NEURAL DESIGN OF CONTESTS AND ALL-PAY AUCTIONS USING MULTI-AGENT SIMULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose a multi-agent learning approach for designing crowdsourcing contests and all-pay auctions. Prizes in contests incentivise contestants to expend effort on their entries, with different prize allocations resulting in different incentives and bidding behaviors. In contrast to auctions designed manually by economists, our method searches the possible design space using a simulation of the multi-agent learning process, and can thus handle settings where a game-theoretic equilibrium analysis is not tractable. Our method simulates agent learning in contests and evaluates the utility of the resulting outcome for the auctioneer. Given a large contest design space, we assess through simulation many possible contest designs within the space, and fit a neural network to predict outcomes for previously untested contest designs. Finally, we apply mirror descent to optimize the design so as to achieve more desirable outcomes. Our empirical analysis shows our approach closely matches the optimal outcomes in settings where the equilibrium is known, and can produce high quality designs in settings where the equilibrium strategies are not solvable analytically.

1 INTRODUCTION

Many economic allocation decisions are determined by a competition for a prize based on expending costly efforts. For example, multiple political candidates may engage in costly political campaigns, but only one candidate wins; though only the winner is rewarded, other candidates cannot recover their expenditure. Similarly, Netflix offered a prize of one million dollars in an open competition to improve its recommender system (Bell & Koren, 2007). Again, only the winning entry gets the prize, but other participants incur the cost of their effort. Such contests are modelled in the economic literature as *all-pay auctions* (Vojnović, 2015), where players simultaneously bid for a fixed prize; the highest bidder receives the prize, and *every* player, including non-winners, pay their bid.

A key question regarding crowdsourcing contests (or equivalently, all-pay auctions) is how to design the contest to optimize the utility achieved by the principal. [1] For instance, should the principal give all the reward to the top entry, or does it make sense to give some of the reward to the top entry, and some for the second best? Earlier research has investigated how different contest designs affect the utility of the principal (Archak & Sundararajan, 2009; DiPalantino & Vojnovic, 2009; Gao et al., 2012; Chawla et al., 2015). Such work examines a specific model of the all-pay auction given as a normal-form game and analytically solves for the Nash equilibrium of the bidding strategy, expressed as a probability distribution over the possible bids. This approach has multiple limitations. First, economists have only managed to solve for the Nash equilibrium under very specific auction designs. Secondly, in many settings, participants are likely to adjust their bidding strategy by using simple learning behaviors based on their experience (Gneezy & Smorodinsky, 2006; Anderson et al., 1998; Nanduri & Das, 2007), so one cannot always assume the Nash equilibrium behaviour as a model of participants' behavior when designing the contest.

Our Contribution: We propose a **machine learning method for designing crowdsourcing contests**, investigating how the principal's utility is affected by the reward allocation. By simulating the behavior of learning participants, and predicting the outcomes of contests with a neural network, our approach constructs a differentiable model for the principal's utility under various contest designs.

[1] The all-pay auction literature sometimes refers to the principal as the *auctioneer*.

Given the model, we then optimize the design by employing mirror descent (Beck & Teboulle, 2003; Nemirovsky & Yudin, 1983), which allows optimizing the design while adhering to the fixed budget of the principal. Our approach is flexible: it can be applied to arbitrary mechanism design problems, including analytically intractable settings. It allows using various models for the behavior of participants. We apply fictitious play (FP) (Brown, 1951; Fudenberg & Kreps, 1993) or independent reinforcement learning (Littman, 1994; Hu et al., 1998; Bu et al., 2008).

We **empirically evaluate our framework on several auction design problems**. We study optimally allocating a fixed reward budget in the setting of contests with rank-order allocation of prizes, where the utility of a submission has diminishing returns in effort.

