Computing Bayes-Nash Equilibria in Combinatorial Auctions with Continuous Value and Action Spaces

Vitor Bosshard[1], Benedikt Bünz[2], Benjamin Lubin[3] and **Sven Seuken[1]**

[1]Department of Informatics, University of Zurich
[2]Department of Computer Science, Stanford University
[3]Questrom School of Business, Boston University
{bosshard,seuken}@ifi.uzh.ch, buenz@cs.stanford.edu, blubin@bu.edu

Abstract

Combinatorial auctions (CAs) are widely used in practice, which is why understanding their incentive properties is an important problem. However, finding Bayes-Nash equilibria (BNEs) of CAs analytically is tedious, and prior algorithmic work has only considered limited solution concepts (e.g. restricted action spaces). In this paper, we present a fast, general algorithm for computing symmetric pure ε-BNEs in CAs with continuous values and actions. In contrast to prior work, we separate the *search phase* (for finding the BNE) from the *verification step* (for estimating the ε), and always consider the full (continuous) action space in the best response computation. We evaluate our method in the well-studied LLG domain, against a benchmark of 16 CAs for which analytical BNEs are known. In all cases, our algorithm converges quickly, matching the known results with high precision. Furthermore, for CAs with quasi-linear utility functions and independently distributed valuations, we derive a theoretical bound on ε. Finally, we introduce the new Multi-Minded LLLLGG domain with eight goods and six bidders, and apply our algorithm to finding an equilibrium in this domain. Our algorithm is the first to find an accurate BNE in a CA of this size.

1 Introduction

A combinatorial auction (CA) is a suitable mechanism to allocate resources in domains where bidders have complex preferences. CAs have been applied in multi-billion dollar domains, including for the sale of spectrum [Cramton, 2013; Weiss *et al.*, 2017] and for procurement [Sandholm, 2013].

Unfortunately, in CAs, the strategyproof VCG mechanism can lead to very low or even zero revenue (despite high competition for the goods), and it incentivizes collusion [Ausubel and Milgrom, 2006]. For these reasons, many CAs conducted in practice do not use VCG but other mechanisms, like core-selecting payment rules [Day and Milgrom, 2008]. However, these rules are not strategyproof. Thus, to understand their properties (incentives, efficiency, etc.) we must study them *in equilibrium* instead of *at truth*.

1.1 Bayes-Nash Equilibria in CAs

Early results for core-selecting auctions were derived in full-information Nash equilibrium (NE) [Day and Raghavan, 2007]. However, many real-world (high-stakes) CAs are only conducted once, and bidders work hard to keep their private information secret, making the full-information setting unrealistic. Instead, a more appropriate assumption is that each bidder knows his own valuation, but only has imperfect information about the other bidders' valuations. In this light, the appropriate solution concept is *Bayes-Nash Equilibrium (BNE)*.

For core-selecting payment rules, some analytical research already exists. Ausubel and Baranov [2013] as well as Goeree and Lien [2016] have independently derived the analytical BNE of the so-called "Quadratic" rule, which is the payment rule most commonly used in practice [Day and Cramton, 2012]. Furthermore, Ausubel and Baranov [2013] have also derived analytical BNEs of three other core-selecting rules. However, finding BNEs by hand is tedious and error-prone, as it typically requires solving challenging differential equations. For this reason, only few analytical results exist, and only for small settings with two goods and three bidders.

1.2 Prior Algorithmic Work on Computing BNEs

Computer scientists have long worked on algorithms for computing equilibria in non-cooperative games. The *Gambit* software package provides a number of algorithms to find NEs and BNEs [McKelvey and McLennan, 1996; McKelvey *et al.*, 2016], but only for finite games (with finite type and action spaces). Solving auction games with even a modest number of types (valuations) and actions quickly becomes infeasible; thus, discretizing the CA on a fine grid is not an option. For this reason, researchers have developed special-purpose algorithms for computing BNEs in infinite games (like CAs). To make the computation tractable, all methods that have been proposed to date are limited in some way: restricting the type space, the action space, the size and complexity of the games, or considering a simpler equilibrium concept. Importantly, all numerical algorithms actually search for an ε-BNE, i.e., a strategy profile where each player can only benefit (in expectation) by at most ε in utility by deviating unilaterally.

One important class of BNE algorithms is based on *iterated best response* (also known as fictitious play). The algorithms proposed by Reeves and Wellman [2004], Vorobeychik and Wellman [2008] and Rabinovich *et al.* [2013] belong to this

class and made important contributions to the literature. To keep the computation manageable, all three algorithms restrict the strategy space: using piecewise linear strategies, multiplicative shading strategies, or a finite set of actions. One limitation of these algorithms is that they can only solve games with restricted strategy spaces, because their ε-BNEs are only valid within the space over which they search for best responses.[1] Extrapolating the BNEs produced by their algorithm to obtain a solution for CAs in the full strategy space would produce incorrect results, and lead to the *false precision problem*, which we discuss in Section 3.

1.3 Overview of our Approach

We present a fast, general algorithm for computing *symmetric pure ε-BNEs in CAs with continuous values and actions*. Our approach is also based on the iterated best response algorithm, but highly optimized to CAs. To handle CAs with continuous value and action spaces we introduce two key ideas, which separate us from prior work: (1) our algorithm is split into a *search phase* to find the BNE (where we operate with a coarse estimate of the ε), and a *verification step* to estimate a robust ε for the found BNE; (2) in every best response calculation, we use the full (continuous) action space.

In the search phase, we use piecewise linear strategies, similar to Reeves and Wellman [2004]. However, the usage of the *verification step* is novel: we verify the found BNE by computing the utility loss at a large but finite number of verification points (values). This provides us with a robust estimate of the ε of the ε-BNEs. In Section 7, we show that for CAs with quasi-linear utilities and independently distributed valuations, we can prove a bound on the true ε over the *full* (continuous) value space (essentially for an infinite number of verification points). Our approach is the first to achieve such a guarantee for infinite games without restricting the strategy space.

In Sections 5 and 6, we offer numerous techniques for reducing the runtime of the BNE search. We benchmark these techniques in the widely used LLG domain, matching known results with high precision. In Section 9, we introduce the new Multi-Minded LLLLGG domain, with eight goods and six bidders, and apply our algorithm to find an estimated 0.0035-BNE in this domain. To the best of our knowledge, our algorithm is the first to find such an accurate BNE in a CA of this size.

