Learning to Bid: Computing Bayesian Nash Equilibrium Strategies in Auctions via Neural Pseudogradient Ascent

Martin Bichler*, Max Fichtl, Stefan Heidekrüger, Nils Kohring, Paul Sutterer
Department of Informatics, Technical University of Munich, Germany, bichler@in.tum.de

Learning equilibria in multi-agent games is challenging because the players' rewards may change depending on the actions of other learning agents. While there has been progress in computing Nash equilibria in complete-information games, little is known about learning equilibria in Bayesian games. Such games are a research frontier with auctions as the best-known example. The key difference between complete-information games and auction games is that we search an equilibrium bid function over a domain of infinitely many valuations, a problem for which no general solution theory is available. We introduce a numerical technique to compute Bayes-Nash equilibria, which is based on neural networks and self-play. The method implements a gradient ascent scheme and approximates the expected utility via Monte Carlo sampling. Training neural networks in this environment is challenging as the payoff functions of individual auctions are discontinuous and nondifferentiable. We solve this problem by leveraging an evolutionary strategy optimization technique that effectively smoothes the objective. This allows us to derive an estimate of the gradient of the smoothed game, and the individual agents adapt their policy by taking a step along their policy gradient. We introduce conditions for which we can certify an ϵ-Bayes-Nash equilibrium and provide extensive numerical experiments. The experimental results show that the method converges quickly to the analytical Bayes-Nash equilibrium in a wide range of auction games including auctions with asymmetric priors or risk aversion, or even in combinatorial auctions with correlation or multiple pure Bayes-Nash equilibria.

Key words: equilibrium learning, neural networks, Bayesian Nash equilibria

1. Introduction

The literature on machine learning largely focuses on single-agent learning. Multi-agent learning has become more popular recently due to the advent of Generative Adversarial Networks and applications in complex competitive game-playing (e.g. Brown and Sandholm (2019), Daskalakis et al. (2017), Silver et al. (2018)). This literature typically focuses on zero-sum, complete-information games. While complete-information games have seen some progress, equilibrium learning for incomplete-information (aka. Bayesian) games is in its infancy. Auction theory is arguably the best-known and practically most relevant application domain of Bayesian games, central to modern economic theory (Klemperer 2000) and with a multitude of applications in the field, ranging from industrial procurement to

treasury auctions and spectrum sales (Klemperer 2004, Milgrom 2017, Bichler and Goeree 2017). Unfortunately, Bayes-Nash equilibria in these games are not well understood.

The derivation of BNE strategies for the first-price and second-price sealed-bid auction led to a comprehensive theoretical framework for the analysis of single-item auctions, a landmark result of economic theory (Vickrey 1961, Krishna 2009). While single-item auctions are well understood and closed-form BNE strategies are known for a variety of auction formats and market environments, we only know equilibrium strategies for very few multi-item auction environments. For example, no explicit characterization of BNE strategies is known in first-price sealed-bid auctions of multiple homogeneous goods (multi-unit auctions), nor in first-price sealed-bid combinatorial auctions where bidders can submit bids on packages of goods (Krishna 2009). But even in simple single-item first-price sealed-bid auctions with specific assumptions on the prior type distributions or the risk attitudes of agents, equilibrium strategies can be hard to derive. Typically, one ends up in a system of partial differential equations and no closed-form solution is available. Numerical techniques to compute Bayes-Nash equilibria would not only be valuable for theory, but they would also be very valuable for the study of practical auction rules.

For finite, complete-information games, we know of the existence of a mixed Nash equilibrium and that the computation is generally PPAD-hard (Daskalakis et al. 2009). For Bayesian games with continuous action space, we neither know whether (possibly mixed) Bayes-Nash equilibria exist in the general case, nor do we know how hard they are to find if they exist. Cai and Papadimitriou (2014) showed that finding a BNE in simultaneous auctions for individual items is hard for PP, a complexity class higher than the polynomial hierarchy and close to PSPACE, and we know little about the complexity of finding BNE in other multi-item auctions (see Section 3).

The theory of learning in games examines what kind of equilibrium arises as a consequence of a process of learning and adaptation, in which agents are trying to maximize their own payoff while learning about the actions of other agents (Fudenberg and Levine 2009). Research on equilibrium learning has largely focused on complete-information normal-form games. So far, there is no comprehensive characterization of games that are "learnable," but there are some important results. For example, it is well-known that *no-regret dynamics* converge to a coarse correlated equilibrium in arbitrary finite games (Jafari et al. 2001, Stoltz and Lugosi 2007, Hartline et al. 2015, Foster et al. 2016). Coarse correlated equilibria (CCE) encompass the set of correlated equilibria (CE) of a finite game. The latter is a nonempty convex polytope which in turn contains the convex hull of the game's Nash equilibria such that we get $\text{NE} \subset \text{CE} \subset \text{CCE}$. The coordination in CE can be implicit via the history of play (Foster and Vohra 1997, Stoltz and Lugosi 2007). In contrast to correlated equilibria, coarse correlated equilibria may contain strictly dominated (pure) strategy profiles with positive probability. This means that while coarse correlated equilibria are learnable via no-regret algorithms,

they are a rather weak solution concept (Viossat and Zapechelnyuk 2013). Therefore, the question is when learning dynamics converge to a Nash equilibrium. Only recently, Mertikopoulos and Zhou (2019) showed conditions for which no-regret learning algorithms result in a necessarily unique Nash equilibrium, if they converge.

Bayesian games, however, have received little attention in equilibrium learning until recently. Such games can be modeled as infinite-dimensional variational inequalities, but solving such problems is challenging and *requires learning a bid function over infinitely many types*. Currently, we are lacking an established solution theory for such problems. Given how hard it is to find Bayes-Nash equilibria even in simple simultaneous single-item auctions in the worst case (Cai and Papadimitriou 2014) it is far from obvious that no-regret dynamics can find a BNE in continuous-type and -action Bayesian games. It is not even clear, how no-regret dynamics would be implemented in such games. However, with an appropriate implementation, the approach is surprisingly successful, as we will show.

1.1. Contributions

We introduce Neural Pseudogradient Ascent (NPGA) as a method to learn ex-ante equilibrium bid functions which allow for continuous type- and continuous action-spaces. We use neural networks to represent the players' bid functions. Neural networks can approximate a wide variety of (equilibrium bid) functions as is well-known from the universal approximation theorem (Hornik 1991). The networks are trained to maximize utility of each of the agents for which we rely on self-play and gradient-based optimization which does not require the expensive computation of best response strategies as it has been suggested in prior work.

Unfortunately, using neural self-play in this environment is all but straightforward: While we assume players' *expected utility* (over the distribution of other players' types) are differentiable in the chosen action, a key challenge is that in auctions, their ex-post utilities given realizations of valuations and other agents' actions have nontrivial discontinuities. Only the latter, however, can be directly observed in the data generated during self-play. As a result, standard ways of gradient computation (i.e., backpropagation from the observed data) fail and necessarily would result in constant-zero bids by all bidders. We address this problem by deriving pseudo-gradients via evolutionary strategy optimization rather than exact gradients via standard learning methods.

We extend recent results on the convergence of projected gradient ascent to Nash equilibria in complete-information games, which provides us with sufficient conditions when we can certify a Bayes-Nash equilibrium. An extensive experimental evaluation on a variety of sealed-bid auction games is provided where we compare against the analytical BNE strategy where it is known. The analysis includes standard single-item first- and second-price sealed-bid auctions, combinatorial auctions and different versions of first- and second-price multi-unit auctions. For single-item auctions, we

analyze symmetric and asymmetric environments with several different priors, as well as auctions with risk-averse agents. In the combinatorial setting, we analyze various types of core-selecting auctions with single-minded agents and possibly correlated priors, but also environments with multi-minded bidders and multiple pure-strategy Bayes-Nash equilibria. In the latter environment with multiple pure Bayes-Nash equilibria, we know that the sufficient conditions for gradient dynamics to converge to a unique Bayes-Nash equilibrium do not hold. Yet, NPGA converges to the payoff-dominant equilibrium. The accuracy of the method and the speed of convergence is very high in all of these environments and we are able to approximate the analytical BNE strategy with high precision whenever it is known. While our evaluation mostly focuses on such environments (to provide an unambiguous benchmark), we also discuss an estimator for approximation quality for environments where this is not the case. Actually, NPGA converges in all the environments that we analyze.

The versatility of NPGA and the number of auction games where we can closely approximate the BNE is remarkable and surprising given the established hardness results for equilibrium computation. NPGA is generic and can easily be adapted to different types of auction games or other Bayesian games. The method relies on no setting-specific domain knowledge beyond the ability to sample outcomes of the (auction) game for given valuation- and strategy-profiles. Importantly, we do not need to make any assumptions on the payoff function, the risk-attitude of bidders, the individual prior value distributions, or independence between bidders' valuations. NPGA exploits GPU hardware acceleration to massively parallelize the computations, which allows us to achieve good approximations within one or two hours even for the most complex combinatorial auction environments in our experiments on a single GPU. Our experiments suggest that NPGA provides a powerful tool for daunting equilibrium computation problems in incomplete-information games that are not yet well understood theoretically.

1.2. Outline

The remainder of this paper is structured as follows: First, in Section 2, we define the problem, introduce necessary notation and terminology, while Section 3 discusses related work. Then, in Section 4, we formally introduce NPGA and discuss a sufficient criterion to certify a strategy profile learned by NPGA as a Bayes-Nash equilibrium. In Section 5 we introduce the experimental design, and in Section 6 we present the empirical results of applying NPGA to sealed-bid auction games, before concluding with a summary of our findings and outlook in Section 7.

2. Preliminaries

A complete-information game is a triple $G = (\mathcal{I}, \mathcal{A}, u)$ where $\mathcal{I} = \{1, \dots, n\}$ describes the set of agents participating in the game. Throughout this paper, we denote by the index $-i$ a profile of types, actions or strategies for all agents but agent i. $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$ is the set of possible action profiles,

with $\mathcal{A}_i$ being sets of actions available to agent $i \in \mathcal{I}$. u is a vector of individual *utility* or *payoff* functions $u_i : \mathcal{A} \to \mathbb{R}$ that assign the game outcome for each action profile. In a *finite* game, each player $i \in \mathcal{I}$ has a finite number of available actions, and the number of agents is finite. Games with continuous action space, as discussed in this paper, are not finite.

A Bayesian game (with incomplete information) is described by a quintuple $G = (\mathcal{I}, \mathcal{A}, \mathcal{V}, F, u)$. $\mathcal{V} = \mathcal{V}_1 \times \cdots \times \mathcal{V}_n$ is the set of *type profiles*, where $\mathcal{V}_i$ describe convex sets of type signals available to agent $i \in \mathcal{I}$. $F : \mathcal{V} \to [0, 1]$ defines a prior probability distribution over type profiles that is assumed to be common knowledge among all agents in the game. By a slight abuse of notation, for any random variable X that depends on F, we will write F_X for its cumulative distribution function and f_X for its probability density function throughout this paper. For example, F_{v_i} denotes the marginal distribution of agent i's type. At the beginning of the game, a type profile $v \sim F$ is drawn and each agent i is informed of their own type $v_i \in \mathcal{V}_i$ only, thus the type constitutes private information based on which each agent chooses an action $b_i \in \mathcal{A}_i$. Each agent's (*ex-post*) utility (or payoff) function is then determined by $u_i : \mathcal{A} \times \mathcal{V}_i \to \mathbb{R}$, i.e. the agents' utilities depend on all agents' actions but only on their own type. Players aim to maximize their individual utility u_i.

In this paper we consider *sealed-bid auctions* on $\mathcal{K} = \{1, \ldots, m\}$. In this setting, agents are commonly referred to as *bidders*, types $v_i \in \mathcal{V}_i$ are called *private valuations* and actions $b_i \in \mathcal{A}_i$ are called *bids*. In single-item and multi-unit sealed-bid auctions, both possible valuations and feasible bids form convex sets $\mathcal{V}_i, \mathcal{A}_i \subseteq \mathbb{R}^m_+$, in combinatorial auctions, these sets are generally convex subsets of $2^{\mathcal{K}}$ unless further restricted. Without loss of generality, we further assume the prior distribution F to be atomless on $\mathcal{V}$. For sake of brevity, we limit the further formal description to single-item auctions ($m = 1$), but the extension to multiple items is straightforward. In each auction, a *valuation profile* $v \in \mathbb{R}^n_+$ is drawn from F. Now, i submits a bid b_i chosen according to some *strategy* or *bid function* $\beta_i : \mathcal{V}_i \to \mathcal{A}_i$ that maps valuations to actions. While randomization over actions via mixed strategies would be possible, the literature on Bayesian auction games focuses on *pure*-strategy equilibria. We will likewise restrict ourselves to pure-strategies here. We denote by $\Sigma_i \equiv \mathcal{A}_i^{\mathcal{V}_i}$ the resulting strategy space of bidder i induced by valuations $\mathcal{V}_i$ and by $\Sigma \equiv \prod_i \Sigma_i$ the space of possible joint strategies. Note that also for deterministic strategies, the spaces Σ_i are infinite-dimensional unless $\mathcal{V}_i$ are finite (in which case the game remains infinite but is finite-dimensional). We will equip Σ_i with the inner product $\langle \cdot, \cdot \rangle_{\Sigma_i} : \Sigma_i \times \Sigma_i \to \mathbb{R}; (\alpha, \beta) \mapsto \mathbb{E}_{v_i \sim F_{v_i}} [\alpha(v_i)^T \beta(v_i)]$ and the norm $\|\beta\|_{\Sigma_i} \equiv \sqrt{\langle \beta, \beta \rangle_{\Sigma_i}}$ such that they form Hilbert spaces. The specific choice of inner product is motivated by deliberations about agents' expected utilities and will become clear below.

When bidders have chosen their bids b_i, the auctioneer collects these bids, applies some *auction mechanism* that determines (a) an allocation $x \in \{0, 1\}^n$, with $x_i = 1$ if and only if agent i wins the

item, or 0 otherwise, and (b) payments $p \in \mathbb{R}^n$ that the agents have to pay to the auctioneer. Given some risk-constants $\rho_i > 0$, we model bidders' utilities by the *risk-adjusted payoffs*

$$u_i \ = (x_i \cdot v_i - p_i)^{\rho_i} \,. \tag{1}$$

Here $\rho_i = 1$ corresponds to the risk-neutral case with quasi-linear utility, whereas $\rho_i < 1$ indicates risk aversion. In the case of multi-unit auctions, each agent is allocated an integer amount between 0 and m units of the item and has valuations for each possible number of allocated units; in combinatorial auctions, valuations and allocations are defined on the set of *bundles* of items.

In non-cooperative game theory on complete-information games, Nash equilibria (NE) form the central equilibrium solution concept. An action profile a^* is a pure-strategy NE of the game $(\mathcal{I}, \mathcal{A}, u)$ if $u_i(a_i^*, a_{-i}^*) \geq u_i(a_i, a_{-i}^*)$ for all $a = (a_i, a_{-i}) \in \mathcal{A}$ and all $i \in \mathcal{I}$. Informally, in a NE no agent has an incentive to deviate unilaterally, given that all other agents also play the equilibrium strategy. Bayesian-Nash equilibria (BNE) extend this notion to incomplete-information games by calculating the expected utility $\overline{u}$ over the conditional distribution of opponent valuations v_{-i}: For a valuation $v_i \in \mathcal{V}_i$, action $b_i \in \mathcal{A}_i$ and fixed opponent strategies $\beta_{-i} \in \Sigma_{-i}$, we denote the *ex-interim utility* of bidder i by

$$\overline{u}_i(v_i, b_i, \beta_{-i}) \equiv \mathbb{E}_{v_{-i}|v_i} \left[u_i \left(v_i, b_i, \beta_{-i}(v_{-i}) \right) \right]. \tag{2}$$

We will also use the shorthand notation $\overline{u}_i(v_i, \beta)$ when $b_i = \beta_i(v_i)$. We further denote the *ex-interim utility loss* of an action b_i that is incurred by not playing the best response action instead, given v_i and β_{-i}, as

$$\overline{\ell}(v_i; b_i, \beta_{-i}) = \sup_{b_i' \in \mathcal{A}_i} \overline{u}_i(v_i, b_i', \beta_{-i}) - \overline{u}_i(v_i, b_i, \beta_{-i}). \tag{3}$$

Note that $\overline{\ell}$ can generally not be observed in online-settings (since best-response actions are unknown and expensive to compute) but is nevertheless useful for theoretical considerations. An *(ex-interim) ϵ-Bayes-Nash Equilibrium (ϵ-BNE)* is a strategy profile $\beta^* = (\beta_1^*, \ldots, \beta_n^*) \in \Sigma$ such that no agent can improve her own ex-interim expected utility by more than $\epsilon \geq 0$ by deviating from the common strategy profile. Thus, in an ϵ-BNE the following holds for all agents $i \in \mathcal{I}$, all her possible types v_i and her chosen bids $b_i^* = \beta_i^*(v_i)$:

$$\overline{\ell}_i \left(v_i; b_i^*, \beta_{-i}^* \right) \leq \epsilon. \tag{4}$$

A 0-BNE is simply called BNE. Thus, in a BNE, every bidder's strategy maximizes her expected ex-interim utility given opponent strategies for every possible type realization $v \in \mathcal{V}$. While BNE are most commonly defined at the *ex-interim* stage of the game, one might also consider *ex-ante* Bayesian equilibria as strategy profiles that concurrently maximize each player's *ex-ante* expected utility $\tilde{u}$ as given by

$$\tilde{u}_i(\beta_i, \beta_{-i}^*) \equiv \mathbb{E}_v \left[u_i(v_i, \beta_i(v_i), \beta_{-i}^*(v_{-i})) \right] = \mathbb{E}_{v_i \sim F_{v_i}} \left[\overline{u}_i(v_i, b_i, \beta_{-i}) \right]. \tag{5}$$

Similarly defining the *ex-ante* utility loss of a given strategy $\beta_i \in \Sigma_i$ by

$$\tilde{\ell}(\beta_i, \beta_{-i}) \equiv \sup_{\beta_i' \in \Sigma_i} \tilde{u}_i(\beta_i', \beta_{-i}) - \tilde{u}_i(\beta_i, \beta_{-i}), \tag{6}$$

an ex-ante κ-BNE $\beta^* \in \Sigma$ can be characterized by the equations $\tilde{\ell}(\beta_i^*, \beta_{-i}^*) \leq \kappa$ for all bidders $i \in \mathcal{I}$.