We first examine contests with few participants and no performance noise, for which earlier research characterized the Nash equilibrium behavior (Baye et al., 1996; Cohen & Sela, 2008; Sisak, 2009). We find that simulating participants' behavior using fictitious play closely agrees with the Nash equilibrium prediction, [2] and our framework identifies a contest design near the optimal design prescribed by the economic equilibrium analysis. [3]

Finally, we investigate contests where the performance of a participant's entry is determined by their exerted effort perturbed by random noise. Such uncertainty can be a more realistic model of contests, but the Nash equilibrium behavior is unknown, highlighting the advantage of using our simulation based approach. We show that designs with multiple prizes outperform awarding a single first prize in terms of principal utility. As the variance of the random noise grows, we find that the optimal designs award larger second prizes, acting to protect bidders against the effect of the noise.

1.1 Optimization Goal and Contest Design Space

We consider maximizing the principal's utility in a crowdsourcing contest (or equivalently, the revenue of the auctioneer in an all-pay auction). We examine contests that award multiple prizes based on the rank ordering of the performance of the participants. For instance, a contest may award a large first prize to the best performing contestant, and a smaller runner-up prize to the second best performer. Offering more prizes could incentivise more participants to exert effort, however a smaller top prize means that the maximum bid possible is also reduced.

Consider a contest with n bidders. The principal decides on a division of a fixed total prize $\bar{w}$. The prize awarded to the k^{th} ranked player is denoted w_k, so $\sum_{k=1}^{n} w_k = \bar{w}$. We insist that prizes are decreasing with rank, i.e. that $w_1 \geq w_2 \geq ... \geq w_n$. Awarding a last-place prize only reduces performance at equilibrium, as it reduces the incentive to exert more effort than other bidders, so we set $w_n = 0$. Bidders each choose a bid (their effort level), and the vector of all bids is denoted as $\mathbf{b}$. Effort is costly, so the payoff for bidder i is their prize winnings minus their effort:

$$s_i(\mathbf{b}) = \sum_{j=1}^{n} w_j x_{i,j}(\mathbf{b}) - b_i$$

where $x_{i,j}(\mathbf{b}) = 1$ when player i's submission is ranked j^{th} in terms of its quality, and 0 otherwise. Allocation is based on the ranking of the *realized* performance of the bidders. Some earlier work considers the realized performance to be deterministic given the bidder's effort (Baye et al., 1996), whereas others model the performance as a nosiy, stochastic, function of the effort (Amegashie, 2006). We also consider the performance q_i as a noisy function of the effort b_i, indicating that participants have uncertainty about the exact effectiveness of their effort in producing high quality work. We model this uncertainty as random additive noise on the effort level: $q_i = \epsilon_i + b_i$, where ϵ_i is a random variable, drawn i.i.d for each contestant. We consider cases where ϵ_i's distribution is either uniform or a Beta distribution ($\alpha = \beta = \frac{1}{2}$)) as well as the noiseless case (i.e. $\epsilon_i = 0$). A bidding strategy σ_i of participant i is a distribution over the bid levels. A set of bidding strategies $\sigma = (\sigma_1, \ldots, \sigma_n)$ is a **Nash equilibrium** if for any bidder i and any alternative strategy $\tilde{\sigma}_i$

[2] FP is only known to converge to a Nash equilibrium in two-player zero-sum games (Papadimitriou & Roughgarden, 2008), and we examine all-pay auctions, which are not zero-sum and have more than two participants. Nonetheless, we empirically show that FP does converge to the Nash equilibrium bidding strategies in the restricted settings where the Nash equilibrium is known.

[3] Simulations using multi-agent reinforcement learners, using simple REINFORCE (Williams, 1992) learners, exhibit noisier behavior. We discuss these results in the appendix.

(alternative distribution over bid levels) we have $s_i(\sigma) \geq s_i(\sigma_{-i}, \tilde{\sigma}_i)$, i.e. given the bidding strategy of others σ_{-i}, no player i wants to unilaterally deviate from their strategy σ_i to any other strategy $\tilde{\sigma}_i$. It is only known how to derive Nash equilibria for specific all-pay auction domains.