2 Preliminaries

2.1 Formal Model

A combinatorial auction (CA) is a mechanism allocating a set M of goods to a set N of bidders. Each bidder has a valuation (also called his *type* in the game theory literature) $v_i(S) : 2^M \mapsto \mathbb{R}_{\geq 0}$, which is a function assigning a value to each bundle $S \subseteq M$ of goods. Bidders may report non-truthful valuations $\hat{v}_i = s_i(v_i)$ to the mechanism, where s_i is the bidder's *strategy*, mapping all her possible valuations

---

[1]Please note that Vorobeychik and Wellman [2008][Sec. 7.4] correctly state this limitation of their algorithm. Rabinovich *et al.* [2013] also handle this issue correctly by only claiming to find the BNE in the "game with the restricted strategy space.". Reeves and Wellman [2004] restrict themselves to a class of auctions where the best response is guaranteed to lie in the restricted strategy space.

to bids. We make the standard assumption that bidders only submit bids on bundles for which their value is nonzero.

The mechanism takes the reported valuation profile $\hat{v}$ and computes an allocation $X(\hat{v}) = (X_1, \ldots, X_n)$ and a payment vector $p(\hat{v}) = (p_1, \ldots, p_n)$, with $X_i \subseteq M$ and $p_i \in \mathbb{R}$. We consider the incomplete information setting: v_i is a random variable drawn from some distribution V_i, and while all bidders know their own valuation, they only know the distributions V_i from which other bidders' valuations are drawn. We let V denote the corresponding joint distribution over all bidders. We study the Bayesian game induced by the simultaneous-move CA. We let $u_i(v_i, \hat{v}_i, \hat{v}_{-i})$ denote bidder i's utility, given his own valuation v_i, report $\hat{v}_i$ and all other bidders' reports $\hat{v}_{-i}$. Note that u_i implicitly encompasses the allocation and payment rule of the mechanism. Given a strategy profile $s = (s_1, \ldots, s_n)$, the expected utility of reporting $\hat{v}_i$ when other bidders are playing strategies s_{-i} is denoted

$$\bar{u}_i(v_i, \hat{v}_i, s_{-i}) := \mathop{\mathbb{E}}_{v_{-i} \sim V_{-i}} \left[u_i(v_i, \hat{v}_i, s_{-i}(v_{-i})) \right]. \quad (1)$$

The best response of bidder i to the other bidders' strategies is $BR_i(v_i, s_{-i}) := \arg \max_{r \in \mathbb{R}_{\geq 0}^{|2^M|}} \bar{u}_i(v_i, r, s_{-i})$. The expected utility loss of reporting $\hat{v}_i$ instead of a best response is $\varepsilon_i(v_i, \hat{v}_i, s_{-i}) := \bar{u}_i(v_i, BR_i(v_i, s_{-i}), s_{-i}) - \bar{u}_i(v_i, \hat{v}_i, s_{-i})$. An (ex-interim) *$\varepsilon$-Bayes-Nash equilibrium (ε-BNE)* is a strategy profile $s^* = (s_1^*, \ldots, s_n^*)$ such that for all i and all v_i in the support of V_i, $\varepsilon_i(v_i, s_i^*(v_i), s_{-i}^*) \leq \varepsilon$. In words, an ε-BNE is a strategy profile such that no bidder has a profitable deviation netting her more than an ε improvement in (absolute) utility.

We use ε-BNE as the solution concept because we use numerical algorithms with limited precision to find the BNEs. Thus, when we *solve* a CA, we mean that we search for an ε-BNE, where ε is a suitably small constant.

Remark 1. *While we present our results using the* absolute *error to establish the notion of an ε-BNE, our approach can alternatively use the* relative *error, by defining the utility loss as* $\bar{u}_i(v_i, BR_i(v_i, s_{-i}), s_{-i}) / \bar{u}_i(v_i, \hat{v}_i, s_{-i}) - 1$. *Only some minor technical adjustments for Theorem 1 are needed.*

2.2 The LLG Domain

We study the performance of our algorithm both in a small domain, where analytical results are available, and later in a novel larger domain (see Section 9). For the former we turn to the widely-used Local-Local-Global (LLG) domain [Ausubel and Milgrom, 2006]. In LLG there are two local bidders, each of whom is interested in a single good, and a global bidder who is interested in the package of both goods. Ausubel and Baranov [2013] study the case where the global bidder is drawn from $\mathcal{U}[0, 2]$, while the local bidders' valuations are distributed according to $F(v) = v^\alpha$ for parameter α and perfectly correlated with probability γ. Within this framing, they provide analytical results for four different core-selecting payment rules (Quadratic, Nearest-Bid, Proxy and Proportional). Adopting their results as our benchmark, we assemble a set of 16 auction settings: 4 payment rules each applied to four domains ($\alpha \in \{1, 2\} \times \gamma \in \{0, 0.5\}$).

When using a core-selecting payment rule in LLG, the global bidder is always truthful. To match the analytical results of Ausubel and Baranov [2013], we search for symmetric

equilibria (though this simplification is not essential to our algorithm). Accordingly, an LLG strategy profile is described by the symmetric local bidder strategy $s_{local} : [0, 1] \mapsto \mathbb{R}_{\geq 0}$.

3 The "False Precision" Problem

In this section, we discuss two limitations of prior algorithmic approaches that could lead to a *false precision problem:* (1) using a restricted action space, and (2) computing ex-ante BNEs instead of ex-interim BNEs.

3.1 Restricted Action Spaces

Many BNE algorithms restrict the action space in some way during the *search* for the ε-BNE, to keep the computation time manageable. However, a problem may arise when the algorithm has converged to the final strategy profile, and the ε of the ε-BNE must be reported. If the final strategy profile is only evaluated in the *restricted action space*, then the computed ε is accurate in the restricted game, but *not* in the game with the full (continuous) action space.

Consider the following simple but striking thought experiment: We search for a BNE in a non-strategyproof CA, restricting the action space to only one action, namely *bidding truthful*. Any iterated best response algorithm will immediately find an ε-BNE with $\varepsilon = 0$, as there is no beneficial deviation. Obviously, this 0-BNE only "survives" in the restricted action space, but not in the full action space.