In fact, given the Bayesian game $G = (\mathcal{I}, \mathcal{A}, \mathcal{V}, F, u)$, one can consider $\tilde{G} = (\mathcal{I}, \Sigma, \tilde{u})$, which is an infinite-dimensional, complete-information game where the strategies β of G form the actions of $\tilde{G}$. We will call $\tilde{G}$ the *ex-ante game* associated with G. The set of ex-ante BNE of G is then equivalent to the set of NE in $\tilde{G}$. Clearly, every exact ex-interim BNE also constitutes an exact ex-ante equilibrium. The reverse holds almost surely, i.e. any ex-ante equilibrium fulfills equation 4, except possibly on a set $V \subset \mathcal{V}$ with $F(V) = 0$. To see this, one may consider the equation $0 = \tilde{\ell}(\beta^*) = \mathbb{E}_{v_i}\left[\overline{\ell}(v_i; \beta_i^*(v_i), \beta_{-i}^*)\right]$ and the fact that $\overline{\ell}(\beta, v_i) \geq 0$ by definition. Importantly, this almost-everywhere equivalence of ex-ante and ex-interim BNE holds for $\epsilon = 0$ but not for strictly positive ϵ. Given an ex-ante κ-BNE, equation 4 with $\epsilon = \kappa$ must only hold in expectation but may be violated for arbitrary many v_i. To delineate this difference between ex-ante and ex-interim approximate equilibria, we will write κ and ϵ to denote their respective approximation bounds.

Finally, differentiability of u, $\overline{u}$ and $\tilde{u}$ will be of importance. As we will see, the ex-post utilities u are crucially *not* differentiable in auctions. Nevertheless, differentiability will often hold for the expected ex-ante and ex-interim utilities. Let us introduce some desirable regularity conditions for theoretical analysis:

Definition 1 (Smooth Bayesian game). We call a Bayesian game *smooth* if the ex-interim utilities $\overline{u}_i(v_i, b_i, \beta_{-i})$ are continuously differentiable with respect to $b_i \in \mathcal{A}_i$ for each $i \in \mathcal{I}$ and a.e. $v \sim F$, all partial derivatives are uniformly bounded by a finite constant $Z < +\infty$:

$$\left\| \frac{\partial \bar{u}_i}{\partial b_{ik}}(v_i, b_i, \beta_{-i}) \right\| \leq Z \qquad \forall i \in \mathcal{I}, b_i \in \mathcal{A}_i \subseteq \mathbb{R}^m, k \in \mathcal{K}, v_i \in \mathcal{V}_i, \beta_{-i} \in \Sigma_{-i}, \tag{7}$$

and the ex-post utilities are square-integrable: There exists $S > 0$, s.t. for all $i \in \mathcal{I}$ and $\beta \in \Sigma$ we have

$$\mathbb{E}_v\left[u_i\left(v_i, \beta_i(v_i), \beta_{-i}(v_{-i})\right)^2\right] \leq S. \tag{8}$$

In smooth Bayesian games, we will write $\nabla \overline{u}_i(v_i, b_i, \beta_{-i}) \equiv (\partial \overline{u}_i(v_i, b_i, \beta_{-i})/\partial b_{ik})_k$ and call it the ex-interim payoff gradient. Furthermore, when G is smooth, the ex-ante gradient $\nabla_{\beta_i} \tilde{u}_i(\beta_i, \beta_{-i}) \in \Sigma_i$, which formally constitutes the Gâteaux derivative in the Hilbert spaces Σ_i, are also guaranteed to exist.

For a concrete example of an auction game, consider the (single-item) *First-Price Sealed-Bid* (FPSB) auction: The highest-bidding agent j (s.t. $\forall i \neq j : b_j > b_i$) wins, is allocated the item and pays her own bid as price $p_j = b_j$, thus her risk-neural ex-post utility will be $u_j = v_j - b_j$. All other agents i neither

get the item, nor do they have to pay anything: $\forall i \neq j : u_i = 0$. In the independent private values model, where all F_{v_i} are independent and identical, it is common to also assume symmetric bid functions $\beta_i \equiv \beta$. The expected utility can then be described as $\overline{u}_i(b_i) = G(\beta^{-1}(b_i))(v_i - b_i)$, where β^{-1} is the inverse bid function and $G(v) = F_{v_i}(v)^{n-1}$ is the common prior of the valuations of all other agents. Assuming this symmetry, setting to zero the payoff gradient $\nabla_{b_i}\overline{u}_i = \frac{g(v_i)}{\beta'(v_i)}(v_i - b_i) - G(v_i)$ results in an ordinary differential equation $\frac{d}{dv}\left(G(v)\beta^*(v)\right) = v \cdot g(v)$ which, in turn, yields a closed-form solution $\beta^*(v)$ to equation (4) (Krishna 2009). More general auction games with asymmetric priors $F_{v_i} \neq F_{v_j}$ or multiple items often lead to systems of partial differential equations for which it might neither be known if a solution exists, nor how to solve them analytically. It is known that monotonicity of the payoff gradient is a sufficient property for Bayesian games to exhibit a unique Bayes-Nash equilibrium (Ui 2016). Unfortunately, it is also difficult to characterize such structural properties for most auctions.

3. Related Literature

In what follows, we survey existing hardness results, approaches to equilibrium learning, and initial research on computing approximate Bayes-Nash equilibria.

3.1. Hardness of Equilibrium Computation

The computation of Nash equilibria has received significant attention after the initial contribution by John Nash on the existence of such equilibria in complete-information normal-form games (Nash et al. 1950). However, it was shown that the problem is PPAD-complete already for 2-agent normal-form games (Daskalakis et al. 2009) and it is hard to approximate (Rubinstein 2016). The computation of Nash equilibria for 3 or more agents is even FIXP-complete, i.e. complete for the class of search problems that can be cast as fixed point computation problems (Etessami and Yannakakis 2007).

Determining whether a pure-strategy BNE exists in a finite Bayesian game is NP-complete and these hardness results also hold if there are only two agents and the game is symmetric (Conitzer and Sandholm 2008). Finding a mixed Bayesian equilibrium in a Bayesian game is, of course, PPAD-hard, but might be even harder, but little is known in general. As indicated in the introduction, Cai and Papadimitriou (2014) show that finding a BNE in simultaneous single-item Vickrey auctions for which the bidders have combinatorial valuations is hard for the class PP (the decision version of ♯P), which is much harder than NP. Even certifying a BNE is PP-hard, which casts doubt on the question whether BNE can at all be predictive in the field. Besides, the authors show that it is even NP-hard to find a strongly Bayesian approximate coarse correlated equilibrium in the simultaneous Bayesian auction game. Note that environments with continuous action space are not finite games, and the existence result by Nash does not carry over. We are not aware of a proof that a possibly mixed Bayesian equilibrium always exists in such games. Athey (2001) showed conditions for pure BNE to

exist, Carbonell-Nicolau and McLean (2018) provided conditions that guarantee the existence of a BNE, while Ui (2016) characterized strong payoff-monotonicity as sufficient condition for uniqueness of BNE in ex-post differentiable continuous-action Bayesian games.

3.2. Equilibrium Learning

Our research is best situated in the literature on equilibrium learning. Learning in complete-information normal form games has a long history and has been extensively studied in game theory and, more recently, multi-agent reinforcement learning. One class of methods is formed by *best response dynamics*. The earliest such method, published by Cournot in 1838 has agents play a pure strategy best response against other agents' strategy used in the previous iteration. In Fictitious Play (FP) (Brown 1951), a best response is instead played against the strategy profile induced by opponents' empirical frequencies of play in all previous iterations. When the *empirical frequencies* of FP converge, the limit constitutes a Nash equilibrium, but the actual play only converges in special cases of normal form games such as potential games (Monderer and Shapley 1996).

Gradient dynamics constitute another class of equilibrium learning algorithms. Generalized Infinitesimal Gradient Ascent (GIGA) (Zinkevich 2003) or GIGA-WoLF (Bowling 2005) are examples of gradient dynamics in normal form games, where in each iteration, for each agent we move a step along the direction of the utility gradient, and then project the resulting point back to the set of feasible mixed strategies. If, aggregating over the stages of the process, the agent's regret grows sublinearly, then there is "no regret" asymptotically. GIGA's total regret is $O(\sqrt{T})$, where T is the number of steps in a repeated strategic game. Hazan et al. (2007) give an algorithm with a total regret of $O(\log(T))$. More recently, motivated by the emergence of Generative Adversarial Networks, there has been a focus on (complete-information) games with continuous action spaces and smooth utility functions (Letcher et al. 2019, Balduzzi et al. 2018, Schäfer and Anandkumar 2019). A result found for many of the studied settings and algorithms is that gradient-based learning rules do not necessarily converge to Nash equilibria and may exhibit cycling behavior, but often achieve no-regret properties and thus converge to Coarse Correlated equilibria (CCE) in complete-information games. An analogous result exists for finite-type (but possibly continuous-action) Bayesian games, where no-regret learners are guaranteed to converge to a Bayesian CCE (Hartline et al. 2015).

Gradient dynamics are known to converge in certain types of normal-form games such as potential games, bilinear games (Singh et al. 2000b), and convex games (Mertikopoulos and Zhou 2019). Letcher et al. (2019) explores gradient dynamics in complete-information continuous-action *differential games*. If ex-post payoffs are twice continuously-differentiable, they find properties such that gradient dynamics converge to at least *local equilibria*. Unfortunately, the ex-post utility in our auction games is not differentiable. More importantly, these techniques are defined for complete-information games with finite-dimensional action-spaces while we search for general functions. Unfortunately, a thorough understanding of the convergence and limiting behaviors in general continuous games is still missing.

3.3. Algorithms for Computing Approximate BNE

Earlier approaches to compute approximate BNE in auctions either comprised solving the set of nonlinear differential equations resulting from the first-order conditions of simultaneous maximization of the bidders' payoffs (Marshall et al. 1994, Bajari 2001), or of restricting the action space, e.g. through discretization (Athey 2001). Armantier et al. (2008) introduced a general BNE-computation method that is based on expressing the Bayesian game as the limit of a sequence of complete-information games but defining this sequence requires setting-specific analysis. More recently, research in machine learning contributed to learning good bidding strategies in repeated revenue-maximizing auctions (Nedelec et al. 2019).

Bosshard et al. (2017, 2020) were the first to compute BNE in more complex combinatorial auctions in two recent papers. Their innovative approach explicitly computes point-wise best-responses in a fine-grained linearization of the strategy space via sophisticated Monte-Carlo integration. Assuming independent priors ($F_{v_i|v_{-i}} = F_{v_i}$) and risk-neutrality of agents, their verification method can guarantee an upper bound ϵ on the ex-interim loss in payoff, thus provably find an ϵ-BNE. The high worst-case ex-interim precision comes at computational cost for more complex environments with multi-minded bidders.

NPGA works entirely different compared to prior best-response algorithms. It learns bid functions for *all possible values* (rather than point-wise) by searching the parameter space of the bid function via ex-ante gradient ascent. It can be adapted to various types of Bayesian games with low reimplementation effort. Importantly, with the massively parallel implementation of NPGA, we can compute approximate BNE even for complex first-price combinatorial auctions in less than 1.5 hours on consumer-grade hardware. NPGA neither requires discretization of the value or action space as in Athey (2001) nor does it rely on twice differentiable payoff or loss functions as required in the literature on differentiable games (Singh et al. 2000a, Letcher et al. 2019). We also do not make assumptions on the risk attitude or independence of the bidders' valuations, making NPGA a remarkably general numerical solver to analyze the equilibria of auction games and other complete- or incomplete-information games.

4. Neural Pseudogradient Ascent

In this section, we will first outline the basic algorithmic approach of NPGA, describe problems with standard backpropagation in neural networks in this context, and show conditions when we can certify an (ex-ante) κ-BNE.

4.1. NPGA as a Gradient Ascent Scheme

NPGA implements an online gradient ascent scheme, a method known from online convex optimization. Therefore, we briefly introduce some relevant terms and definitions. In online convex optimization, an agent makes decisions in an iterative decision process. Online algorithms might make several feedback

assumptions: This could be full information about the function u to be optimized, or merely oracle feedback on the n-th order derivatives of the function. Examples are bandit feedback where the agent observes $u(b)$ for specific $b \in \mathcal{A}$ or gradient feedback, where the agent observes $\nabla u(b)$. This feedback might also be noisy where some noise term is added to the gradient at each step. The latter is the type of feedback relevant for NPGA.

After making a decision, the decision maker observes the feedback from the oracle and suffers a loss, which can be adversarially chosen. A central performance criterion is *regret*, which is the difference between the total cost she has incurred and that of the best static decision in hindsight. An algorithm performs well if its total regret grows sublinearly, i.e. the loss in each stage asymptotically approaches zero.

Definition 2 (No-regret learner). Given a complete-information game $G = (\mathcal{I}, \mathcal{A}, u)$ that is played repeatedly, we call a sequence of action profiles $(b^t)_{t\in\mathbb{N}}$ a no-regret sequence if for each agent i the regret against any fixed action $b'_i \in \mathcal{A}_i$ vanishes to zero,

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} u_i(b'_i, b^t_{-i}) - u_i(b^t) \;=\; 0.$$

One of the best-known approaches in convex optimization is (projected) gradient ascent. At each stage, the algorithm takes a step towards the gradient of the objective. Then the resulting point is projected back onto the problem's feasible region, and the process repeats. When faced with a different objective function at each step as is the case in online convex optimization, this results in the *online gradient ascent* algorithm.

Online gradient ascent enjoys an $O(\sqrt{T})$ regret bound in the presence of stochastic first-order feedback, and is known to be a no-regret learner (Zinkevich 2003), as is dual averaging (Nesterov 2009). Dual averaging aggregates all previous gradient feedback in dual space, the result is then mirrored onto the problem's feasible region. A new gradient observation is generated, and the process repeats. The mirroring step is determined by a strongly convex regularizer. With the squared Euclidean norm as regularizer, dual averaging is equivalent to online gradient ascent. We'll draw on both methods in what follows.

As indicated, NPGA implements an online gradient ascent algorithm with stochastic first-order feedback. We provide a succinct description of an online gradient ascent scheme in Algorithm 1 and a detailed pseudo-code for NPGA later in Algorithm 2. Note that we want to find a Bayes-Nash equilibrium strategy β_i^* in the ex-ante state of the game, i.e. for a continuum of types v_i that a bidder can have. Crucially, this means that we want to perform projected gradient ascent in strategy space Σ and with respect to the ex-ante utilities $\tilde{u}$, i.e. understanding the game in algorithm 1 to be the ex-ante game $\tilde{G}$. At each time t, every agent $i \in \mathcal{I}$ receives a (noisy) estimate $\hat{\nabla}\tilde{u}_i$ of her individual

(ex-ante) payoff gradient at the current strategy profile. The agents simultaneously take a step along this gradient estimate and then chose a bid for the next stage and continue playing. Feasibility in each iteration is ensured in online gradient descent by including a Euclidean projection Proj_Σ onto the the set of feasible strategies. However, as strategies and gradients are infinite-dimensional vectors in a Hilbert space, this gradient ascent scheme in Algorithm 1 cannot be implemented directly.