Given the realized performance of each contestant, the principal receives a utility as a function u of the maximum performance, i.e. $u(\max_i q_i)$. The utility function u describes how the performance of the bidders translates into value to the principal [4]. We consider the case of diminishing marginal returns on effort, modelled by a logarithmic utility function. Diminishing returns can also be used to model risk-aversion of the principal. We model a fixed entry cost of b that does not contribute to the solution quality [5] Finally, we assume that the principal has some existing default solution with a utility of 0. If no bid is better, the principal uses the default solution and receives a utility of 0. Hence our principal's utility function is $u(q) = \max(\log(a(q - b)), 0)$, where a is a scale factor.

Goal: we seek to find the allocation $w = (w_1, \ldots, w_n)$ of the total prize that maximizes the principal's expected utility $\mathbb{E}_\pi(u(\max_i q_i))$ (given how participants would behave in the resulting contest).

Multiple Nash equilibria may exist in rank-allocation auctions. We focus on the *symmetric* case, where all bidders use the same strategy, a distribution over bids between 0 and the maximum prize available. In the noiseless case, theoretical analysis of the symmetric Nash is possible; for fewer than 5 bidders the density function of the symmetric equilibrium can be derived exactly, while for larger numbers of bidders it can only be sampled from. We present this analysis in appendix C.

2 METHODS

Our approach for automating the contest design process is illustrated in Figure 1. Shortly, we simulate agent learning in contests under various designs and record the resulting utilities. Next, we generalize from the training data by fitting a parameterized mapping from designs to utilities. Because the mapping is differentiable, it allows gradient based optimization in the continuous space of designs, so we apply an optimization procedure to identify the optimal design under the model. We provide a detailed discussion of our method, with a formal description in Algorithm 2.

We begin by investigating a set $\mathcal{D}$ of possible possible contest designs. As discussed in Section 1.1, a design for n bidders is characterized by the reward distribution $\mathbf{w} = (w_1, \ldots, w_n)$, lying on the simplex (i.e. $\sum_{i=1}^{n} w_i = 1$). Given a design $d \in \mathcal{D}$, our framework simulates how agents would learn to bid under this design. For the simulation, we use fictitious play (Brown, 1951), one of the most prominent models for how an agent may learn and adapt their strategy; We also discuss other alternatives such as independent multi-agent reinforcement learning (Littman, 1994). We emphasize that our method is flexible and may use any model for agent learning in our simulation.

For a design $d \in \mathcal{D}$, the output of the simulation are the *bidding strategies* σ_d of agents under this design, where σ_d is a distribution over the bid levels. Given the bidding strategies σ_d and contest simulation, we can also determine the expected utility u_d for the principal, as given in Section 1.1 (the subscript d indicates the bidding strategies and the principal's utility depend on the contest design d). By performing the simulation for many designs $d_1, \ldots, d_k$ chosen from the design space $\mathcal{D}$, we obtain a simulation dataset $\{(d_i, u_{d_i})\}_{i=1}^{k}$, where $d_i \in \mathcal{D}$ is a design and u_{d_i} is the expected utility the simulation shows it would generate for the principal (shown in the left of Figure 1).

Using the simulation dataset, we train a differentiable model to predict the principal's utility u_d under a contest design $d \in \mathcal{D}$ (including designs not observed during training). In other words, the true model for the principal's utility is a function $m : \mathcal{D} \to \mathcal{R}$, mapping any possible contest design in $\mathcal{D}$ to the utility it would provide to the principal. We approximate m using a neural network, trained on simulation data, yielding the approximate function $m_\theta : \mathcal{D} \to \mathcal{R}$ (where θ are model parameters). We use a simple feedforward network trained on many auction designs (see full details regarding architecture and training in Appendix B), depicted in the middle of Figure 1.

