocation can be formulated as an integer program (IP), given polynomial prices. However, the size of the IP scales poorly with the price cardinality: with just 10 items and 5 agents the program requires hundreds of constraints even with just quadratic prices, in the worst case. From a modularity standpoint, it is also undesirable to have a core part of the auction—provisional allocation—depend so closely on price structure.

To address both these issues, we propose a generic approach to revenue maximization based on the idea of restricting the set of possible allocations. Because we no longer ensure $\mathbf{Y}^t \in S(p^t)$, this means that we lose the theoretical efficiency guarantee provided by Proposition 1. It is nonetheless still possible to provide a worst-case approximation guarantee, given below, and our experiments will show that the approach is very effective in practice.

Let $\mathcal{B}_i^t$ be the collection of bundles that have been bid on by agent i up to round t —the collection of “observed” bundles. For convenience, we initialize this collection with the empty bundle in Line 2 of Algorithm 1. We apply a natural restriction: The seller may only allocate to agent i a bundle from this collection, namely a bundle that has been explicitly bid on. More precisely, the seller is restricted to the set $\tilde{\mathcal{F}}^t = \{\mathbf{Y} \in \mathcal{F} : Y_i \in \mathcal{B}_i^t, i \in N\}$ when computing the provisional allocation; note that we use a superscript t because the feasible set now evolves with the rounds. Finding the revenue-maximizing allocation is still an integer programming problem, formulated as follows. For each $i \in N$ and $X \in \mathcal{B}_i^t$, we introduce indicator variable $x_i(X) \in \{0, 1\}$ to denote whether X is allocated to i in the current round.

$$\max_{x \text{ binary}} \sum_{i \in N} \sum_{X \in \mathcal{B}_i^t} p^t(X) x_i(X) \quad (6)$$

$$\text{s.t.} \quad \sum_{i \in N} \sum_{X \in \mathcal{B}_i^t : a \in X} x_i(X) \leq 1 \quad \forall a \in M \quad (7)$$

$$\sum_{X \in \mathcal{B}_i^t} x_i(X) = 1 \quad \forall i \in N \quad (8)$$

As long as the number of rounds stays limited (i.e., convergence is fast enough), this is a tractable problem for

modern IP solvers. The number of constraints stays constant at $m + n$, and the number of variables is at most nt . Importantly, the IP and its Objective (6), in particular, do not depend in any way on the price structure: One only needs to evaluate prices over the observed bundles for each agent. Note that Constraints (8) can always be satisfied as equalities because we ensured that $\emptyset \in \mathcal{B}_i^t$ for each agent i by the way the collections of observed bundles are initialized in Line 2. In Line 7, the history of bidded bundles $\mathcal{B}^{t-1}$ is passed to the SupplyQuery subroutine to accommodate this alternative approach to revenue maximization.

The program itself represents a set packing problem, for which IP methods are particularly effective (Andersson et al. 2000, Sandholm 2006), so we refer to this provisional allocation scheme as *set-packing* allocation. There is now the risk of terminating with an inefficient allocation because condition $\mathbf{Y}^t \in S(p^t)$ of the clearing price definition is no longer confirmed. The following example illustrates how inefficiencies can arise.

Example 3. There are three items $\{a, b, c\}$ and three agents with valuations $(\{a, b\}, 2) \oplus (\{a\}, 1)$, $(\{b, c\}, 2) \oplus (\{b\}, 1)$, and $(\{a, c\}, 2) \oplus (\{c\}, 1)$. The auction is initialized with prices $p^1 = (0, 0, 0)$ and provisional allocation $\mathbf{Y}^1 = (\emptyset, \emptyset, \emptyset)$. The auction uses constant price increment $\eta = 0.5$ and discount $\epsilon = 0.01$. In the first round, under zero prices, the agents best-respond with bundles $\{a, b\}$, $\{b, c\}$, and $\{a, c\}$, respectively. Following the update rule (5), the excess demand for each item is two so item prices are increased to $p^2 = (1, 1, 1)$ given the price increment. At this point, the auction has only observed bids on $\{a, b\}$, $\{b, c\}$, and $\{a, c\}$ from each agent. Under set-packing allocation, it maximizes revenue under the restriction that it can only allocate these bundles. Breaking ties in favor of the first agent (which makes no difference to the overall conclusion), the next provisional allocation is $\mathbf{Y}^2 = (\{a, b\}, \emptyset, \emptyset)$. The assigned bundles maximize the agents' utilities at prices p^2 , so the auction terminates. The value of the final allocation is two, whereas the value of an efficient allocation such as $(\{a\}, \{b\}, \{c\})$ is three. There are multiple efficient allocations, but all of them involve at least one singleton bundle. If the auction had been aware that it could also allocate these singleton bundles, which are not observed during the bidding process, it would have been able to reach an efficient allocation.

Although the true test of set-packing allocation lies in its empirical performance, some mild theoretical guarantees are possible. The following auxiliary result holds independently of price structure, for any iterative auction that follows the template described thus far.

Proposition 2. *Under single-minded agents, an iterative auction with set-packing allocation obtains an efficient allocation upon termination. Under general valuations, it*

obtains an n -approximation to the efficient allocation upon termination.

The result above presumes $\epsilon = 0$. For $\epsilon > 0$, the result holds with the usual adjustment of an additive error of $n\epsilon$ to the efficiency guarantee. With single-minded agents, all bundles that could arise in an efficient allocation are discovered immediately in the first round, so the difficulties illustrated in Example 3 do not arise. The n -approximation for general valuations is not a strong guarantee. Its main purpose is to confirm that under set-packing allocation the efficiency of the auction upon termination cannot be arbitrarily bad. It is also important to stress that the result does not guarantee that the auction will terminate (the “upon termination” state may not be reached)—this depends on the structure of prices and valuations.

We emphasize that the provisional allocation is computed with respect to the current round prices p^t for each bundle $X \in \mathcal{B}_i^t$ and not the price at which X was first or last bid on. Therefore, the provisional allocation $\mathbf{Y}^t$ is just an “offer” on the part of the seller, and there is no commitment on the part of agent i to accept $\mathbf{Y}_i^t$ regardless of its previous bids, unless it explicitly accepts by bidding $X_i = \mathbf{Y}_i^t$ in the current round. However, if our auction were used as a first phase for the CCA, then it would be important to record and carry over the last bid price for each $X \in \mathcal{B}_i^t$ for use in the second phase.

3.4. Extension: Activity Rules

An important feature of real-world clock auctions is the imposition of activity rules designed to curtail “bid sniping” behavior, where bidders wait until the final rounds to place large bids in order to avoid revealing valuation information to their competitors. This goes against the goal of price discovery in a clock auction. Activity rules such as those for the CCA impose constraints that bids should be consistent with straightforward bidding according to *some* underlying valuation function, since true valuations are not publicly known.⁶

Suppose that an agent places a bid on bundle X^t in the current round t , under prices p^t . The bid satisfies the *revealed-preference activity rule* with respect to a previous round $s < t$ if

$$p^t(X^t) - p^t(X^s) \leq p^s(X^t) - p^s(X^s), \quad (9)$$

where X^s is the bundle that the agent bid on in round s , and p^s were the prices in that round. Note that our auction is entirely compatible with applying restrictions of this form as an activity rule: We simply need to be able to evaluate the prices of various bundles. The constraints are valid regardless of price monotonicity. Ausubel and Baranov (2020) have advocated for chaining constraints of the form (9) so that revealed preferences satisfy the generalized axiom of revealed preference (GARP). An

agent's bids satisfy GARP if and only if the following system of inequalities has a solution

$$v^t - p^t(X^t) \geq v^s - p^t(X^s),$$

where s, t range over all rounds, and the v^s, v^t are non-negative variables. Again, verifying that these inequalities have a solution is simply a matter of evaluating the prices of various bundles.

Although price structure has no bearing on the application of revealed-preference activity rules, it does have implications for how much observers can learn about true valuations from the auction bids. Under the stronger GARP activity rule, bids are guaranteed to be rationalized by some valuation function, using a construction originally from Afriat (1967). Using the solution to the inequalities, we form the inferred valuation function

$$v(X) = \min_t \{v^t + p^t(X) - p^t(X^t)\}, \quad (10)$$

where the minimum is taken over all auction rounds t . Note that as the degree of polynomial prices increases, the class of valuations that can be modeled by (10) grows larger. In other words, by restricting the price degree, we limit the amount of information that can be learned about bidder valuations. The fact that iterative auctions can conceal information is one reason they are sometimes favored over their sealed-bid counterparts (Rothkopf et al. 1990), and in this respect, our adaptive-price auction may have advantages over bundle-price alternatives.

4. Adaptive Pricing

The polynomial-price auction described thus far keeps the monomial structure $\mathcal{M}$ fixed across all rounds. In practice, it can be difficult to know a priori whether linear prices will suffice to clear the market. On the other hand, using a fully expressive, m -degree polynomial is impractical for even a moderate number of items. Here we introduce an adaptive version of the polynomial price auction that augments the price structure as the auction progresses. We first develop the theory behind price expansion assuming that each agent i bids its *entire demand set* $D_i(p^t)$ at each round rather than just a *single element* of the demand set as described in Section 3.1. We later explain how the approach can be adapted to handle bidding on a singleton bundle from the demand set at each round, which places a lighter computational burden on bidders.

Although our auction is a member of the subgradient class by the way its price increment is computed, the techniques we develop here for price expansion draw on ideas from primal-dual algorithms. In particular, the “restricted primal” that we introduce in this section

is exactly the restricted primal that one would formulate in a primal-dual auction, but here it is only used to augment price structure (if needed), not to compute a price increment.

4.1. Cutting Planes

The key insight is that expanding the polynomial prices with a new monomial term corresponds to generating a cut for the associated allocation problem, toward computing an integer solution. We begin by formulating the efficient allocation problem as a linear program, where all variables are continuous (as opposed to an integer program as we had previously). We introduce a continuous variable $x_i(X)$ for each $i \in N$ and $X \in \mathcal{B}_i^t$, and we also introduce a continuous variable $y(\mathbf{Y})$ for each $\mathbf{Y} \in \tilde{\mathcal{F}}^t$. Recall that $\mathcal{M}$ represents the current set of monomials in the polynomial prices. An LP formulation of the efficient allocation problem is as follows. With a slight abuse of notation, we write $\mathbf{Y} \supseteq Z$ when there is some Y_i in allocation $\mathbf{Y} = (Y_1, \dots, Y_n)$ such that $Y_i \supseteq Z$.

$$\max_{x \geq 0, y \geq 0} \sum_{i \in N} \sum_{X \in \mathcal{B}_i^t} v_i(X) x_i(X) \quad (11)$$

$$(P) \quad \text{s.t.} \quad \sum_{i \in N} \sum_{X \in \mathcal{B}_i^t: X \supseteq Z} x_i(X) = \sum_{\mathbf{Y} \in \tilde{\mathcal{F}}^t: \mathbf{Y} \supseteq Z} y(\mathbf{Y}) \quad \forall Z \in \mathcal{M} \quad (12)$$

$$\sum_{X \in \mathcal{B}_i^t} x_i(X) = 1 \quad \forall i \in N \quad (13)$$

$$\sum_{\mathbf{Y} \in \tilde{\mathcal{F}}^t} y(\mathbf{Y}) = 1 \quad (14)$$

This LP is closely related to formulations given by Bikhchandani and Ostroy (2002) and Abernethy et al. (2016). To form the dual problem, we associate a variable $w(Z)$ to each constraint in (12), a variable π_i to each agent constraint in (13), and a variable π^s to the constraint over allocations in (14). These dual variables are all unrestricted. The dual to the allocation problem is a pricing problem, captured by the following LP.

$$\min_{\pi, w} \sum_{i \in N} \pi_i + \pi^s \quad (15)$$

$$(D) \quad \text{s.t.} \quad \pi_i \geq v_i(X) - p(X) \quad \forall i \in N, X \in \mathcal{B}_i^t \quad (16)$$

$$\pi^s \geq \sum_{i \in N} p(Y_i) \quad \forall \mathbf{Y} \in \tilde{\mathcal{F}}^t \quad (17)$$

Here, $p(X)$ above is shorthand for

$$p(X) = \sum_{Z \in \mathcal{M}: Z \subset X} w(Z),$$

which is the desired polynomial structure. We refer to the primal allocation LP as (P) and its dual pricing LP as (D). Using standard duality arguments from Bikhchandani and Ostroy (2002), one can show that polynomial prices with the current structure $\mathcal{M}$ can clear the market if and only if (P) has an *integer* optimal solution. The clearing prices can be obtained from the optimal

solution to (D): The dual variables $w(Z)$ associated with each constraint in (12) provide the monomial coefficients. From this perspective, expanding the prices by introducing another monomial Z is equivalent to introducing a constraint in (12), strengthening the formulation and cutting off fractional solutions.