Algorithm 1: Gradient Ascent scheme

Input: complete-information game $G = (\mathcal{I}, \mathcal{A}, u)$, initial action profile $b^0 \in \mathcal{A}$, step size sequence η_t,

for $t := 0, 1, 2, \ldots$ **do**

$\forall i:$	observe $\hat{\nabla} u_i(b^t)$;	`// observe noisy payoff gradients`
$\forall i:$	$b_i^{t+1} := \text{Proj}[b_i^t + \eta_t \hat{\nabla} u_i]$;	`// take projected gradient step`

end

Instead, we model each agent's bid function (or strategy) $\beta_i(v_i)$ by a neural network, called *policy network* $\pi_i(v_i; \theta_i)$, which determines bids $b_i = \pi_i(v_i; \theta_i)$. The policy network is determined by an appropriate neural network architecture and by a parameter vector θ_i, which represents d_i weights and biases of the network. Thus, for the space of possible parameters we have $\Theta_i \subseteq \mathbb{R}^{d_i}$. These finitely many network parameters can now be considered the actions in yet another finite-dimensional complete-information game.

Definition 3 (Proxy game). Let $G = (\mathcal{I}, \mathcal{A}, \mathcal{V}, F, u)$ be a Bayesian game with ex-ante utilities $\tilde{u}_i$ and let its strategy functions be implemented by some neural network: $\beta_i(v_i) \equiv \pi_i(v_i; \theta_i)$ with parameters from some finite-dimensional vector space $\Theta_i \subseteq \mathbb{R}^{d_i}$. The resulting game on parameters $\Gamma = (\mathcal{I}, \Theta, \tilde{u})$ with $\Theta \equiv \prod_i \Theta_i$ is a *finite-dimensional* complete-information game, which is called the *proxy game*.

Here we slightly abuse notation and write $\tilde{u}_i(\theta_i, \theta_{-i}) \equiv \tilde{u}_i(\pi_i(\cdot\,; \theta_i), \pi_{-i}(\cdot\,; \theta_{-i}))$. Importantly, common neural network architectures, like the ones we will use, have been shown to be able to approximate arbitrarily well any possible (sufficiently regular) function β_i, thus the equilibria in $\tilde{G}$ and Γ are likewise equivalent (up to an error imposed by the approximation capacity of the chosen neural net architecture).

At a high level, in NPGA, agents observe their individual payoff gradient with respect to θ_i under the current strategy profile and apply a small update to their neural network parameters θ_i (not to the bids b_i^t directly) that will lead to an improvement in payoff. With the NPGA policy network, we have transformed the search for bid functions β in an infinite-dimensional strategy space Σ to a search in a $d \equiv \sum_i d_i$ dimensional parameter space Θ of the neural networks.

Next, we will discuss the projection step onto the feasible region: In NPGA, we ensure feasibility of strategies directly by choosing a neural network architecture that only produces feasible bids. In our implementation, this means applying a ReLU activation in the output layer, which ensures that bids will always be non-negative. Thus, for any parameter $\theta_i \in \mathbb{R}^{d_i}$ and valuation $v_i \in \mathcal{V}_i$, $b_i = \pi_i(v_i; \theta_i)$ will become a feasible (non-negative) bid $b_i \in \mathcal{A}_i$ with $0 \leq b_i < \infty$, and we can therefore identify $\Theta_i \equiv \mathbb{R}^{d_i}$. The projection step of online projected gradient ascent is thereby superfluous and not required in our implementation.

The final, and most important, question is how the feedback oracle $\hat{\nabla}\tilde{u}_i$ in this online optimization can be implemented and computed in self-play.

4.2. Evolutionary Strategy Pseudogradients

The canonical way of implementing the gradient update would now be to calculate the payoff gradients $\nabla_\theta \tilde{u}$ via sampling ex-post outcomes and applying stochastic gradients via backpropagation (Rumelhart et al. 1986): If the u_i were ex-post differentiable, the policy gradients would be given by

$$\nabla_{\theta_i} \tilde{u}_i(\theta_i, \beta_{-i}) = \mathbb{E}_v \left[\nabla_{\theta_i} \pi(v_i; \theta_i) \nabla_{b_i} u_i(v_i; b_i, \beta_{-i}(v_{-i}))|_{b_i = \pi_i(v_i;\theta_i)} \right], \tag{9}$$

and we could stochastically sample the right hand side to get unbiased gradient estimates. However, auction mechanisms are not ex-post differentiable, thus the rightmost gradient term in (9) is not well-defined. In the following, we demonstrate that learning via backpropagation necessarily fails in auction games, even on ex-post differentiable intervals of u, and we instead propose direct ex-ante gradient estimation via evolutionary strategies, an important ingredient of NPGA. Exact gradients $\nabla_b u$ of the ex-post utility for a fixed valuation and opponent strategy profile (v_i, β_{-i}) lead to problems in gradient updates, because agent i's ex-post utility will generally be discontinuous in her action. This is inherent as allocations are discrete: An agent either wins or doesn't win an item. On any given segment of the action space that corresponds to a specific allocation, gradients for all agents will *always* be non-positive (in all coordinates), as shown in Figure 1 for the one-dimensional single-item case. Backpropagation will thus lead to a steady decrease of bids in every iteration.

Evolutionary strategies (ES) present an alternative method for gradient estimation in neural networks that is more applicable in our setting. Most recently, ES has been proposed as a competitive alternative to backpropagation in MDP-based reinforcement learning (Salimans et al. 2017). In ES, the parameter vector θ of the model is perturbed randomly P times, for example by adding P i.i.d. zero-mean, σ^2-variance Gaussian noise terms $\varepsilon_1, \ldots, \varepsilon_P$. The resulting P perturbed models are evaluated with respect to their "fitness" $\varphi_p \in \mathbb{R}$ and the model is ultimately updated with a step size η in the direction of the weighted average of the P noise vectors ε_p with more desirable perturbations being weighted higher than less desirable ones:

$$\theta^{t+1} \equiv \theta^t + \frac{\eta}{\sigma^2 P} \sum_{p=1}^{P} \varphi_p \varepsilon_p. \tag{10}$$

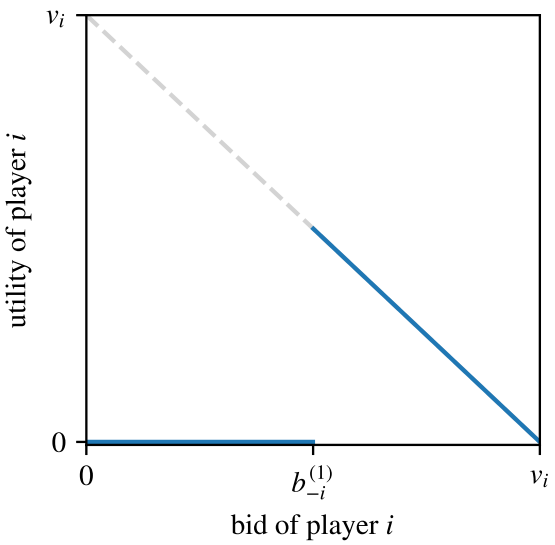


Figure 1 **Ex-post utility function $u_i(b_i)$ in First-Price Sealed-Bid Auction for fixed opponent bids b_{-i} with highest opponent bid $b_{-i}^{(1)}$. For any given bid b_i, the gradient $\nabla u_i(b_i)$ will be zero whenever the agent is not winning the item, and negative whenever she is. When all agents update their strategies using analytical gradients, they will thus eventually all bid zero.**

While Salimans et al. (2017) mainly motivate this alternative update with the need for large scale parallelization across CPU clusters and computational deficiencies of backpropagation in distributed hardware environments, the method also exhibits an important property that is crucial in our context. The *finite* perturbations solve the problem of inconsistent gradient signals at discontinuities of the utility, and serve as a smoothing technique. If an agent is barely losing an auction, a small perturbation resulting in a higher bid will also result in the agent winning the auction, thus providing a positive "pseudo-gradient" signal. The noise-hyperparameter $\sigma > 0$ shall be just large enough to serve this purpose. At the same time, for $\sigma \to 0$, the ES pseudogradient is in expectation identical to the analytical infinitesimal gradient (whenever the latter is well-defined). In NPGA, we thus use ES pseudogradients to implement the stochastic first-order feedback in Algorithm 1.

In our actual implementation, we extend the basic ES algorithm from Salimans et al. (2017) with two common practices from reinforcement learning and optimization: (a) we use the agent's utility in the previous iteration $\tilde{u}_i^{t-1}$ as a baseline parameter to reduce sampling variance in the fitness function, and (b) we replace the pseudo-gradient update with a generalized gradient ascent method to smoothen the learning trajectories. In particular, we found that Momentum or Adam (Kingma and Ba 2015) work well. A complete description of NPGA is provided in Algorithm 2.

Finally, in NPGA, we extend online gradient ascent in one more way: We sample large *batches* of valuation profiles (valuations for all agents) $v_h = (v_{1,h}, \ldots, v_{n,h})$ simultaneously and calculate the

Algorithm 2: Neural Pseudogradient Ascent using Evolutionary Strategy gradients

Input: agents $i \in \mathcal{I}$ with initial policy $\beta_i^0 := \pi_i(\cdot; \theta_i^0)$, with initial parameters θ_i^0; ES population size P; ES noise stddev σ; learning rate η; batch size H

for $t := 1, 2, \ldots$ **do**

Sample a batch $(v_h)_{h=1..H}$ of valuation profiles, with $v_h \sim F$

Calculate joint utility in current strategy profile:

$$\tilde{u}^{t-1} := \frac{1}{H} \sum_h \tilde{u} \left(\beta^{t-1}(v_h) \right)$$

for each agent $i \in \mathcal{I}$ **do**

Sample P perturbations of agent i's current policy:

$$\pi_{i;p} := \pi_i(\cdot; \theta_p)$$

with $\theta_p := \theta_i^{t-1} + \varepsilon_p$ and $\varepsilon_p \sim \mathcal{N}(0, \sigma^2 I)$ i.i.d. $\forall p \in [P]$

For each p, evaluate the fitness by playing against current opponents:

$$\varphi_p := \frac{1}{H} \sum_h u_i \left(\pi_{i;p}(v_{h,i}), \beta_{-i}^{t-1}(v_{h,-i}) \right) - \underbrace{\tilde{u}_i^{t-1}}_{\text{baseline}}$$

Calculate ES pseudogradient as fitness-weighted perturbation noise:

$$\nabla^{ES} := \frac{1}{\sigma^2 P} \sum_p \varphi_p \varepsilon_p$$

Perform a gradient update step on the current policy:

$$\Delta\theta_i^t := \eta^t \nabla^{ES} \tilde{u}_i^{t-1}(\beta^{t-1}(v)), \quad \theta_i^t := \theta_i^{t-1} + \Delta\theta_i^t, \quad \beta_i^t := \pi_i(\cdot; \theta_i^t)$$

end

end

resulting utilities and payoff gradient estimates in parallel leveraging modern hardware accelerators like GPUs. With a batch-size $H \gg 1$, we can therefore vastly reduce the variance of the gradient estimate in each iteration and thus the number of necessary learning iterations in comparison to standard online gradient ascent which expects sequential sampling. Overall, one innovation in NPGA is how we exploit GPU-acceleration to perform parallel computations over the entire type space in Bayesian games. While neural networks are suitable to such GPU-accelerated computation by design, this requires appropriate batched implementations of the auction formats.

4.3. Certification of BNE

The interaction of gradient dynamics and neural networks leads to a fairly complex algorithm such that an analysis is obviously challenging. Until very recently, convergence of no-regret dynamics to Nash equilibria was only known for restricted types of finite games. We draw on a very recent result by Mertikopoulos and Zhou (2019) on the convergence of no-regret learners to Nash equilibria in *finite-dimensional continuous-action and concave* complete-information games.

THEOREM 1 **(Mertikopoulos and Zhou (2019), Theorem 4.1)**. *Suppose that dual averaging is run with stochastic gradient estimates satisfying zero mean, and finite mean squared error, and a step-size sequence that is square summable but not summable and produces the sequence* $(a^t)_{t\in T}$ *of action profiles. If a complete-information game with finite-dimensional continuous action space* $\mathcal{A}\subseteq\mathbb{R}^d$ *is (pseudo-)concave and the sequence of pure strategy profiles* $(a_i^t)_{t\in T}$ *converges to* $a_i^*\in\mathcal{A}_i$ *for all* $i\in\mathcal{I}$ *with positive probability, then* a^* *is a Nash equilibrium.*

Convergence of no-regret algorithms to Nash equilibria is remarkable, because such algorithms otherwise only converge to coarse-correlated equilibria (Blum and Mansour 2007, Stoltz and Lugosi 2007), and the same is true for Bayesian games with finite type spaces (Hartline et al. 2015), as we discussed in Section 3.2.

In contrast to Mertikopoulos and Zhou (2019) we search for an (ex-ante) Bayes-Nash equilibrium function β^* in a Bayesian game rather than a complete-information game. The ex-ante state of the game can be interpreted as a complete-information game $\tilde{G}$ with an infinite-dimensional action space Σ. At first sight, the infinite-dimensional action space is in contrast to the assumptions in Mertikopoulos and Zhou (2019). However, instead of finding an equilibrium bid function in an infinite-dimensional action space, we approximate this bid function with a neural network, which has finitely many continuous parameters. We can consider the search for parameters as a finite-dimensional complete-information game Γ.

In order to draw on Theorem 1, we require the proxy game Γ to be concave. Assume that the ex-ante game $\tilde{G}$ is itself concave.[1] Then we desire that concavity of $\tilde{G}$ carry over to concavity of the finite-dimensional proxy game Γ, where the utility depends only on the network parameters θ. To achieve this, we rely on a special class of *convex neural networks* for our theoretical analysis, which is different from standard architectures used in practice or in our experiments.

[1] Unfortunately, analytical verification of concavity in Bayesian games is elusive in general. Without setting-specific theoretical analysis (which turns out to be difficult even for simple single-item settings), one may perform point-wise checks for concavity but these only provide weak evidence as they cannot guarantee adherence on the entire action-space in infinite games. Given the positive experimental results in Section 6, we suppose that payoff-gradient monotonicity to be satisfied in many auction games.

Definition 4 (Convex neural network). We call a network architecture $\pi_i : \mathcal{V}_i \times \Theta_i \to \mathcal{A}_i$ a *convex neural network*, or simply *convex*, if for every convex objective function $g : \Sigma_i \to \mathbb{R}$, the function $\theta_i \mapsto g(\pi(\cdot, \theta_i))$ is convex.

Finally, we assume an NPGA policy network to be a convex neural network with a few properties:

Definition 5 (NPGA policy network). An NPGA policy network $\pi_i : \mathcal{V}_i \times \Theta_i \to \mathcal{A}_i$ is a *concave neural network*, with the following properties:

1. *Lipschitz-continuous dependence of the network on its parameters:* The networks π_i depend Lipschitz-continuously on the parameters θ_i in the following sense: There exists some $L > 0$, such that for all $i \in \mathcal{I}$ and θ_i, θ_i' we have

$$\mathbb{E}_{v_i}\left[\|\pi_i(v_i, \theta_i) - \pi_i(v_i, \theta_i')\|\right] \leq L\|\theta_i - \theta_i'\|. \tag{11}$$

2. *Approximability of Σ_i by Θ_i:* There exists a $\delta > 0$, such that for all $i \in \mathcal{I}$ and $\beta_i \in \Sigma_i$ there exist parameters $\theta_i \in \Theta_i$ such that

$$\mathbb{E}_{v_i}\left[\|\beta_i(v_i) - \pi_i(v_i, \theta_i)\|\right] \leq \delta. \tag{12}$$

Since we assume that for fixed β_{-i} the function $\beta_i \in \Sigma_i \mapsto \tilde{u}_i(\beta_i, \beta_{-i})$ is concave, it follows that the utilities in the finite-dimensional game Γ, given by $\theta \mapsto \tilde{u}_i(\pi_i(\cdot, \theta_i), \pi_{-i}(\cdot, \theta_{-i}))$ are also concave in θ_i, if π_i is convex. Thus, if the networks π_i are convex, concavity of G carries over to concavity of Γ. Importantly, although most common neural network architectures are *not* convex, networks that implement Definitions 4 and 5, i.e. which induce a convex optimization problem on θ, have been shown to exist and to have universal approximation properties (Bach 2017). This means, they are able to implement any reasonable bid function β_i with arbitrary precision (depending on network size), which justifies the assumption in equation (12) of approximability.

Note that Mertikopoulos and Zhou (2019) focus on the dual averaging method (Nesterov 2009) in their analysis of no-regret learning in complete-information games. Dual averaging as well as projected online gradient descent as described in Algorithm 1 include a projection step, where the result of a gradient step is projected back to the set of feasible actions. Here, we apply the theorem to the simultaneous search for optimal neural net parameters of all agents. As discussed earlier, the ReLU output activations in the NPGA policy network ensure that $b_i = \pi_i(v_i; \theta_i)$ are non-negative and thus the resulting bids are feasible for any θ_i in the NPGA policy network, such that no projection is required. Therefore, our analysis is based on simultaneous gradient ascent as it is widely used in neural networks and which is known to be no-regret when producing feasible actions (Zinkevich 2003).