More realistically, consider a simultaneous second-price auction of two goods as described in [Rabinovich *et al.*, 2013, Section 6]. There are two symmetric bidders with single-dimensional type $\theta \in U[0, 1]$. We consider the setting where the bidders' utility for good A is 0.7θ, for good B is θ, and for the bundle $\{A, B\}$ is 1.6θ. Rabinovich *et al.* [2013] analyze the game with a restricted action space where only bids of $0, 0.25, 0.5, 0.75$ or 1 are allowed. For this game, they find a (0.001)-BNE. To illustrate what would happen if one tried to naïvely extrapolate from the restricted action space to the full action space, we computed a best response to their BNE in the full (continuous) action space.[2] Not surprisingly, the bidders deviate largely, and ε increases from 0.001 to 1.61.

Both examples illustrate that, if one is interested in finding the BNE of the game with the full (continuous) action space, then this needs to be handled explicitly. Our algorithm addresses this by considering the full action space in the best response calculation (at individual valuations). In Section 7, we later show that we can bound the ε of our algorithm, even over the whole value space, for a specific class of CAs.

3.2 Ex-ante vs. ex-interim BNEs

The second problem with "false precision" refers to the equilibrium concept being used: whether an *ex-ante* or an *ex-interim* ε-BNE is being computed. Interestingly, this issue mostly shows up when the strategy space is restricted to simple multiplicative or additive shading strategies (as in Lubin and Parkes [2009], Schneider *et al.* [2015], and Lubin *et al.* [2015]).

The underlying assumption of a one-parameter shading strategy is that *one* shading factor is applied uniformly, across

---

[2]For this, we used a relative instead of an absolute ε in our algorithm, because this is also what Rabinovich *et al.* [2013] used.

Algorithm 1: Iterated Best Response (with Verification)

> **Data:** Mechanism $\mathcal{M}$, Distribution V of bidder valuations
> **Result:** ε-BNE strategy profile
> 1 $s :=$ truthful strategies
> 2   **repeat**
> 3   **foreach** bidder i **do**
> 4 $s_i' :=$ BestResponseStrategy$(\mathcal{M}, V, s_{-i})$
> 5   **end**
> 6 $\tilde{\varepsilon} :=$ UtilityLoss(s, s')
> 7 $s :=$ Update(s, s')
> 8   **until** Converged$(\tilde{\varepsilon})$
> 9 $\varepsilon :=$ Verification(s)
> 10   **return** (s, ε)

---

all types. Thus, it is not possible for each type to play its preferred strategy, but all types must jointly choose a strategy to play. This gives rise to the notion of an *ex-ante ε-BNE:* every bidder only knows the distribution of its own type, the best response in the ε-BNE is enforced to be the same for all types, and the ε bounds the *average* benefit from deviating across all types. This is in contrast to the *ex-interim ε-BNE* we compute: each bidder knows its own type, the best response in the ε-BNE is computed separately for each type, and the ε bounds the maximum benefit from deviating across all types.

Sometimes, an auction designer may truly be interested in an ex-ante rather than an ex-interim BNE. However, when using multiplicative/additive shading strategies, the use of an ex-ante BNE arises mostly as an artifact of the choice of the strategy space, and is typically not otherwise justified. This is important for two reasons. First, ex-interim BNEs are arguably more realistic/interesting, because it makes sense to assume that bidders know their own valuation. Second, ex-interim BNEs provide significantly stronger guarantees, with ε bounding the maximum any type can gain from deviating unilaterally, while ex-ante BNEs only bound the average gain.

4 BNE Algorithm Framework

Our algorithmic framework has two key properties that distinguish it from prior work. First, we separate the *search phase* (finding the BNE) from the *verification step* (robustly estimating the ε of the found BNE). Second, when computing best responses, we consider the full (continuous) action space.

4.1 The Search Phase

The Iterated Best Response Algorithm. At the core of our algorithm's search phase is the well-known iterated best response algorithm (also known as fictitious play [Brown, 1951]), presented in Algorithm 1. This algorithm proceeds in rounds. In each round, each bidder's new strategy is computed via BestResponseStrategy as a response to the strategy profile from the previous round. The algorithm terminates when the utility loss across all bidders is small enough.

Modeling Strategies. To instantiate Algorithm 1, we must specify the strategy space to be used in the

`BestResponseStrategy` calculation. Ideally this would be the space of all functions, but this is infeasible. Instead, we use a restricted strategy space, parameterized by a finite set of *control points*. While the *search* is over such a restricted space, we later verify that our BNEs are valid without this restriction.

There are many restricted strategy spaces that one might use (e.g., piecewise constant functions, splines, etc); in this work, we adopt piecewise-linear functions for our strategy space. Note that when using piecewise-linear strategies, the control points are simply elements of the value space. We find piecewise-linear strategies to be particularly attractive as they can approximate any bounded function with a sufficient number of control points. For LLG, we use 160 control points (unless otherwise noted), as this is sufficient for convergence to $\varepsilon = 0.00001$ in all auction settings.

Pointwise Best Responses. To perform the `BestResponseStrategy` step under piecewise linear strategies, we need to maximize a bidder's expected utility as a function of her report at each control point individually. This requires calculating *pointwise* best responses at individual valuations v_i, defined as:

$$BR_i(v_i, s_{-i}) := \arg\max_{r \in \mathbb{R}_{\geq 0}^{|2^M|}} \bar{u}_i(v_i, r, s_{-i}(v_{-i})) \quad (2)$$

where s_{-i} is the strategy profile of the other bidders from the previous round. Finding the expected utility $\bar{u}_i$ requires solving a computationally challenging integral. We devote Section 5 to methods for making this calculation and the `Update` step in Algorithm 1 practical. Then, in Section 6, we show how to effectively implement the stopping criterion `Converged`.

Remark 2. *The function to be maximized in Equation (2) may be non-convex and/or non-differentiable, especially with high-dimensional strategy spaces. Furthermore, it is not given analytically, but only in black-box form. Thus, numerical methods may only find a local optimum. Given this, we employ a sophisticated version of* pattern search *to compute best responses that are as accurate as possible, even in multiple dimensions (see Section 5.5).*

4.2 Verification Step: Estimating the ε

An ε-BNE requires a bound on the utility loss over the entire value space. However, since the value space is continuous, it is not possible to check the loss for all individual valuations. Our algorithm might achieve a small utility loss *at the control points*, but not necessarily at values *between* control points, where strategies are interpolated linearly and not directly optimized. To cope with this, we employ a `Verification` step in our algorithm, as shown in Algorithm 1. In the verification step, we estimate the maximum utility loss as precisely as possible by taking a very fine, evenly-spaced grid of verification points (1, 000 in our case) and computing a best response at each of those points, using twice the number of Monte Carlo samples and twice the number of function evaluations in our pattern search (which will be introduced in Section 5). The maximum loss we find in this way is an estimate for the true ε. Note that due to the stochastic nature of our algorithm, the verification step should only be run once.

