

Fast Iterative Combinatorial Auctions via Bayesian Learning

Gianluca Brero
University of Zurich
brero@ifi.uzh.ch

Sébastien Lahaie
Google Research
slahaie@google.com

Sven Seuken
University of Zurich
seuken@ifi.uzh.ch

Abstract

Iterative combinatorial auctions (CAs) are often used in multi-billion dollar domains like spectrum auctions, and speed of convergence is one of the crucial factors behind the choice of a specific design for practical applications. To achieve fast convergence, current CAs require careful tuning of the price update rule to balance convergence speed and allocative efficiency. Brero and Lahaie (2018) recently introduced a Bayesian iterative auction design for settings with single-minded bidders. The Bayesian approach allowed them to incorporate prior knowledge into the price update algorithm, reducing the number of rounds to convergence with minimal parameter tuning. In this paper, we generalize their work to settings with no restrictions on bidder valuations. We introduce a new Bayesian CA design for this general setting which uses Monte Carlo Expectation Maximization to update prices at each round of the auction. We evaluate our approach via simulations on CATS instances. Our results show that our Bayesian CA outperforms even a highly optimized benchmark in terms of clearing percentage and convergence speed.

1 Introduction

In a combinatorial auction (CA), a seller puts multiple indivisible items up for sale among several buyers who place bids on packages of items. By placing multiple package bids, a buyer can express complex preferences where items are complements, substitutes, or both. CAs have found widespread applications, including for spectrum license allocation (Cramton, 2013), the allocation of TV advertising slots (Goetzendorf et al., 2015), and industrial procurement (Sandholm, 2013).

Practical auctions often employ *iterative* designs, giving rise to *iterative combinatorial auctions*, where bidders interact with the auctioneer over the course of multiple rounds. A well-known example is the combinatorial clock auction (CCA) which has been used by many governments around the world to conduct their spectrum auctions, and it has generated more than \$20 Billion in revenue since 2008 (Ausubel and Baranov, 2017). The CCA consists of two phases: an initial clock phase used for price discovery, followed by a sealed-bid phase where bidders can place additional bids.

One of the key desiderata for a CA is its speed of convergence because each round can involve costly computations

and business modeling on the part of the bidders (Kwasnica et al., 2005; Milgrom and Segal, 2013; Bichler, Hao, and Adomavicius, 2017). Large spectrum auctions, for example, can easily take more than 100 bidding rounds.¹ To lower the number of rounds, in practice many CAs use aggressive price updates (e.g., increasing prices by 5% to 10% every round), which can lead to low allocative efficiency (Ausubel and Baranov, 2017). Thus, the design of iterative CAs that are highly efficient but also converge in a small number of rounds still remains a challenging problem.

1.1 Machine Learning in Auction Design

AI researchers have studied this problem from multiple angles. One early research direction has been to employ machine learning (ML) techniques in preference elicitation (Lahaie and Parkes, 2004; Blum et al., 2004). In a related thread of research, Brero, Lubin, and Seuken (2017, 2018) integrated ML into a CA design, but they used value queries instead of demand queries (prices).

In recent work, Brero and Lahaie (2018) proposed a Bayesian price-based iterative CA that integrates prior knowledge over bidders' valuations to achieve fast convergence and high allocative efficiency. In their design, the auctioneer maintains a model of the buyers' valuations which is updated and refined as buyers bid in the auction and reveal information about their values. The valuation model is used to compute prices at each round to drive the bidding process forward. However, their design has two substantial limitations: (i) it only works with single-minded bidders (i.e., each bidder is only interested in one bundle), and (ii) it only allows for Gaussian models of bidder valuations (even if other models are more suitable and accurate). These limitations are fundamental, because their design relies on both assumptions to obtain an analytical form for the price update rule.

Similarly to Brero and Lahaie (2018), Nguyen and Sandholm (2014, 2016) studied different ways to determine prices in reverse auctions based on probabilistic knowledge on bidders' values. However, in contrast to the setting studied by Brero and Lahaie (2018), in these papers bidders' valuations were not combinatorial, and the auctioneer was allowed to propose personalized prices to each bidder.

¹See for example: <https://www.ic.gc.ca/eic/site/smt-gst.nsf/eng/sf11085.html>

1.2 Overview of our Approach

In this paper, we generalize the approach by Brero and Lahaie (2018). We propose a new, general Bayesian CA that can make use of any model of bidders' valuations and, most importantly, can be applied without any restrictions on the true valuations. At the core of our new auction design is a modular price update rule that only relies on samples from the auctioneer's valuation model, rather than a specific analytic form as used by Brero and Lahaie (2018). We provide a new Bayesian interpretation of the price update problem as computing the most likely clearing prices given the current valuation model. This naturally leads to an Expectation-Maximization (EM) algorithm to compute modal prices, where valuations are latent variables. The key technical contributions to implement EM are (i) a generative process to sample from the joint distribution of prices and valuations (the expectation) and (ii) linear programming to optimize the approximate log likelihood (the maximization).

We evaluate our general Bayesian CA on instances from the Combinatorial Auction Test Suite (CATS), a widely used instance generator for CAs (Leyton-Brown, Pearson, and Shoham, 2000). We first consider single-minded valuations, and compare against the Brero and Lahaie (2018) design. The performance of our general Bayesian CA design matches theirs in terms of clearing percentage and speed of convergence, even though their design is specialized to single-minded bidders. Next, we evaluate our design in settings with general valuations, where we compare it against two very powerful benchmarks that use a subgradient CA design with a non-monotonic price update rule. Our results show that, on average (across multiple CATS domains), our general Bayesian CA outperforms the benchmark auctions in terms of clearing percentage and convergence speed.