The linear program (P) therefore provides, at least in principle, a test to check whether the current price structure is expressive enough to support an efficient allocation: solve it and check whether the solution is integer. The problem is that the valuations v_i in the objective are not available to the auctioneer in the first place.

4.2. Restricted Primal

To test the clearing ability of the current price structure at any given round, we instead work with an alternative LP called the *restricted primal* (RP). This LP shares the same variables and feasible set (12)–(14) as (P), but its objective only depends on the current demand sets and supply set.

$$(RP) \quad \begin{aligned} & \max_{x \geq 0, y \geq 0} \sum_{i \in N} \sum_{X \in D_i(p^t)} x_i(X) + \sum_{Y \in \tilde{S}^t(p^t)} y(Y) \\ & \text{s.t.} \quad (12)–(14) \end{aligned} \quad (18)$$

Here, the mapping $\tilde{S}^t$ is defined analogously to the supply set in (3), with the restriction that $Y \in \tilde{\mathcal{F}}^t$. (Recall that $\tilde{\mathcal{F}}^t$ is the set of feasible allocations restricted to bid bundles B_i^t for each agent i .) Intuitively, this objective is trying to maximize the trade at the current prices, in the sense of finding an allocation that contains as many demanded bundles as possible, as well as being in the seller's supply set. Given the agents' demand sets at the current prices p^t , this objective can in principle be formulated. There is still the difficulty that there may be too many variables $y(Y)$ (i.e., one for each $Y \in \tilde{\mathcal{F}}^t$) to explicitly enumerate. In practice, we can use the standard technique of column generation to solve the LP without enumerating all of these variables (Bertsimas and Tsitsiklis 1997, chapter 6). This involves finding a variable $y(Y)$ with positive reduced cost, which can be found via an integer program very similar to the one used to compute the provisional allocation. We defer the details of column generation to Appendix B.

The following key result establishes the crucial property of (RP): It allows us to test whether the original formulation (P) has an integer optimal solution, assuming that the current prices p^t have converged to an optimal solution for (D).

Theorem 1. *Let (π, w) be a feasible solution to (D), used to obtain demand and supply sets and form (RP). Then (x, y) and (π, w) are optimal solutions to (P) and (D), respectively, if and only if (x, y) is an optimal solution to (RP) with value $n + 1$.*

Note that the theorem makes no assumptions on whether the primal solution (x, y) is integer. The following corollary leads to a test for price expansion.

Corollary 1. *Form (RP) based on the current prices p^t . If the solution to (RP) has optimal value strictly less than $n + 1$, then prices p^t are not clearing prices. If the solution to (RP) has optimal value $n + 1$ attained at an integer solution, then prices p^t are clearing prices.*

The corollary justifies using the solution to (RP) to test for the need for price expansion. We describe the price expansion policy based on this result first for the case where agents bid their entire demand sets (*demand-set bidding*), and next adapt it to the case of bidding on singleton demanded bundles (*singleton-demand bidding*), which is used in our actual auction implementation.

4.2.1. Demand-Set Bidding. According to Corollary 1, there are three cases:

1. If the optimal value is $n + 1$ and the solution to (RP) is integer, then the auction has converged to clearing prices.

2. If the optimal solution to (RP) has value strictly less than $n + 1$, then the auction proceeds by updating prices as in any other round—The fact that p^t are not clearing prices ensures that the price update is nonzero.

3. The remaining case occurs when the optimal value is $n + 1$, but the solution to (RP) is fractional. In this case, we can make progress by adding extra constraints in (12) in order to cut off the fractional solution, which is equivalent to price expansion, as discussed.

4.2.2. Singleton-Demand Bidding. In this case, it is not possible to directly check whether the optimal value of (RP) is $n + 1$, so the policy is as follows:

1. If the termination criterion holds, then the auction has converged to clearing prices.

2. Otherwise, formulate and solve (RP) using just the singleton bundles as demand sets. If the solution is integer, perform a price update, otherwise cut off the fractional solution using price expansion.

Note that if the termination criterion holds, then the current allocation represents an integer solution to (RP) with value $n + 1$, so case 1 under singleton-demand bidding is equivalent to case 1 under demand-set bidding. Case 2 under singleton-demand bidding can deviate from the demand-set bidding policy in one respect: The optimal solution to (RP) is fractional but has value strictly less than $n + 1$. Under demand-set bidding, this fractional solution would be ignored and prices simply updated. Under singleton-demand bidding, we use a more conservative policy that cuts off fractional solutions via price expansion even when it is not strictly necessary.

4.3. Price Expansion

The preceding results on integrality detection were proved in the context of polynomial prices for concreteness, but in fact the proofs are not tied to this structure and would easily extend to other price structures.⁷ The key property is that the price structure should arise, via duality, from a set of feasibility constraints analogous to (12) relating demand-side variables x_i to supply-side variables y . Day (2018) discusses many interesting alternatives along these lines. Naturally, expanding prices (i.e., generating a cut) does depend on the specific choice of price structure, and we address this now for polynomial prices.

We propose the following method for generating a cut and thereby construct an algorithm to back the `ExpandPrices(·)` call in Line 13 of Algorithm 1. We provide a listing for this in Algorithm 2. We will reference the algorithm steps as we detail our method for cut generation. We first check that the current round

constitutes the beginning of a new epoch (Line 1); otherwise, price expansion is skipped. At the start of a new epoch, the restricted primal (RP) is solved and the result is recorded in x and y (Line 2). Note that the objective here differs slightly from (18) because the algorithm assumes access to only a single demanded bundle per agent, rather than the entire demand set. If x and y are integral (Line 3), then following the singleton-demand bidding policy the price structure is kept the same (Line 4)—The auction will proceed with a price update as usual.

When x or y contain at least one fractional entry we need to expand the prices by finding a constraint of the form (12) that is violated by the current solution. To do so, we first obtain the current set of constraints in (12) with fractional variables, denoted $\mathcal{A} \subseteq \mathcal{M}$ (Line 6):

$$\mathcal{A} = \{Z \in \mathcal{M} : \exists i \in N, X \in \mathcal{B}_i^t \text{ where } X \supseteq Z, x_i(X) \in (0, 1)\} \cup \{Z \in \mathcal{M} : \exists Y \in \tilde{\mathcal{F}}^t \text{ where } Y \supseteq Z, y(Y) \in (0, 1)\}. \quad (19)$$

Algorithm 2 (Price Expansion)

Input: Demanded bundles $\mathbf{X}^t$; Observed bundles $\mathbf{B}^t$; Price monomials $\mathcal{M}$; Price coefficients w^t ; Current round t ; Epoch length ϕ

Output: Updated price monomials and coefficients $(\mathcal{M}, w^t)$

Function `ExpandPrices( $\mathbf{X}^t, \mathbf{B}^t, \mathcal{M}, w^t, t, \phi$ )`:

```

1  if  $t \bmod \phi = 0$  then                                     ▷ Begin new epoch
2  Solve (RP) using column generation:                         ▷ See Appendix B
                                                                ▷ (18) Max trade
    max  $\sum_{i \in N} x_i(X_i^t) + \sum_{Y \in \tilde{\mathcal{F}}^t(p^t)} y(Y)$ 
    s.t.  $\sum_{i \in N} \sum_{X \in \mathcal{B}_i^t : X \supseteq Z} x_i(X) = \sum_{Y \in \tilde{\mathcal{F}}^t : Y \supseteq Z} y(Y) \quad \forall Z \in \mathcal{M}$    ▷ (12) Balance demand & supply
         $\sum_{X \in \mathcal{B}_i^t} x_i(X) = 1 \quad \forall i \in N$                                        ▷ (13) Aggregate demand
         $\sum_{Y \in \tilde{\mathcal{F}}^t} y(Y) = 1$                                            ▷ (14) Aggregate supply

3  if  $x$  and  $y$  are integral then                               ▷ Current structure sufficient
4  | return  $(\mathcal{M}, w^t)$                                            ▷ Price monomials and coefficients
5  end
6   $\mathcal{A} \leftarrow \{Z \in \mathcal{M} : \exists i \in N, X \in \mathcal{B}_i^t \text{ where } X \supseteq Z, x_i(X) \in (0, 1)\} \cup$ 
     $\{Z \in \mathcal{M} : \exists Y \in \tilde{\mathcal{F}}^t \text{ where } Y \supseteq Z, y(Y) \in (0, 1)\}$    ▷ (19) Const. w/ frac. vars.

7   $\mathcal{C}_0 \leftarrow \bigcup_{Z \in \mathcal{A}} \left( \bigcup_{i \in N} \{X \in \mathcal{B}_i^t : x_i(X) > 0, X \supseteq Z\} \cup \{Y|_Z : Y \in \tilde{\mathcal{F}}^t, y(Y) > 0, Y \supseteq Z\} \right)$    ▷ (20) Candidate new monomials

8   $\mathcal{C} \leftarrow \left\{ Z' \in \mathcal{C}_0 : \sum_{i \in N} \sum_{X \in \mathcal{B}_i^t : X \supseteq Z'} x_i(X) \neq \sum_{Y \in \tilde{\mathcal{F}}^t : Y \supseteq Z'} y(Y) \right\}$    ▷ (21) Restrict to monomials violating (12)

9   $\mathcal{C} \leftarrow \text{MonomialSelection}(\mathcal{C})$                                ▷ Select monomials, for example, by (22) (optional)
10  $\mathcal{M} \leftarrow \mathcal{M} \cup \mathcal{C}$                                            ▷ Expand monomial set
11  $w^t(Z') \leftarrow 0 \quad \forall Z' \in \mathcal{C}$                                ▷ Initialize new price coefficients
12 end
13 return  $(\mathcal{M}, w^t)$                                            ▷ Price monomials and coefficients
    
```

Next, we use $\mathcal{A}$ to form a set of candidate monomials for expansion denoted $\mathcal{C}$ (Line 7). To form this set, we first iterate over all constraints that contain fractional variables (corresponding to monomials $Z \in \mathcal{A}$), and from each constraint record the bundles for which there is an associated positive (but not necessarily fractional) demand variable x or supply variable y , according to

$$\mathcal{C}_0 = \bigcup_{Z \in \mathcal{A}} \left(\bigcup_{i \in N} \{X \in \mathcal{B}_i^t : x_i(X) > 0, X \supseteq Z\} \cup \{\mathbf{Y}|_Z : \mathbf{Y} \in \tilde{\mathcal{F}}^t, y(\mathbf{Y}) > 0, \mathbf{Y} \supseteq Z\} \right), \quad (20)$$

where $\mathbf{Y}|_Z$ denotes the unique bundle Y_i in a feasible allocation $\mathbf{Y} \supseteq Z$ such that $Y_i \supseteq Z$. (Such a bundle is unique as bundles in a feasible $\mathbf{Y}$ are pairwise disjoint.) We next restrict to those monomials violating (12), namely those monomials that would cut off the current fractional primal solution (Line 8):

$$\mathcal{C} = \left\{ Z' \in \mathcal{C}_0 : \sum_{i \in N} \sum_{X \in \mathcal{B}_i^t : X \supseteq Z'} x_i(X) \neq \sum_{\mathbf{Y} \in \tilde{\mathcal{F}}^t : \mathbf{Y} \supseteq Z'} y(\mathbf{Y}) \right\}. \quad (21)$$

The set of monomials $\mathcal{C}$ that generate a cut can in principle be empty at this stage of the algorithm. In that case, it is no longer possible to make progress with anonymous polynomial prices, and a switch to personalized prices is needed—we defer the details of this case to the next section. If $\mathcal{C}$ is nonempty, on the other hand, then there may be several monomials $Z' \in \mathcal{C}$ that all generate a cut. We could introduce all of them, or to be more conservative use a heuristic to select one among them. We capture this in the algorithm by the optional call `MonomialSelection`($\cdot$) on Line 9; the only requirement on this subroutine is that it should return at least one bundle $Z' \in \mathcal{C}$. For our evaluation we considered several heuristics: choosing a Z' with minimum degree $|Z'|$; choosing a Z' with maximum degree $|Z'|$; and choosing a Z' with maximum absolute constraint violation in (12) according to

$$\left| \sum_{i \in N} \sum_{X \in \mathcal{B}_i^t : X \supseteq Z'} x_i(X) - \sum_{\mathbf{Y} \in \tilde{\mathcal{F}}^t : \mathbf{Y} \supseteq Z'} y(\mathbf{Y}) \right|. \quad (22)$$

These all performed similarly, so in our simulations in Section 7, we will report on results using the latter heuristic.