5 Computing Best Responses

In this section, we focus on the computation of pointwise best responses as formalized in Equation (2). This optimization problem is performed in the inner loop of our algorithm (called thousands of times). Thus, making the best response computation as fast as possible is essential to keep the runtime of the overall algorithm manageable. Computing the expected utility $\bar{u}_i$ for a single report $\hat{v}_i$ requires solving the integral

$$\bar{u}_i(v_i, \hat{v}_i, s_{-i}) = \int_{v_{-i}} u_i(v_i, \hat{v}_i, s_{-i}(v_{-i})) f_{-i}(v_{-i}) \, dv_{-i}, \quad (3)$$

where f_{-i} is the joint PDF associated with the distribution of valuations of all the other bidders, V_{-i}. In this paper, we approximate the value of Equation (3) via Monte Carlo (MC) integration (i.e., numerical integration via random sampling) because it is robust to discontinuities and scales to high-dimensional spaces. This is important when we turn to larger CAs (our ultimate goal) in Section 9. However, we first consider the simpler LLG setting because it will enable us to compare our ε-BNE solutions against known analytical results in Section 8.

Remark 3. *The integral in Equation (3) in the simple LLG setting is only two-dimensional and thus may alternatively be solved by numerical quadrature. We also ran our full algorithm with numerical quadrature in LLG, and achieved an additional 3-fold speed-up. But to keep the presentation of the algorithmic techniques comparable throughout the paper, we evaluate their performance using MC integration exclusively.*

In the following, we present a baseline algorithm for computing best responses, and then offer a series of improvements, each building upon the last. Runtime results for finding a 0.00001-BNE are presented in Table 1.[3] Several of our techniques make a trade-off between speed and accuracy, so it is important to evaluate their effectiveness *as a whole* to capture how changes in accuracy affect the convergence rate of the overall algorithm. Therefore, we measure the runtime of the entire algorithm, not just a single best response calculation. Furthermore, to avoid conflating our runtime measurements with the accuracy of the algorithm's stopping condition, we *omit* the stopping criterion and instead run the algorithm for a large fixed number of iterations. Then, in an ex-post facto analysis, we compute the iteration and runtime at which the estimated ε crossed the 0.00001 threshold for the first time. In Section 6, we will study effective stopping criteria.

Remark 4. *Even though Monte Carlo integration is random, we keep the seed fixed to make our results exactly reproducible.*

5.1 Naive Monte Carlo Algorithm

We first present a basic algorithm where we create an evenly-spaced grid of control points over the value space. To maximize the bidder's expected utility at each control point we use Brent search [Brent, 1971], a commonly-used form of unconstrained optimization, and we use Monte Carlo integration to find the expectation. In most CAs, expected utility will be zero when a bidder bids too little to win, and positive above this, with a discontinuity at the boundary. In the LLG domain, it

---

[3]Single-threaded runs on a 2.8Ghz Intel Xeon E5-2680 v2.

Algorithm	Average Iterations	Average Runtime	Speedup Factor
Naive MC †	20.0	5991.8 s	
+ Quasi-R.Num. (baseline)	9.1	1296.2 s	-
+ Common R. Num.	8.4	117.9 s	11.0 x
+ Adaptive Dampening	5.5	70.6 s	18.4 x
+ Pattern Search	5.6	29.4 s	44.1 x
+ Statistical Tests	5.8	22.8 s	57.0 x
+ Adaptive Control Points	5.8	5.3 s	243.2 x

Table 1: Runtimes to achieve an estimated 0.00001-BNE for several algorithms, averaged over our 16 auction settings.
† Even with $200,000$ samples, Naive MC achieves only a 0.0004-BNE and is thus not suitable as a baseline.

is straightforward to find this boundary, enabling us to sample only from the positive region in our integration. We perform this optimization even in the basic algorithm presented here.

After each best response computation, we perform a *dampened update*, by making the current strategy a combination of the previous strategy and the best response: $(1 - w) \cdot s_i + w \cdot BR_i(v_i, s_{-i})$, for update weight $w = 0.5$. This reduces the risk of overshooting the equilibrium strategy, and thus avoids oscillations around the solution without convergence.[4] This basic algorithm fails to converge to our target $\varepsilon = 0.00001$ even using $200,000$ MC samples. We therefore include the time to reach a 0.0004-BNE in Table 1 for reference, but use our first enhancement, presented next, as our baseline.

5.2 Quasi-Random Numbers

Because an ε-BNE is defined by the worst-case utility loss over all valuations, an equilibrium is only found when any error in the best response calculation is below the ε-threshold for all control points of every bidder. Thus, an error on *any* control point of our strategy profile can prevent the algorithm from converging. This can produce the counter-intuitive effect that increasing the number of control points without increasing sampling accuracy can actually decrease convergence.

These types of errors arise in Monte Carlo integration through the *variance* in the sample estimate. Clearly, any reduction in variance we can obtain will enhance the algorithm. One effective method for reducing variance is to replace standard *pseudo-random* numbers with *quasi-random* numbers in the sampling process [Morokoff and Caflisch, 1995]. Quasi-random numbers are low discrepancy sequences that cover the sampled region more evenly than the same quantity of random numbers. In our implementation we use a multi-dimensional Sobol sequence; this modification enables convergence to our target of $\varepsilon = 0.00001$ with only 100,000 samples. Using quasi-random numbers we converge in 9.1 iterations and 1296.2 seconds on average (see Table 1).