Practical Considerations and Incentives. One can view our Bayesian iterative CA as a possible replacement for the clock phase of the CCA. Of course, in practice, many other questions (beyond the price update rule) are also important. For example, to induce (approximately) truthful bidding in the clock phase, the design of good *activity rules* play a major role (Ausubel and Baranov, 2017). Furthermore, the exact payment rule used in the supplementary round is also important, and researchers have argued that the use of the Vickrey-nearest payment rule, while not strategyproof, induces good incentives in practice (Cramton, 2013). Our Bayesian CA, like the clock phase of the CCA, is not strategyproof. However, if our design were used in practice in a full combinatorial auction design, then we envision that one would also use activity rules, and suitably-designed payment rules, to induce good incentives. For this reason, we consider the incentive problem to be orthogonal to the price update problem. Thus, for the remainder of this paper, we follow prior work (e.g., Parkes (1999)) and assume that bidders follow myopic best-response (truthful) bidding throughout the auction.

2 Preliminaries

The basic problem solved by an iterative combinatorial auction is the allocation of a set of items, owned by a seller,

among a set of buyers who will place bids for the items during the auction. Let m be the number of items and n be the number of buyers. The key features of the problem are that the items are indivisible and that bidders have preferences over sets of items, called bundles. We represent a bundle using an m -dimensional indicator vector for the items it contains and identify the set of bundles as $\mathcal{X} = \{0, 1\}^m$. We represent the preferences of each bidder i with a non-negative valuation function $v_i : \mathcal{X} \rightarrow \mathbb{R}_+$ that is private knowledge of the bidder. Thus, for each bundle $x \in \mathcal{X}$, $v_i(x)$ represents the willingness to pay, or *value*, of bidder i for obtaining x . We denote a generic valuation profile in a setting with n bidders as $v = (v_1, \dots, v_n)$. We assume that bidders have no value for the null bundle, i.e., $v_i(\emptyset) = 0$, and we assume free disposal, which implies that $v_i(x') \geq v_i(x)$ for all $x' \geq x$.

At a high level, our goal is to design a combinatorial auction that computes an allocation that maximizes the total value to the bidders. An *iterative* combinatorial auction proceeds over rounds, updating a provisional allocation of items to bidders as new information about their valuations is obtained (via the bidding), and updating prices over the items to guide the bidding process. Accordingly, we next cover the key concepts and definitions around allocations and prices.

An *allocation* is a vector of bundles, $a = (a_1, \dots, a_n)$, with a_i being the bundle that bidder i obtains. An allocation is *feasible* if it respects the supply constraints that each item goes to at most one bidder.² Let $\mathcal{F} \subset \mathcal{X}^n$ denote the set of feasible allocations. The total value of an allocation a , given valuation profile v , is defined as

$$V(a; v) = \sum_{i \in [n]} v_i(a_i), \quad (1)$$

where the notation $[n]$ refers to the index set $\{1, \dots, n\}$. An allocation a^* is *efficient* if $a^* \in \arg \max_{a \in \mathcal{F}} V(a; v)$. In words, the allocation is efficient if it is feasible and maximizes the total value to the bidders.

An iterative auction maintains *prices* over bundles of items, which are represented by a price function $\theta : \mathcal{X} \rightarrow \mathbb{R}_+$ assigning a price $\theta(x)$ to each bundle $x \in \mathcal{X}$. Even though our design can incorporate any kind of price function θ , our implementations will only maintain prices over items, which are represented by a non-negative vector $p \in \mathbb{R}_+^m$; this induces a price function over bundles given by $\theta(x) = \sum_{j \in [m]} p_j x_j$. Item prices are commonly used in practice as they are very intuitive and simple for the bidders to parse (see, e.g., Ausubel et al. (2006)).³

Given bundle prices θ , the *utility* of bundle x to bidder i is $v_i(x) - \theta(x)$. The bidder's *indirect utility* at prices θ is

$$U(\theta; v_i) = \max_{x \in \mathcal{X}} \{v_i(x) - \theta(x)\} \quad (2)$$

i.e., the maximum utility that bidder i can achieve by choosing among bundles from \mathcal{X} .

²We assume that there is one unit of each item for simplicity, but our work extends to multiple units without complications.

³We emphasize that, although the framework generalizes conceptually to any kind of price function θ , complex price structures may bring additional challenges from a computational standpoint.

On the seller side, the *revenue* of an allocation a at prices θ is $\sum_{i \in [n]} \theta(a_i)$. The seller's *indirect revenue* function is

$$R(\theta) = \max_{a \in \mathcal{F}} \left\{ \sum_{i \in [n]} \theta(a_i) \right\} \quad (3)$$

i.e., the maximum revenue that the seller can achieve among all feasible allocations.

Market Clearing. We are now in a position to define the central concept in this paper.

Definition 1. *Prices θ are clearing prices if there exists a feasible allocation $a \in \mathcal{F}$ such that, at bundle prices θ , a_i maximizes the utility of each bidder $i \in [n]$, and a maximizes the seller's revenue over all feasible allocations.*

We say that prices θ support an allocation a if the prices and allocation satisfy the conditions given in the definition. The following important facts about clearing prices follow from linear programming duality (see Bikhchandani and Ostroy (2002) as a standard reference for these results).

1. The allocation a supported by clearing prices θ is efficient.
2. If prices θ support some allocation a , they support every efficient allocation.
3. Clearing prices minimize the following objective function:

$$W(\theta; v) = \sum_{i \in [n]} U(\theta; v_i) + R(\theta). \quad (4)$$

The first fact clarifies our interest in clearing prices: they provide a certificate for efficiency. An iterative auction can update prices and check the clearing condition by querying the bidders, thereby solving the allocation problem without necessarily eliciting the bidders' complete preferences. The second fact implies that it is possible to speak of clearing prices without specific reference to the allocation they support. The interpretation of clearing prices as minimizers of (4) in the third fact will be central to our Bayesian approach.⁴

Clearing Potential. By linear programming duality (Bikhchandani et al., 2001), we have

$$W(\theta; v) \geq V(a; v)$$

for all prices θ and feasible allocations a , and the inequality is tight if and only if θ are clearing and a is efficient. In the following, we will therefore make use of a "normalized" version of (4):

$$\hat{W}(\theta; v) = \left(W(\theta; v) - V(a^*; v) \right) \geq 0. \quad (5)$$

It is useful to view (5) as a potential function that quantifies how close prices θ are to clearing prices; the potential can reach 0 only if there exist clearing prices. We refer to function (5) as the *clearing potential* for the valuation profile v which will capture, in a formal sense, how likely a price function θ is to clearing the valuation profile v within our Bayesian framework.