Given m_θ, we aim to identify designs resulting in high utility for the principal; Our goal is thus to "reverse engineer" the model, seeking inputs causing the model to output a high value reflecting

[4] In the noiseless case, when u is invertible this is equivalent to specifying a cost function $c(b_i)$ on the bid where $c = u^{-1}$, as done in previous work (Vojnović, 2015).

[5] For instance in the Netflix competition, contestants had to perform some work just to enter the contest, e.g. downloading data or reading documentation, efforts that provide no value to the principal.

high utility to the principal. The model is differentiable, so we can calculate the gradient of the output with regard to the *inputs* $\nabla_{\mathbf{w}} m_\theta(\mathbf{w})$, allowing gradient-based optimization. A key challenge here is that the input design $(w_1, \ldots, w_n)$ must respect the principal's budget, i.e. $\sum_{i=1}^{n} w_i = \bar{w}$. As illustrated on the right of Figure 1, we perform the optimization while adhering to the principal's budget by employing a form of entropic mirror ascent (Beck & Teboulle, 2003), given in Algorithm 1 below. We now describe the data generation (Step 1) and design optimization (Step 3) in more detail.

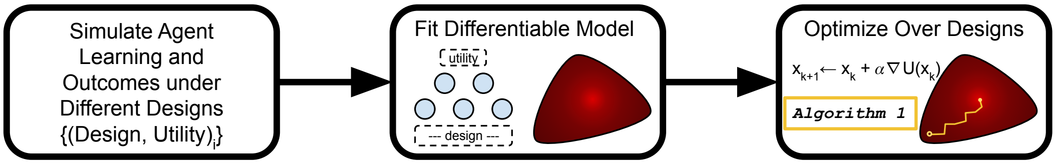

Figure 1: Diagram of the contest design process (Algorithm 2). Step 1) Simulate contests and agent learning to determine the utility of possible designs. Each data point represents an (contest design, utility) pair. Step 2) Fit a deep network to predict utility given contest design. Step 3) Optimize the output (utility) over the input (contest design) of the deep network to find the optimal design.

2.1 Data Generation

We generate data to train the model m_θ by simulating the learning process of agents in contests of a given design. The simulated contest receives bids as input and returns the rewards earned by the participants, as well as the principal's revenue. We use FP (Brown, 1951) as a model of agent learning. In FP, each the agents adjust a distribution over discrete bid levels by computing the best response to historical play. We use FP as it is a well-established model of agent learning in strategic settings. However, we emphasize that there are many alternative algorithms that can be used as the simulation method in our framework.

Multi-agent independent RL (MARL) is an appealing simulation alternative. We discuss how our empirical results change as we switch from FP to MARL in the appendix. Various surveys contain a detailed comparison of FP, MARL and other methods (Claus & Boutilier, 1998; Shoham et al., 2003; Yang & Gu, 2004). In 3-bidder auctions, we found that MARL agreed with Nash for large first prizes ($\geq 0.8\bar{w}$), but deviated significantly from Nash with smaller first prizes ($\leq 0.7\bar{w}$)

2.2 Design Optimization

As discussed in Section 1.1, the contest design space is a convex set, the simplex $\sum_{i=1}^{n} w_i = \bar{w}$. Entropic Mirror Ascent (Beck & Teboulle, 2003) is a non-euclidean gradient ascent method from the convex optimization literature, specifically designed for simplex constraints. The optimizer update rule for a design, w is: $\mathbf{w} \leftarrow \texttt{softmax}(\log(\mathbf{w}) + \eta \nabla m_\theta(\mathbf{w}))$ where $m_\theta(\mathbf{w})$ represents the neural model's predicted utility for input design $\mathbf{w}$. By inspection, w remains on the simplex after the update and $\log(\mathbf{w})$ is always defined as long as $\mathbf{w} = \mathbf{w}_0$ is initialized to the interior of the simplex.