Proposition 1. *Let G be a smooth and concave Bayesian game, and $\pi_i(\,\cdot\,; \theta_i)$ be NPGA policy networks with $\Theta_i = \mathbb{R}^{d_i}$. Suppose that the updates in NPGA are chosen as $\Delta\theta_i^t \equiv \eta^t \nabla^{ES} \tilde{u}^{t-1}(\beta^{t-1}(v))$, where the step sizes $(\eta^t)_{t\in\mathbb{N}}$ are square summable but not summable. If the sequence $(\theta^t)_{t\in\mathbb{N}}$ produced*

by NPGA converges to $\theta^* \in \Theta$ with positive probability, then the strategies $\pi_i(\,\cdot\,;\theta_i^*)$ form a κ-Bayes-Nash-Equilibrium of G where $\kappa \in \mathcal{O}(\delta + L\sqrt{d}\sigma)$, σ is the smoothing-parameter of the NPGA algorithm and $d = \max_i d_i$ is the maximum dimension of the parameter spaces Θ_i.

Note that the ex-ante error κ can be divided into an $\mathcal{O}(\delta)$ part, which simply says that the error is linear in the "density" δ of the network, and a $\mathcal{O}(L\sqrt{d}\sigma)$ part, which describes the error that comes from the fact that we do not use exact gradients. Further note that the factor $\sqrt{d}$ here should not be interpreted as some "curse of dimensionality": The smoothing parameter σ denotes the magnitude of perturbation of a *single parameter* of the parameter vector $\theta_i \in \Theta_i$. Thus, if we keep σ constant, but increase the number of parameters d_i, the total magnitude of perturbation increases. So, in order to keep the "total perturbation" constant, σ should be chosen proportional to $1/\sqrt{d}$.

Proof of Proposition 1: Recall once again that the ex-ante state of the game can be seen as a complete-information game $\tilde{G}$ with an infinite-dimensional action space Σ. Now, we replace players' bid functions by NPGA policy networks $\pi_i(\,\cdot\,,\theta_i)$, which approximate β_i, which yields the proxy game $\Gamma = (\mathcal{I}, \Theta, \tilde{u})$ as described above. Note that Γ is a concave game by Definition 4.

The NPGA algorithm follows a gradient ascent scheme with respect to the $\tilde{u}_i(\theta_i, \theta_{-i})$, where gradients are approximated by ∇^{ES}. Note that Theorem 1 requires unbiased gradient estimates. However, the pseudo-gradients computed in the NPGA are in general *not* unbiased with respect to $\tilde{u}_i(\theta_i, \theta_{-i})$. Indeed, they are (in expectation) exact gradients of the "smoothed" utilities $\tilde{u}_i^\sigma(\theta_i, \theta_{-i}) = \mathbb{E}_{\varepsilon \sim \mathcal{N}(0,\sigma^2 I)}\left[\tilde{u}_i(\theta_i + \varepsilon, \theta_{-i})\right]$. This defines yet another finite-dimensional game $\Gamma^\sigma = (\mathcal{I}, \Theta, \tilde{u}^\sigma)$.

Lemma 1. *The gradient estimates ∇^{ES} in NPGA are unbiased and have finite mean squared error with respect to the smoothed utilities $\tilde{u}_i^\sigma$ of the game Γ^σ.*

One can easily check, that for a convex function $f : \mathbb{R}^d \to \mathbb{R}$, also $x \mapsto \mathbb{E}_{\varepsilon \sim \mathcal{N}(0,\sigma^2)}[f(x+\varepsilon)]$ is convex: For $x, y \in \mathbb{R}^d$ and $\lambda \in [0,1]$ we have

$$\mathbb{E}_{\varepsilon \sim \mathcal{N}(0,\sigma^2)}\left[f(\lambda x + (1-\lambda)y + \varepsilon)\right] = \mathbb{E}_\varepsilon\left[f(\lambda(x+\varepsilon) + (1-\lambda)(y+\varepsilon)\right] \tag{13}$$

$$\leq \lambda\, \mathbb{E}_\varepsilon[f(x+\varepsilon)] + (1-\lambda)\, \mathbb{E}_\varepsilon[f(y+\varepsilon)]. \tag{14}$$

Thus, $\tilde{u}^\sigma$ satisfies all the assumptions from Theorem 1. Consequently, if NPGA converges in the smoothed game Γ^σ, we found strategies θ^* that yield zero loss by Theorem 1. Now, since $\tilde{u}_i^\sigma(\theta_i, \theta_{-i})$ is close to $\tilde{u}_i(\theta_i, \theta_{-i})$, these strategies have a small loss in Γ as well:

Lemma 2. *Consider the utility loss $\tilde{\ell}_i^\Gamma(\theta_i, \theta_{-i})$ of agent i with respect to the utility function $\tilde{u}_i$ in the finite-dimensional proxy game Γ, and the utility loss $\tilde{\ell}_i^\sigma(\theta_i, \theta_{-i})$ with respect to the smoothed utility $\tilde{u}_i^\sigma$ in the game Γ^σ. Then*

$$\tilde{\ell}_i^\Gamma(\theta_i, \theta_{-i}) \leq \tilde{\ell}_i^\sigma(\theta_i, \theta_{-i}) + 2ZL\sqrt{d_i}\sigma.$$

Since the loss of θ^* in Γ^σ is zero, this gives $\tilde{\ell}_i^\Gamma(\theta_i^*, \theta_{-i}^*) \leq ZL\sqrt{d_i}\sigma$. Finally, as we assume approximability of any function in Σ_i by the network π_i (Assumption (12) in Definition 5), we can use a similar argument as in Lemma 2 to show that the loss of θ^* in the original infinite-dimensional game remains small: Let $\beta_i \in \Sigma_i$ be a best response to θ_{-i}^*. Notice that now β_i is chosen from the whole strategy space Σ_i, and is in general not induced by network parameters θ_i. However, by assumption, there exist network parameters $\tilde{\theta}_i$ such that

$$\mathbb{E}_{v_i}\left[\|\beta_i(v_i) - \pi_i(v_i, \tilde{\theta}_i)\|\right] \leq \delta.$$

Consequently,

$$\tilde{u}_i(\beta_i, \theta_{-i}^*) - \tilde{u}_i(\tilde{\theta}_i, \theta_{-i}^*) = \mathbb{E}_{v_i}\left[\bar{u}_i(v_i, \beta_i(v_i), \theta_{-i}^*) - \bar{u}_i(v_i, \tilde{\theta}_i, \theta_{-i}^*)\right] \leq Z\,\mathbb{E}_{v_i}\left[\|\beta_i(v_i) - \pi_i(v_i, \tilde{\theta}_i)\|\right] \leq Z\delta.$$

It follows that we can bound the ex-loss in the game G the following way:

$$\begin{aligned}\tilde{\ell}_i(\theta_i^*, \theta_{-i}^*) &= \tilde{u}_i(\beta_i, \theta_{-i}^*) - \tilde{u}_i(\theta_i^*, \theta_{-i}^*) = (\tilde{u}_i(\beta_i, \theta_{-i}^*) - \tilde{u}_i(\tilde{\theta}_i, \theta_{-i}^*)) + (\tilde{u}_i(\tilde{\theta}_i, \theta_{-i}^*) - \tilde{u}_i(\theta_i^*, \theta_{-i}^*)) \\ &\leq Z\delta + \tilde{\ell}_i^\Gamma(\theta_i^*, \theta_{-i}^*) \leq Z\delta + 2ZL\sqrt{d}\sigma = \mathcal{O}(\delta + L\sqrt{d}\sigma)\end{aligned}$$

Proofs for the Lemmata 1 and 2 are given in the appendix. This completes the proof of Proposition 1.

■

5. Experimental Design

In order to evaluate the versatility of NPGA to learn BNE in auction games, we explore a large number of auction settings with single or with multiple items. Table 1 provides an overview of the auction games that we analyze. For most of them, analytical BNE are known and can serve as an unambiguous benchmark. However, we also analyze some settings where no BNE are known. In these cases, in order to evaluate the quality of the outcome, we estimate the utility loss $\tilde{\ell}$ incurred by not playing a best response to the other agents strategies.

5.1. The Auctions

We begin with single-item auctions for which we consider two different payment rules, namely second-price sealed-bid (a.k.a. Vickrey/VCG auctions) and first-price sealed-bid (FPSB); we consider both risk-neutral and risk-averse bidders with symmetric priors, either following a uniform or a Gaussian distribution. In addition, we explore two cases with asymmetric uniform priors, one with overlapping and the other with non-overlapping support of the individual type spaces.

We continue with multi-unit auctions in which bidders compete for multiple homogeneous units. The standard payment rules for selling multiple units include "pay-your-bid" (discriminatory-price) and uniform-price (all items are sold at the same price). For the sale of m identical objects $\mathcal{K} = \{1, \ldots, m\}$,

Type	Payment rule	Agents
Single-item	VCG	risk-neutral
	FPSB	symmetric risk-neural, uniform prior
	FPSB	symmetric risk-neural, Gaussian prior
	FPSB	symmetric risk-averse, uniform prior
	FPSB	asymmetric risk-neutral, uniform prior
Multi-unit	discriminatory	risk-neutral, uniform prior
	uniform	risk-neutral, uniform prior
Combinatorial	VCG	risk-neutral, uniform prior
(LLG, 2 items, single-minded)	Core (nearest-VCG)	risk-neutral, uniform prior
(independent values)	Core (nearest-zero)	risk-neutral, uniform prior
	Core (nearest-bid)	risk-neutral, uniform prior
	FPSB*	risk-neutral, uniform prior
(LLG, 2 items, single-minded)	Core (nearest-VCG)	risk-neutral, uniform prior
(correlated values)	Core (nearest-zero)	risk-neutral, uniform prior
	Core (nearest-bid)	risk-neutral, uniform prior
(LLLLGG, 6 items, multi-minded)	FPSB*	risk-neutral, uniform prior
(independent values)	nearest-VCG*	risk-neutral, uniform prior
(Split-award, 2 items, multi-minded)	FPSB	risk-neutral, uniform prior

Table 1 Overview of auction settings learned with NPGA. For settings with the asterix (*) no analytical BNE is known

the type profile and action profile of each bidder fulfills $\mathcal{V}_i, \mathcal{A}_i \subset \mathbb{R}^m$, leading to a total number of $n \cdot m$ bids and allocations $x \in \{0,1\}^{n \times m}$. In each of the auctions, the items are awarded to the bidders corresponding to the m-highest bids. Each bid-component corresponds to the bidders' willingness to pay for one additional unit. The utility in equation (1) extends naturally to

$$u_i = \left(\sum_{k=1}^{m} x_{i,k} \cdot (v_{i,k} - p_{i,k}) \right)^{\rho_i}. \tag{15}$$

A detailed introduction to these standard multi-unit auctions can be found in chapters 12 and 13 of Krishna (2009).

Finally, we analyze three specific forms of combinatorial auctions. The first is known as a local-local-global (LLG) auction with two items $k = 1, 2$; two local bidders $i = 1, 2$ who are interested only in the item i; and one global bidder, interested in the set of both items $\{1, 2\}$ as perfect complements. Agents in the LLG setting are single-minded, i.e. they are interested in one bundle only. In this setting, we analyze environments both with independent priors $F = \prod_i F_{v_i}$, and with correlated priors $F_{v_i|v_{-i}} \neq F_{v_i}$, and consider FPSB and Vickrey-Clarke-Groves (VCG) as well as three *core-selecting* payment rules for which Ausubel and Baranov (2018) provide closed form equilibrium solutions. These payment rules are called the nearest-VCG, the nearest-zero, and the nearest-bid core-selecting payment rules. We analyze these auctions in the independent private values model, but also allow for correlation (Ausubel and Baranov 2018).

An extension to LLG is the LLLLGG setting with 6 items, 4 local bidders, each interested in two bundles of size 2, and 2 global bidders, each interested in 2 bundles of size 4. All bundles are partially overlapping. Bidders in this setting are multi-minded and $\mathcal{A}_i = \mathbb{R}_+^2$, significantly increasing the complexity. A detailed description can be found in Bosshard et al. (2020), who introduced the environment as a benchmark for a complex auction environment for which no analytical BNE is known.

The third combinatorial auction exhibits two pure BNE, i.e. bidders face a coordination problem. The model has been analyzed in the context of split-award procurement auctions (Anton and Yao 1992, Kokott et al. 2019). In first-price split-award auctions with two identical units, two suppliers (bidders) who have diseconomies of scale compete by offering prices for taking one or both units (Kokott et al. 2019). The auctioneer either buys one lot (50%) from each supplier or two lots (100%) from a single supplier at the offered prices based on a FPSB payment rule, depending on which award is cheaper. The equilibrium selection problem makes this environment particularly interesting.

5.2. Experimental Procedures

The focus of the experiments are the learned bidding strategies $\pi(\,\cdot\,;\theta)$, and their parameter vectors θ. In our implementation, we use fully connected policy networks with 2 hidden layers of 10 nodes each, using SeLU activation functions (Klambauer et al. 2017) in the hidden layers and a ReLU activation function in the output layer[2]. Instead of standard gradient ascent, we apply the Adam optimization algorithm with default parameters (see Kingma and Ba (2015)) and a learning rate of 0.001. In each iteration we generate $P = 64$ perturbations of the network π_i for ES gradient estimation, using zero-mean Gaussian noise with a standard deviation of $\sigma = 1/d_i$ (as suggested by Salimans et al. (2017)). We use batch sizes of $H = 2^{18}$ chosen such that the largest (10-bidder) settings would fit into available GPU memory. In the presence of asymmetries or multiple items, degenerate initializations (e.g., when some players *never* win) can impede convergence. To alleviate this, we force close-to-truthful initializations by pre-training the networks towards the truthful strategy using supervised learning (RMSE-loss, 500 steps of vanilla stochastic gradient descent). The network architecture performs well throughout most of the settings presented here. We did not perform setting-specific hyperparameter tuning to allow for comparable results. There are possibilities to improve the performance of our results when tuning the hyperparameters for a specific environment.

When possible, we take advantage of symmetry in the game and implement self-play with model sharing, i.e. symmetric bidders share a common policy $\pi_i(\,\cdot\,;\theta)$ with identical parameter vector θ. As a result, the inner for-loop in Algorithm 2 has to be computed only once each time, leading to

[2] Note that this architecture does not adhere to Definition 4, but parameter search in neural networks usually works remarkably well even when the induced problem is nonconvex. The exact reasons for this remain poorly understood.

a considerable speedup in settings with many symmetric bidders. In the asymmetric first-price and the local-global combinatorial settings, the weak and strong bidder and local and global bidders, respectively, have different policy networks.

We implemented the auctions using the PyTorch framework (Paszke et al. 2017) with a focus on computing many auctions in parallel. All experiments were performed on a single Nvidia Geforce RTX 2080Ti GPU with 5000 iterations (unless noted otherwise) and 10 replications of every setting provided in Table 1.

5.3. Evaluation Criteria

In order to assess the ability of NPGA to learn different auction settings and their BNE, we report a number of evaluation criteria introduced in the following. In particular, we compare the performance to the analytical BNE β^*, whenever known. For auction settings where the analytical BNE is unknown, we estimate the utility loss of not playing a best response to the opponents actions. It is important to stress that even though we use the analytical BNE in our evaluation, no knowledge of β^* is used in training: NPGA only uses information achieved from self-play of the network π_i against other players' networks π_{-i}. Evaluations of π_i against the (opponent) analytical BNE β^*_{-i} are independent of the learning algorithm and only performed for analysis.

To estimate ex-ante utility in self-play, we use the sample-mean of ex-post utilities achieved in a batch which is an unbiased estimator:

$$\hat{u}(\beta_i, \beta_{-i}) \equiv \frac{1}{H} \sum_h u_i(v_{h;i}, \beta(v_h)) \approx \tilde{u}_i(\beta_i, \beta_{-i}) \tag{16}$$

Whenever an analytical BNE, β^*, is known we implement it and sample the *BNE-utility* of a player $\hat{u}_i(\beta^*) \approx \tilde{u}_i(\beta^*)$ as well as the utility of player i's *learned strategy* β_i *against the BNE* $\hat{u}_i(\beta_i, \beta^*_{-i}) \approx \tilde{u}_i(\beta_i, \beta^*_{-i})$ with a batch size of $H_{\text{eval}} = 2^{22}$. With this batch size, the standard error of the sampled BNE utility is on the order of $\Omega(10^{-4})$, so this provides a natural limit to the precision of the results we report. Whenever additionally a closed-form solution of $\tilde{u}_i(\beta^*)$ (and not just β^*) is known, we use the closed form formulation for higher precision. We use the following metric as our main evaluation criterion whenever available:

1. The *relative utility loss* incurred by not playing the BNE strategy profile β^* but individually deviating with the learned strategy β_i:

$$\mathcal{L}(\beta_i) \equiv 1 - \hat{u}_i(\beta_i, \beta^*_{-i}) / \hat{u}_i(\beta^*) \tag{17}$$

For a near-perfect strategy, the fractional term approaches 1 and the loss vanishes. It's possible that a learned strategy leads to a utility very close to that of the BNE strategy ($\mathcal{L}(\beta_i)$ very close to zero), yet there are differences of β_i and β^*_i in the action space. Therefore, a second metric is the RMSE.