⁴We emphasize that none of the results above assert the *existence* of clearing prices of the form $\theta(x) = \sum_{j \in [m]} p_j x_j$. The objective in (4) is implicitly defined in terms of item prices p , and for general valuations there may be no item prices that satisfy the clearing price condition (Gul and Stacchetti, 2000).

Algorithm 1: Bayesian Auction Framework

- 1 **Input:** Prior beliefs $Q^0(v)$.
 - 2 $\theta^0 =$ initial prices, $t = 0$.
 - 3 **repeat**
 - 4 $t \leftarrow t + 1$.
 - 5 Observe each bidder i 's demanded bundle b_i^t at θ^{t-1} .
 - 6 Compute revenue-maximizing allocation a^t at θ^{t-1} .
 - 7 **Belief update:** Use each bidder i 's demand b_i^t and $Q^{t-1}(v)$ to derive $Q^t(v)$.
 - 8 **Price update:** Use $Q^t(v)$ to derive new prices θ^t .
 - 9 **until** $a_i^t = b_i^t$ for each bidder $i \in [n]$.
-

3 The Bayesian Auction Framework

We now describe the Bayesian auction framework introduced by Brero and Lahaie (2018) (see Algorithm 1). At the beginning of the auction, the auctioneer has a prior belief over bidder valuations which is modeled via the probability density function $Q^0(v)$. First, some initial prices θ^0 are quoted (Line 2). It is typical to let θ^0 be "null prices" which assign price zero to each bundle. At each round t , the auctioneer observes the demand b_i^t of each bidder i at prices θ^{t-1} (Line 5), and computes a revenue maximizing allocation a^t at prices θ^{t-1} (Line 6). The demand observations are used to update the beliefs $Q^{t-1}(v)$ to $Q^t(v)$ (Line 7), and new prices θ^t reflecting new beliefs are quoted (Line 8). This procedure is iterated until the revenue maximizing allocation matches bidders' demand (Line 9), which indicates that the elicitation has determined clearing prices.

In this paper, we build on the framework introduced by Brero and Lahaie (2018) and generalize their approach to (i) handle any kind of priors and (ii) apply it to settings with no restrictions on the bidders' valuations. This requires completely new instantiations of the *belief update rule* and the *price update rule*, which are the main contributions of our paper, and which we describe in the following two sections.

4 Belief Update Rule

In this section, we describe our belief modeling and updating rule, based on Gaussian approximations of the belief distributions (which proved effective in our experiments). We emphasize, however, that the belief update component of the framework is modular and could accommodate other methods like expectation-propagation or non-Gaussian models. A complete description of the rule is provided in Algorithm 2.

The auctioneer first models the belief distribution via the probability density function $Q(v) = \prod_{i=1}^n Q_i(v_i)$. As the rounds progress, each bidder bids on a finite number of bundles (at most one new bundle per round). Let B_i be the set of bundles that bidder i has bid on up to the current round. We model Q_i as

$$Q_i(v_i) = \prod_{b_i \in B_i} Q_i(v_i(b_i)). \quad (6)$$

Note that $Q_i(\cdot)$ assigns equal probability to any two valuations that assign the same values to bundles in B_i . We model

Algorithm 2: Belief Update Rule

```

1 Input: Beliefs  $Q^{t-1}(v)$ , demand  $b_i^t$  of each bidder
    $i \in [n]$ .
2 foreach  $i \in [n]$  do
3   if  $b_i^t \neq \emptyset$  then
4     if  $b_i^t \notin B_i$  then
5        $B_i \leftarrow B_i \cup \{b_i^t\}$ .
6        $Q_i^{t-1}(v_i(b_i^t)) = Q(v^0(b_i^t))$ .
7     end
8      $Q_i^t(v_i(b_i^t)) \approx$ 
9        $\Phi(\beta(v_i(b_i^t) - \theta^{t-1}(b_i^t))) \cdot Q_i^{t-1}(v_i(b_i^t))$ .
10    foreach  $b_i \in B_i \setminus \{b_i^t\}$  do
11       $Q_i^t(v_i(b_i)) = Q_i^{t-1}(v_i(b_i))$ .
12    end
13  else
14    foreach  $b_i \in B_i$  do
15       $Q_i^t(v_i(b_i)) \approx$ 
16         $\Phi(\beta(\theta^{t-1}(b_i) - v_i(b_i))) \cdot Q_i^{t-1}(v_i(b_i))$ .
17    end
18  end
19 Output: Updated beliefs  $Q^t(v)$ .

```

the density over each $b_i \in B_i$ using a Gaussian:

$$Q_i(v_i(b_i)) = \mathcal{N}(\mu_i(b_i), \sigma_i(b_i)), \quad (7)$$

where $\mathcal{N}(\mu, \sigma)$ denotes the density function of the Gaussian distribution with mean μ and standard deviation σ . By keeping track of separate, independent values for different bundles bid on, the auctioneer is effectively modeling each bidder's preferences using a multi-minded valuation. However, as this is just a model, this does not imply that the bidders' true valuations are multi-minded over a fixed set of bundles.