5.3 Common Random Numbers

In the best response computation, we repeatedly compare the expected utility of two different actions. If X and Y are the random variables representing the utility associated

with two actions, then we want to determine if $\mathbb{E}[X] - \mathbb{E}[Y]$ is greater or smaller than zero. By linearity of expectation we can compute $\mathbb{E}[X - Y]$ instead and get the same result with lower variance. This idea is implemented by using the same sequence of samples to compute both $\mathbb{E}[X]$ and $\mathbb{E}[Y]$. The samples used for both integrals are pairwise perfectly correlated, but still quasi random when considering each of the integrals in isolation. Adding this technique, we get convergence to our target ε using only 10,000 samples, i.e., 10% of the samples needed by the baseline, resulting in a 11.0-fold speedup. Note that we get more than a 10-fold speedup because, in addition to saving a factor 10 in the best response computation, this change also makes it converge in slightly fewer iterations.

5.4 Adaptive Dampening of Strategy Updates

To obtain more consistent convergence, we employ *adaptive dampening* instead of the constant update factor $w = 0.5$ used in the baseline.[5] This sets the weights w dynamically, based on how close to a solution we expect to be:

$$w = \frac{2}{\pi} \arctan(c \cdot \varepsilon_i) \cdot (w_{max} - w_{min}) + w_{min}, \quad (4)$$

where c is a new constant and $\varepsilon_i = \varepsilon_i(v_i, s_i(v_i), s_{-i})$ is the utility loss of playing s_i at v_i. This creates a weight between w_{min} and w_{max}, separately for each control point. Adding this technique results in an 18.4-fold speedup over the baseline.

5.5 Pattern Search

In the best response calculation, the function being maximized is an integral computed via Monte Carlo, and is thus very expensive. To reduce these costs, we replace the Brent search with *pattern search*, which requires many fewer function evaluations. Pattern search is a type of hierarchical local search that evaluates a number of points around the current point according to a fixed pattern. If a better solution is found, it moves the center of the pattern there and continues searching. If not, it decreases the size of the pattern and continues searching at the current point. This process repeats until a sufficiently small scale has been achieved. We further tune the *precision* of the algorithm: when we are far away from an ε-BNE, then high precision is wasteful, but when we are close to an ε-BNE, then we need more precision to converge with high accuracy. We adjust the required precision to match the context using a method that is similar to that used for adaptive dampening in Section 5.4. Overall, this method is much cheaper to compute than Brent search when high precision is unnecessary, and almost as accurate when it *is* needed. Using pattern search results in a 44.1-fold speedup over the baseline.

5.6 Statistical Tests in Pattern Search

When using pattern search to find the best response, we do not care about the *actual utility* achieved by different actions, but only about *ordinal* information: which of the evaluated pattern points has the largest utility. Thus, we only need to draw enough samples to be reasonably certain which of the actions we are considering is best. We therefore modify our algorithm

---

[4]This phenomenon is typical of any procedure that iteratively searches for fixed points. For domains where convergence is difficult, it is necessary to set w adaptively, which we address in Section 5.4.

[5]Note that we are not the first to use adaptive forms of dampening (see, e.g., [Fudenberg and Levine, 1995; Lubin and Parkes, 2009]).

Algorithm	Average Iterations	Average Runtime	Speedup Factor
Naive	7.2	1094.4 s	
Naive-every-5 (baseline)	9.4	257.0 s	-
Adaptive	8.3	148.9 s	1.7 x

Table 2: Runtimes for algorithm with intrinsic stopping, excluding the verification step.

to first draw a small number of samples, and then apply a two sided t-test [Cramér, 1947] to determine if one of the actions is better with $p < 0.01$. If the t-test passes, we immediately continue to the next step in our pattern search, having saved many samples. If not, we draw additional samples. Adding this approach enables us to draw as few as 10% of the samples for some utility comparisons, resulting in a 57.0-fold speedup.

5.7 Adaptive Control Point Placement

BNE strategies often have regions of both high and low curvature, and thus, using an equal spacing of control points is inefficient because it requires many unnecessary points in straight regions to have sufficient accuracy in curved regions. To avoid this, we initialize our algorithm with an evenly spaced grid of a few control points (10 in our experiments). We then repeatedly place additional points at the midpoint of those segments where the second derivative is largest. This allows us to reduce the total number of control points without increasing ε. We obtain convergence in all 16 auction settings with only 40 control points instead of the 160 required at baseline, resulting in an overall 243.2-fold speedup.

6 Intrinsic Stopping Criterion

In Section 5, we employed an *extrinsic stopping criterion* for the search phase as an experimental tool to focus on the performance of the best response computation. But when used in practice, our algorithm needs to use an *intrinsic stopping criterion* to determine when the target ε has been reached.

To this end, we employ a *two-loop* approach inside our algorithm. The *inner loop* corresponds to the standard BNE search, while the *outer loop* performs a higher precision best response computation to estimate ε more precisely. Only if the outer loop estimates ε to be small enough do we move into the verification step. Otherwise, we return to the inner loop. In practice, when choosing the precision of the outer loop, there is an application-specific trade-off between the algorithm's runtime and the probability that the verification step will fail. In our experiments, we set the outer loop to have the same high precision as the verification step itself, to avoid the case where verification fails (with very high probability).

Given that the outer loop is much more expensive than the inner loop (here, as expensive as the verification step, with $1,000$ instead of 40 control points, and twice as many MC samples and pattern search points) we must avoid running the outer loop too frequently. We tested three intrinsic stopping criteria. For the *Naive* algorithm, we simply go into the outer loop after every iteration of the inner loop. This algorithm is so expensive that we do not use it as our baseline (first row of Table 2). Instead, we use as our baseline an approach where we only go into the outer loop every 5 iterations (second row of Table 2). Finally, our *adaptive algorithm* instead transitions

from the inner to the outer loop based on its current estimate of the ε from the inner loop of the algorithm. To account for the lower accuracy of the inner loop, we use $0.8 \cdot \varepsilon$ as the target for the inner loop. Furthermore, when the outer loop fails, we require at least two inner loop iterations before going into the outer loop again. This approach yields a 1.7-fold improvement (see Table 2). In all three variants of our stopping criterion, the strategy profiles we find pass the verification step.

7 A Theoretical Bound on ε

So far, we have estimated ε numerically by computing the utility loss at a finite number of valuations. In this section, we show that, in some auction settings, we can derive a theoretical bound on ε over the entire value space, thus proving formally that a strategy profile is in fact a *true* ε-BNE. We also show that our numerical estimates for ε are essentially identical to the theoretical bound.