2. The probability-weighted root mean squared error of β_i and β_i^* in the action space, which approximates the L_2 norm of these two functions:[3]

$$\text{RMSE}(\beta_i) \equiv \sqrt{\frac{1}{H}\sum_h \left(\beta_i\left(v_{h;i}\right) - \beta_i^*(v_{h;i})\right)^2} \approx \sqrt{\mathbb{E}_{v_i}\left[\left(\beta_i\left(v_i\right) - \beta_i^*(v_i)\right)^2\right]} = \|\beta_i - \beta^*\|_{\Sigma_i} \tag{18}$$

In cases where no analytical BNE is available, we estimate the utility losses $\tilde{\ell}$ and $\overline{\ell}$ that were introduced in Section 2. Estimating these requires access to best response strategies for a given opponent strategy profile. For a given valuation profile v_h in a batch, we consider an equidistant grid of W possible bids $b_w \in \mathcal{A}_i$ of player i on the support of F_{v_i},[4] which in the single-item case corresponds to possible bids $b_w \in [0, V_i]$ where $V_i = \max \mathcal{V}_i$. For each of these possible bids b_w we evaluate their ex-interim utility against the current opponent strategy profile, $\overline{u}_i(v_{h;i}; b_w, \beta_{-i})$ (again via Monte-Carlo sampling). We then estimate $\overline{\ell}_i(v_{h;i}, b_w, \beta_{-i})$ by considering the best of these W bids as the best response at $v_{h;i}$:

$\hat{\lambda}_{h;i}(v_{h;i}; b_{h;i}; \beta_{-i}) \equiv \max_w \overline{u}_i(v_{h;i}; b_w, \beta_{-i}) - \overline{u}_i(v_{h;i}, b_{h;i}, \beta_{-i})$. Now we can define the following two metrics:

3. The estimated ex-ante utility loss of a learned strategy profile β:

$$\hat{\ell}_i(\beta_i) = \frac{1}{H}\sum_h \hat{\lambda}_{h;i}(v_{h;i}; \beta_i(v_{h;i}); \beta_{-i}) \approx \tilde{\ell}_i(\beta_i; \beta_{-i}) \tag{19}$$

4. The estimated worst-case ex-interim utility loss of a learned strategy profile β:

$$\hat{\epsilon}_i = \max_h \hat{\lambda}_{h;i}(v_{h;i}; \beta_i(v_{h;i}); \beta_{-i}) \approx \sup_{v_i} \overline{\ell}(v_i; \beta_i(v_i), \beta_{-i}) = \underbrace{\inf \epsilon}_{\text{s.t. eq. 4 holds for } i} \tag{20}$$

Then $\max_i \hat{\epsilon}_i$ can be considered an estimate of the smallest ϵ, such that β constitutes an ex-interim ϵ-BNE and $\kappa \equiv \max_i \hat{\ell}_i$ is our best estimate for an ex-ante κ-BNE.

Both $\hat{\ell}_i$ and $\hat{\epsilon}_i$ are imperfect: First, their computation for n players requires evaluating $n \cdot H^2 \cdot W$ auctions (for each $v_{i,h}$, W candidates for b_i are evaluated over H values of b_{-i}). We again leverage concurrent computation on the GPU and use $H = 2^{12}$, $W = 2^{13}$ in our reports. Still, this is considerably less efficient than a learning iteration, which requires the evaluation of only $n \cdot P \cdot K$ auctions. Second, both estimators are optimistically biased estimates of $\tilde{\ell}$ and the "true" ϵ, respectively. This is because both ex-ante and ex-interim loss take into account many, but only finitely many, potential ex-interim best responses b_i', and we can consider only finitely many potential worst-case valuations v_i. Nevertheless, these estimates are a useful decision aid and similar to those used in Bosshard et al. (2020). Based on these estimates, we can now compute a relative ex-ante utility loss without the analytical BNE available:

[3] For multidimensional actions, $m > 1$, we flatten the vectors for simplicity, as there is no necessity for separate evaluation in the presented settings.

[4] For priors with unbounded support, we use the $99.9th$ percentile as the "maximum" value.

5. The estimated relative ex-ante utility loss incurred by not playing a best response, approximating the relative utility loss (equation 17) in the absence of known BNE:

$$\hat{\mathcal{L}}(\beta_i) = 1 - \frac{\hat{u}_i(\beta)}{\hat{u}_i(\beta) + \hat{\ell}_i(\beta)} \tag{21}$$

Note that while opponents play a BNE strategy when computing $\mathcal{L}(\beta_i)$ they play their learned strategy when computing $\hat{\mathcal{L}}(\beta_i)$. As a result, $\mathcal{L}(\beta_i)$ can be larger than $\hat{\mathcal{L}}(\beta_i)$.[5]

Additionally, we report the runtime per iteration and, for all the experiments presented here, we set the maximum number of iterations to 5,000, and for more complex experiments to 20,000, a-priori. To indicate convergence in learning, we report stationarity once the maximum difference of the last three consecutive measurements[6] of $\hat{\ell}$ decreases below 0.0001, which can also constitute a termination criterion.

6. Results

A full table of NPGA's performance according to those criteria can be found in Table 10 in the Appendix. It allows for a compact comparison of all experiments' results while we focus on more specific criteria when reporting the individual results. In the following subsections we start by describing the analytical BNE solution, followed by a summary of the experimental results.

6.1. Vickrey-Clarke-Groves (VCG) Auctions

In any VCG auction, bidding truthful is a dominant strategy, therefore also a BNE (Vickrey 1961).

Result 1 *NPGA converges to the BNE in only a few iterations because of pre-training to truthful bidding. Without pre-training NPGA also learns a close approximation of the truthful strategy, but after significantly more iterations.*

Due to the pre-training to bidding truthful, NPGA is already in or very close to the BNE in all experiments with VCG auctions, such as the single-item or in combinatorial auctions. We saw convergence after just a few iterations in all Vickrey auctions (single-item and combinatorial), for all priors (uniform and Gaussian), and for any number of bidders (up to 10 in the single-item case). We thus omit detailed quantitative results for VCG auctions for brevity. When we did not pre-train to the truthful strategy, NPGA also learned a close approximation to the truthful strategy, but after serveral 100s to even thousands of iterations.

[5] For the specific environment of combinatorial auctions with independent priors and risk-neutral agents, Bosshard et al. (2020) report an ex-interim utility loss that is different. We also consider risk aversion and correlation in our experiments. Besides, NPGA takes long to converge for very low valuations. These lead to wins very rarely such that there is not much that can be learned in every round. Errors in the bidding strategy for very low values have very little impact on the overall utility as the probability of winning is low and the payoff is low. This is why we focus on ex-ante utility loss.

[6] $\hat{\ell}$ is computed every 100 iterations.

6.2. Single-Item Auctions

We consider symmetric as well as asymmetric single-item auctions. For both, we report the FPSB only, as the second-price or Vickrey auction is covered in 6.1. In FPSB auctions, unique BNE are known for n risk-neutral ($\rho = 1$) players with arbitrary but symmetric prior valuations, n risk-averse ($\rho < 1$) bidders under symmetric uniform priors (Menezes and Monteiro 2005), as well as for 2 players with asymmetric uniform valuation distributions with a stronger and a weaker player (Plum 1992).

Result 2 *Considering symmetric bidders, both risk averse and risk neutral, with uniform distributed valuations, NPGA converges to the BNE within a few minutes or up to 2h depending on the number of bidders; with Gaussian distributed valuations NPGA doesn't always meet the convergence criterion in time. The relative utility loss $\mathcal{L}$ is <0.35% (uniform) and <1.3% (Gaussian) considering only few bidders and $\leq 1\%$ (uniform) and <3.2%(Gaussian) considering 10 bidders. The learned bid strategy from NPGA closely approximates the analytical BNE.*

With asymmetric bidders and overlapping valuations NPGA learns a strategy with $\mathcal{L}$ <0.7% for both, the weak and the strong bidder. With non-overlapping valuations NPGA has difficulties learning a good strategy for the weak bidder and results in a strategy with $\mathcal{L}$ <0.7% for the strong bidder and 20% for the weak bidder.

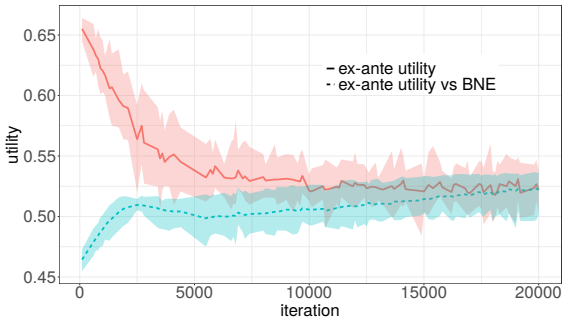
6.2.1. Symmetric We ran experiments in the symmetric environments with uniform and Gaussian distributed valuations for 2, 3, 5 and 10 bidders each. In the uniform-prior case, we considered risk-neutral $r = 1$ and risk-averse $r = 0.5$ bidders, with a Gaussian prior only risk-neutral bidders. In all of these settings, we measure bidders' utilities based on self-play in NPGA as well as when unilaterally playing their learned strategy against the analytical BNE after each observation. We observe convergence according to our convergence criterion described in Section 5.3 for uniform distributed valuations but not always for normal distributed valuations, yet the utility as well as utility loss is low and stable after around 18,000 iterations. One reason is that we focus on the absolute difference of the ex-ante utility loss which varies more if valuations are higher as in this setting; however, relying on a relative measurement leads to high fluctuation for minimal changes in settings with low valuations. Another reason is the sample variance which, in future, can be reduced by an increased batch size or more advanced sampling techniques.

Table 2 presents the utility loss incurred when playing a learned strategy against the analytical BNE. When learning with this 'standard' architecture without any hyper-parameter tuning, NPGA achieves a relative utility loss of less than 1.3% in all but one setting.

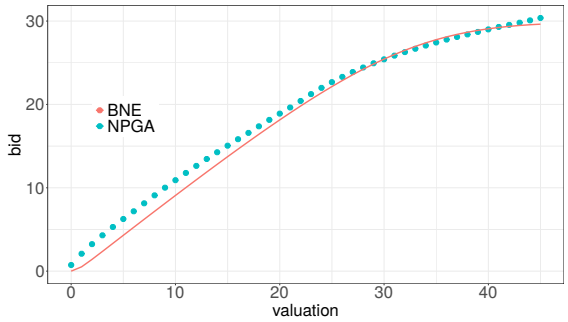
As an example, Figure 2 presents the learning behavior (a) and learning result (b) for NPGA in a FPSB auction with symmetric-normal prior distributions for 10 bidders. First, all bidders' utility decreases because they are forced to increase their bids given the others bidding behavior (left). After

Table 2 **Results of NPGA learning in first-price auctions with symmetric bidders averaged over 10 runs. If not all runs satisfied the convergence criterion, the number of runs that fulfilled the criterion are noted as subscript. [1]After 5,000 iterations. [2]After 20,000 iterations. [3]Only the weaker 0.0005 criterion is met.**

Setting	bidders	Relative loss $\mathcal{L}$ mean	(stdev)	conv. (iters)	sec per iter
Uniform risk-neutral[1] $\mathcal{U}(0,10)$ $\rho=1$	**2**	0.0001	0.0009	1,350	0.31
	3	0.0017	0.0006	870	0.40
	5	0.0034	0.0020	900	0.46
	10	0.0084	0.011	720	0.73
Uniform risk-averse[2] $\mathcal{U}(0,10)$ $\rho=0.5$	**2**	0.0011	0.0004	5,720	0.46
	3	0.0006	0.0003	2,100	0.52
	5	0.0012	0.0011	4,170	0.63
	10	0.0100	0.0068	8,410	0.92
Gaussian risk-neutral[2] $\mathcal{N}(15,100)$ $\rho=1$	**2**	0.0015	0.0011	–	0.31
	3	0.0037	0.0043	$7,050_2$[3]	0.39
	5	0.0129	0.0135	$11,700_5$[3]	0.44
	10	0.0314	0.0212	$9,675_8$[3]	0.68



(a) Trajectory of NPGA utility of learning opponents against each other (solid red line) and NPGA utility of learning opponents individually evaluated against the analytical BNE strategy (dashed blue line) over 10 runs of 20,000 iterations each. Shaded area around the lines illustrate the minimum and maximum.

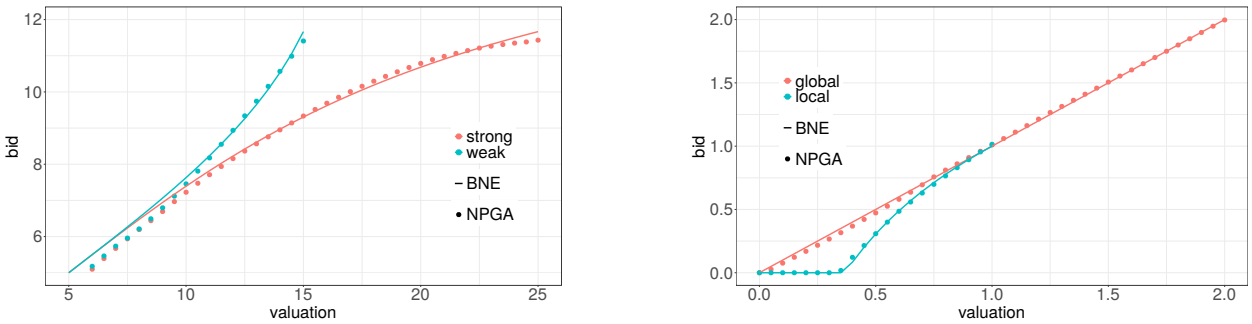


(b) Typical result of bids learned by NPGA (blue dotted line) compared to the BNE strategy (red line) after 20,000 iterations.

Figure 2 **NPGA in 10-bidder FPSB with symmetric Gaussian valuation priors.**

about 10,000 iterations the utility remains relatively stable at around 0.525, however, they are not done learning. The dashed blue line shows the utility when the learned strategy plays against bidders following the analytical BNE: the bidders improve still after iteration 10,000 and converge in utility only after iteration 18,000 to around 0.52.

The resulting bid function is shown in 2b. The dotted blue line indicates the learned strategy and the solid red line indicates the BNE strategy. We often see behavior where bidders do not conform to the equilibrium strategy for low valuations, particularly in settings with a high number of bidders. This results from the fact that in equilibrium, a bidder with a low random valuation will almost never win an auction, and even if she does, the utility gained is minuscule. As a result, the learning with



(a) Asymmetric first-price auction with overlapping valuations.

(b) LLG with nearest-zero core payment rule.

Figure 3 **Learned strategies with NPGA. Solid lines correspond to analytical equilibria and dotted lines to the learned strategy.**

very low valuations converges very slowly to the analytical BNE. However, bidders learn the correct bid level for more frequent valuations very quickly.

6.2.2. Asymmetric Asymmetric prior distributions are harder to analyze analytically. We again chose an environment where the analytical BNE is known with two bidders having uniform prior distributions with overlapping valuations on $(5, 15)$ and $(5, 25)$, respectively (Plum 1992). In equilibrium, the weaker bidder learns to bid more aggressively than the strong bidder.

Table 3 **Average NPGA utilities achieved in asymmetric first-price setting with overlapping valuations. Aggregated over 10 runs of 20,000 iterations each.**

	Utility	**Utility NPGA**		**Rel. loss**
	in BNE	self-play	vs BNE	$\mathcal{L}$
Weak Bidder	0.9694	0.9475	0.9628	0.0069
Strong Bidder	5.0688	5.0754	5.0535	0.0030

NPGA again does not meet the convergence criterion, yet learns a stable strategy yielding good results. The relative utility loss is <0.7% for the strong and the weak bidder. Aggregated performance results over 10 model runs are displayed in Table 3. Figure 3a shows an example of the learned bid functions for both bidders.

Maskin and Riley (2000) and Lebrun (2006) prove that the Bayes–Nash equilibrium of asymmetric first-price auctions is also unique. This uniqueness requires the assumption that a buyer never bids above her value. More recently, Kaplan and Zamir (2015) showed that relaxing this assumption results in additional equilibria (although, no buyer wins by overbidding). We ran NPGA in their setting also with bidders that have non-overlapping uniform valuations, $\mathcal{V} = [0, 5] \times [6, 7]$, see Table 4.

In spite of the equilibrium selection problem in this game, the bidding tends to converge to the lower priced equilibria (denoted BNE 1 and 2 in Kaplan and Zamir (2015)). Only the weaker bidder

Table 4 Average NPGA utilities achieved in asymmetric first-price setting with non-overlapping valuations. Aggregated over 10 runs of 20,000 iterations each. Compared against the first equilibrium of Kaplan and Zamir (2015).