We now describe how the auctioneer updates $Q^{t-1}(v)$ to $Q^t(v)$ given the bids observed at round t . We assume the auctioneer maintains a Gaussian distribution with density $Q(v^0(x)) = \mathcal{N}(\mu^0(x), \sigma^0(x))$ over the value a generic bidder may have for every bundle $x \in \mathcal{X}$. To update beliefs about bidder valuations given their bids, the auctioneer needs a probabilistic model of buyer bidding. According to myopic best-response bidding, at each round t , bidder i would report a utility-maximizing bundle $b_i^t \in \mathcal{X}$ at current prices θ^t . In other words, the buyer would bid on b_i^t with probability 1 if the bundle maximizes her utility at price θ^t , and 0 otherwise (ties can be broken arbitrarily). This kind of bidding model is incompatible with Gaussian modeling because it contradicts full support: all bundle values $v_i(b_i^t) < \theta(b_i^t)$ must have probability 0 in the posterior. To account for this, we relax the sharp myopic best-response model to probit best-response, a common random utility model under which the probability of bidding on a bundle is proportional to its utility (Train, 2009). Specifically, we set the probability that bidder i bids on b_i^t at prices θ^{t-1} proportional to

$$\Phi(\beta(v_i(b_i^t) - \theta^{t-1}(b_i^t))), \quad (8)$$

where Φ is the cumulative density function of the standard Gaussian distribution and $\beta > 0$ is a scalar parameter that controls the extent to which Φ approximates myopic best-response bidding. As $\beta \rightarrow \infty$, probability mass concentrates on the highest-utility bundle.

Given a bid on bundle $b_i^t \neq \emptyset$ in round t , the auctioneer first records the bundle in B_i if not already present (Line 5) and sets $Q_i^{t-1}(v_i(b_i^t))$ to $Q(v^0(b_i^t))$ (Line 6). The belief distribution over value $v_i(b_i^t)$ is updated to

$$\begin{aligned} Q_i^t(v_i(b_i^t)) &= \mathcal{N}(\mu_i^t(b_i^t), \sigma_i^t(b_i^t)) \\ &\approx \Phi(\beta(v_i(b_i^t) - \theta^{t-1}(b_i^t))) \cdot Q_i^{t-1}(v_i(b_i^t)) \\ &= \Phi(\beta(v_i(b_i^t) - \theta^{t-1}(b_i^t))) \cdot \mathcal{N}(\mu_i^{t-1}(b_i^t), \sigma_i^{t-1}(b_i^t)) \end{aligned}$$

(Line 8). To approximate the right-most term with a Gaussian, we use simple moment matching, setting the mean and variance of the updated Gaussian to the mean and variance of the right-hand term, which can be analytically evaluated for the product of the Gaussian cumulative density function Φ and probability density function \mathcal{N} (see for instance Williams and Rasmussen (2006)). This is a common online Bayesian updating scheme known as *assumed density filtering*, a special case of expectation-propagation (Opper and Winther, 1998; Minka, 2001).

If $b_i^t = \emptyset$, then we update the value of *every* bundle $b_i \in B_i$ with an analogous formula (Line 14), except that the probit term is replaced with

$$\Phi(\beta(\theta^{t-1}(b_i) - v_i(b_i))) \quad (9)$$

to reflect the fact that declining to bid indicates that $v_i(b_i) \leq \theta^{t-1}(b_i)$ for each bundle $b_i \in B_i$.

5 Price Update Rule

In this section, we describe our price update rule. A complete description of the rule is provided in Algorithm 3. The key challenge we address in this section is how to derive ask prices from beliefs over bidders' valuations. We transform this problem into finding the mode of a suitably-defined probability distribution over prices, and we then develop a practical approach to computing the mode via Monte Carlo Expectation Maximization.

We seek a probability density function $P(\theta)$ over prices whose maxima are equal to those prices that will most likely be clearing under $Q(v)$. As a first attempt, consider an induced density function over clearing prices as given by

$$P(\theta) \propto \int_v \mathbf{1}\{\hat{W}(\theta; v) = 0\} Q(v) dv, \quad (10)$$

with $P(\theta, v) \propto \mathbf{1}\{\hat{W}(\theta; v) = 0\} Q(v)$ as the associated joint density function. Recall that $\hat{W}(\theta; v) = 0$ if and only if prices θ are clearing for valuations v . Thus, under function (10), prices θ get assigned all the probability density of the configurations v for which they represent clearing prices.

Although this approach is natural from a conceptual standpoint, it may lead to problems when the Bayesian auction uses specific price structures (e.g., item prices) that cannot clear any valuation in the support of $Q(v)$. It is then useful to introduce price distributions such that, for each price function

θ , $P(\theta, v) > 0$ for all possible configurations v . To obtain a suitable price density function we approximate $P(\theta)$ with

$$P_\lambda(\theta) \propto \int_v e^{-\lambda \hat{W}(\theta; v)} Q(v) dv. \quad (11)$$

The approximation is motivated by the following proposition.

Proposition 1. *Assume that $Q(v)$ allows us to define a probability density function over prices via Equation (10). Then, for every price function θ ,*

$$\lim_{\lambda \rightarrow \infty} P_\lambda(\theta) = P(\theta). \quad (12)$$

Proof. We prove the convergence of $P_\lambda(\theta)$ to $P(\theta)$ by separately showing the convergence of the numerator $\int_v e^{-\lambda \hat{W}(\theta; v)} Q(v) dv$ to $\int_v \mathbf{1}\{\hat{W}(\theta; v) = 0\} Q(v) dv$ and of the normalizing constant $\int_\theta \int_v e^{-\lambda \hat{W}(\theta; v)} Q(v) dv d\theta$ to $\int_\theta \int_v \mathbf{1}\{\hat{W}(\theta; v) = 0\} Q(v) dv d\theta$. We will only show the convergence of the numerator, as the convergence of the normalizing constant follows from very similar reasoning.