The simplex constraint for $\mathbf{w} = (w_1, \ldots, w_n)$ is insufficient. Having prizes that are not monotonically decreasing in rank gives participants an incentive to attempt to obtain a lower rank (as they get a higher prize for exerting less effort). Hence, we want designs with strictly monotonically decreasing prizes and zero last prize (giving a prize to the lowest quality submission is wasteful, causing lower efforts). We propose a modified entropic mirror ascent procedure to constrain iterates to this region of the simplex with a transformation. For example, in an (n=4) four bidder contests, let $w = [z_1 + z_2 + z_3, z_2 + z_3, z_3, 0]$ where $z_i > 0$. z_i denotes the marginal increase of the prize from that of rank $i-1$ to that of rank i. This sequence w is strictly monotonically decreasing. The simplex constraint implies $z_1 + 2z_2 + 3z_3 = 1$. Let e be the vector of coefficients, e.g., $e = [1, 2, 3]$, and define $\tilde{z}_i = e_i z_i$. Then $\tilde{z}$ lives on a simplex. We can run entropic mirror descent on $\tilde{z}$ and transform back to z with $z = \tilde{z}/e$. We formally express this idea in the transformation given in Algorithm 1 where $\odot$ and $\oslash$ denote element-wise multiplication and division respectively, Δ_{int}^{n} denotes the interior of the $n{-}1$ dimensional simplex, $w[i{:}j] = [w_i, \ldots, w_{j-1}]$, $\texttt{softmax}(y) = \frac{e^{y_i}}{\sum_j e^{y_j}}$, $\texttt{rev}$ reverses an array, and $\texttt{cumsum}(y)$ denotes the cumulative sum, i.e., $[y_1, y_1 + y_2, \ldots, \sum_j y_j]$.

Algorithm 1: Monotonic Entropic Mirror Ascent (`M-EMA`)

INPUT: initial design $\mathbf{w}_0$, with $\mathbf{w}_0[0{:}n{-}1] \in \Delta_{int}^{n-1}$, utility neural net $m_\theta(\mathbf{w})$: $\mathbb{R}^{n-1} \to \mathbb{R}$, number of optimization iterations K, learning rate η.
$z = \mathbf{w}[0{:}n{-}1] - \mathbf{w}[1{:}n]$
$e = [1, 2, \ldots n-1]$
for $k = 1 : K$ **do**
  $z_{k+1} \leftarrow \texttt{softmax}(\log(e \odot z_k) + \eta \nabla_z m_\theta(z_k) \oslash e) \oslash e$
end for
$\mathbf{w}_K[0{:}n{-}1] = \texttt{rev}(\texttt{cumsum}(\texttt{rev}(z)))$
OUTPUT: $\mathbf{w}_K$

2.3 Contest Design Using Simulation, Learning and Optimization

Algorithm 2 describes the overall contest design optimization method, described informally in the beginning of Section 2. It samples contest designs (we use a Dirichlet distribution $Dir_{n-1}(\alpha{=}1)$), uses FP to simulate agent learning on each design, trains a neural network for predicting the principal's revenue and finally uses Algorithm 1 to optimize the design.

Algorithm 2: Neural Auction Design

INPUT: n learning agents (bidders), utility metric, presumed optimal design $\mathbf{w}_0$, with $\mathbf{w}_0[0{:}n{-}1] \in \Delta_{int}^{n-1}$, empty dataset $\mathcal{D}$, number of auction simulations M, untrained neural network $m_\theta(\mathbf{w})$, number of optimization iterations K.
for $m = 1 : M$ **do**
  Sample design $d \in \mathcal{D}$: $\mathbf{w}[0{:}n{-}1] \sim \texttt{Dir}_{n-1}(\alpha{=}1)$, $\mathbf{w}[n{-}1] = 0$. Sort x (decreasing).
  Use FP to simulate contest until convergence, obtaining participant strategies σ.
  Fix agent strategies σ, simulate contests to compute expected auctioneer's utility $u_d(\sigma)$.
  Record (design d, expected utility u_d) pair to dataset $\mathcal{K}$.
end for
Train neural network, $m_\theta(\mathbf{w})$, to regress on M samples from $\mathcal{K}$.
Optimize design using Algorithm 1, $\mathbf{w}^* \leftarrow \texttt{M-EMD}(\mathbf{w}_0, m_\theta)$
OUTPUT: $\mathbf{w}^*$