	Utility	**Utility NPGA**		**Rel. loss**
	in BNE	self-play	vs BNE	$\mathcal{L}$
Weak Bidder	0.0334	0.0245	0.0270	0.1940
Strong Bidder	2.0332	1.9456	2.0206	0.0062

has difficulties finding a particular strategy for low valuations, because bids in this range are far from competitive to any opposing bids, which leads to higher errors. The high relative loss of approximately 20% of the weak bidder is due to his strategic disadvantage in this game. She rarely wins and thus rarely has an opportunity to learn in this specific setting.

6.3. Multi-Unit Auctions

A natural extension to single-unit auctions are multi-unit auctions of multiple homogeneous goods. Typically, the valuations of each bidder are assumed i.i.d. and marginally decreasing (Krishna 2009). For a number of multi-unit auction environments the analytical BNE is known. This includes the discriminatory (first-price) auction with $n = m = 2$ (Engelbrecht-Wiggans and Kahn 1998a) and the case with constant marginal valuations for $n = 2$ bidders and an arbitrary number of items m (Anwar 2007). Furthermore, for the uniform-price scheme, symmetric distributions of valuations, $n = 2$ and $m = 3$, where bidders are only interested in two of the units each, the equilibrium strategy can be stated explicitly (Bresky 2008).

Result 3 *In a multi-unit auction in the independent private values model with a VCG or uniform pricing rule NPGA learns a strategy that is payoff equivalent to the BNE ($\mathcal{L} < 0.1\%$); learning converges after only 330 iterations on average. With a discriminatory pricing rule NPGA learns a strategy with an estimated ex-ante utility loss of $\hat{\mathcal{L}} < 3\%$ and the convergence criterion is met after 5,860 iterations on average.*

Table 5 NPGA performance achieved after 5,000 (Discriminatory: 20,000) iterations in standard multi-unit settings with $m = n = 2$. BNE was not available for the discriminatory price auction as it's only known implicitly.

Payment Rule	**n**	**Rel. loss $\mathcal{L}$**	**RMSE**	$\hat{\mathcal{L}}$
Discriminatory	**2**	–	–	0.0245
VCG	**2**	-0.0008	0.0250	0.0003
Uniform	**2**	0.0005	0.0695	0.0000

The case of $n = m = 2$ for the uniform pricing-scheme is of special interest, because there is a continuum of equilibria (Krishna 2009, Bresky 2008, Engelbrecht-Wiggans and Kahn 1998b). Under the uniform payment rule units are sold to the m highest bids all at the same price of the $(k+1)$-highest

bid. For these auctions in general it is known that bidders are incentivized to shade some of their bids. Assuming identically and independently uniformly distributed valuations $\mathcal{V}_i = \{v_i \in [0,1]^2 : v_{i,1} \geq v_{i,2}\}$, a BNE is given by the strategy profile $\beta^* = (\beta_1^*, \beta_2^*)$ with

$$\beta_i^*(v_i) = (v_{i,1}, 0), \quad i = 1, 2.$$

As a result each bidder wins one unit at a price of zero in equilibrium (Krishna 2009, p. 195). Another equilibrium strategy is given by $\beta_i^{**}(v_i) \equiv (1, 0)$, resulting in the same allocations and payments as in the previous equilibrium. Here again, a utility improvement assuming the opponent keeps playing according to β_i^{**} is impossible, since the price for one unit cannot be lowered and trying to win both units would require bidding $(1, 1)$, thus raising the price to one without even being certain of winning both units.

When initializing with (approximately) truthful pre-training, NPGA learns to bid zero on the second unit remarkably fast (within the first 100 iterations), while the first bid remains close to truthful. As soon as this state of cooperation is reached, the value of the first bid becomes extraneous in terms of the payoff. The reward signal in self-play diminishes, because marginal changes in strategy no longer affect the payoff. This prevents convergence and the first bid is exposed to small random adjustments due to the stochasticity in valuations and gradient estimates, especially when learning with Adam. However, there is a mean relative utility loss of only $\mathcal{L} \leq 0.0005$ over ten runs with 5,000 iterations each.[7]

6.4. Combinatorial Auctions

Multi-item auctions of multiple heterogeneous goods have received significant attention in the past decade in the form of combinatorial auctions. In particular, core-selecting combinatorial auctions have become popular for their use in spectrum auctions world-wide (Bichler and Goeree 2017). For several environments an analytical BNE is available. We consider three specific settings of a combinatorial auction. The first two are the well-known local-local-global (LLG) environment (Goeree and Lien 2016) with single-minded bidders, who are only interested in one package. Next, we analyze an extension, the LLLLGG model which was introduced by Bosshard et al. (2020) as a benchmark for equilibrium computation. For the latter no analytical BNE is known. The third model is the split-award procurement auction analyzed in Anton and Yao (1992) and Kokott et al. (2019).

Result 4 *In the LLG auction, NPGA converges to the BNE with a relative utility loss of less than 0.24% in all six settings with core payment rules (known BNE) for independent and dependent values;*

[7] The utility loss $\mathcal{L}$ is sometimes (here for VCG) negative. This minimal error arises because we have a closed-form solution of the BNE strategy, but not of the utility. The sampling of utility can lead to very small errors.

also for a FPSB payment rule it converges with an estimated relative utility loss of <1%. In the LLLLGG auction, NPGA converges in utility and estimated utility loss for both, a first-price and nearest-vcg payment rule. It achieves a relative utility loss of $\hat{\mathcal{L}} < 0.01$ *for both local and global bidders. In the split-award auction, there are multiple BNE, but NPGA converges consistently converges to the payoff-dominant BNE.*

6.4.1. Local-Local-Global Auction (LLG) Ausubel and Baranov (2018) provide closed form solutions for the LLG environment with independent and correlated valuations considering the following payment rules: a VCG rule, a nearest-VCG rule, a nearest-zero (or proxy) rule, and a nearest-bid rule. As in Ausubel and Baranov (2018), we consider the local-local-global (LLG) setting with two local bidders, interested in winning one item only with a uniform prior of $\mathcal{U}(0,1)$, and one global bidder, interested in winning only the two items as perfect complements whose valuations are drawn from $\mathcal{U}(0,2)$. For all three core payment rules (a nearest-VCG rule, a nearest-zero (or proxy) rule, a nearest-bid rule) the global bidder is bidding truthfully in the BNE (Ausubel and Baranov 2018). The local bidders BNE strategies differ in each payment rule.

Figure 3b illustrates the BNE strategy for the nearest-zero payment rule with independent valuations indicated by the lines. The dotted line represents the learned strategy. In general, NPGA converges to the BNE in all core payment rules with independent valuations after only a few hundred iterations (see Table 6) and also converges with correlated valuations (see Table 7). The computation of estimated relative utility loss is not straightforward considering correlated valuations since the sampling of each bidder's valuations depend on each other. Therefore, we did not compute the estimated relative utility loss for this setting and can not report on the convergence criterion being met. However, the more relevant utility loss (playing against BNE) is so low that we assume convergence.

Considering the FPSB without known BNE, the relative utility loss for playing a strategy learned by NPGA is minuscule for independent valuations. Note that we do not know an analytical solution for first-price combinatorial auctions in this settings.

6.4.2. LLLLGG Auction An extension to the LLG auction is the LLLLGG auction with 4 local and 2 global bidders, 6 items and each bidder interested in 2 bundles, for which no analytical BNE are known.

As shown for first-price payment rule in Figure 4, the bidders' utility converges fast to around 0.24 (local bidders) and 0.18 (global bidders) and the utility losses drop quickly. However, NPGA continues learning and the final convergence criterion is met after 4,300 iterations.

Solving the LLLLGG auction considering a nearest-vcg payment rule is much more computational demanding because it requires solving a linear- and a subsequent quadratic optimization problem (Day and Cramton 2012). Because NPGA solves many thousand auctions at once, we implemented a

Table 6 **Results of NPGA in the LLG auctions with independent valuations.[1]Only the weaker 0.0005 criteria is met.**

payments	bidder	$\mathcal{L}$	RMSE	$\hat{\mathcal{L}}$	conv. iters	sec per iter
nearest-vcg	**locals**	0.0011	0.0050	0.0018	320	0.84
	global	0.0000	0.0269	0.0000		
nearest-bid	**locals**	-0.0013	0.0073	0.0026	540	0.79
	global	0.0000	0.0424	0.0000		
nearest-zero	**locals**	-0.0007	0.0078	0.0018	630	0.79
	global	0.0000	0.0088	0.0000		
FPSB	**locals**	–	–	0.0062	850[1]	0.65
	global	–	–	0.0038		

Table 7 **Results of NPGA after in the LLG auctions with correlated valuations.**

payments	bidder	$\mathcal{L}$	RMSE	sec per iter
nearest-vcg	**locals**	-0.0007	0.0042	0.80
	global	0.0000	0.0305	
nearest-bid	**locals**	0.0023	0.0064	0.83
	global	0.0000	0.0498	
nearest-zero	**locals**	0.0006	0.0059	0.81
	global	0.0000	0.0072	

Table 8 **Results of NPGA after 5,000 (1,000) iterations in the LLLLGG first-price (nearest-vcg) auction. Results are averages over 10 (2) replications and the standard deviation displayed in brackets.**

payments	bidder	$\hat{\ell}$	$\hat{\epsilon}$	$\hat{\mathcal{L}}$	sec per iter
first-price	**locals**	0.0015 (0.0003)	0.0110 (0.0025)	0.0085 (0.0015)	0.97
	globals	0.0010 (0.0002)	0.0077 (0.0016)	0.0041 (0.0007)	(0.005)
nearest-vcg	**locals**	0.0013 (0.0003)	0.0052 (0.0012)	0.0065 (0.0016)	275.22
	globals	0.0011 (0.0006)	0.0100 (0.0060)	0.0063 (0.0033)	(0.670)

solver that allows to solve batches of quadratic optimization problems on the GPU. Nonetheless, we had to reduce the number of experiments to run all experiments in a reasonable time.[8]

Even for the much more computational demanding nearest-vcg rule we meet the weaker 0.0005 convergence criterion after only 600 iterations. Both, local and global bidders, show a very small estimated ex-ante relative utility loss of $\hat{\mathcal{L}} < 0.007$, indicating little incentive to deviate from the learned strategy.

[8] We adjusted the parameters for LLLLGG with nearest-vcg (compared to the original). Number of runs: 2 (10), number of training iterations: 1000 (5000 or 20000), population size: 32 (64), training batch size: $H = 2^{14}$ ($H = 2^{18}$), utility loss batch size: $H = 2^{7}$ ($H = 2^{12}$), utility grid size: $W = 2^{8}$ ($W = 2^{13}$)

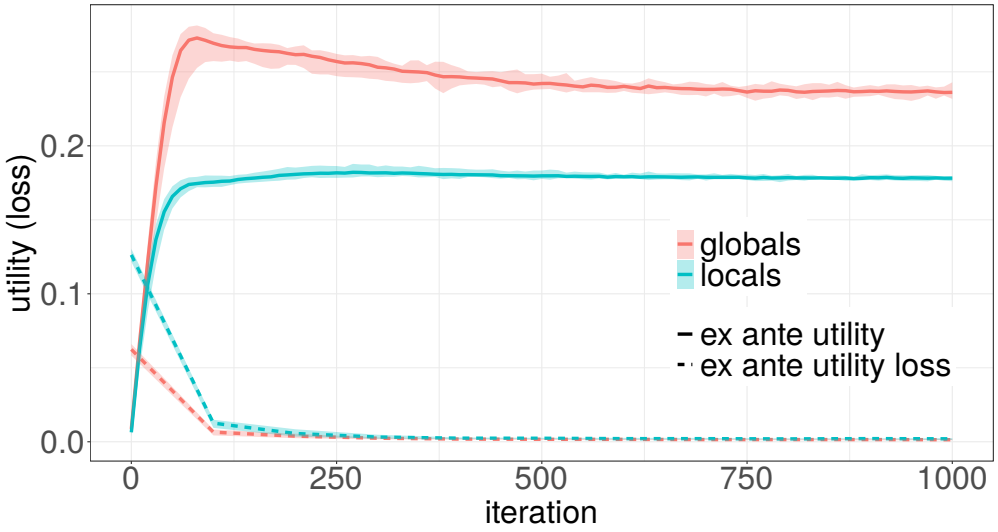


Figure 4 **Ex-ante utility $\tilde{u}$ and loss $\tilde{\ell}$ of in NPGA self-play in the LLLLGG first-price auction. Shaded area and line show min, max, mean over 10 repetitions.**

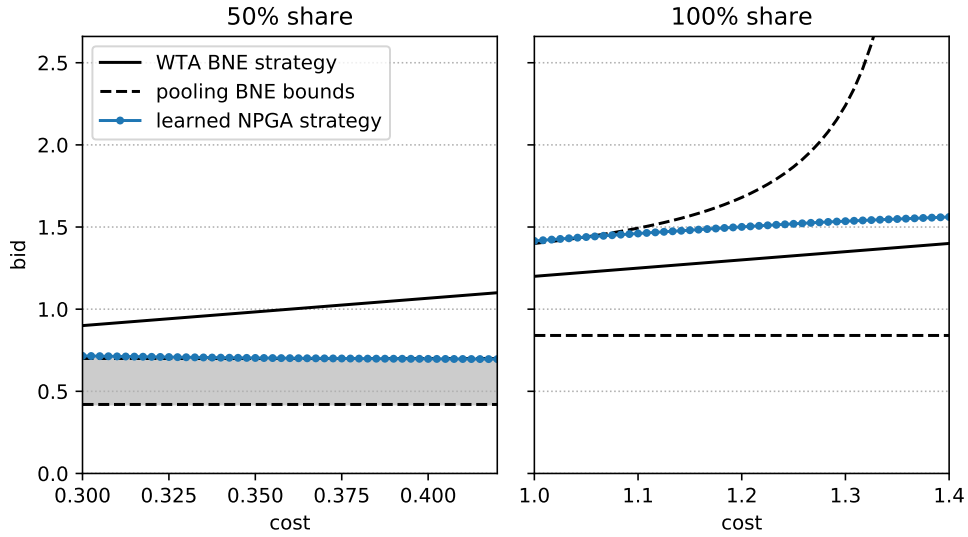


Figure 5 **The figure depicts the WTA equilibrium (solid line), bounds for range of efficient pooling equilibria (grey shaded), and NPGA strategies (dotted lines) for the first-price split-award auction. As the NPGA strategy is within the continuum of efficient pooling equilibria, two bidders playing according to this strategy always end up with a split contract for one lot each.**

6.4.3. Split-Award Auction A particularly interesting environment of a combinatorial auction with multi-minded bidders has been analyzed in Anton and Yao (1992) and later in Kokott et al. (2019). This is an environment with multiple pure BNE, which makes it particularly interesting for our analysis due to the BNE selection problem.

The model is a reverse auction and it is described by the bidders' type (or cost) distribution

$$\mathcal{V}_i = \{v_i \in \mathbb{R}^2 | v_{i,1} \sim F,\ v_{i,2} = C \cdot v_{i,1}\}, \quad i = 1, 2,$$

where $v_{i,1}$ corresponds to the cost of two 50% lots (or items) and the *efficiency parameter* C corresponds to the fraction of total costs for one of the lots.[9] The model describes diseconomies of scale in the production costs. Bayes-Nash equilibria can be categorized into (a) the economically inefficient "winner-takes-all" (WTA) equilibrium where one bidder wins both lots or goods, and (b) a continuum of efficient pooling equilibria (grey shaded in Figure 5) where both suppliers coordinate and each bidder wins one good at a high pooling price (Anton and Yao 1992). The equilibrium with the highest bids on one lot out of all the efficient pooling equilibria is the payoff-dominant strategy for each bidder. Apart from these two pure-strategy Nash equilibria, hybrid equilibria are known to exist and there might even be mixed equilibria in nondeterministic strategies, which makes this setting strategically challenging.

Figure 5 depicts the analytically known equilibria alongside a representative strategy learned via NPGA. Running NPGA multiple times, it always converges to a state close to a pooling pure-strategy BNE: the bidders cooperate in the split equilibrium. Similar to our analysis in the uniform-price auctions in Section 6.3, once the bidders have agreed on cooperation, the non-price-determining bid becomes subject to slight random changes as there is no "reward signal". The relative ex-ante utility loss $\hat{\mathcal{L}}$ falls below 2% (see Table 9). Again, the root mean squared error of the winning bid compared to that of the analytical pure BNE strategy shows close proximity in action space.

Table 9 Results of NPGA in the split-award setting. Standard deviations are stated in parentheses; aggregated over 40 runs of 20,000 iterations each. In all runs the strategies learned were closer to the payoff-dominant pooling equilibrium.