Given that $\hat{W}(\theta; v) \geq 0$ for any θ and $\hat{W}(\theta; v) = 0$ only if θ is clearing for v , we have that, for each v, θ ,

$$\lim_{\lambda \rightarrow \infty} e^{-\lambda \hat{W}(\theta; v)} Q(v) = \mathbf{1}\{\hat{W}(\theta; v) = 0\} Q(v). \quad (13)$$

To achieve convergence in the integral form, we note that, as $e^{-\lambda \hat{W}(\theta; v)}$ varies between 0 and 1, $e^{-\lambda \hat{W}(\theta; v)} Q(v)$ is bounded by the integrable probability density function $Q(v)$. This allows us to obtain convergence in the integral form via Lebesgue's Dominated Convergence Theorem. \square

We can now interpret $P_\lambda(\theta)$ as the marginal probability density function of

$$P_\lambda(v, \theta) \propto e^{-\lambda \hat{W}(\theta; v)} Q(v). \quad (14)$$

The standard technique for optimizing density functions like (11), where latent variables are marginalized out, is Expectation Maximization (EM). The EM algorithm applied to P_λ takes the following form:

- **E step:** At each step τ , we compute

$$\mathbb{E}_{P_\lambda(v|\theta^{\tau-1})} \left[\log P_\lambda(v, \theta) \right],$$

where

$$P_\lambda(v | \theta) = \frac{e^{-\lambda \hat{W}(\theta; v)} Q(v)}{\int_{v'} e^{-\lambda \hat{W}(\theta; v')} Q(v') dv'}. \quad (15)$$

- **M step:** Compute new prices

$$\theta^\tau \in \arg \max_\theta \mathbb{E}_{P_\lambda(v|\theta^{\tau-1})} \left[\log P_\lambda(v, \theta) \right].$$

In general, it may be infeasible to derive closed formulas for the expectation defined in the E step. To overcome this issue, we use the Monte Carlo version of the EM algorithm introduced by Wei and Tanner (1990). For the E step, we provide a sampling method that correctly samples from $P_\lambda(v|\theta)$, the conditional probability density function of valuations obtained from (14). For the M step, we use linear programming to optimize the objective, given valuation samples.

Algorithm 3: Price Update Rule

```

1 Input: Current beliefs  $Q(v)$ .
2  $\theta^0 =$  initial prices,  $\tau = 0$ .
3 repeat
4    $\tau \leftarrow \tau + 1$ .
5   foreach  $k \in [\ell]$  do
6     repeat
7       Set resample = 0.
8       Draw  $v^{(k)}$  from  $Q(v)$ .
9       Set resample = 1 with probability
           $1 - e^{-\lambda \hat{W}(\theta^{\tau-1}; v^{(k)})}$ .
10    until resample = 0.
11  end
12  Compute  $\theta^\tau \in \arg \min_\theta \sum_{k \in [\ell]} \hat{W}(\theta; v^{(k)})$ .
13 until  $\|\theta^\tau - \theta^{\tau-1}\| / \|\theta^{\tau-1}\| < \varepsilon$ .
14 Output: Prices  $\theta^\tau$ .

```

Monte Carlo EM. Our Monte Carlo EM algorithm works as follows: At each step τ ,

- Draw samples $v^{(1)}, \dots, v^{(\ell)}$ from $P_\lambda(v|\theta^{\tau-1})$ (Lines 5-11).
- Compute new prices

$$\begin{aligned} \theta^\tau &\in \arg \max_\theta \frac{1}{\ell} \sum_{k \in [\ell]} \log P_\lambda(v^{(k)}, \theta) \\ &\in \arg \max_\theta \frac{1}{\ell} \sum_{k \in [\ell]} \log Q(v^{(k)}) - \lambda \hat{W}(\theta; v^{(k)}) \\ &\in \arg \min_\theta \sum_{k \in [\ell]} \hat{W}(\theta; v^{(k)}) \end{aligned}$$

(Line 12). Note that each θ^τ can be derived via linear programming. Indeed, using equation (5), we have that the minimizers of $\sum_{k \in [\ell]} \hat{W}(\theta; v^{(k)})$ correspond to the minimizers of

$$\sum_{k \in [\ell]} \sum_{i \in [n]} U(\theta; v_i^{(k)}) + \ell R(\theta) \quad (16)$$

which can be derived via the following linear program (LP):

$$\begin{aligned} &\text{minimize}_{\theta, \pi_{ik} \geq 0, \pi \geq 0} \sum_{k \in [\ell]} \sum_{i \in [n]} \pi_{ik} + \ell \pi \quad (17) \end{aligned}$$

$$\begin{aligned} \text{s.t. } \pi_{ik} &\geq v_i^{(k)}(b_i) - \theta(b_i) \quad \forall i \in [n], k \in [\ell], b_i \in B_i, \\ \pi &\geq \sum_{i \in [n]} \theta(a_i) \quad \forall a \in \mathcal{F}. \end{aligned}$$

Note that, at any optimal solution for LP (17), each variable π_{ik} equals the indirect utility (2) of bidder i in sample k at prices θ , while π equals the seller's indirect revenue (3). Under item prices, θ can be parametrized via m variables $p_j \geq 0$, and the last set of constraints reduces to $\pi \geq \sum_{j \in [m]} p_j$. Furthermore, as discussed in Section 4, the size of each B_i cannot be larger than the number of rounds. However, note that both the number of constraints and the number of variables π_{ik} are

proportional to the number of samples ℓ . In our experimental evaluation we confirm that it is possible to choose the sample size ℓ to achieve both a good expectation approximation and a tractable LP size.

Sampling Valuations from Posterior (Lines 5-9). To sample from the posterior $P_\lambda(v | \theta)$, we use the following generative model of bidder valuations. This generative model is an extension of the one provided by Brero and Lahaie (2018), itself inspired by Sollich (2002).

Definition 2. [Generative Model] *The generative model over bidder valuations, given prices θ , is defined by the following procedure:*

- Draw $v^{(k)}$ from $Q(v)$ (Line 8).
- Resample with probability $1 - e^{-\lambda \hat{W}(\theta; v^{(k)})}$ (Line 9).⁵

The following proposition confirms that the generative model correctly generates samples from (15).

Proposition 2. *The samples generated by our generative model have probability defined via the density function*

$$P_{gen}(v) \propto e^{-\lambda \hat{W}(\theta; v)} Q(v), \quad (18)$$

which corresponds to $P_\lambda(v | \theta)$.