The algorithm continues by updating the monomial set (Line 10) and initializing the new price coefficients to zero (Line 11). Finally, it returns the updated price monomials and coefficients (Line 13).

Remark 2. The literature on integer programming offers many cutting-plane techniques to strengthen an LP formulation and achieve integer solutions, such as Gomory cuts (Gomory 1963, Balas et al. 1996, Wolsey and Nemhauser 1999). The issue with using generic

cutting-plane techniques in our context is that the cuts may not be of the form (12), breaking the polynomial price structure in the dual. Our approach is more in the spirit of defining relaxation hierarchies for zero-one programming problems (Sherali and Adams 1990, Laurent 2003), in that the allocation problem (P) represents such a hierarchy as the degree of the polynomial is increased.

4.4. Personalized Pricing

For the sake of simplicity, we have up to now described the adaptive-price auction using anonymous prices only. In some instances, it may not be enough to add extra monomial terms: Personalized prices may be needed to reach market clearing. The next result establishes how this scenario can be detected.

Theorem 2. Assume that the optimal solution (x, y) to (RP) is fractional and has value $n + 1$. If price expansion via Algorithm 2 fails to generate a cut, then anonymous clearing prices do not exist.

According to this theorem, if Algorithm 2 fails to identify a monomial that generates a constraint violation—More precisely, if the state is $\mathcal{C} = \emptyset$ in Line 8—then a switch to personalized prices is needed. The simplest way to achieve this is for the auction to instantiate n copies of the current price function, and this is the approach we use in our simulations.

However, a more gradual approach is possible using polynomial prices. By redefining a monomial as a pair consisting of a bundle of items and a subset of agents (rather than just the former), it is possible to apply price terms to specific agents rather than all of them at once (as in anonymous prices). Specifically, the set of monomials is now a collection $\mathcal{M} \subseteq 2^M \times 2^N$, where each monomial is of the form (Z, L) where Z is a bundle of items and L is a subset of agents. The personalized price of a bundle Z to agent i becomes

$$p_i(Z) = \sum_{(Z', L) \in \mathcal{M} : Z' \subseteq Z, i \in L} w(Z', L). \quad (23)$$

In the cutting plane approach to price expansion, the main change is to replace Constraints (12) in (P) with the more general constraints

$$\sum_{i \in L} \sum_{X \in \mathcal{B}_i^t : X \supseteq Z} x_i(X) = \sum_{\mathbf{Y} \in \tilde{\mathcal{F}}^t : \mathbf{Y}|_Z \supseteq Z, i \in L} y(\mathbf{Y}) \quad \forall (Z, L) \in \mathcal{M}, \quad (24)$$

which by duality leads to polynomial prices of the form (23). If $L = N$ in each monomial, this recovers anonymous prices, whereas if L is always a singleton agent, this yields fully personalized prices. In between, we have prices where certain monomials are shared among subsets of agents, which is potentially a more parsimonious way of capturing personalized prices (using fewer coefficients to describe the price functions for all agents) that we have not encountered elsewhere in the literature.

5. Worked Example

In this section, we demonstrate how the adaptive-price auction proceeds in the setting of Example 1, where there are four single-minded bidders and three items. The first three bidders are interested in combinations of two items, with value 3, and the last is interested in all items, with value 4. Recall that in this example linear prices (and even quadratic prices) cannot clear the market. We use an epoch length of five rounds for price expansion.

Table 1 lists, for each round of the auction, the price of each bidder's desired bundle along with the surplus from purchasing the bundle. When surplus is positive, we mark this in bold, to indicate that the bidder places a bid. Because the bundles all pairwise intersect, choosing an allocation amounts to choosing a single winning bidder in this example. It turns out that the revenue-maximizing allocation always allocates to the last bidder (interested in all items), so we mark its price in bold at each round to emphasize that it represents the provisional allocation. Comparing the bundles that have bolded price or surplus allows one to check whether the termination condition holds (i.e., whether there is balance between demand and supply).

Prices are initially linear, and item prices are initialized to 0.1. In the first round, demand exceeds supply for each item so the item prices increase, and the reverse happens in the second round. This pattern continues with diminishing price increments until round 5

(the chosen epoch length), where the auction attempts price expansion.

5.1. Price Expansion

To perform price expansion, the auction forms and solves the restricted primal (RP), whose main constraints are the balance constraints (12). There is one constraint for each of the current price monomials a, b, c :

$$\begin{aligned}x_1(ab) + x_2(ac) + x_4(abc) &= y(\emptyset, \emptyset, \emptyset, abc) + y(ab, \emptyset, \emptyset, \emptyset) \\&\quad + y(\emptyset, ac, \emptyset, \emptyset) \quad (\text{monomial } a) \\x_1(ab) + x_3(bc) + x_4(abc) &= y(\emptyset, \emptyset, \emptyset, abc) + y(ab, \emptyset, \emptyset, \emptyset) \\&\quad + y(\emptyset, \emptyset, bc, \emptyset) \quad (\text{monomial } b) \\x_2(ac) + x_3(bc) + x_4(abc) &= y(\emptyset, \emptyset, \emptyset, abc) + y(\emptyset, ac, \emptyset, \emptyset) \\&\quad + y(\emptyset, \emptyset, bc, \emptyset) \quad (\text{monomial } c).\end{aligned}$$

To understand these constraints, let us examine the first one for monomial a . Its left-hand side has all x (i.e., demand) variables that contain a , and the right-hand side has all y (i.e., allocation) variables that allocate a . More generally, the monomial could be some combination of items.

The RP also has the more straightforward Constraints (13)–(14). Note that Constraints (13) involve variables of the form $x_i(\emptyset)$, which are not simply auxiliary variables; if the price of a buyer's bundle exceeds its value, then $\emptyset$ is the surplus-maximizing bundle and

Table 1. Progress of Our Adaptive-Price Auction on the Four-Bidder, Three-Item Example 1

Round	Bundle Value	Bidder 1	Bidder 2	Bidder 3	Bidder 4	Price Coefficients			
		ab 3	ac 3	bc 3	abc 4	$w(a)$	$w(b)$	$w(c)$	$w(abc)$
1	Price	0.2	0.2	0.2	0.3	0.1	0.1	0.1	.
	Surplus	2.8	2.8	2.8	3.7				
2	Price	4.2	4.2	4.2	6.3	2.1	2.1	2.1	.
	Surplus	−1.2	−1.2	−1.2	−2.3				
3	Price	2.78	2.78	2.78	4.17	1.39	1.39	1.39	.
	Surplus	0.22	0.22	0.22	−0.17				
4	Price	3.94	3.94	3.94	5.91	1.97	1.97	1.97	.
	Surplus	−0.94	−0.94	−0.94	−1.91				
5	Price	2.94	2.94	2.94	4.41	1.47	1.47	1.47	.
	Surplus	0.06	0.06	0.06	−0.41				
After an epoch of five rounds, the auction tests for price expansion. Monomial abc is generated with a default coefficient of $w(abc) = 0$.									
6	Price	3.84	3.84	3.84	5.76	1.92	1.92	1.92	0.0
	Surplus	−0.84	−0.84	−0.84	−1.76				
7	Price	3.02	3.02	3.02	4.12	1.51	1.51	1.51	−0.41
	Surplus	−0.02	−0.02	−0.02	−0.12				
8	Price	2.26	2.26	2.26	2.6	1.13	1.13	1.13	−0.79
	Surplus	0.74	0.74	0.74	1.4				
9	Price	3.68	3.68	3.68	4.73	1.84	1.84	1.84	−0.79
	Surplus	−0.68	−0.68	−0.68	−0.73				
In the final round, demand balances supply. Note that a negative $w(abc)$ is needed to obtain clearing prices.									
10	Price	3.02	3.02	3.02	3.41	1.51	1.51	1.51	−1.12
	Surplus	−0.02	−0.02	−0.02	0.59				

$x_i(\emptyset)$ would then appear in the objective of the RP. In Constraints (14), the y variables are summed over all feasible allocations, which becomes tedious to write out even in this small example. In practice, the LP needs to be solved using column generation because of the large number of y variables (see Appendix B).

The objective of the RP sums over all surplus-maximizing bundles and revenue-maximizing allocations. In round 5, this corresponds to

$$x_1(ab) + x_2(ac) + x_3(bc) + x_4(\emptyset) + y(\emptyset, \emptyset, \emptyset, abc).$$

The optimal solution to the RP here sets $x_1(ab) = x_1(\emptyset) = 0.5$, $x_2(ac) = x_2(\emptyset) = 0.5$, $x_3(bc) = x_3(\emptyset) = 0.5$, $x_4(\emptyset) = 1$, $y(\emptyset, \emptyset, \emptyset, abc) = 1$, and all remaining variables are set to zero. Because this solution is fractional, we perform price expansion. All the constraints have at least one fractional variable, so the first step is to collect the nonempty monomials corresponding to all nonzero variables (both the x demand variables and y supply variables): $\{ab, ac, bc, abc\}$. Then for each we can test whether generating the monomial cuts off the fractional solution. In this instance, any monomial works. In the specific example shown in Table 1, we apply the heuristic that selects a monomial with the maximum absolute constraint violation (22), leading to the unique choice of monomial abc and the new constraint:

$$x_4(abc) = y(\emptyset, \emptyset, \emptyset, abc) \quad (\text{monomial } abc).$$

Here the left-hand side is zero and the right-hand side is one, for an absolute violation of one, whereas the absolute violation for the other monomials is 0.5. The monomial abc is added to the polynomial prices and appears with an initial coefficient of $w(abc) = 0$ in the next round.

5.2. Termination

The prices continue to adjust to balance supply and demand. In this example, this means that only bidder 4 should demand its bundle. To achieve this and to ensure that the prices of the other bidders' bundles stay above their value of three, $w(abc)$ needs to become negative. In rounds 6 and 7, this price adjustment happens because bidder 4 declines to bid (i.e., it bids $\emptyset$), but it wins in the revenue-maximizing allocation, so according to the price update rule (5), $w(abc)$ is decreased. In round 8, bidder 4 both bids and wins, so $w(abc)$ does not change. In round 9, prices are too high so no bidder places a bid, and all price coefficients are decreased. In round 10, there is finally balance between demand and supply, and the auction terminates at the efficient allocation.

6. Experimental Setup

6.1. Domain Generators

The envisioned advantage of an adaptive-price auction is that it should achieve high efficiency, within a moderate number of rounds, across a range of application

domains and a variety of valuation structures on the part of buyers. Importantly, the valuation structures may not be known a priori. Accordingly, we make use of two different generators to obtain valuation profiles for our experiments. The first is the combinatorial auction test suite (CATS), a standard generator that has been used extensively to test both single-shot and iterative auction designs (Leyton-Brown et al. 2000). The second is the quadratic model originally proposed by de Vries and Vohra (2003), which was motivated by an analysis of bidding in spectrum auctions (Ausubel et al. 1997). The empirical CA literature has focused almost entirely on instances or generators of multiminded valuations for experimental work. We believe this is simply because current CA implementations are tailored for multiminded valuations and not because these valuations are expressive enough to cover all interesting applications. Part of our motivation with adaptive pricing is to encourage work on a wider array of valuation classes.

For all our auction implementations (including the benchmarks), interaction between the seller and buyers occurs solely through a *demand query* interface (Blumrosen and Nisan 2010). In a demand query, the seller quotes a price function to a buyer (using some encoding like bundle or polynomial prices), and the buyer responds with a utility-maximizing bundle at the quoted prices in accordance with straightforward bidding.⁸ Depending on the structure of valuations and prices, this may involve complicated optimization logic on the part of the buyer, but from the perspective of the auction this is a black box. In the adaptive-price auction, demand queries are issued in Line 8 of Algorithm 1.

6.1.1. Multiminded Valuations. CATS generates instances from the class of multiminded valuations, which take the form $(X_1, \alpha_1) \oplus \dots \oplus (X_k, \alpha_k)$. These are also known as XOR valuations (Nisan 2006). CATS allows one to specify the number of items, but not the number of bids k per valuation. Instead, one specifies the total number of bids across all valuations in the generated set of agents. With multiminded valuations, demand queries are conveniently computed, given any price representation, by evaluating the prices of the k bundles and finding the most preferred bundle in the collection. We assume that a multiminded agent only bids on one of its k bundles of interest (i.e., it never includes extraneous items in its bid). In principle any general valuation can be represented as a multiminded valuation, but in practice, we are limited to representations with a polynomial number of bids k .

CATS instances are meant to be economically motivated, representing stylized models of allocation problems involving truck routes, network bandwidth, pollution rights, or real estate. We use the default CATS parameter settings to generate the instances throughout. We consider three generator distributions for our

experiments: regions, arbitrary, and paths. We did not consider the scheduling distribution in CATS because it is known to generate instances that are easy to clear with linear prices (Leyton-Brown et al. 2000).