	Utility in BNE	**Utility NPGA**		$\mathcal{L}$	**Winning bid RMSE**
		self-play	vs BNE		
Pooling	0.3400	0.3584 (0.01)	0.3337 (0.00)	0.0185 (0.00)	0.0351 (0.00)

Let us finally make a remark on the auction settings for which Cai and Papadimitriou (2014) showed that finding a BNE is PP-hard. Their paper describes a larger set of sub-additive valuations and a full exploration of different subsets (such as XOS or submodular valuations is beyond the scope of this paper). However, we did explore the most basic CA setting with item-bidding for which the complexity results hold: All bidders have additive valuations, with the exception of one bidder who has unit demand. For up to three items and three bidders, we achieve a estimated relative utility loss $\hat{\mathcal{L}}(\beta_i)$ of less than 2.5%.

[9] In our experiments we set parameters $F = \mathcal{U}(1.0, 1.4)$ with $C = 0.3$, being consistent to previous work on the split-award auction (Kokott et al. 2019). Due to the linear relationship of the costs, the policies can be defined either as $\beta : \mathbb{R} \to \mathbb{R}^2$ or as $\beta : \mathbb{R}^2 \to \mathbb{R}^2$. Experimental results are presented from the first choice.

7. Conclusions

For most auction settings, no Bayes-Nash equilibrium is known and the equilibrium analysis, especially of multi-item auctions, is very challenging. We introduce a very general numerical method to learn Bayes-Nash equilibria in continuous type and action Bayesian games and focus on auctions in our analysis. Neural Pseudogradient Ascent is based on neural networks and self-play. The method implements a gradient ascent algorithm and approximates the expected utility via Monte Carlo sampling where we run many auctions with different value draws with slight perturbations of the weights in the policy network. This allows us to derive an estimate of the payoff gradient, and the individual bidders adapt their policy by taking a step along their payoff gradient.

Training neural networks in such an environment is challenging as the ex-post payoff function for individual batches of auctions played against a fixed set of opponents is discontinuous. Rather than standard backpropagation, we use Evolutionary Strategies to smoothen the utility function and derive the pseudo-gradients of the payoff function. We show conditions for which we can certify that NPGA finds a ε-Bayes-Nash equilibrium. Note that NPGA neither makes assumptions on the parametric form of the utility or bid functions nor on the independence of valuations and it does not require discretization of the value or bid space.

Our experimental results show that Neural Pseudogradient Ascent finds Bayes-Nash equilibria in a large variety of different auction games. NPGA leverages the potential of GPUs to parallelize the Monte Carlo sampling. The method scales well with an increasing number of parameters, finding approximate Bayes-Nash equilibria in auction settings with up to ten bidders and multiple items within minutes on a single GPU. Monte Carlo experiments can be used to compute the relative loss compared to an approximate best response in environments where the analytical solution is unknown and we find very low estimated utility loss in these environments as well. Interestingly, NPGA converged even in those environments where payoff monotonicity cannot hold (i.e., those with multiple pure Bayes-Nash equilibria). These results stand in contrast to the worst-case complexity results in Cai and Papadimitriou (2014), for which NPGA also converged in initial experiments. We conjecture that for auction games with well-behaved priors and quasi-linear utility functions, pure Bayes-Nash equilibria describe large "areas of attraction" in the utility landscape such that the likelihood is high for gradient dynamics to converge to payoff-dominant Bayes-Nash equilibria even though it can be very hard to find equilibria in the worst case. A comprehensive analysis of conditions when we can expect gradient dynamics to converge in auction games in general is obviously challenging and beyond a single article. Apart from its practical relevance in the equilibrium analysis of auction games, our results reemphasize the importance of research on gradient dynamics in games.

References

J. J. Anton and D. A. Yao. Coordination in split award auctions. *The Quarterly Journal of Economics*, 107 (2):681–707, 1992.

A. W. Anwar. Equilibria in Multi-Unit Discriminatory Auctions. *The B.E. Journal of Theoretical Economics*, 7(1), Jan. 2007. ISSN 1935-1704. doi: 10.2202/1935-1704.1327. URL https://www.degruyter.com/view/j/bejte.2007.7.1/bejte.2007.7.1.1327/bejte.2007.7.1.1327.xml.

O. Armantier, J.-P. Florens, and J.-F. Richard. Approximation of nash equilibria in bayesian games. *Journal of Applied Econometrics*, 23(7):965–981, 2008.

S. Athey. Single crossing properties and the existence of pure strategy equilibria in games of incomplete information. *Econometrica*, 69(4):861–889, 2001.

L. M. Ausubel and O. Baranov. Core-Selecting Auctions with Incomplete Information. page 23, 2018.

F. Bach. Breaking the Curse of Dimensionality with Convex Neural Networks. *Journal of Machine Learning Research*, 18(19):1–53, 2017. ISSN 1533-7928. URL http://jmlr.org/papers/v18/14-546.html.

P. Bajari. Comparing competition and collusion: a numerical approach. *Economic Theory*, 18(1):187–205, 2001.

D. Balduzzi, S. Racaniere, J. Martens, J. Foerster, K. Tuyls, and T. Graepel. The mechanics of n-player differentiable games. *arXiv preprint arXiv:1802.05642*, 2018.

M. Bichler and J. K. Goeree. *Handbook of spectrum auction design*. Cambridge University Press, 2017.

A. Blum and Y. Mansour. Learning, Regret Minimization, and Equilibria. In N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, editors, *Algorithmic Game Theory*, pages 79–102. Cambridge University Press, Cambridge, 2007. ISBN 978-0-511-80048-1. doi: 10.1017/CBO9780511800481.006. URL https://www.cambridge.org/core/product/identifier/CBO9780511800481A051/type/book_part.

V. Bosshard, B. Bünz, B. Lubin, and S. Seuken. Computing bayes-nash equilibria in combinatorial auctions with continuous value and action spaces. In *IJCAI*, pages 119–127, 2017.

V. Bosshard, B. Bünz, B. Lubin, and S. Seuken. Computing Bayes-Nash Equilibria in Combinatorial Auctions with Verification. *Journal of Artificial Intelligence Research*, Dec. 2020. URL http://arxiv.org/abs/1812.01955.

M. Bowling. Convergence and no-regret in multiagent learning. In *Advances in neural information processing systems*, pages 209–216, 2005.

M. Bresky. Properties of Equilibrium Strategies in Multiple-Unit, Uniform-Price Auctions. *SSRN Electronic Journal*, 2008. ISSN 1556-5068. doi: 10.2139/ssrn.1488832. URL http://www.ssrn.com/abstract=1488832.

G. W. Brown. Iterative solution of games by fictitious play. *Activity analysis of production and allocation*, 13 (1):374–376, 1951.

N. Brown and T. Sandholm. Superhuman AI for multiplayer poker. *Science*, 365(6456):885–890, Aug. 2019. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.aay2400. URL https://science.sciencemag.org/content/365/6456/885.

Y. Cai and C. Papadimitriou. Simultaneous Bayesian auctions and computational complexity. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 895–910, 2014.

O. Carbonell-Nicolau and R. P. McLean. On the existence of nash equilibrium in bayesian games. *Mathematics of Operations Research*, 43(1):100–129, 2018.

V. Conitzer and T. Sandholm. New complexity results about nash equilibria. *Games and Economic Behavior*, 63(2):621–641, 2008.

A.-A. Cournot. *Recherches sur les principes mathématiques de la théorie des richesses.* 1838. URL https://gallica.bnf.fr/ark:/12148/bpt6k6117257c.

C. Daskalakis, P. Goldberg, and C. Papadimitriou. The Complexity of Computing a Nash Equilibrium. *SIAM Journal on Computing*, 39(1):195–259, Jan. 2009. ISSN 0097-5397. doi: 10.1137/070699652. URL https://epubs.siam.org/doi/abs/10.1137/070699652.

C. Daskalakis, A. Ilyas, V. Syrgkanis, and H. Zeng. Training gans with optimism. *arXiv preprint arXiv:1711.00141*, 2017.

R. W. Day and P. Cramton. Quadratic Core-Selecting Payment Rules for Combinatorial Auctions. *Operations Research*, 60(3):588–603, June 2012. ISSN 0030-364X. doi: 10.1287/opre.1110.1024. URL https://pubsonline.informs.org/doi/abs/10.1287/opre.1110.1024.

R. Engelbrecht-Wiggans and C. M. Kahn. Multi-Unit Pay-Your-Bid Auctions with Variable Awards. *Games and Economic Behavior*, 23(1):25–42, 1998a. ISSN 08998256. doi: 10.1006/game.1997.0599. URL https://linkinghub.elsevier.com/retrieve/pii/S0899825697905996.

R. Engelbrecht-Wiggans and C. M. Kahn. Multi-unit auctions with uniform prices. *Economic Theory*, 12(2): 227–258, 1998b.

K. Etessami and M. Yannakakis. On the complexity of nash equilibria and other fixed points (extended abstract). In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '07, page 113–123, USA, 2007. IEEE Computer Society. ISBN 0769530109. doi: 10.1109/FOCS.2007.48. URL https://doi.org/10.1109/FOCS.2007.48.

D. J. Foster, Z. Li, T. Lykouris, K. Sridharan, and E. Tardos. Learning in games: Robustness of fast convergence. In *Advances in Neural Information Processing Systems*, pages 4734–4742, 2016.

D. P. Foster and R. V. Vohra. Calibrated learning and correlated equilibrium. *Games and Economic Behavior*, 21(1-2):40, 1997.

D. Fudenberg and D. K. Levine. Learning and equilibrium. *Annu. Rev. Econ.*, 1(1):385–420, 2009.

J. K. Goeree and Y. Lien. On the impossibility of core-selecting auctions. *Theoretical Economics*, 11(1): 41–52, 2016. ISSN 1555-7561. doi: 10.3982/TE1198. URL https://onlinelibrary.wiley.com/doi/abs/10.3982/TE1198.

J. Hartline, V. Syrgkanis, and E. Tardos. No-Regret Learning in Bayesian Games. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3061–3069. Curran Associates, Inc., 2015. URL http://papers.nips.cc/paper/6016-no-regret-learning-in-bayesian-games.pdf.

E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

A. Jafari, A. Greenwald, D. Gondek, and G. Ercal. On no-regret learning, fictitious play, and nash equilibrium. In *ICML*, volume 1, pages 226–233, 2001.

T. R. Kaplan and S. Zamir. Multiple equilibria in asymmetric first-price auctions. *Economic Theory Bulletin*, 3(1):65–77, 2015.

D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, Jan. 2015. URL http://arxiv.org/abs/1412.6980. arXiv: 1412.6980.

G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter. Self-Normalizing Neural Networks. *arXiv:1706.02515 [cs, stat]*, June 2017. URL http://arxiv.org/abs/1706.02515.

P. Klemperer. Why every economist should learn some auction theory. *Available at SSRN 241350*, 2000.

P. Klemperer. *Auctions: theory and practice*. Princeton University Press, 2004.

G.-M. Kokott, M. Bichler, and P. Paulsen. The beauty of Dutch: Ex-post split-award auctions in procurement markets with diseconomies of scale. *European Journal of Operational Research*, 278(1):202–210, 2019.

V. Krishna. *Auction Theory*. Academic press, 2009.

B. Lebrun. Uniqueness of the equilibrium in first-price auctions. *Games and Economic Behavior*, 55(1): 131–151, 2006.

A. Letcher, D. Balduzzi, S. Racanière, J. Martens, J. N. Foerster, K. Tuyls, and T. Graepel. Differentiable game mechanics. *Journal of Machine Learning Research*, 20(84):1–40, 2019.

R. C. Marshall, M. J. Meurer, J.-F. Richard, and W. Stromquist. Numerical analysis of asymmetric first price auctions. *Games and Economic Behavior*, 7(2):193–220, 1994.

E. Maskin and J. Riley. Asymmetric auctions. *The review of economic studies*, 67(3):413–438, 2000.

F. M. Menezes and P. K. Monteiro. *An Introduction to Auction Theory*. OUP Oxford, 2005.

P. Mertikopoulos and Z. Zhou. Learning in games with continuous action sets and unknown payoff functions. *Mathematical Programming*, 173(1-2):465–507, 2019.

P. Milgrom. *Discovering prices: auction design in markets with complex constraints*. Columbia University Press, 2017.

D. Monderer and L. S. Shapley. Potential games. *Games and economic behavior*, 14(1):124–143, 1996.

J. F. Nash et al. Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 36 (1):48–49, 1950.

T. Nedelec, N. El Karoui, and V. Perchet. Learning to bid in revenue maximizing auction. In *Companion Proceedings of The 2019 World Wide Web Conference*, pages 934–935, 2019.

Y. Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical programming*, 120(1): 221–259, 2009.

A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

M. Plum. Characterization and computation of Nash-equilibria for auctions with incomplete information. *International Journal of Game Theory*, 20(4):393–418, 1992.

A. Rubinstein. Settling the complexity of computing approximate two-player Nash equilibria. *arXiv:1606.04550 [cs]*, June 2016. URL `http://arxiv.org/abs/1606.04550`.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, Oct. 1986. ISSN 1476-4687. doi: 10.1038/323533a0. URL `https://www.nature.com/articles/323533a0`. Number: 6088 Publisher: Nature Publishing Group.

T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever. Evolution Strategies as a Scalable Alternative to Reinforcement Learning. *arXiv:1703.03864 [cs, stat]*, Mar. 2017. URL `http://arxiv.org/abs/1703.03864`.

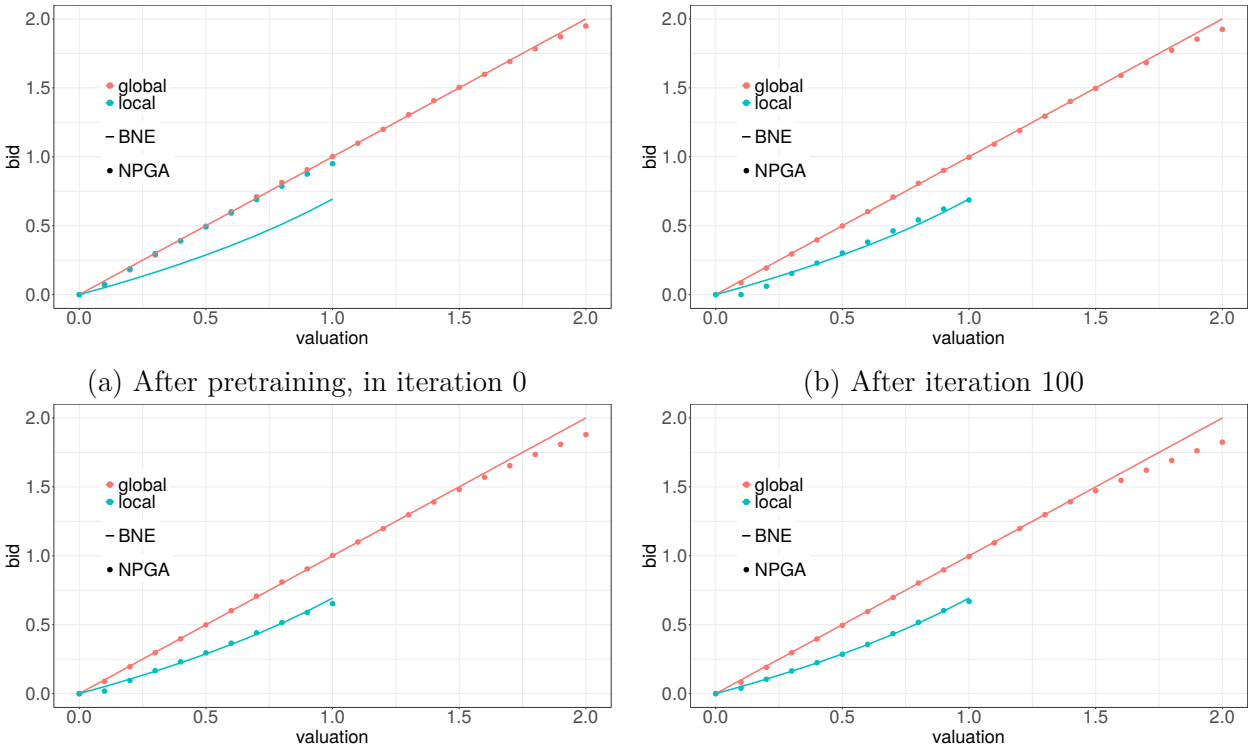
F. Schäfer and A. Anandkumar. Competitive gradient descent. In *Advances in Neural Information Processing Systems*, pages 7623–7633, 2019.

D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419):1140–1144, Dec. 2018. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.aar6404. URL `http://www.sciencemag.org/lookup/doi/10.1126/science.aar6404`.

S. Singh, M. Kearns, and Y. Mansour. Nash Convergence of Gradient Dynamics in Iterated General-Sum Games. *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI2000)*, June 2000a. URL `http://arxiv.org/abs/1301.3892`.

S. P. Singh, M. J. Kearns, and Y. Mansour. Nash convergence of gradient dynamics in general-sum games. In *UAI*, pages 541–548, 2000b.