---

3 Experiments

Section 1.1 describes various assumptions one can make regarding the performance noise model and the utility of the principal in crowdsourcing contests. We applied our proposed framework to optimize the design of crowdsourcing contests under various such assumptions. In all our experiments, we consider the principal's utility function to be the one given in Section 1.1, $u(q) = \max(\log(a(q-b)), 0)$, which reflects a risk averse principal, with a minimal quality bar.

Section 3.1 shows empirical results for a domain with three or four bidders, and noiseless performance. As discussed in Section 1.1, in this model the symmetric Nash equilibrium strategy is *known*. Our analysis shows that the FP simulation results in agent behavior that is extremely close to the Nash equilibrium prediction. Further, after fitting a differentiable model and optimizing the design using Algorithm 2 we obtain the same optimal design prescribed by equilibrium based analysis.

Section 3.2 considers settings where the equilibrium behavior is *not known*, so standard economic techniques struggle to recommend an optimal design. We consider settings with 10 participants and various performance noise models, and apply our framework to identify the optimal design. We show that our designs award prize money to a few top entrants. As the variance of performance noise increases, optimal designs award more prizes, and larger prizes to the runner-up in the contest.

3.1 Models With Known Equilibrium Behaviour

Consider a setting with three or four bidders, and with no performance noise. The first step in our framework is simulating agents who learn from repeated interaction in the contest, by applying FP.

We first investigate whether the predictions of FP agree with the Nash equilibrium strategies. In general FP *may not* converge to a Nash equilibrium as an all-pay auction is *not* a constant-sum or dominance solveable game (Jafari et al., 2001; Shamma & Arslan, 2005). Furthermore, the solution found with fictitious play is to a discrete version of the auction (where bids take one of a discrete set of values), whereas the analytic solution is for the case where bids can take any real value.

The top left of Figure 2 shows the symmetric Nash equilibrium bidding strategy, as the cummulative distribution function (CDF) of the distribution over bid levels, under multiple three bidder contest designs, characterized by the prize for the top rank (the remainder prize goes to the second rank, and the prize for the third rank is zero). The remaining plots of Figure 2 each examine one design (characterized by the first rank prize), and plot the bid CDFs of the Nash equilibrium versus those outputted by FP. Figure 2 shows that the FP output closely matches the Nash equilibrium.

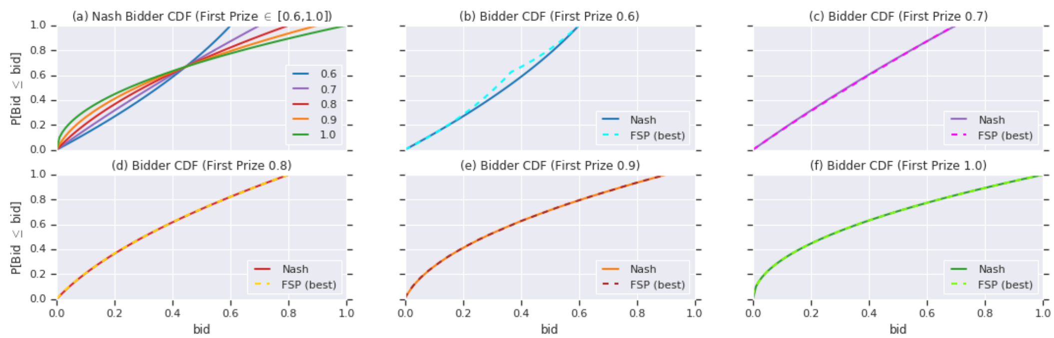

Figure 2: Bidding strategy CDFs. The top left plot shows the CDFs for each of five different contest designs assuming the bidder plays the Nash distribution. The first prize is listed in the legend; the second prize equals one minus the first; no prize is given to the third bidder. The remaining plots compare the Nash CDF with the CDF learned by fictitious self-play for different first prize amounts.