6.1.2. Quadratic Valuations. In their survey of methods for single-shot CAs, de Vries and Vohra (2003) proposed the *quadratic* model of agent valuations. These valuations are encoded by specifying a value ω_a for each item $a \in M$, as well as a “synergy set” $\Gamma \subseteq M$ and a possibly agent-specific synergy multiplier μ_i . The value of a generic bundle $Z \in \mathcal{Z}$ is evaluated as

$$v_i(Z) = \sum_{a \in Z} \omega_a + \mu_i \sum_{a, b \in \Gamma \cap Z} \omega_a \omega_b. \quad (25)$$

Like quadratic prices, quadratic valuations can be represented by a degree-2 polynomial. The quadratic model is based on the analysis of Ausubel et al. (1997) spectrum auction data from the Federal Communications Commission, which found that complementarity between pairs of licenses is captured by multiplying their populations.

In our experiments, we use $\mu_i = 1/2$ for all agents and hold the size of the synergy set constant at $m/2$ throughout. As others have noted, one concern with quadratic valuations is that an excess of synergy can lead to uninteresting efficient allocations with just one winner (Bichler et al. 2009). To mitigate this, we work with *capped* quadratic valuations of the form $v_i(Z; \kappa) = \max\{v_i(Z') : Z' \subseteq Z, |Z'| \leq \kappa\}$, where $1 \leq \kappa \leq m$ is a parameter bounding the maximum size of a bundle bid on. In our experiments we used $\kappa = m/2$ for this cap. Computing the value of a bundle given a capped quadratic valuation is fast and straightforward by evaluating (25). However, computing the response to a demand query given prices requires mixed-integer programming.

Our choice of quadratic valuations may seem to favor our adaptive-price auction using polynomial prices because a quadratic valuation can be represented with a quadratic polynomial. Indeed, personalized quadratic prices can clear the market under quadratic valuations.⁹ However, this is no longer the case under *capped* quadratic valuations; above the bundle cap size, these valuations rather behave like multiminded valuations (the value of larger bundles is imputed by free disposal). In our experimental implementation, we use the heuristic of maximum constraint violation to generate monomials—which does not a priori favor small or large monomials—to remain agnostic to the kinds of valuations the auction may be facing. This is in keeping with our goal of developing an auction that works well without a priori knowledge of the bidders’ valuation structure.

To summarize, in our experiments we use $\mu_i = 1/2$, a cap of $\kappa = m/2$, and synergy set size $|\Gamma| = m/2$ throughout. The synergy set Γ is drawn uniformly from M , whereas each item coefficient ω_a for $a \in M$ is drawn uniformly from $[0, 1]$, following de Vries and Vohra (2003).

6.2. Baseline Mechanisms

In our experiments we evaluate our adaptive-price auction empirically both in absolute terms and by comparison with a suite of baseline mechanisms. For our first baseline, we use *iBundle* (Parkes 1999). We select *iBundle* not only because it is widely cited in the literature but also because it is an exemplar of a mechanism with pure bundle pricing. Because of the expressiveness of bundle prices, *iBundle* is known to perform well in settings with multiminded combinatorial bidders with strongly peaked preferences, precisely the type of valuations produced by CATS. However, as we will see, *iBundle* may require an extremely large number of rounds in settings where valuations are less peaked, such as the quadratic generator. We consider the *iBundle* variant with personalized bundle prices, which provides a guarantee that approximate clearing prices can always be computed by *iBundle* in a finite (but possibly very large) number of rounds.

For our second baseline we implemented the clock phase of the CCA (Ausubel and Baranov 2017), a design that has been widely used in practice, for example in the 2014 Canadian 700 MHz Auction, which raised more than \$5 Billion CAD (Industry Canada 2015). Because we are interested in studying the properties of iterative pricing mechanisms, we omit the CCA’s last-and-final package phase. Such a phase could be bolted onto any of the mechanisms proposed here and would likely further improve the results. But this would conflate the effect of the final round with the capability of the iterative phase and thereby complicate interpretation of the results. To emphasize that we are focused on the clock phase, we call this mechanism the Linear Clock auction to highlight its price structure. One challenge in implementing an ascending clock auction is to determine the appropriate clock step size. Set the size too small, and the auction will take too many rounds. Set it too large, and the prices may overshoot, resulting in under-demand. For our baseline we implement the approach commonly used in practice that uses an *increasing* step size (Ausubel and Baranov 2017), as a heuristic for finding reasonable increments over the rounds.

Next, we turn to analyzing our own mechanism. Recall the two major components of our design: set-packing allocation and adaptive pricing. We first seek to capture the effect of set-packing in isolation by studying it in simpler implementations that omit the adaptive pricing aspect. To this end, we include a linear-price mechanism that maximizes revenue over the entire set of feasible allocations, which is computationally feasible under linear prices (i.e., a basic auction without either set-packing allocation or adaptive pricing). We compare this to a linear-price mechanism that employs our set-packing allocation (but not adaptive pricing). We call these mechanisms *linear exact* and

linear packing, respectively. We note that the guarantee of Proposition 1 applies to the linear exact variant if it converges, but either variant may fail to converge if linear prices are not expressive enough to clear the market. Finally, we include our actual proposed *adaptive* auction mechanism.

6.3. Parametrization and Computation

All the auctions we implemented make use of an additive discount ϵ on provisional allocation prices; a smaller ϵ should achieve higher efficiency, according to Proposition 1, but possibly at the expense of slower convergence. For *iBundle*, this ϵ also serves as the fixed price increment by design (Parkes 1999). The adaptive-price auction and the linear clock auction use a separate step size schedule controlled by a scaling parameter denoted c . We describe how the ϵ and c parameters are set in our experiments for a fair comparison of the various auctions.

As the valuations generated by the different CATS distributions can vary by orders of magnitude, ϵ and c need to be scaled appropriately for each experimental run. Let V be the median bid value, for multiminded valuations generated by CATS, or the largest bundle value across agents, for quadratic valuations.¹⁰ The actual additive discount used in an auction run is ϵV . We used the same $\epsilon = 0.05$ across all auctions, so that they all share the same efficiency guarantee from Proposition 1.

For our adaptive-price auction (and the linear-price variants) we use the *diminishing* step size schedule $\eta^t = cV/\sqrt{t}$, where recall from the price update rule (5) that η^t is the multiplicative factor applied to excess demand when updating a polynomial coefficient in round t . In the linear clock auction, the price of an item is multiplied by $(1 + c)$ if there is excess demand for the item in the current round; this leads to an *increasing* step size across rounds.

Experiments were run under c parameter settings of 0.25%, 0.5%, 1%, 2%, 4%, 8%, and 16%. When presenting the results, we select the c scaling parameter for each auction type and domain that yields the highest median *efficiency per round* over the runs. (This scaling does not apply to *iBundle*, which sets its step size

based on the discount ϵ .) We found that simply tuning c to optimize median efficiency sometimes led to exceedingly small price increments and therefore a large number of rounds, notably for the linear clock auction. Tuning for efficiency per round leads to a more balanced performance along both efficiency and rounds to convergence. We provide the specific values used in Appendix C.

Furthermore, in our adaptive-price auction, we test for the need for price expansion after a fixed number of rounds called an *epoch*, denoted as ϕ in Line 1 of Algorithm 2. We evaluated epoch lengths of 1, 5, 10, and 20 and report on results using an epoch of 10. For all auctions, we imposed a cap of 1,000 as the maximum number of rounds and a maximum runtime of three hours for each auction run. If an auction run reached either the round or runtime limit, we consider that auction instance to have failed to converge and clear the market, corresponding to the check in Line 15 of Algorithm 1.

We use 30 items for all our auction instances. At this scale it is impractical to encode a general valuation by listing the value of every possible bundle, because there are more than a billion bundles. However, as previously mentioned, CATS provides multiminded valuations of tractable size. We generated 150 bids in all our CATS instances. For quadratic instances, we generated five agents. We ran 100 auction instances for all valuation domains. The auctions were implemented in Python 2.7, and the integer programs were solved using CPLEX 12 on a 12-core 2.93-GHz Xeon machine. We provide open-source access to our implementation at <https://github.com/blubin/Adaptive-Price-CA>.

7. Simulation Results

In this section, we evaluate our adaptive-price auction empirically both in absolute terms and by comparison with the suite of baseline mechanisms described in Section 6.2.

7.1. Clearing

Table 2 shows the percentage of instances that reached clearing prices for each mechanism under each of our four domains of study. Besides the clearing percentage, we also supply the modal reason for any clearing failure,

Table 2. Percentage of Auctions That Cleared for $\epsilon = 0.05$ and 30 Items

Domain	<i>iBundle</i>	Linear clock	Linear exact	Linear packing	Adaptive
Arbitrary	100 (Cleared)	0 (Under Demand)	1 (Max Rounds)	11 (Max Rounds)	99 (Cleared)
Paths	100 (Cleared)	0 (Under Demand)	22 (Max Rounds)	96 (Cleared)	99 (Cleared)
Regions	100 (Cleared)	0 (Under Demand)	30 (Max Rounds)	64 (Cleared)	97 (Cleared)
Quadratic	0 (Max Time)	0 (Under Demand)	91 (Cleared)	65 (Cleared)	99 (Cleared)
Mean	75	0	36	59	98

Notes. The best value in each row is bold. Percentages are taken over the 100 runs in each sample. The modal final state among these runs is provided in parentheses.

detailed below. In our tables, the best-performing auction mechanism for each domain is highlighted in bold.

We observe that *iBundle* is able to clear all of the CATS distributions, which is expected because it was designed for multiminded valuations. However, it is unable to clear any of the quadratic instances as it always times out (recall that we apply a timeout of three hours for auction runs). Quadratic valuations do not have short multiminded representations so the set of bundles on which agents bid becomes very large. This is a known failure mode of bundle-price auctions, which arises even under additive valuations.

Unlike the other auctions, the linear CCA does not necessarily balance supply and demand upon termination; we record this final state as “under demand.” This is not unexpected, because the full CCA includes a supplementary round (recall that the linear clock mechanism is the first phase of the CCA). Here, however, we are interested in the informativeness of the iterative phase prices; these results provide some evidence that they are not as informative as our alternatives. Bichler et al. (2013) have previously observed that the efficiency of the linear CCA can be arbitrarily low.

The linear exact mechanism is able to clear only a modest number of the CATS instances, given its restrictive pricing structure, with most instances running out of rounds (recall that we apply a cap of 1,000 on the number of rounds). It performs much better on the quadratic valuations where it clears 91% of the instances. By contrast, the linear packing auction clears a much larger percentage of the CATS instances. This shows that the allocative “set-packing” restriction we deploy for provisional allocation (for computational reasons) can result in a higher clearing rate.

Finally, we turn to our adaptive auction, which clears at least 97% of the instances across all of the domains. It is never easy to compare mechanisms across multiple domains; in our tables, we provide the mean as a simple summary metric. This is conservative, in that it overweights the CATS domains that play to *iBundle*’s strength. Even with this weight handicap, adaptive still significantly outperforms the other mechanisms’ overall clearing rate, achieving 98% on average.

7.2. Efficiency

We provide statistics on final efficiency levels in Table 3. As expected, *iBundle* is highly efficient on the CATS instances but struggles on quadratic, where it achieves only 74.1% efficiency; overall, it achieves 93.3% efficiency averaged across the four domains. The monotone price update restriction in the linear clock auction causes it to fare poorly in all settings, achieving an average of only 60.1% efficiency. We note that the real-world CCA has a second phase to help ameliorate the effects of inefficiencies in the first phase, but our goal here is to achieve high efficiency in a simpler, single-phase setup.

The linear exact auction shines in the quadratic setting, where it achieves the highest efficiency of all the mechanisms, at 96.7%. However, like linear clock, it has poor performance on the CATS domains, with an average of only 68.0% over these domains. By contrast, the linear packing auction performs significantly better in the CATS domains and worse on the quadratic domain, yielding an average efficiency of 90.6%. Our adaptive auction does not obtain the highest efficiency in any single domain (except paths where it ties linear packing), but its consistency across the different domains makes it the best on average at 97.1%, a significant lift over *iBundle*.