Proof. We denote the probability of resampling as

$$r = 1 - \int_v e^{-\lambda \hat{W}(\theta; v)} Q(v) dv. \quad (19)$$

The probability that v will be drawn after h attempts is

$$P(v, h) = e^{-\lambda \hat{W}(\theta; v)} Q(v) r^h. \quad (20)$$

Thus, we have that

$$P_{gen}(v) = \sum_{h=0}^{\infty} P(v, h) \propto e^{-\lambda \hat{W}(\theta; v)} Q(v). \quad (21)$$

□

The λ relaxation of the price density function has interesting computational implications in our sampling procedure. The larger the λ (i.e., the better the approximation of Equation (10)), the larger the probability of resampling. Thus, using smaller λ will speed up the sampling process at a cost of lower accuracy. From this perspective, λ can serve as an annealing parameter that should be increased as the optimal

⁵Note that, to determine whether to resample $v^{(k)}$, one needs to compute the optimal social welfare $V(a^*; v^{(k)})$ via $\hat{W}(\theta; v^{(k)})$ (see Equation (5)), which may require a computationally costly optimization. This is not the case in our implementation as each $v_i^{(k)}$ is a multi-minded valuation over a small set of bundles B_i . Alternatively, one can use the “unnormalized” $W(\theta; v^{(k)})$ as a proxy for $\hat{W}(\theta; v^{(k)})$, as in the generative model proposed by Brero and Lahaie (2018) for single-minded bidders. When the optimal social welfare is not varying too much across different samples, this trick provides a good approximation of our generative model. Furthermore, it also did not prevent Brero and Lahaie (2018) from obtaining very competitive results.

solution is approached. However, while a larger λ increases the probability of finding clearing prices under density function $Q(v)$, it does not necessarily lead to better clearing performances in our auctions. Indeed, $Q(v)$ is affected by the auctioneer’s prior beliefs which may not be accurate. In particular, when the observed bids are unlikely under $Q(v)$, it can be useful to decrease λ from round to round. In our experimental evaluations, we will simply use $\lambda = 1$ and scale valuations between 0 and 10 as (implicitly) done by Brero and Lahaie (2018). This also keeps our computation practical. We defer a detailed analysis of this parameter to future work.

6 Empirical Evaluation

We evaluate our Bayesian auction via two kinds of experiments. In the first set of experiments we consider single-minded settings where we compare our auction design against the one proposed by Brero and Lahaie (2018). These experiments are meant to determine how many samples at each step of our Monte Carlo algorithm we need to draw to match their results. In the second set of experiments we consider multi-minded settings. Here, we will compare our auction design against non-Bayesian baselines.

6.1 Experiment Set-up

Settings. We evaluate our Bayesian auction on instances with 12 items and 10 bidders. These instances are sampled from four distributions provided by the Combinatorial Auction Test Suite (CATS): *paths*, *regions*, *arbitrary*, and *scheduling* (Leyton-Brown, Pearson, and Shoham, 2000). Each instance is generated as follows. First, we generate an input file with 1000 bids over 12 items. Then we use these bids to generate a set of bidders. To generate multi-minded bidders, CATS assigns a dummy item to bids: bids sharing the same dummy item belong to the same bidder. To generate single-minded bidders we simply ignore the dummy items. In each CATS file, we partition the bidders into a *training set* and a *test set*. The training set is used to generate the prior Gaussian density functions $Q(v^0(x)) = \mathcal{N}(\mu^0(x), \sigma^0(x))$ over bundle values $v^0(x)$ which are used to initialize the auctioneer beliefs in our auction. Specifically, we fit a linear regression model using a Gaussian process with a linear covariance function which predicts the value for a bundle as the sum of the predicted value of its items. The fit is performed using the publicly available GPML Matlab code (Williams and Rasmussen, 2006). Each bid of each bidder in the training set is considered as an observation. We generate the actual auction instance by sampling 10 bidders uniformly at random from the test set. We repeat this process 300 times for each distribution to create 300 auction instances which we use for the evaluation of our auction designs.

Upper Limit on Rounds. As mentioned in Section 2, the implementation of our Bayesian auction is based on item prices given by $\theta(x) = \sum_{j \in [m]} p_j x_j$, where $p_j \geq 0$. Item prices may not be expressive enough to support an efficient allocation in CATS instances (Gul and Stacchetti, 2000). We therefore set a limit of 100 rounds for each elicitation run and record reaching this limit as a failure to clear the market. Note

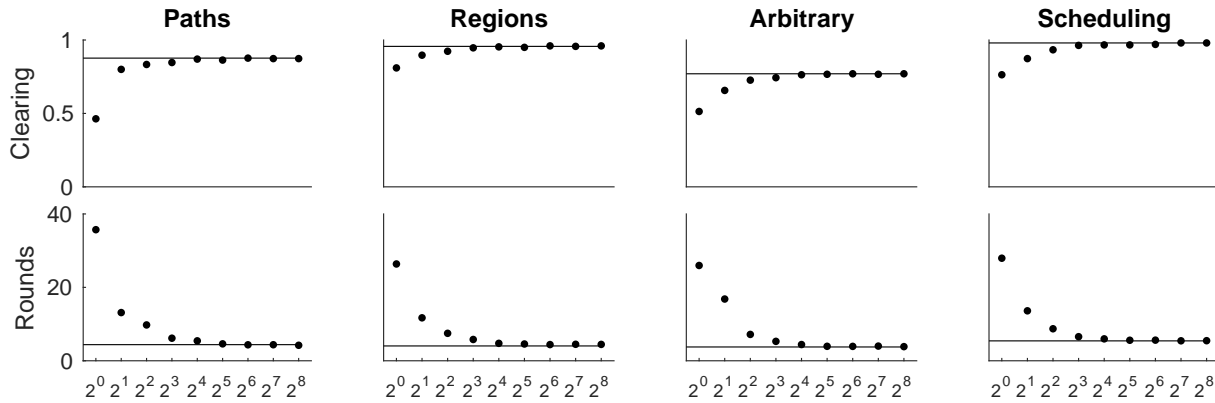


Figure 1: Average number of cleared instances and rounds under different number of samples ℓ in our Monte Carlo Expectation Maximization Algorithm. The horizontal lines indicate the baseline performance of Bayes^{SM} (Brero and Lahaie, 2018).

that, under this limit, some instances will only be cleared by some of the auctions that we test. To avoid biases, we always compare the number of rounds on the instances cleared by all auctions we consider.⁶

Non-Bayesian Baselines. We compare our auction design against non-Bayesian baselines that are essentially clock auctions that increment prices according to excess demand, closely related to clock auctions used in practice like the CCA, except that prices are not forced to be monotone. Because the Bayesian auctions are non-monotone, we consider non-monotone clock auctions a fairer (and stronger) comparison. The baseline clock auctions are parametrized by a single scalar positive step-size which determines the intensity of the price updates. We refer to these auctions as *subgradient auctions* (SG-Auctions), as they can be viewed as subgradient descent methods for computing clearing prices.