7.1 Deriving the Theoretical Bound

Our theorem requires the following two assumptions:

Assumption 1 (Quasi-linear Utilities). *Utility functions are quasi-linear, i.e. $u_i = v_i(X_i) - p_i$.*

Assumption 2 (Independently Distributed Valuations). *The valuations v_i are mutually independent random variables.*

Assumption 1 is standard in auction theory, and not very restrictive. In contrast, Assumption 2 is more restrictive. By excluding all CAs with interdependent valuations, it implies that our theorem does not apply to eight of the 16 settings we defined in Section 2.2.

Before we can state our theorem, we need three more definitions. For a bidder i, a *grid* G_i is a partition of the bidder's value space into rectangular cells, where each cell C is characterized by its lowest and highest corners w_i and w_i', written as $C[w_i, w_i') \in G_i$. The open parenthesis denotes that the valuations on the upper boundary of the grid cell are explicitly excluded.[6] A strategy s_i is *piecewise constant on some grid G_i* if, for all cells $C[w_i, w_i')$ of G_i, for all valuations $v_i \in C[w_i, w_i')$, it holds that $s_i(v_i) = s_i(w_i)$. Finally, we let $u_i^{\text{BR}}(v_i) := \bar{u}_i(v_i, BR_i(v_i, s_{-i}^*), s_{-i}^*)$ denote the best response utility at valuation v_i.

Theorem 1. *Let s^* be a strategy profile of a CA with quasi-linear utilities and independently distributed valuations, with each strategy s_i^* piecewise constant on grid G_i. Then s^* is an ε-BNE with*

$$\varepsilon = \max_{i \in N} \max_{\substack{w_i, w_i': \\ C[w_i, w_i') \in G_i}} u_i^{BR}(w_i') - u_i^{BR}(w_i) + \varepsilon_i(w_i, s_i^*(w_i), s_{-i}^*) \quad (5)$$

Proof. See Appendix A of the full version of this paper. □

In words, the theorem states that for each grid cell of each bidder i, we derive a local bound on the utility loss of all valuations v_i contained in that cell. This bound does not depend on v_i, but only on the lowest and highest corners defining the cell. The worst such bound among all cells and all bidders is an upper bound for ε, over the whole value space.

---

[6]To simplify the presentation, in the definition of a *grid* we have not discussed what to do with the points at the upper boundary of the value space. To be technically correct, the grid definition should include all lower dimensional grid cells at this boundary as well.
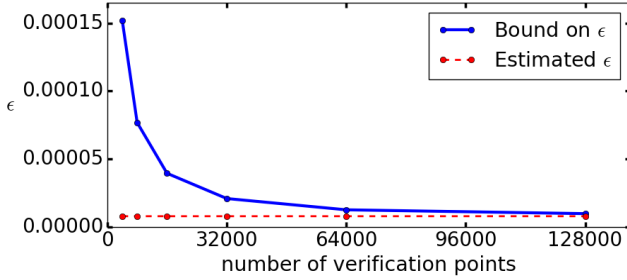
Figure 1: Maximum utility loss ε as estimated by our final algorithm from Section 6 compared to the proven bound on ε over the whole value space. Each data point is an average over eight auction settings.

Discussion/Application. Interestingly, Theorem 1 is constructive: we can use this result to obtain a strategy profile s^* and a bound ε, together with a proof that s^* is a true ε-BNE. For this, we consider an arbitrary strategy profile s (a candidate BNE). We transform the strategies in s such that they are piecewise constant on a grid: we take a regular grid over the value space and let $s_i^*(v_i) := s_i(w_i)$ for each v_i contained in cell $C[w_i, w_i']$. Then we compute a best response to the resulting strategy profile s^* at all grid points w_i, and finally we compute ε according to (5).

Remark 5. *Note that to formally apply Theorem 1, we would technically need to be able to calculate the exact best response utility u_i^{BR}. However, the best response algorithm we described in Section 5 is not guaranteed to find the global optimum (because it uses pattern search). We could substitute a best response algorithm with global convergence guarantees as used in [Vorobeychik and Wellman, 2008], though it should be noted that even such an algorithm is not guaranteed to produce an exact best response in finite computation time.*

7.2 The Bound on ε vs. the Estimated ε

In Section 4.2, we estimated the ε by computing the utility loss on finitely many verification points. With Theorem 1 at our disposal, we can now check the accuracy of these estimates. For this, we consider the 8 of our original 16 auction settings that are subject to Theorem 1, namely those with independently distributed valuations (i.e. $\gamma = 0$). We take the strategy profiles obtained as described in Sections 5 and 6 and compute both the estimated ε as well as the ε bound (via Equation (5)), using n evenly-spaced verification points in both cases.

In Figure 1, we show the estimated ε and the bound on ε, varying the number of verification points on the x-axis. We observe that the bound on ε converges towards the estimated ε, which remains practically constant. At $128,000$ verification points, the theoretical bound guarantees an ε that is below our target of 0.00001. This illustrates how attractive our estimated ε is: It is highly accurate and only uses a tiny fraction of the verification points that the theoretical bound requires.

8 Comparison to Analytical Results

The ε-BNE concept provides guarantees regarding the maximum *benefit* any bidder can gain from a unilateral deviation. However, it does not provide any guarantees regarding the *distance* between the ε-BNE and the true BNE. Nevertheless,

Mechanism	CA Domain			
	$\alpha = 1.0$ $\gamma = 0.0$	$\alpha = 1.0$ $\gamma = 0.5$	$\alpha = 2.0$ $\gamma = 0.0$	$\alpha = 2.0$ $\gamma = 0.5$
Quadratic	0.0016	0.0015	0.0027	0.0009
Nearest-Bid	0.0021	0.0032	0.0038	0.0031
Proxy	0.0021	0.0018	0.0026	0.0019
Proportional	0.0014	0.0016	0.0021	0.0009

Table 3: L_∞ distance between our ε-BNEs and analytical results from the literature for our set of 16 auction settings.

an auction designer may be interested in knowing this distance. We therefore determined the L_∞ distance between the 0.00001-BNEs we find in LLG and the analytical results from Ausubel and Baranov [2013] (see Table 3). We observe that all strategies we find are within 0.0038 of the corresponding analytical solution, showing that, in LLG, our algorithm indeed often finds an ε-BNE close to the exact BNE.