7.3. Rounds and Runtime

For an auction to be practical, it must terminate in a reasonable number of rounds. In practice, there is a tradeoff between having a modest number of rounds through aggressive price updates and having high efficiency. Table 4 reports on the number of rounds required by each auction. Recall that we report results with the step size scaling parameter c set such that each mechanism yields the highest median efficiency per round. As expected, *iBundle* requires only a modest number of rounds in CATS but a large number for quadratic. In fact, as *iBundle* always times out on quadratic instances, its average of 833.8 rounds only indicates how many rounds it was able to complete in three hours, on average. By contrast, the linear exact auction requires only a modest number of rounds in quadratic and a large number in CATS. The linear packing

Table 3. Efficiency of the Auctions for $\epsilon = 0.05$ and 30 Items

Domain	<i>iBundle</i>	Linear clock	Linear exact	Linear packing	Adaptive
Arbitrary	99.6 (0.1)	46.8 (1.5) ^a	49.8 (1.9) ^a	81.0 (1.3) ^a	96.1 (0.4)
Paths	99.8 (0.0)	62.8 (1.0) ^a	87.9 (1.0) ^a	99.9 (0.0)	99.9 (0.0)
Regions	99.4 (0.1)	47.8 (1.5) ^a	66.4 (3.0) ^a	93.9 (1.1)	99.1 (0.2)
Quadratic	74.1 (0.5) ^a	83.3 (0.8) ^a	96.7 (0.6)	87.5 (1.8)	93.2 (0.9)
Mean	93.3	60.1	75.2	90.6	97.1

Notes. The best value in each row is bold. Entries are the mean of 100 runs with standard errors in parentheses.

^aLess than half the instances cleared.

Table 4. Number of Auction Rounds for $\epsilon = 0.05$ and 30 Items

Domain	iBundle	Linear clock	Linear exact	Linear packing	Adaptive
Arbitrary	56.9 (1.3)	24.8 (0.1) ^a	999.4 (0.6) ^a	906.4 (27.6) ^a	159.2 (16.1)
Paths	57.9 (1.2)	81.5 (9.4) ^a	898.9 (21.1) ^a	223.2 (20.8)	127.7 (12.0)
Regions	56.2 (1.1)	25.1 (0.2) ^a	778.6 (36.1) ^a	446.6 (43.1)	147.9 (19.0)
Quadratic	833.8 (0.9) ^a	24.2 (0.2) ^a	115.7 (28.4)	379.8 (46.2)	58.3 (11.1)
Mean	251.2	38.9	698.1	489.0	123.3

Notes. The best value in each row is bold. Entries are the mean of 100 runs with standard errors in parentheses.

^aLess than half the instances cleared.

auction requires a relatively large number across all the domains. Linear clock uses only a very small number of rounds given its monotonic price updates. However, as we saw previously, this low number of rounds betrays very low efficiency. Finally, the adaptive auction requires a relatively modest number of rounds in all domains, requiring only 123.3 on average. This is the second-smallest number of rounds on average, behind only the inefficient linear clock auction.

It is worth pointing out that the number of rounds is also closely related to the runtime needs of the auction. Although there are several nested optimization problems that the auctioneer must solve in the adaptive auction (due to constraint generation), it still achieves the second-smallest overall runtime because of its low number of rounds, needing only 99.1 seconds to clear a market on average, over all instances across the four domains. Because of its low number of rounds and its simplicity, the lowest runtime was achieved by linear clock at 85.3 seconds on average over all instances.

7.4. Revenue

Table 5 shows the revenue generated under each of the auctions and domains, listed as a percentage of the optimal social welfare. Unlike the previous metrics, there is no ideal level for revenue as it simply represents a split of the surplus between the buyers and seller; therefore, we do not bold any numbers in the table. This said, we observe that iBundle produces 76.7% on average over the CATS domains where it clears. This is similar to the 79.5% obtained by adaptive on average over all domains and substantially more revenue than that achieved by any of the other three mechanisms.

7.5. Price Structure

To provide some detail on how the adaptive auction achieves the rates of clearing, efficiency, and convergence we have seen thus far, we examine the structure of the nonlinear prices it constructs. We begin by noting that anonymous prices suffice to clear all the auction runs in all of the domains except for the paths domain. By contrast, 18 runs require personalized prices to clear in paths. Next, Figure 1 provides box plots of the degree of the polynomial prices used, across each of the domains. (For personalized prices, the degree is defined as the maximum degree across all the agent-specific prices.) We observe that the auction uses an average degree of 8.0 across domains. However, in the paths domain, it uses much smaller degree prices. As we have observed, paths often requires personalized prices to clear, and thus for this domain, a high degree of nonlinearity is not the limiting factor.

In addition, we are also interested in knowing how many price terms participants must consider when bidding in the auction, as illustrated in Figure 2. To facilitate comparison, we include only anonymous prices in the figure; in Appendix D, we show that the personalized prices required by 18 instances in the paths domain have a similar term distribution for each bidder. We observe that the adaptive auction requires on average 51.1 terms in the paths domain (it adds more, relatively lower degree terms there) and on average 41.9 terms in the rest of the auctions. We note that 30 of these terms are the standard item prices because there are 30 items in our domains. Therefore, the auction typically adds only a modest number of nonlinear offsets to the linear prices in order to achieve the significant

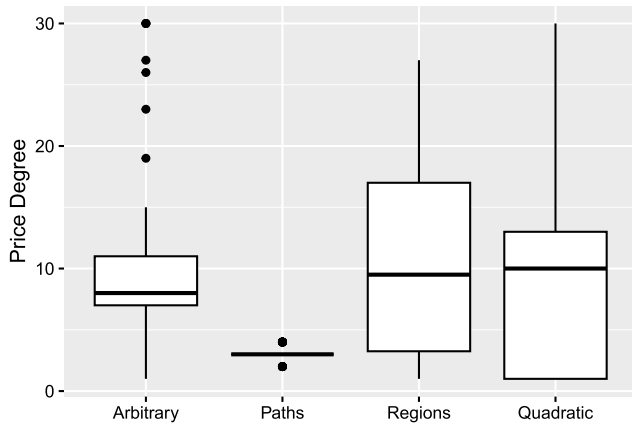
Table 5. Revenue of the Auctions for $\epsilon = 0.05$ and 30 Items

Domain	iBundle	Linear clock	Linear exact	Linear packing	Adaptive
Arbitrary	81.5 (0.5)	30.3 (1.4) ^a	34.0 (2.3) ^a	10.7 (2.8) ^a	84.9 (0.5)
Paths	72.6 (0.6)	34.3 (0.9) ^a	62.8 (1.0) ^a	76.0 (0.6)	78.4 (0.6)
Regions	75.9 (0.6)	29.0 (1.3) ^a	46.6 (3.0) ^a	61.1 (3.3)	80.5 (0.8)
Quadratic	64.1 (0.5) ^a	34.7 (1.1) ^a	74.9 (1.5)	51.5 (3.8)	74.1 (0.9)
Mean	73.5	32.1	54.6	49.8	79.5

Note. Entries are the mean of 100 runs with standard errors in parentheses.

^aLess than half the instances cleared.

Figure 1. Distribution of the Degree of the Final Prices in the Adaptive Auction with $\epsilon = 0.05$ and 30 Items



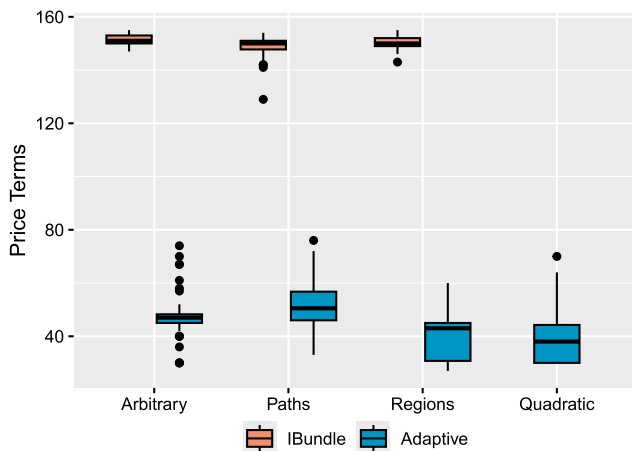
Note. Each box represents 100 samples from the domain.

improvements in performance we have observed. The plot also includes data on the number of bundle-price terms required by *iBundle*. For the CATS domains, this is a reasonable but comparatively large number at 150.2 on average; we omit the quadratic data in the figure, where *iBundle* times out after adding 2,649.1 terms on average.

7.6. Robustness

Evidence from the laboratory indicates that bidders may bid heuristically in package auctions, for example, by restricting their attention to a subset of their bundle space (Scheffel et al. 2012). To provide evidence for the robustness of our mechanism to nonstraightforward bidding, in Appendix E, we repeat the experiments of this section using bidders who fail to respond perfectly

Figure 2. (Color online) Distribution of the Number of Terms in Final Anonymous Prices in the Adaptive Auction and Final Prices in *iBundle*



Note. *iBundle* fails to clear instances in the Quadratic domain, and we therefore omit this data point.

to demand queries, as might happen to bounded-rational bidders or bidders who make a mistake. Specifically, we model bidders as responding randomly between their first-best and second-best bundles. We find that the adaptive auction is highly robust in both the arbitrary and regions domains, where it still clears 90% and 86% of the instances, respectively. By contrast, in the paths domain, many of the multiminded valuations consist of small and largely disjoint bundles. As a consequence, the second-best bundle may have few items in common with the first best, limiting the effectiveness of gradient-based price updates. Accordingly, all the subgradient-based mechanisms we study fail to clear the paths domain under the bidding heuristic considered. However, despite this, the adaptive auction achieves 89.0% efficiency in paths and even higher efficiency of 97.6% and 98.6% in arbitrary and regions, respectively.

7.7. Monotonicity

In Appendix F, we provide both representative price trajectories and distributional information on the fraction of price changes that are increasing (versus decreasing) in the adaptive auction. We observe that prices generally have decreasing trajectories, especially in the early rounds. However, within this overall trend, prices on a given bundle can rise round-on-round, especially in later rounds as the prices converge to market-clearing prices. We leave to future work an in-depth laboratory-based investigation of the efficacy of the mechanism under “behavioral” bidders that may deviate from straightforward bidding, and on the relative advantages of monotone versus nonmonotone price trajectories.

8. Conclusions

This work introduced the first iterative CA design with fully adaptive pricing, along with a practical implementation of the auction using polynomial prices. The design provably detects the need for price expansion to ensure that clearing prices exist and correctly chooses a monomial expansion to guarantee progress, drawing on a connection between price expansion and cutting plane methods.

We conducted a series of experiments to evaluate our adaptive-price auction against linear-price and bundle-price auctions. Our results show that, under multiminded valuations from the CATS generator, the adaptive-price auction is competitive with *iBundle*, which is specifically designed to handle such valuations. Under quadratic valuations, we found that the adaptive-price auction is competitive with its fixed linear-price variants and the clock phase of the CCA, clearing almost all instances with high efficiency. Notably, the benchmarks perform poorly when the domains are switched, failing to clear most instances under the round or time limit, whereas our auction performs well across all of the domains. We also investigated

the robustness of our mechanism to bidders that only respond with approximate best responses rather than exactly and found the mechanism to outperform the benchmarks in two of our domains and match them in the third.

Given its high rates of market clearing across valuation domains, the adaptive-price auction obviates the need for a supplementary sealed-bid phase to correct any under- or over-demand. Our design is core-selecting but does not ensure convergence to a *bidder-optimal* point in the core, and this is a worthwhile aspect for future work (Day and Milgrom 2008). If VCG or bidder-optimal core payments are important for certain applications, our adaptive-price auction is entirely compatible with a second sealed-bid phase as in the CCA because it admits the same kind of revealed preference activity rules. Because the domains where iterative CAs are deployed are high stakes, such as auctions for spectrum, natural resources, or procurement, improving allocative efficiency even modestly through innovations such as those presented here can potentially result in tens or even hundreds of millions of dollars of social welfare gains.

Our mechanism also offers concrete benefits to the auctioneer and participants by reducing the information they require for running and participating in the auction, respectively. Specifically, based on our experimental results, adaptive pricing frees the auctioneer from relying on any a priori knowledge of valuation structures in order to ensure high efficiency and fast convergence. At the same time, adaptive pricing keeps the pricing structure simple, with only a relatively small number of nonlinear terms to consider, which lowers the computational burden faced by bidders. The auction also relies on personalized prices only when needed; in our experiments we found that only one out of the four valuation domains required a switch to personalized prices to clear the market.

We see several avenues for follow-on work. First, given that the adaptive-price auction can handle valuations with complete support, it would be interesting to identify new domains that can take advantage of the clearing capability of this new mechanism. Second, it would be important to understand (e.g., via laboratory experiments) how bidders cope with specific features of our design, like the interpretation of polynomial prices or price nonmonotonicity (especially in the presence of budget constraints). Finally, it would be interesting to consider alternative adaptive-pricing structures beyond polynomial prices. Our theoretical work supports customizing the price structure specifically to the domain, which could enable a more intuitive mapping of aspects of the domain to prices, leading to higher efficiency and faster convergence.