To optimize these subgradient auctions we run them 100 times on each setting, each time using a different step-size parameter spanning the interval from zero to the maximum bidder value. When then consider the following baselines:

⁶Alternatively, one could allow for more than 100 rounds on each instance where item clearing prices are found by any of the tested auctions. However, relaxing the cap on the number of rounds can lead to outliers with a very high number of rounds which can drastically affect our results.

- Subgradient Auction Distribution (SG-Auction^D): the subgradient auction using the step-size parameter that leads to the best performance *on average* over the auction instances generated from any given distribution.
- Subgradient Auction Instance (SG-Auction^I): the subgradient auction using the step-size parameter that leads to the best performance *on each individual auction instance*.

Note that these baselines are designed to be extremely competitive compared to our Bayesian auctions. In particular, in SG-Auction^I, the auctioneer is allowed to run 100 different subgradient auctions on each instance and choose the one that cleared with the lowest number of rounds.

6.2 Results for Single-Minded Settings

We now compare our Bayesian auction, denoted $\text{Bayes}^{\text{GEN}}$, against the one proposed by Brero and Lahaie (2018) (which is limited to single-minded settings), denoted Bayes^{SM} , and against the non-Bayesian baselines.

We first consider different versions of our auction design where we vary the number of samples ℓ used at each step of the Monte Carlo EM algorithm. As shown in Figure 1, our general Bayesian auction is competitive with Bayes^{SM} (Brero and Lahaie, 2018) starting from $\ell = 2^4$ (note that this is true for all distributions even though they model very different domains). This low number of samples allows us to solve the linear program presented in (17) in a few milliseconds. For the remainder of this paper, we will use $\ell = 2^7$.

	Paths		Regions		Arbitrary		Scheduling	
	Clearing	Rounds	Clearing	Rounds	Clearing	Rounds	Clearing	Rounds
SG-Auction ^D	84%	19.5 (1.0)	89%	24.8 (1.2)	65%	35.1 (1.8)	94%	21.0 (1.2)
SG-Auction ^I	88%	8.6 (0.4)	95%	15.9 (0.7)	75%	21.3 (1.0)	97%	11.9 (0.6)
$\text{Bayes}^{\text{GEN}}$	88%	5.2 (0.4)	96%	4.6 (0.3)	77%	4.5 (0.3)	98%	6.3 (0.3)
Bayes^{SM} (Brero and Lahaie, 2018)	88%	4.9 (0.2)	96%	4.3 (0.2)	77%	4.2 (0.3)	98%	6.1 (0.3)

Table 1: Comparison of different auction designs in settings with single-minded bidders. Clearing results are averaged over 300 auction instances. Rounds results are averaged over those instances that were cleared by all four auctions (which always included more than 100 instances for each distribution). Standard errors are reported in parentheses.

	Paths		Regions		Arbitrary		Scheduling	
	Clearing	Rounds	Clearing	Rounds	Clearing	Rounds	Clearing	Rounds
SG-Auction ^D	46%	22.0 (1.44)	75%	28.8 (1.4)	34%	36.2 (2.6)	51%	31.2 (2.2)
SG-Auction ^I	49%	9.3 (0.6)	81%	19.0 (0.9)	42%	27.4 (2.1)	59%	21.2 (1.4)
Bayes ^{GEN}	47%	11.5 (1.3)	83%	8.3 (0.6)	47%	9.7 (0.6)	57%	18.8 (1.4)

Table 2: Comparison of different auction designs in settings with multi-minded bidders. Clearing results are averaged over 300 auction instances. Rounds results are averaged over instances that were cleared by all three auctions (which always included more than 100 instances for each distribution). Standard errors are reported in parentheses.

As we can see from Table 1, both Bayesian designs dominate the non-Bayesian baselines in terms of cleared instances while also being significantly better in terms of number of rounds.

6.3 Results for Multi-Minded Settings

We now evaluate our Bayes^{GEN} auction in settings where bidders are multi-minded. As we can observe from Table 2, our Bayesian auction outperforms both baselines in terms of clearing and rounds (on average, over the different distributions). In Figure 2, we present the distributions of results for the auction rounds using box plots. Note that our Bayesian auction always significantly outperforms SG-Auction^D, and it outperforms SG-Auction^I for three out of the four distributions. Furthermore, note that, while SG-Auction^I and SG-Auction^D present very heterogeneous behavior across different distributions, our Bayesian design is much more consistent, with a third quartile that is always below 25 rounds.

7 Conclusion

In this paper, we have presented a Bayesian iterative CA for general classes of bidder valuations. Our framework allows the auctioneer to make use of any prior information and any model of bidder values to propose new ask prices at each round of the auction. At the core of our auction is a practical Monte Carlo EM algorithm to compute the most

likely clearing prices based on the bidders’ revealed information. Our auction design is competitive against the design proposed by Brero and Lahaie (2018) over single-minded valuations, for which the latter was specially designed. For general valuations, our auction design (without any special parameter tuning) outperforms a very powerful subgradient auction design with carefully tuned price increments.

Our work gives rise to multiple promising research directions that can leverage and build on our framework. The most immediate next step is to investigate different valuation models within the belief update component. In this paper, we considered Gaussian models to better compare against prior work, but the analytic convenience of these models is no longer needed; for instance, it may be the case that different kinds of models work best for the different CATS distributions, and such insights could give guidance for real-world modeling. Another intriguing direction is to handle incentives within the framework itself, rather than rely on a separate phase with VCG or core pricing. Future work could investigate whether the auction’s modeling component can gather enough information to also compute (likely) VCG payments.