Remark 6. *Originally, we found a discrepancy between the ε-BNEs reached by our algorithm and the analytical BNEs in Ausubel and Baranov [2013]. We contacted the authors, who promptly confirmed a small mistake in their analytical results and provided us with the correct BNEs, which we have incorporated into Table 3. Such a correction highlights the value of having numerical techniques for finding BNEs, even in simple domains, as a complement to analytical methods.*

9 The Multi-Minded LLLLGG Domain

We next introduce the new Multi-Minded LLLLGG domain, which represents a significant increase in complexity over LLG. The domain has 8 goods $A - H$, and 6 bidders, each of which is interested in two bundles, as enumerated in Table 4. Each bidder L_i draws its two bundle values from $\mathcal{U}[0, 1]$, while bidders G_1 and G_2 draw their two bundle values

Bidder	Bundle 1	Bundle 2
L_1	AB	BC
L_2	CD	DE
L_3	EF	FG
L_4	GH	HA
G_1	ABCD	EFGH
G_2	CDEF	GHAB

Table 4: The Multi-Minded LLLLGG domain has 8 goods and 6 bidders. Each bidder is interested in exactly two bundles.

from $\mathcal{U}[0, 2]$; all draws are independent. We apply our algorithm to find an ε-BNE for the Quadratic rule in this domain. At this domain complexity, we must employ constraint generation [Day and Raghavan, 2007; Bünz *et al.*, 2015] to make finding prices computationally feasible in practice. Because the domain exhibits significant symmetries, we can search for symmetric equilibria where all local bidders play one strategy and global bidders play another. However, unlike in LLG, these strategies are two-dimensional. Thus, the strategy profile is described by a pair of strategies $s_{local} : [0, 1]^2 \mapsto \mathbb{R}_{\geq 0}^2$ and $s_{global} : [0, 2]^2 \mapsto \mathbb{R}_{\geq 0}^2$. Even though these symmetries can be exploited to reduce the dimensionality of the problem, computing the expected utility of a given report remains very expensive, requiring a 10-dimensional integral to be solved: the V_{-i} space consists of 2 bundle values for each of 5 other bidders, a dimensionality only amenable to MC approaches.
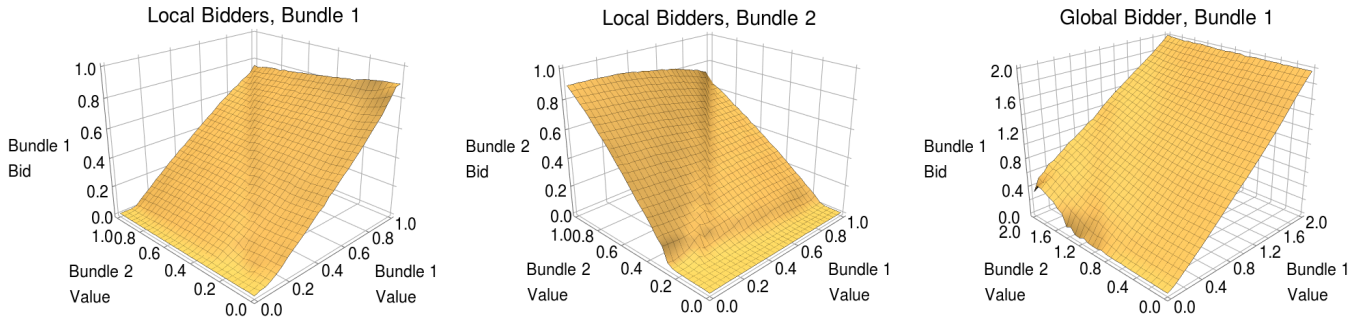
Figure 2: The equilibrium of the Multi-Minded LLLLGG domain. Left and center show the BNE strategies for the local bidders for bundles 1 and 2. Right shows the global bidders' BNE strategy for bundle 1 (the other bundle being symmetric).

We run our algorithm using a 30×30 grid of control points. For each iteration, we update all points in the grid as described in Section 5, using a maximum of 200,000 Monte Carlo samples to compute the expected utility (the exact number varies due to the usage of statistical tests). Computing the best response over a single bidder's grid of valuations takes several hundred core-hours. Consequently, running the 80 iterations needed to find an ε-BNE took us $54,381$ core-hours of computation time. The final strategy profile we found is an estimated 0.0037-BNE, and is illustrated in Figure 2. Using our technique from Section 7, we use a 120×120 grid of verification points to prove that it is a true 0.0218-BNE. Performing a single best response computation on this larger grid required another $54,192$ core-hours. Note that this is the first BNE calculation in such a complex domain with such rich strategies.

Discussion. In Figure 2, we observe that global bidders overbid on modestly valued bundles when their other bundle has a very high value. This manipulation makes sense as it shifts the VCG reference point: Suppose that bidders G_1, L_3 and L_4 win bundles $ABCD$, EF and GH respectively. Consider G_1: overstating the value for $EFGH$ will cause the VCG reference point to increase for L_3 and L_4. And if the value for $ABCD$ is very high for G_1, then this manipulation carries little risk of accidentally winning the low-valued $EFGH$ bundle, and it decreases the expected payment for $ABCD$.

10 Generality and Extensions

Our approach is very general and can be applied to a variety of settings. For example, our algorithm allows for arbitrary joint distributions of valuations, which cannot be handled by the analytical methods in Ausubel and Baranov [2013] and Goeree and Lien [2016] or by the numerical methods in Rabinovich *et al.* [2013]. A natural way to model such joint distributions is via *copulae* [Sklar, 1959; Lubin *et al.*, 2017].

Another direction we are already pursuing is to study the effect of bidders *overbidding* on bundles for which they have no value, as identified by Beck and Ott [2013]. Such strategy spaces are more challenging to evaluate, because they are inherently multi-dimensional. However, the inclusion of overbidding in LLG is less complex than the new Multi-Minded LLLLGG domain we presented in Section 9.

We also highlight the study of asymmetric equilibria which

has largely been absent from prior work due to its complexity, but which can be directly studied using our approach: all that is required is to track all bidders separately and to distinctly initialize their strategies. The former adds only linearly more computational effort, and the latter introduces a requirement to run the algorithm multiple times with random restarts.

Finally, we note that the theoretical bound derived in Section 7 does not apply to CAs with correlated bidders, which is an obvious limitation. To recover a similar result in the general case, it would be necessary to constrain the bidders' valuations and the mechanism to be nicely behaved in some sense. It may be possible to derive a bound for weakly correlated values, for specific CAs, but not for general classes of CAs. Investigating this is an interesting subject for future work.