Acknowledgments

This paper is a significantly extended version of work presented at the ACM Conference on Economics and Computation (Lahaie and Lubin 2019); the authors thank the

conference reviewers for comments. The authors thank the editors and anonymous reviewers for insightful comments and advice and Robert Day for feedback. Any remaining errors are our own. The authors are listed alphabetically; all authors contributed equally to this work.

Appendix A. Omitted Proofs

Proof of Proposition 1. The argument is standard, and we provide a proof for completeness. Let $\mathbf{Y}'$ be an arbitrary feasible allocation. By the definition of clearing prices, we have that $v_i(Y_i) - p_i(Y_i) \geq v_i(Y'_i) - p_i(Y'_i) - \epsilon$ for all $i \in N$, and we also have $\sum_{i \in N} p_i(Y_i) \geq \sum_{i \in N} p_i(Y'_i)$. Summing up all these inequalities yields $\sum_{i \in N} v_i(Y_i) \geq \sum_{i \in N} v_i(Y'_i) - n\epsilon$. $\square$

Proof of Proposition 2. For any given round t , let $\tilde{\mathcal{F}}^t$ denote the set of feasible allocations considered by the auction for revenue-maximization purposes, based on the bids placed thus far. With set-packing allocation, the auction selects an allocation from the supply set (3) where $\mathcal{F}$ is replaced by $\tilde{\mathcal{F}}^t$. By reasoning analogous to the proof of Proposition 1, this implies that the auction terminates with an allocation that maximizes efficiency within the restricted feasible set $\tilde{\mathcal{F}}^t$.

Assume that the agents are single minded and interested in bundles $X_1, \dots, X_n$. Each of these bundles is bid on given zero prices in the first round and therefore considered for allocation by the auction in all subsequent rounds. In particular, $\tilde{\mathcal{F}}^t$ contains an efficient allocation in every round beyond the first, which proves the first claim.

Assume now that agents have general valuations. Let $X_1, \dots, X_n$ be the bundles bid on by the agents in the first round. As prices are zero in the first round, it is the case that $v_i(X_i) = \max_{Z \in \mathcal{Z}} v_i(Z)$ for all $i \in N$, by straightforward bidding. Let k be the agent for which $v_k(X_k)$ is maximal. Beyond the first round, $\tilde{\mathcal{F}}^t$ always contains the allocation where agent k obtains X_k , and the others receive $\emptyset$. The efficiency of this allocation is

$$v_k(X_k) \geq \frac{1}{n} \sum_{i=1}^n v_i(X_i) \geq \frac{1}{n} \sum_{i=1}^n v_i(Y_i),$$

where $(Y_1, \dots, Y_n)$ is any efficient allocation among the agents. The first inequality follows from $v_k(X_k) \geq v_i(X_i)$ for all $i \in N$, and the second from $v_i(X_i) \geq v_i(Y_i)$ for all $i \in N$. As the auction terminates with an allocation that is at least as efficient as this n -approximation, this proves the second claim. $\square$

Proof of Theorem 1. Note that given coefficient variables w defining prices p in the dual (D), the setting of the other variables is implied because at the optimum we must have $\pi_i = \max_{X \in \mathcal{B}_i} v_i(X) - p(X)$ for all $i \in N$, and $\pi^s = \max_{Y \in \tilde{\mathcal{F}}^t} \sum_{i \in N} p(Y_i)$. Therefore, in reasoning about (D), it is sufficient to know w only, and the remaining variables are implied.

First recall the complementary slackness conditions. Let (x, y) and (π, w) be feasible solutions to (P) and (D), respectively. They have the same objective value if and only if

$$x_i(X) > 0 \Rightarrow \pi_i = v_i(X) - p_i(X) \Rightarrow X \in D_i(p)$$

$$y(\mathbf{Y}) > 0 \Rightarrow \pi^s = \sum_{i \in N} p_i(Y_i) \Rightarrow \mathbf{Y} \in \tilde{\mathcal{S}}^t(p),$$

where p is the price function defined by coefficients w . By

weak duality, the value of (P) is always no more than the value of (D), so this implies that both solutions are optimal.

Note that (P) and (RP) share the same feasible set, so a feasible solution for one is feasible for the other. To prove the forward direction, let (x, y) be an optimal solution to (RP) of value $n + 1$, formed from feasible (D) solution (π, w) . We have that

$$\sum_{X \in D_i(p)} x_i(X) \leq \sum_{X \in \mathcal{B}_i^t} x_i(X) = 1$$

for each agent i and

$$\sum_{Y \in \tilde{S}^t(p)} y(Y) \leq \sum_{Y \in \tilde{\mathcal{F}}^t} y(Y) = 1.$$

As the objective value of (RP) is $n + 1$, each of the inequalities above must hold with equality. Thus, for any i and $x_i(X) > 0$, the variable must contribute to the objective, and $X \in D_i(p)$. An analogous argument for the seller shows that $y(Y) > 0$ implies $Y \in \tilde{S}^t(p)$. Therefore, (x, y) and (π, w) satisfy complementary slackness and are optimal for (P) and (D), respectively.

For the reverse direction, assume (x, y) and (π, w) are optimal for (P) and (D), respectively. As mentioned, (x, y) is also feasible for (RP). By complementary slackness, $x_i(X) > 0 \Rightarrow X \in D_i(p)$, and therefore

$$\sum_{X \in D_i(p)} x_i(X) = \sum_{X \in \mathcal{B}_i^t} x_i(X) = 1.$$

Similarly, $y(Y) > 0 \Rightarrow Y \in \tilde{S}^t(p)$ and therefore

$$\sum_{Y \in \tilde{S}^t(p)} y(Y) = \sum_{Y \in \tilde{\mathcal{F}}^t} y(Y) = 1.$$

This implies that the objective value of (RP) at solution (x, y) is $n + 1$, which must be optimal. $\square$

Proof of Corollary 1. Let w^t be the current price coefficients and complete the feasible dual solution to (π^t, w^t) as explained in the proof of Theorem 1. Suppose that the optimal solution (x, y) to (RP) has value strictly less than $n + 1$, and assume for the sake of contradiction that p^t are clearing prices. By complementary slackness, this implies that (π^t, w^t) is an optimal solution to (D). Let (x', y') be an optimal solution to (P). Then by Theorem 1, (x', y') is an optimal solution to (RP) with value $n + 1$. But this contradicts the premise that (x, y) was an optimal solution to (RP) with value strictly less than $n + 1$. Thus (π^t, w^t) cannot be optimal for (D), and p^t are not clearing prices.

For the second statement, suppose (x, y) is an integer optimal solution to (RP) with value $n + 1$. Then by Theorem 1, (x, y) and (π^t, w^t) are optimal solutions to (P) and (D), respectively. By complementary slackness, the prices p^t are therefore clearing prices. $\square$

Proof of Theorem 2. Suppose that the optimal solution to (RP) is fractional and that the price expansion process fails to find a cut for all constraints associated with current monomials $Z \in \mathcal{M}$. Then observe that we can introduce all possible monomials $Z \subseteq M$ into $\mathcal{M}$, and solution (x, y) still remains feasible for (RP). By Theorem 1, (x, y) is thus an optimal solution to (P) where constraints for all $Z \subseteq M$ are present. On the other hand, as the provisional allocation does not balance supply and demand, an integer solution to (P) does not exist. The LP formulation (P) with

constraints for all $Z \subseteq M$ is equivalent to the second-order formulation from Bikhchandani and Ostroy (2002). By their theorem 2, anonymous clearing prices exist if and only if their second-order formulation has an integer optimal solution, which completes the result. $\square$

Appendix B. Solving the Restricted Primal

Here we provide the implementation details of solving (RP) via column generation. Recall that it is impractical to enumerate all variables $y(Y)$ for $Y \in \tilde{\mathcal{F}}^t$ at the outset. We initialize the LP with the single variable $y(Y^t)$ associated with the provisional allocation. Once solved, we obtain dual variables $\tilde{w}$, $\tilde{\pi}_i$ (for $i \in N$) and $\tilde{\pi}^s$ in analogy with the variables in (D). Let $\tilde{p}$ be the prices defined by coefficients $\tilde{w}$. We then seek a column (i.e., variable) $y(Y)$ with positive *reduced cost*, defined as

$$\text{reduced cost of } y(Y) = \begin{cases} 1 - \tilde{\pi}^s + \sum_{i \in N} \tilde{p}(Y_i) & \text{if } Y \in \tilde{S}^t(p^t) \\ -\tilde{\pi}^s + \sum_{i \in N} \tilde{p}(Y_i) & \text{otherwise.} \end{cases}$$

If there is no such column, then we have solved (RP). Here $-\tilde{\pi}^s$ is just a constant, so generating a column with largest reduced cost amounts to finding an allocation that maximizes revenue with respect to $\tilde{p}$, with a bonus of one for allocations in $\tilde{S}^t(p^t)$. Recall that we know the optimal revenue with respect to p^t because we computed Y^t ; let r^* be this optimum. To generate a column, we can adapt the IP for revenue maximization, used to compute the provisional allocation. We introduce another binary variable $z \in \{0, 1\}$ and an additional constraint with a sufficiently large constant M . The constraint ensures that the z variable takes on a value of one only if the IP selects an allocation from $\tilde{S}^t(p^t)$.

$$\max_{\substack{x \text{ binary} \\ z \text{ binary}}} \sum_{i \in N} \sum_{X \in \mathcal{B}_i^t} \tilde{p}^t(X) x_i(X) + z - \tilde{\pi}^s \quad (\text{B.1})$$

$$\text{s.t.} \quad \sum_{i \in N} \sum_{X \in \mathcal{B}_i^t : a \in X} x_i(X) \leq 1 \quad \forall a \in M \quad (\text{B.2})$$

$$\sum_{X \in \mathcal{B}_i^t} x_i(X) \leq 1 \quad \forall i \in N \quad (\text{B.3})$$

$$r^* - \sum_{i \in N} \sum_{X \in \mathcal{B}_i^t} p^t(X) x_i(X) \leq M(1 - z) \quad (\text{B.4})$$

If the objective value of this IP is nonpositive, we are done—There is no column with positive reduced cost. Otherwise, we can add column $y(Y^*)$ to the (RP) formulation where Y^* is the allocation defined by the optimal binary $x_i(X)$ variables.

Appendix C. Parameter Settings for c in Reported Results

In Section 6.3, we introduced a c parameter as part of the specification of the pricing step size for the adaptive auction and its fixed linear exact and linear packing variants. Recall that we use a diminishing step size schedule $\eta^t = cV/\sqrt{t}$, where t indexes the round and where V is the median bid value for multiminded valuations generated by CATS or the largest bundle value across agents for quadratic valuations. In Table C.1, we list the values for c

Table C.1. The c Parameter Used in Determining the Step Size in Each Auction

Domain	Linear clock	Linear exact	Linear packing	Adaptive
Arbitrary	0.08	0.08	0.08	0.02
Paths	0.08	0.04	0.08	0.16
Regions	0.08	0.02	0.04	0.02
Quadratic	0.08	0.01	0.02	0.04

Note. For each experimental setup, we report using the value of c that results in the greatest median efficiency per round over the samples in the experiment.

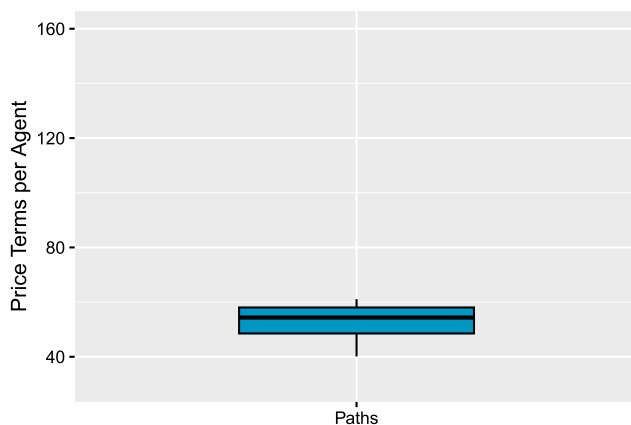
used for the results reported in Section 7; these are the values with the highest median efficiency per round.

Appendix D. Number of Personalized Price Terms in the Paths Domain

Figure 2 in Section 7 shows the number of anonymous price terms in the adaptive auction for several domains. However, for the paths domain, the auction switches to personalized prices in 18 of the 100 runs. Accordingly, for completeness, we provide the number of price terms *per bidder* in Figure D.1.