Acknowledgments

Part of this research was supported by the SNSF (Swiss National Science Foundation) under grant #156836.

References

- Ausubel, L. M., and Baranov, O. 2017. A Practical Guide to the Combinatorial Clock Auction. *The Economic Journal* 127(605).
- Ausubel, L. M.; Cramton, P.; Milgrom, P.; et al. 2006. The Clock-proxy Auction: A Practical Combinatorial Auction Design. In Cramton, P.; Shoham, Y.; and Steinberg, R., eds., *Combinatorial Auctions*. MIT Press. chapter 5.
- Bichler, M.; Hao, Z.; and Adomavicius, G. 2017. Coalition-Based Pricing in Ascending Combinatorial Auctions. *Information Systems Research* 28(1):159–179.
- Bikhchandani, S., and Ostroy, J. M. 2002. The Package Assignment Model. *Journal of Economic theory* 107(2):377–406.
- Bikhchandani, S.; de Vries, S.; Schummer, J.; and Vohra, R. V. 2001. Linear Programming and Vickrey Auctions. *IMA Volumes in Mathematics and its Applications* 127:75–116.
- Blum, A.; Jackson, J.; Sandholm, T.; and Zinkevich, M. 2004. Preference Elicitation and Query Learning. *Journal of Machine Learning Research* 5:649–667.

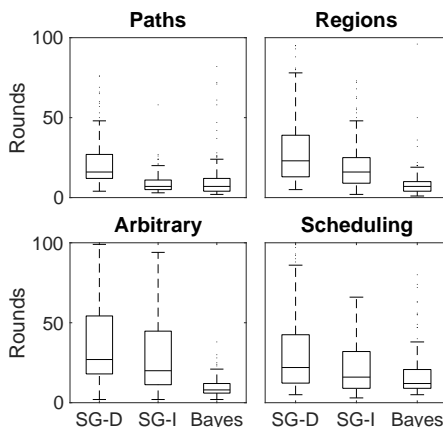


Figure 2: Auction rounds under multi-minded bidders. The box plot provides the first, second and third quartiles; the whiskers are at the 10th and 90th percentile.

- Brero, G., and Lahaie, S. 2018. A Bayesian Clearing Mechanism for Combinatorial Auctions. In *Proceedings of the 32nd AAAI Conference of Artificial Intelligence*, 941–948.
- Brero, G.; Lubin, B.; and Seuken, S. 2017. Probably Approximately Efficient Combinatorial Auctions via Machine Learning. In *Proceedings of the 31st AAAI Conference of Artificial Intelligence*, 397–405.
- Brero, G.; Lubin, B.; and Seuken, S. 2018. Combinatorial Auctions via Machine Learning-based Preference Elicitation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 128–136.
- Cramton, P. 2013. Spectrum Auction Design. *Review of Industrial Organization* 42(2):161–190.
- Goetzendorf, A.; Bichler, M.; Shabalin, P.; and Day, R. W. 2015. Compact Bid Languages and Core Pricing in Large Multi-item Auctions. *Management Science* 61(7):1684–1703.
- Gul, F., and Stacchetti, E. 2000. The English Auction with Differentiated Commodities. *Journal of Economic theory* 92(1):66–95.
- Kwasnica, A. M.; Ledyard, J. O.; Porter, D.; and DeMartini, C. 2005. A New and Improved Design for Multiobject Iterative Auctions. *Management Science* 51(3):419–434.
- Lahaie, S., and Parkes, D. C. 2004. Applying Learning Algorithms to Preference Elicitation. In *Proceedings of the 5th ACM Conference on Electronic Commerce*, 180–188.
- Leyton-Brown, K.; Pearson, M.; and Shoham, Y. 2000. Towards a Universal Test Suite for Combinatorial Auction Algorithms. In *Proceedings of the 2nd ACM Conference on Electronic Commerce*, 66–76. ACM.
- Milgrom, P., and Segal, I. 2013. Designing the US Incentive Auction.
- Minka, T. P. 2001. *A Family of Algorithms for Approximate Bayesian Inference*. Ph.D. Dissertation, Massachusetts Institute of Technology.
- Nguyen, T.-D., and Sandholm, T. 2014. Optimizing Prices in Descending Clock Auctions. In *Proceedings of the 15th ACM Conference on Economics and Computation*, 93–110. ACM.
- Nguyen, T.-D., and Sandholm, T. 2016. Multi-Option Descending Clock Auction. In *Proceedings of the 2016 International Conference on Autonomous Agents and Multiagent Systems*, 1461–1462. International Foundation for Autonomous Agents and Multiagent Systems.
- Opper, M., and Winther, O. 1998. A Bayesian Approach to Online Learning. *Online Learning in Neural Networks* 363–378.
- Parkes, D. C. 1999. iBundle: an Efficient Ascending Price Bundle Auction. In *Proceedings of the 1st ACM Conference on Electronic Commerce*, 148–157. ACM.
- Sandholm, T. 2013. Very-Large-Scale Generalized Combinatorial Multi-Attribute Auctions: Lessons from Conducting \$60 Billion of Sourcing. In Vulkan, N.; Roth, A. E.; and Neeman, Z., eds., *The Handbook of Market Design*. Oxford University Press. chapter 1.
- Sollich, P. 2002. Bayesian Methods for Support Vector Machines: Evidence and Predictive Class Probabilities. *Machine learning* 46(1-3):21–52.
- Train, K. E. 2009. *Discrete Choice Methods with Simulation*. Cambridge University Press.
- Wei, G. C., and Tanner, M. A. 1990. A Monte Carlo Implementation of the EM Algorithm and the Poor Man’s Data Augmentation Algorithms. *Journal of the American Statistical Association* 85(411):699–704.
- Williams, C. K., and Rasmussen, C. E. 2006. *Gaussian Processes for Machine Learning*. The MIT Press.