11 Conclusion

In this paper, we have introduced a fast, general algorithm for finding symmetric pure ε-BNEs in CAs with continuous values and actions. We have presented multiple techniques to significantly improve the best response calculation and to make the strategy updates adaptive, leading to an overall 243.2-fold speed-up over the baseline algorithm. Furthermore, we have presented techniques for the algorithm to intrinsically determine when it has converged to its target ε without expending inordinate computation. In contrast to prior work, we separate the *search phase* of the algorithm from the *verification step*, and we use the full action space when *verifying* the found ε-BNE. Both ideas help us avoid the *false precision problem*.

We have first verified the accuracy of the resulting algorithm in the well-known LLG domain, where analytical benchmark results are known. We have then shown the power of our algorithm by providing the first ε-BNE with expressive strategies in a novel, large CA domain with six goods and eight multi-minded bidders. We believe this algorithm will facilitate the study of many mechanisms and domains that were not previously amenable to analytic or algorithmic analysis.

Acknowledgements

References

[Ausubel and Baranov, 2013] Lawrence Ausubel and Oleg Baranov. Core-Selecting Auctions with Incomplete Information. Working Paper, University of Maryland, December 2013.

[Ausubel and Milgrom, 2006] Lawrence Ausubel and Paul Milgrom. The lovely but lonely Vickrey auction. In Peter Cramton, Yoav Shoham, and Richard Steinberg, editors, *Combinatorial Auctions*, pages 17–40. MIT Press, 2006.

[Beck and Ott, 2013] Marissa Beck and Marion Ott. Incentives for overbidding in minimum-revenue core-selecting auctions. Working paper, Stanford University, 2013.

[Brent, 1971] Richard P. Brent. An algorithm with guaranteed convergence for finding a zero of a function. *The Computer Journal*, 14(4):422–425, 1971.

[Brown, 1951] George W Brown. Iterative solution of games by fictitious play. *Activity Analysis of Production and Allocation*, 13(1):374–376, 1951.

[Bünz et al., 2015] Benedikt Bünz, Sven Seuken, and Benjamin Lubin. A faster core constraint generation algorithm for combinatorial auctions. In *Proceedings of the 29th Conference on Artificial Intelligence (AAAI)*, 2015.

[Cramér, 1947] Harald Cramér. *Mathematical Methods of Statistics*. Princeton University Press, 1947.

[Cramton, 2013] Peter Cramton. Spectrum auction design. *Review of Industrial Organization*, 42(2):161–190, 2013.

[Day and Cramton, 2012] Robert W. Day and Peter Cramton. Quadratic core-selecting payment rules for combinatorial auctions. *Operations Research*, 60(3):588–603, 2012.

[Day and Milgrom, 2008] Robert W. Day and Paul Milgrom. Core-selecting package auctions. *International Journal of Game Theory*, 36(3):393–407, 2008.

[Day and Raghavan, 2007] Robert W. Day and S. Raghavan. Fair payments for efficient allocations in public sector combinatorial auctions. *Management Science*, 53(9):1389–1406, 2007.

[Fudenberg and Levine, 1995] D. Fudenberg and D. K. Levine. Consistency and cautious fictitious play. *Journal of Economic Dynamics and Control*, 19:1065–1089, 1995.

[Goeree and Lien, 2016] Jacob Goeree and Yuanchuan Lien. On the impossibility of core-selecting auctions. *Theoretical Economics*, 11:41–52, 2016.

[Lubin and Parkes, 2009] Benjamin Lubin and David Parkes. Quantifying the strategyproofness of mechanisms via metrics on payoff distributions. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 349–358, 2009.

[Lubin et al., 2015] Benjamin Lubin, Benedikt Bünz, and Sven Seuken. New core-selecting payment rules with better fairness and incentive properties [extended abstract]. In *Proceedings of the 3rd Conference on Auctions, Market Mechanisms and Applications (AMMA)*, 2015.

[Lubin et al., 2017] Benjamin Lubin, Benedikt Bünz, and Sven Seuken. Designing core-selecting payment rules: A computational search approach, 2017. Working Paper.

[McKelvey and McLennan, 1996] R. D. McKelvey and A. McLennan. Computation of equilibria in finite games. In H. M. Amman, D. A. Kendrick, and J. Rust, editors, *Handbook of Computational Economics*, volume 1, pages 87–142. Elsevier, 1996.

[McKelvey et al., 2016] R. D. McKelvey, A. M. McLennan, and T. L. Turocy. Gambit: Software tools for game theory, version 16.0.0. http://www.gambit-project.org, 2016.

[Morokoff and Caflisch, 1995] William J Morokoff and Russel E Caflisch. Quasi-monte carlo integration. *Journal of Computational Physics*, 122(2):218–230, 1995.

[Rabinovich et al., 2013] Zinovi Rabinovich, Victor Naroditskiy, Enrico H. Gerding, and Nicholas R. Jennings. Computing pure bayesian-nash equilibria in games with finite actions and continuous types. *Artificial Intelligence*, 195:106–139, 2013.

[Reeves and Wellman, 2004] Daniel Reeves and Michael Wellman. Computing best-response strategies in infinite games of incomplete information. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 470–478, 2004.

[Sandholm, 2013] Tuomas Sandholm. Very-large-scale generalized combinatorial multi-attribute auctions: Lessons from conducting $60 billion of sourcing. In Nir Vulkan, Alvin E. Roth, and Zvika Neeman, editors, *The Handbook of Market Design*, chapter 16. Oxford University Press, 2013.

[Schneider et al., 2015] Mark Schneider, Robert W. Day, and Robert Garfinkel. Risk aversion and loss aversion in core-selecting auctions. *Decision Support Systems*, 79:161–170, 2015.

[Sklar, 1959] M Sklar. *Fonctions de répartition à n dimensions et leurs marges*. Université Paris 8, 1959.

[Vorobeychik and Wellman, 2008] Yevgeniy Vorobeychik and Michael Wellman. Stochastic search methods for nash equilibrium approximation in simulation-based games. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1055–1062, 2008.

[Weiss et al., 2017] Michael Weiss, Benjamin Lubin, and Sven Seuken. Sats: A universal spectrum auction test suite. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 51–59, 2017.