Appendix E. Robustness to Imperfect Bidder Responses

As a robustness check, we also evaluated how our mechanism performs in the case where bidders fail to respond to demand queries accurately. Such imperfect responses could be due to bounded rationality in the form of informational or computational limitations on the part of bidders, or they could be due to a strategic choice on their part. Bichler et al. (2009) propose several alternative bidding strategies for their own robustness analysis of linear price auctions. Here we adopt their *heuristic* bidding agent, which bids on a randomly selected subset of their most profitable bundles. They implement bidders that bid on 3 out of 10 or 5 out of 20 of the most profitable

Figure D.1. (Color online) Distribution of the Number of Terms per Bidder for the Instances Where the Adaptive Auction Switches to Personalized Prices

Note. There are 18 such instances among the 100 samples from the paths domain with 30 goods and $\epsilon = 0.05$.

Table E.1. Percentage of Auctions That Cleared for $\epsilon = 0.05$ and 30 Items Under Heuristic Bidding

Domain	Linear exact	Linear packing	Adaptive
Arbitrary	0 (max rounds)	5 (max rounds)	90 (cleared)
Paths	0 (max rounds)	0 (max rounds)	0 (max rounds)
Regions	9 (max rounds)	34 (max rounds)	86 (cleared)
Mean	3	13	59

Notes. The best value in each row is in bold. Percentages are taken over the 100 runs in each sample. The modal final state among these runs is provided in parentheses.

Table E.2. Number of Auction Rounds for $\epsilon = 0.05$ and 30 Items Under Heuristic Bidding

Domain	Linear exact	Linear packing	Adaptive
Arbitrary	1,000.0 (0.0) ^a	953.0 (20.6) ^a	291.3 (29.9)
Paths	999.1 (0.7) ^a	1,000.0 (0.0) ^a	1,000.0 (0.0) ^a
Regions	940.7 (20.1) ^a	721.3 (40.3) ^a	296.2 (33.2)
Mean	979.9	891.4	529.2

Notes. The best value in each row is bold. Entries are the mean of 100 runs with standard errors in parentheses.

^aLess than half the instances cleared.

Table E.3. Efficiency of the Auctions for $\epsilon = 0.05$ and 30 Items Under Heuristic Bidding

Domain	Linear exact	Linear packing	Adaptive
Arbitrary	46.6 (1.9) ^a	80.8 (1.2) ^a	97.6 (0.3)
Paths	71.6 (0.6) ^a	91.5 (0.4) ^a	89.0 (0.5) ^a
Regions	56.3 (2.3) ^a	91.5 (1.1) ^a	98.6 (0.4)
Mean	58.1	88.0	95.1

Notes. The best value in each row is bold. Entries are the mean of 100 runs with standard errors in parentheses.

^aLess than half the instances cleared.

bundles. In our auction bidders only bid on singleton bundles at each round, so we adopt the simplest form of this strategy where a bidder randomly bids on one out of its two most profitable bundles.

In Table E.1, we present the clearing results when bidders adopt this heuristic strategy. In both the arbitrary and regions domains, the adaptive mechanism is still able to clear most of the instances. However, in the paths domain, none of the instances clear within the 1,000 round limit. The situation is no better for the baseline

Table E.4. Revenue of the Auctions for $\epsilon = 0.05$ and 30 Items Under Heuristic Bidding

Domain	Linear exact	Linear packing	Adaptive
Arbitrary	23.8 (1.9) ^a	9.2 (2.5) ^a	87.9 (1.3)
Paths	25.2 (0.7) ^a	31.0 (0.9) ^a	28.5 (0.7) ^a
Regions	29.5 (2.5) ^a	50.2 (3.4) ^a	81.6 (1.5)
Mean	26.2	30.1	66.0

Note. Entries are the mean of 100 runs with standard errors in parentheses.

^aLess than half the instances cleared.

mechanisms: Very few of the instances clear in either the linear exact or linear packing auctions. We believe that the difficulty in clearing paths stems from the fact that multiminded valuations from that domain tend to consist of small bundles with very little overlap (often disjoint). For arbitrary and regions, however, relevant bundles are larger and have greater overlap. Therefore, for the latter two domains choosing the second-best bundle still leads to a “gradient-related” price increment (i.e., with positive inner product with the true subgradient), which means that a subgradient auction can still make progress. This is not the case with paths, where the second-best bundle may have nothing in common with the true best-response.

Consistent with the clearing results, in Table E.2, we observe a modest number of rounds in both arbitrary and regions for the adaptive auction, whereas paths uses all 1,000 available rounds in all 100 runs. Both the linear exact and linear packing mechanisms use an extremely large number of rounds in all three domains.

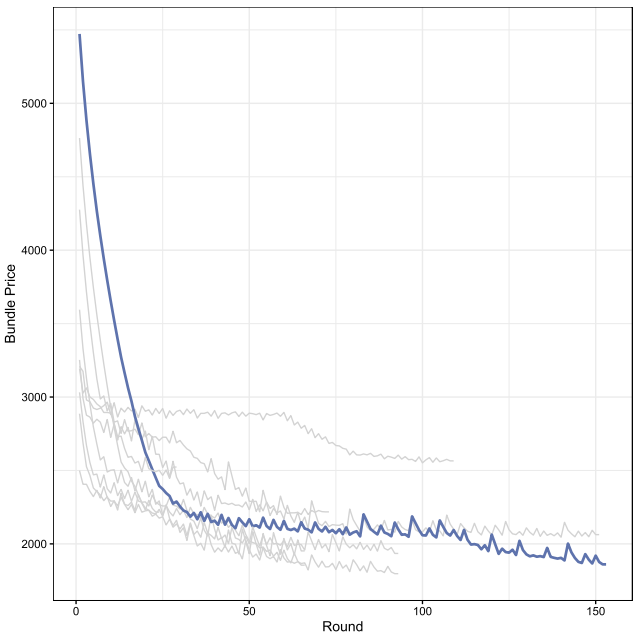
Table E.3 shows the efficiency of each of the mechanisms in each of the domains. Consistent with the above clearing data, we observe high efficiency for the adaptive auction in both the arbitrary and regions domains, where it outperforms both the benchmarks. In paths, despite not clearing, both linear packing and adaptive obtain approximately 90% efficiency, with linear packing edging out adaptive.

Table E.4 shows the revenue of each of the mechanisms in each of the domains. Here we see that adaptive produces high revenue in both arbitrary and regions; it produces less revenue in paths where the clearing condition is not met.

Appendix F. Price Monotonicity

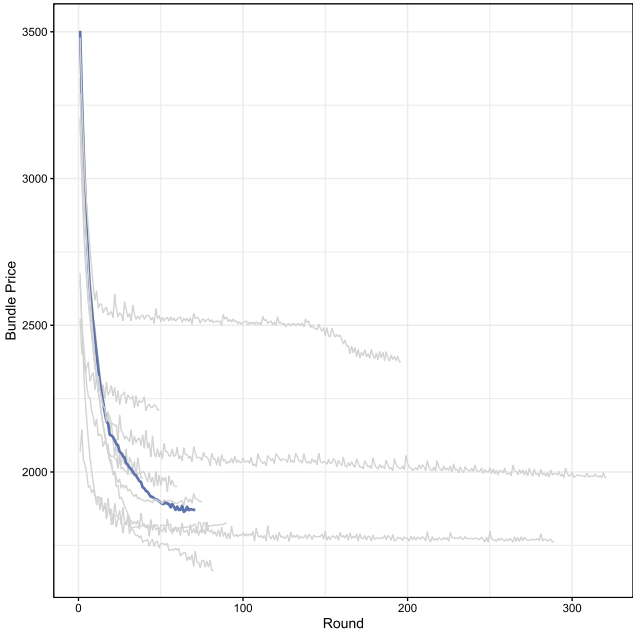
Although the adaptive mechanism does not implement monotone prices, the level of monotonicity observed empirically is

Figure F.1. (Color online) Price Trajectories of the Most Valuable Bundles in 10 Randomly Selected Runs from Among the 100 in the Arbitrary Domain



Note. The trajectory with the largest range is bolded.

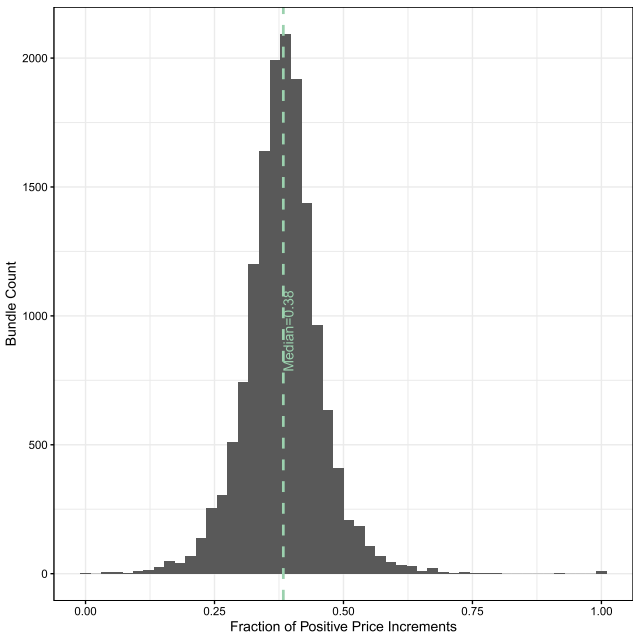
Figure F.2. (Color online) Price Trajectories of the Most Valuable Bundles in 10 Randomly Selected Runs from Among the 100 in the Regions Domain



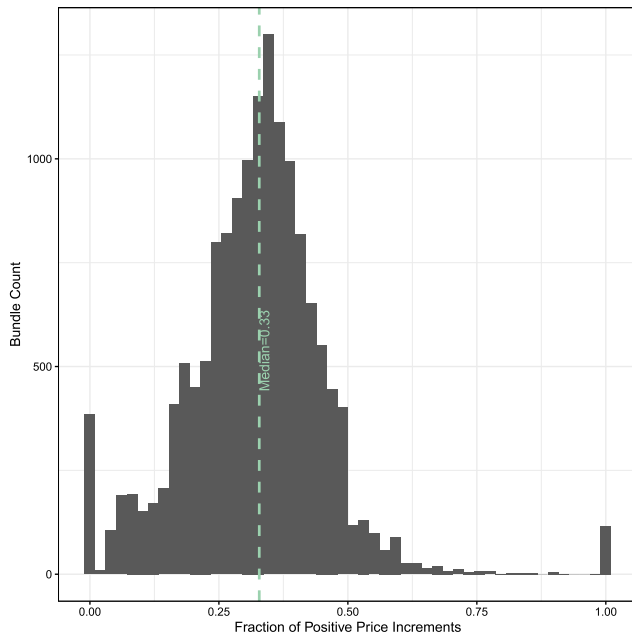
Note. The trajectory with the largest range is bolded.

of potential interest. As an illustration, we provide the price trajectories of the most valuable bundles in 10 randomly selected runs from the arbitrary and regions domains in Figures F.1 and F.2, respectively. We omit paths here due to its use of personalized prices. In each figure, the trajectory

Figure F.3. (Color online) Distribution of Round-on-Round Price Changes That Are Positive in the Arbitrary Domain



Note. The dashed line indicates the median.

Figure F.4. (Color online) Distribution of Round-on-Round Price Changes That Are Positive in the Regions Domain

Note. The dashed line indicates the median.

with the largest range is bolded for easier viewing. We observe that the price trajectories are generally negative (i.e., they approximate descending auctions). However, prices may still modestly rise in any given round.

To quantify this effect, we compute, for each bundle that was bid on, the fraction of rounds that see a positive price change over all runs of arbitrary and regions. Figures F.3 and F.4 plot the distributions of these values. The medians of these distributions are 0.38 and 0.33, respectively, indicating that most price changes are negative.

Endnotes

¹ In the literature these are also called “market-clearing prices” or “competitive equilibrium prices.” We use the term “clearing prices” throughout for the sake of brevity.

² When the price function takes the form of item prices, this is called a Walrasian equilibrium. In this case, the condition $\mathbf{Y} \in S(\mathbf{p})$ is equivalent to ensuring that all items with positive price are allocated, and it is usually stated this way (Gul and Stacchetti 1999).

³ This corresponds to the XOR bidding language (Nisan 2000), but it can equally well be used to represent prices.

⁴ A valuation v_i is super-additive if $v_i(Z \cup Z') \geq v_i(Z) + v_i(Z')$ for $Z \cap Z' = \emptyset$.

⁵ This fact is typically not made explicit in the literature on primal-dual auctions. For their primal-dual auction, de Vries et al. (2007) assume that valuations are integer, which means that using a single-dollar price increment will not overshoot. But in practice this can lead to a very large number of rounds, because the price increment is not calibrated to the actual scale of the valuations, as Schefel et al. (2011) observe in their experimental evaluation of the primal-dual auction.