G. Stoltz and G. Lugosi. Learning correlated equilibria in games with compact sets of strategies. *Games and Economic Behavior*, 59(1):187–208, 2007.



(a) After pretraining, in iteration 0

(b) After iteration 100

(c) After iteration 1000

(d) After iteration 5000

Figure 6 **Learned strategies in LLG with nearest-bid payment Rule. Solid lines correspond to analytical equilibria and dotted lines to the learned strategy.**

T. Ui. Bayesian nash equilibrium and variational inequalities. *Journal of Mathematical Economics*, 63: 139–146, 2016.

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

Y. Viossat and A. Zapechelnyuk. No-regret dynamics and fictitious play. *Journal of Economic Theory*, 148 (2):825–842, 2013.

M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 928–936, 2003.

Appendix A: Summary Table

Experiment	Valuations	n	$\mathcal{L}$	RMSE	estimated util. loss			conv.	time
					$(\hat{\ell})$	$(\hat{\epsilon})$	$(\hat{\mathcal{L}})$	(iters)	(sec/iter)
Single-Item Symmetric	**Uniform** $\mathcal{U}(0,10)$ $\rho=1$	**2**	0.0001	0.0072	0.0011	0.0059	0.0065	1,350	0.31
		3	0.0017	0.0104	0.0007	0.0051	0.0082	870	0.40
		5	0.0034	0.0194	0.0005	0.0053	0.0138	900	0.46
		10	0.0084	0.0303	0.0003	0.0047	0.0288	720	0.73
	Uniform[1] $\mathcal{U}(0,10)$ $\rho=0.5$	**2**	0.0011	0.0057	0.0012	0.0065	0.0051	5,720	0.46
		3	0.0006	0.0069	0.0008	0.0048	0.0062	2,100	0.52
		5	0.0012	0.0161	0.0006	0.0066	0.0096	4,170	0.63
		10	0.0100	0.0383	0.0005	0.0085	0.0212	8,410	0.93
	Gaussian[1] $\mathcal{N}(15,100)$ $\rho=1$	**2**	0.0015	0.3684	0.0443	0.4394	0.0083	–	0.31
		3	0.0037	0.4478	0.0225	0.9723	0.0082	$3{,}800_2$[2]	0.39
		5	0.0129	0.8819	0.0176	1.7324	0.0135	$3{,}825_5$[2]	0.45
		10	0.0314	1.8801	0.0118	2.1660	0.0220	$4{,}300_8$[2]	0.68
1-Item Asym.[1] **Overlapping**	$\mathcal{U}(5,15)$	**weak**	0.0069	0.3178	0.0403	0.2365	0.0410	–	0.61
	$\mathcal{U}(5,25)$	**strong**	0.0030	0.2872	0.0379	0.1897	0.0074		
1-Item Asym.[1] **Non-Overlapping**	$\mathcal{U}(0,5)$	**weak**	0.1943	1.1534	0.0006	0.0281	0.0231	3,530	0.71
	$\mathcal{U}(6,7)$	**strong**	0.0062	0.1426	0.0000	0.0000	0.0000		
Multi-Unit	**Discriminatory**[1]	**2**	–	–	0.0087	0.0570	0.0245	5,860	1.14
	Uniform	**2**	0.0005	0.0695	0.0000	0.0000	0.0000	330	1.43
	VCG	**2**	-0.0008[3]	0.0250	0.0001	0.0031	0.0003	310	1.54
Combinatorial LLG independent values $(\gamma=0.0)$	**nearest-vcg**	**locals**	0.0011	0.0050	0.0002	0.0009	0.0018	320	0.84
		global	0.0000	0.0269	0.0000	0.0001	0.0000		
	nearest-bid	**locals**	-0.0013[3]	0.0073	0.0003	0.0013	0.0026	540	0.79
		global	0.0000	0.0424	0.0000	0.0001	0.0000		
	nearest-zero	**locals**	-0.0007[3]	0.0078	0.0002	0.0019	0.0018	630	0.79
		global	0.0000	0.0088	0.0000	0.0001	0.0000		
	FPSB	**locals**	–	–	0.0009	0.0031	0.0062	850^2	0.65
		global	–	–	0.0016	0.0064	0.0038		
Combinatorial LLG correlated values $(\gamma=0.5)$	**nearest-vcg**	**locals**	-0.0007[3]	0.0042	–	–	–	–	0.80
		global	0.0000	0.0305	–	–	–		
	nearest-bid	**locals**	0.0023	0.0064	–	–	–	–	0.83
		global	0.0000	0.0498	–	–	–		
	nearest-zero	**locals**	0.0006	0.0059	–	–	–	–	0.81
		global	0.0000	0.0072	–	–	–		
Combinatorial LLLLGG	**FPSB**	**locals**	–	–	0.0015	0.0109	0.0085	4,300	0.97
		globals	–	–	0.0010	0.0077	0.0040		
	nearest-vcg	**locals**	–	–	0.0013	0.0052	0.0065	600^2	275.00
		globals	–	–	0.0011	0.0098	0.0063		
Combinatorial[1] **Split-Award**	**Pooling**	**2**	0.0185	0.0351[4]	0.0011	0.0124	0.0031	$3{,}497_{39}$[2]	0.73

Table 10 Summary of all experiments' results averaged over 10 runs (except Split-Award is averaged over 40 runs). If not all runs satisfied the convergence criterion, the number of runs that fulfilled the criterion are noted as subscript [1]Setting performed with 20,000 iterations instead of 5,000. [2]Only the weaker 0.0005 criteria is met. [3]Relative utility loss is negative only due to sampling variance. It must be $\mathcal{L} \geq 0$. [4]Only the relevant bid component is used.

Appendix B: Additional Proofs

Lemma 1 *The gradient estimates ∇^{ES} in NPGA are unbiased and have finite mean squared error with respect to the smoothed utilities $\tilde{u}_i^\sigma$ of the game Γ^σ.*

Proof. We consider the smoothed ex-ante utility $\tilde{u}_i^\sigma$. For fixed $\sigma > 0$, we have

$$\tilde{u}_i^\sigma(\theta_i, \theta_{-i}) := \mathbb{E}_{\varepsilon \sim \mathcal{N}(0,\sigma^2 I)}[\tilde{u}_i(\theta_i + \varepsilon, \theta_{-i})].$$

This is equal to the convolution of $\tilde{u}_i$ with a Gaussian kernel in the i-th coordinate. As was noted by Salimans et al. (2017), its (exact) gradient with respect to θ_i is thus given by

$$\nabla_{\theta_i} \tilde{u}_i^\sigma(\theta_i, \theta_{-i}) = \frac{1}{\sigma} \mathbb{E}_{\varepsilon \sim \mathcal{N}(0,I)}[\varepsilon(\tilde{u}_i(\theta_i + \sigma\varepsilon, \theta_{-i}) - \tilde{u}_i(\theta_i, \theta_{-i}))].$$

By the substitution $\varepsilon' = \sigma\varepsilon$, we see by the transformation formula that

$$\nabla_{\theta_i} \tilde{u}_i^\sigma(\theta_i, \theta_{-i}) = \frac{1}{\sigma^2} \mathbb{E}_{\varepsilon \sim \mathcal{N}(0,\sigma^2 I)}[\varepsilon(\tilde{u}_i(\theta_i + \varepsilon, \theta_{-i}) - \tilde{u}_i(\theta_i, \theta_{-i}))].$$

If we approximate this term by taking P independent samples $\varepsilon_p \sim \mathcal{N}(0, \sigma^2 I)$, we get

$$\nabla_{\theta_i} \tilde{u}_i^\sigma(\theta_i, \theta_{-i}) \approx \frac{1}{P\sigma^2} \sum_p \varepsilon_p(\tilde{u}_i(\theta_i + \sigma\varepsilon_p, \theta_{-i}) - \tilde{u}_i(\theta_i, \theta_{-i})).$$

In the same way, we can approximate $\tilde{u}_i$ by sampling H valuation profiles v_h with respect to the distribution the valuations are drawn from:

$$\tilde{u}_i(\theta_i + \sigma\varepsilon_p, \theta_{-i}) \approx \frac{1}{H} \sum_h u_i(v_{h,i}, \pi_i(v_{h,i}, \theta_i + \sigma\varepsilon_p), \pi_{-i}(v_{h,-i}, \theta_{-i})).$$

The combination of these approximations is exactly how ∇^{ES} is computed in Algorithm 2:

$$\nabla^{ES}\tilde{u}_i(\theta_i, \theta_{-i}) = \frac{1}{PH\sigma^2} \sum_p \varepsilon_p \sum_h u_i(v_{h,i}, \pi_i(v_{h,i}, \theta_i + \sigma\varepsilon_p), \pi_{-i}(v_{h,-i}, \theta_{-i})) - u_i(v_{h,i}, \pi_i(v_{h,i}, \theta_i), \pi_{-i}(v_{h,-i}, \theta_{-i})).$$

Since we sample independently and with respect to the original distributions, the approximation is in expectation equal to the true gradient. Thus, the approximation is unbiased with respect to the smoothed utilities $\tilde{u}_i^\sigma$. ∇^{ES} also has finite mean squared error: Define

$$X_{p,h} = \varepsilon_p \left(u_i(v_{h,i}, \pi_i(v_{h,i}, \theta_i + \varepsilon_p), \pi_{-i}(v_{h,-i}, \theta_{-i})) - u_i(v_{h,i}, \pi_i(v_{h,i}, \theta_i), \pi_{-i}(v_{h,-i}, \theta_{-i}))\right).$$

Because of Equation (8) in Definition 1, we have

$$\mathbb{E}_v[u_i(v_{h,i}, \pi_i(v_{h,i}, \theta_i + \varepsilon_p), \pi_{-i}(v_{h,-i}, \theta_{-i}))^2] \leq S \text{ and}$$
$$\mathbb{E}_v[u_i(v_{h,i}, \pi_i(v_{h,i}, \theta_i), \pi_{-i}(v_{h,-i}, \theta_{-i}))^2] \leq S.$$

This implies $\mathbb{E}[X_{p,h}^2] \leq 4S\,\mathbb{E}[\|\varepsilon\|^2] = 4Sd_i\sigma^2$, where we used the inequality $(a-b)^2 \leq 2a^2 + 2b^2$. Since $\nabla^{ES}\tilde{u}_i(\theta_i, \theta_{-i}) = \frac{1}{PH\sigma^2}\sum_{p,h} X_{p,h}$, we have that

$$\mathbb{E}\left[\nabla^{ES}\tilde{u}_i(\theta_i, \theta_{-i})^2\right] = \frac{1}{P^2H^2\sigma^2}\,\mathbb{E}\left[\left(\sum_{p,h} X_{p,h}\right)^2\right] = \frac{1}{\sigma^2}\,\mathbb{E}\left[\left(\sum_{p,h}\frac{X_{p,h}}{PH}\right)^2\right] \leq$$
$$\leq \frac{1}{PH\sigma^2}\,\mathbb{E}\left[\sum_{p,h} X_{p,h}^2\right] \leq \frac{1}{PH\sigma^2} 4PHd_i\sigma^2 S = 4Sd_i < \infty.$$

Consequently, our gradient estimate has finite mean squared error. ∎

Lemma 2 *Consider the utility loss $\tilde{\ell}_i^{\Gamma}(\theta_i, \theta_{-i})$ of agent i with respect to the utility function $\tilde{u}_i$ in the finite-dimensional game Γ, and the utility loss $\tilde{\ell}_i^{\sigma}(\theta_i, \theta_{-i})$ with respect to the smoothed utility $\tilde{u}_i^{\sigma}$ in the game Γ^{σ}. Then*

$$\tilde{\ell}_i^{\Gamma}(\theta_i, \theta_{-i}) \leq \tilde{\ell}_i^{\sigma}(\theta_i, \theta_{-i}) + 2ZL\sqrt{d_i}\sigma.$$

Proof. We start by bounding the difference between the utilities of the game Γ and the game Γ^{σ}. To be precise, we prove the following bound:

$$|\tilde{u}_i(\theta_i, \theta_{-i}) - \tilde{u}_i^{\sigma}(\theta_i, \theta_{-i})| \leq ZL\sqrt{d_i}\sigma \tag{22}$$

for arbitrary strategies θ. By definition, $\tilde{u}_i^{\sigma}(\theta_i, \theta_{-i}) = \mathbb{E}_{\varepsilon \sim \mathcal{N}(0,\sigma^2 I)}[\tilde{u}_i(\theta_i + \varepsilon, \theta_{-i})]$. Since $\tilde{u}_i(\theta_i, \theta_{-i}) = \mathbb{E}_{\varepsilon \sim \mathcal{N}(0,\sigma^2 I)}[\tilde{u}_i(\theta_i, \theta_{-i})]$, we have the inequality

$$|\tilde{u}_i(\theta_i, \theta_{-i}) - \tilde{u}_i^{\sigma}(\theta_i, \theta_{-i})| \leq \mathbb{E}_{\varepsilon \sim \mathcal{N}(0,\sigma^2 I)}[|\tilde{u}_i(\theta_i + \varepsilon, \theta_{-i}) - \tilde{u}_i(\theta_i, \theta_{-i})|]. \tag{23}$$

Next, we show that for fixed ε, $|\tilde{u}_i(\theta_i + \varepsilon, \theta_{-i}) - \tilde{u}_i(\theta_i, \theta_{-i})| \leq ZL\|\varepsilon\|$. We compute

$$|\tilde{u}_i(\theta_i + \varepsilon, \theta_{-i}) - \tilde{u}_i(\theta_i, \theta_{-i})| \leq \mathbb{E}_{v_i}\left[|\bar{u}_i(v_i, \pi_i(v_i, \theta_i + \varepsilon), \theta_{-i}) - \bar{u}_i(v_i, \pi_i(v_i, \theta_i), \theta_{-i})|\right]$$

Since by assumption, $\bar{u}_i$ is differentiable with respect to b_i and the differential is uniformly bounded by Z (Equation (7) in Definition 1), we have for every ε

$$|\bar{u}_i(v_i, \pi_i(v_i, \theta_i + \varepsilon), \theta_{-i}) - \bar{u}_i(v_i, \pi_i(v_i, \theta_i), \theta_{-i})| \leq \left\| \frac{\partial \bar{u}_i}{\partial b_i} \right\|_{\infty} \|\pi_i(v_i, \theta_i + \varepsilon) - \pi_i(v_i, \theta_i)\| \leq Z\|\pi_i(v_i, \theta_i + \varepsilon) - \pi_i(v_i, \theta_i)\|.$$

Consequently, by Assumption 11 in Definition 5,

$$|\tilde{u}_i(\theta_i + \varepsilon, \theta_{-i}) - \tilde{u}_i(\theta_i, \theta_{-i})| \leq Z\,\mathbb{E}_{v_i}\left[\|\pi_i(v_i, \theta_i + \varepsilon) - \pi_i(v_i, \theta_i)\|\right] \leq ZL\|\varepsilon\|$$

which implies by Equation (23)

$$|\tilde{u}_i(\theta_i + \varepsilon, \theta_{-i}) - \tilde{u}_i^{\sigma}(\theta_i, \theta_{-i})| \leq ZL\,\mathbb{E}_{\varepsilon \sim \mathcal{N}(0,\sigma^2 I)}[\|\varepsilon\|] \leq ZL\sqrt{d_i}\sigma.$$

This proves equation (22). Now let $\tilde{\theta}_i$ be a best response to θ_{-i} in the game Γ. Then

$$\begin{aligned}
&\tilde{\ell}_i^{\Gamma}(\theta_i, \theta_{-i}) = \tilde{u}_i(\theta_i^*, \theta_{-i}) - \tilde{u}_i(\theta_i, \theta_{-i}) = \\
&= (\tilde{u}_i(\theta_i^*, \theta_{-i}) - \tilde{u}_i^{\sigma}(\theta_i^*, \theta_{-i})) + (\tilde{u}_i^{\sigma}(\theta_i^*, \theta_{-i}) - \tilde{u}_i^{\sigma}(\theta_i, \theta_{-i})) + (\tilde{u}_i^{\sigma}(\theta_i, \theta_{-i}) - \tilde{u}_i(\theta_i, \theta_{-i})) \leq \\
&\leq ZL\sqrt{d_i}\sigma + \tilde{\ell}_i^{\sigma}(\theta_i, \theta_{-i}) + ZL\sqrt{d_i}\sigma = \tilde{\ell}_i^{\sigma}(\theta_i, \theta_{-i}) + 2ZL\sqrt{d_i}\sigma.
\end{aligned}$$

■

Acknowledgments

We're grateful for funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – BI 1057/1-8. We thank Vitor Bosshard, Ben Lubin, Panayotis Mertikopoulos, Sven Seuken, Takashi Ui, and Felipe Maldonado for valuable feedback on earlier versions, and two former students in our group: Kevin Falkenstein, for a separate implementation of initial algorithms, and Anne Christopher, for developing a fast batched QP solver which was applied in computing the LLLLGG auctions. All errors are of course ours.