⁶ Early activity rules for the simultaneous ascending auction for spectrum licenses followed a point-based scheme to encourage

early aggressive bidding (Milgrom 2000). Our auction is incompatible with such rules because it uses non-monotonic prices. However, it is now recognized that point-based rules still allow for undesirable bidding behavior like “parking” (Ausubel and Baranov 2014).

⁷ It is possible to prove the results at the level of an abstract price structure, but this does not provide any additional insights and comes at the cost of substantial extra notation.

⁸ For tie-breaking purposes, the demand query might also include a proposal bundle X , which the agent must return as the response to the query if X maximizes its utility. We use this variant in our implementation, along with the stipulation that the agent may take an ϵ -discount on the price of the proposed bundle X when computing the best-response.

⁹ This follows because the vector of quadratic valuations, taken as personalized prices, represents clearing prices.

¹⁰ V is used for calibration and the median is robust and effective for multiminded valuations. Because quadratic valuations are defined over so many bundles, computing the median is not straightforward, so we use the maximum.

References

- Abernethy J, Lahaie S, Telgarsky M (2016) Rate of price discovery in iterative combinatorial auctions. *Proc. 17th Conf. Econom. Comput.*, 809.
- Afriat SN (1967) The construction of utility functions from expenditure data. *Internat. Econom. Rev.* 8(1):67–77.
- Andersson A, Tenhunen M, Ygge F (2000) Integer programming for combinatorial auction winner determination. Werner B, ed. *Proc. 4th Internat. Conf. Multi-Agent Systems* (IEEE, New York), 39–46.
- Ausubel LM (2006) An efficient dynamic auction for heterogeneous commodities. *Amer. Econom. Rev.* 96(3):602–629.
- Ausubel LM, Baranov OV (2014) Market design and the evolution of the combinatorial clock auction. *Amer. Econom. Rev.* 104(5):446–451.
- Ausubel LM, Baranov OV (2017) A practical guide to the Combinatorial Clock auction. *Econom. J.* 127:F334–F350.
- Ausubel LM, Baranov OV (2020) Revealed preference and activity rules in dynamic auctions. *Internat. Econom. Rev.* 61(2):471–502.
- Ausubel LM, Cramton P (2010) Virtual power plant auctions. *Utility Policy* 18(4):201–208.
- Ausubel LM, Milgrom PR (2002) Ascending auctions with package bidding. *Frontiers Theoret. Econom.* (The Berkeley Electronic Press, Berkeley, CA), 1–42.
- Ausubel LM, Cramton P, Milgrom P (2006) The clock-proxy auction: A practical combinatorial auction design. Cramton P, Shoham Y, Steinberg R, eds. *Combinatorial Auctions* (MIT Press, Cambridge, MA), 115–138.
- Ausubel LM, Cramton P, McAfee RP, McMillan J (1997) Synergies in wireless telephony: Evidence from the broadband PCS auctions. *J. Econom. Management Strategy* 6(3):497–527.
- Balas E, Ceria S, Cornuéjols G, Natraj N (1996) Gomory cuts revisited. *Oper. Res. Lett.* 19(1):1–9.
- Baldwin E, Klemperer P (2019) Understanding preferences: “Demand types”, and the existence of equilibrium with indivisibilities. *Econometrica* 87(3):867–932.
- Bertsimas D, Tsitsiklis JN (1997) *Introduction to Linear Optimization*, vol. 6 (Athena Scientific, Nashua, NH).
- Bichler M, Goeree JK (2017) *Handbook of Spectrum Auction Design* (Cambridge University Press, Cambridge, UK).
- Bichler M, Fux V, Goeree JK (2019) Designing combinatorial exchanges for the reallocation of resource rights. *Proc. Natl. Acad. Sci. USA* 116(3):786–791.
- Bichler M, Shabalin P, Pikovsky A (2009) A computational analysis of linear price iterative combinatorial auction formats. *Inform. Systems Res.* 20(1):33–59.

- Bichler M, Shabalin P, Ziegler G (2013) Efficiency with linear prices? A game-theoretical and computational analysis of the combinatorial clock auction. *Inform. Systems Res.* 24(2):394–417.
- Bikhchandani S, Ostroy JM (2002) The package assignment model. *J. Econom. Theory* 107(2):377–406.
- Bikhchandani S, de Vries S, Schummer J, Vohra RV (2001) Linear programming and Vickrey auctions. *IMA Volume Math. Appl.* 127:75–116.
- Blum A, Jackson J, Sandholm T, Zinkevich M (2004) Preference elicitation and query learning. *J. Machine Learn. Res.* 5(Jun):649–667.
- Blumrosen L, Nisan N (2010) On the computational power of demand queries. *SIAM J. Comput.* 39(4):1372–1391.
- Brero G, Lubin B, Seuken S (2017) Probably approximately efficient combinatorial auctions via machine learning. *Proc. 31st AAAI Conf. Artificial Intelligence* (AAAI Press, Washington, DC), 397–405.
- Brero G, Lubin B, Seuken S (2018) Combinatorial auctions via machine learning-based preference elicitation. Lang J, ed. *Proc. 27th Joint Internat. Conf. Artificial Intelligence* (AAAI Press, Washington, DC), 128–136.
- Candogan O, Ozdaglar A, Parrilo P (2018) Pricing equilibria and graphical valuations. *ACM Trans. Econom. Comput.* 6(1):2:1–2:26.
- Cramton P (2013) Spectrum auction design. *Rev. Industrial Organ.* 42(2):161–190.
- Day R (2018) Linear prices in combinatorial auctions. Bichler M, ed. *Proc. INFORMS Workshop Math. Optim. Market Design* (INFORMS, Catonsville, MD).
- Day R, Milgrom P (2008) Core-selecting package auctions. *Internat. J. Game Theory* 36(3):393–407.
- de Vries S, Vohra RV (2003) Combinatorial auctions: A survey. *INFORMS J. Comput.* 15(3):284–309.
- de Vries S, Schummer J, Vohra RV (2007) On ascending Vickrey auctions for heterogeneous objects. *J. Econom. Theory* 132(1):95–118.
- Goetzendorf A, Bichler M, Shabalin P, Day RW (2015) Compact bid languages and core pricing in large multi-item auctions. *Management Sci.* 61(7):1684–1703.
- Gomory RE (1963) An algorithm for integer solutions to linear programs. Graves RL, Wolfe P, eds. *Recent Advances in Mathematical Programming* (McGraw-Hill, New York), 269–302.
- Gul F, Stacchetti E (1999) Walrasian equilibrium with gross substitutes. *J. Econom. Theory* 87(1):95–124.
- Gul F, Stacchetti E (2000) The English auction with differentiated commodities. *J. Econom. Theory* 92(1):66–95.
- Hohner G, Rich J, Ng E, Reid G, Davenport AJ, Kalagnanam JR, Lee HS, et al. (2003) Combinatorial and quantity-discount procurement auctions benefit Mars, Incorporated and its suppliers. *INFORMS Interfaces* 33(1):23–35.
- Industry Canada (2015) Canadian 700MHz auction. Retrieved May 7, <https://ised-isde.canada.ca/site/spectrum-management-telecommunications/en/spectrum-allocation/auctions/700-mhz-2014/final-results-700-mhz-auction-2014>.
- Kelso AS, Crawford VP (1982) Job matching, coalition formation, and gross substitutes. *Econometrica* 50(6):1483–1504.
- Kwasnig AM, Ledyard JO, Porter D, DeMartini C (2005) A new and improved design for multiobject iterative auctions. *Management Sci.* 51(3):419–434.
- Lahaie S, Lubin B (2019) Adaptive-price combinatorial auctions. Immorlica N, Johari R, eds. *Proc. 20th Conf. Econom. Comp.* (ACM, New York), 749–750.
- Lahaie S, Parkes DC (2004) Applying learning algorithms to preference elicitation. Feigenbaum J, Seltzer M, eds. *Proc. 5th Conf. Electronic Comm.* (ACM, New York), 180–188.
- Lahaie S, Parkes DC (2009) Fair package assignment. Das S, Ostrovsky M, Pennock D, Szymanski B, eds. *Proc. 1st Conf. Auctions Market Mechanisms Appl.* (Springer Nature, London).
- Laurent M (2003) A comparison of the Sherali-Adams, Lovász-Schrijver, and Lasserre relaxations for 0–1 programming. *Math. Oper. Res.* 28(3):470–496.
- Ledyard JO, Olson M, Porter D, Swanson JA, Torma DP (2002) The first use of a combined-value auction for transportation services. *INFORMS Interfaces* 32(5):4–12.
- Lehmann B, Lehmann D, Nisan N (2006) Combinatorial auctions with decreasing marginal utilities. *Games Econom. Behav.* 55(2):270–296.
- Leyton-Brown K, Milgrom P, Segal I (2017) Economics and computer science of a radio spectrum reallocation. Mason JM, Tygar D, eds. *Proc. Natl. Acad. Sci. USA* 114(28):7202–7209.
- Leyton-Brown K, Pearson M, Shoham Y (2000) Toward a universal test suite for combinatorial auction algorithms. Mason JM, Tygar D, eds. *Proc. 2nd Conf. Electronic Comm.* (ACM, New York), 66–76.
- Milgrom P (2000) Putting auction theory to work: The simultaneous ascending auction. *J. Political Econom.* 108(2):245–272.
- Mishra D, Parkes DC (2007) Ascending price Vickrey auctions for general valuations. *J. Econom. Theory* 132(1):335–366.
- Nisan N (2000) Bidding and allocation in combinatorial auctions. Mason JM, Tygar D, eds. *Proc. 2nd Conf. Electronic Comm.* (ACM, New York), 1–12.
- Nisan N (2006) Bidding languages. Cramton P, Shoham Y, Steinberg R, eds. *Combinatorial Auctions* (MIT Press, Cambridge, MA), 400–420.
- Nisan N, Segal I (2006) The communication requirements of efficient allocations and supporting prices. *J. Econom. Theory* 129(1):192–224.
- Paes Leme R (2017) Gross substitutability: An algorithmic survey. *Games Econom. Behav.* 106:294–316.
- Parkes DC (1999) iBundle: An efficient ascending price bundle auction. Feldman S, Wellman M, eds. *Proc. 1st Conf. Electronic Comm.* (ACM, New York), 148–157.
- Parkes DC (2006) Iterative combinatorial auctions. Cramton P, Shoham Y, Steinberg R, eds. *Combinatorial Auctions* (MIT Press, Cambridge, MA), 96–149.
- Pikovsky A, Shabalin P, Bichler M (2006) Iterative combinatorial auctions with linear prices: Results of numerical experiments. Wu KL, ed. *Proc. 8th IEEE Internat. Conf. Enterprise Comput. E-Commerce E-Services* (IEEE Computer Society, Washington, DC), 39.
- Porter D, Rassenti S, Roopnarine A, Smith V (2003) Combinatorial auction design. *Proc. Natl. Acad. Sci. USA* 100(19):11153–11157.
- Rothkopf MH, Teisberg TJ, Kahn EP (1990) Why are Vickrey auctions rare? *J. Political Econom.* 98(1):94–109.
- Sandholm T (2006) Optimal winner determination algorithms. Cramton P, Shoham Y, Steinberg R, eds. *Combinatorial Auctions* (MIT Press, Cambridge, MA), 337–368.
- Sandholm T (2013) Very-large-scale generalized combinatorial multi-attribute auctions: Lessons from conducting \$60 billion of sourcing. Vulkan N, Roth AE, Neeman Z, eds. *The Handbook of Market Design* (Oxford University Press, Oxford, UK).
- Scheffel T, Ziegler G, Bichler M (2012) On the impact of package selection in combinatorial auctions: An experimental study in the context of spectrum auction design. *Experiment. Econom.* 15(4):667–692.
- Scheffel T, Pikovsky A, Bichler M, Guler K (2011) An experimental comparison of linear and nonlinear price combinatorial auctions. *Inform. Systems Res.* 22(2):346–368.
- Schneider S, Shabalin P, Bichler M (2010) On the robustness of nonlinear personalized price combinatorial auctions. *Eur. J. Oper. Res.* 206(1):248–259.
- Sherali HD, Adams WP (1990) A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM J. Discrete Math.* 3(3):411–430.
- Varian HR (1992) *Microeconomic Analysis* (W. W. Norton & Company, New York).
- Walras L (1874) *Éléments d'économie politique pure, ou, Théorie de la richesse sociale* (L. Corbaz, Lausanne, Switzerland).
- Weisstener J, Seuken S (2020) Deep learning-powered iterative combinatorial auctions. Conitzer V, Sha F, eds. *Proc. 34th AAAI Conf. Artificial Intelligence* (AAAI Press, Washington, DC), 2284–2293.
- Wolsey LA, Nemhauser GL (1999) *Integer and Combinatorial Optimization*, vol. 55 (John Wiley & Sons, Hoboken, NJ).