The next phase in our pipeline takes the dataset of simulation outcomes under various designs, and trains a neural network to predict the principal's utility in any given contest design (attempting to generalize to unsimulated designs). Figure 3 compares the principal's utility under Nash bidding against the prediction of our trained model for various designs (characterized by the first rank reward, shown on the x-axis). We observe that the simulation results for the principal's utility are consistently very slightly below the Nash-based analytical solution. The model has an almost perfect fit to the simulation results. The final step of our method is optimizing the contest design given the model (Algorithm 1). The optimal design is marked in Figure 3, for both the Nash-based curve and our method. These match almost perfectly (the location on the x-axis is almost identical), indicating our method finds the same optimal design as prescribed by the Nash equilibrium analysis.

Finally, we explore a four bidder contest to investigate the effect of possible designs on the principal's utility. Figure 4 shows a heatmap for the principal's utility for possible designs. The x-axis is the reward w_1, the prize for the first rank, and the y-axis is the reward w_2 for the second rank (the lowest rank gets no reward $w_4 = 0$, and the third rank reward is $w_3 = w - w_1 - w_2$). Figure 4 shows that the utility is fairly robust to designs with a high first prize, i.e., $w_1 \in [0.7 - 0.9]$ and third prizes $w_3 < .1$. However, good designs with a low first prize (e.g. $w_1 < 0.7$) offer no reward to the third rank. This indicates that in settings with many participants we might expect a greater distribution of reward across top prizes, but the principal's utility may be fairly robust around the optimal design.

3.2 Models With Unknown Equilibrium Behaviour

We explore contests where the each participant's bid is perturbed by random noise to yield their performance. We consider noise following either a uniform or a Beta($\frac{1}{2}, \frac{1}{2}$) distribution. Due to the noise distribution, the Nash equilibrium bidding strategy is not known for this setting. We apply our method on such contests, and investigate how the optimal design is affected by the noise distribution.

Figure 5 shows heatmaps of the principal's utility in the noisy setting (uniform noise on the left and Beta noise on the right), under different contest designs. Similarly to Figure 4 the axes are the w_1 and w_2, the last prize is $w_4 = 0$, and $w_3 = 1 - w_1 - w_2$. Figure 5 shows that as more noise is

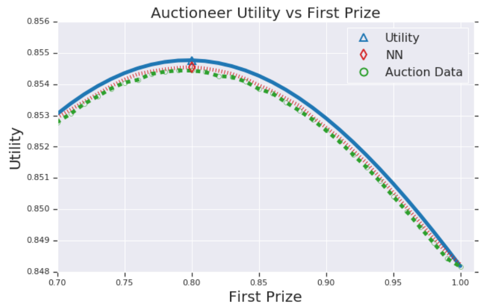

Figure 3: Principal's utility as a function of the first rank prize, as given by the analytical Nash CDFs, the simulation data, and the trained neural network. Markers denote maxima.

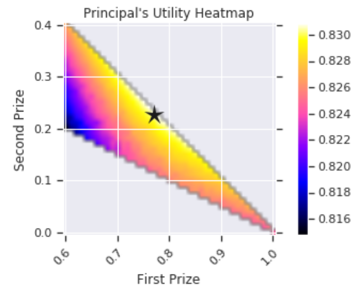

Figure 4: Heatmap for the principal's utility for various four bidder contest designs. The star marks the optimal design.

introduced to the bids, the optimal designs tighten around more evenly distributing reward across the top two bids (in both cases in the optimal design $w_3 = 0$). In other words, as the noise increases, the optimal design transfers more reward from the top rank to the one below it.

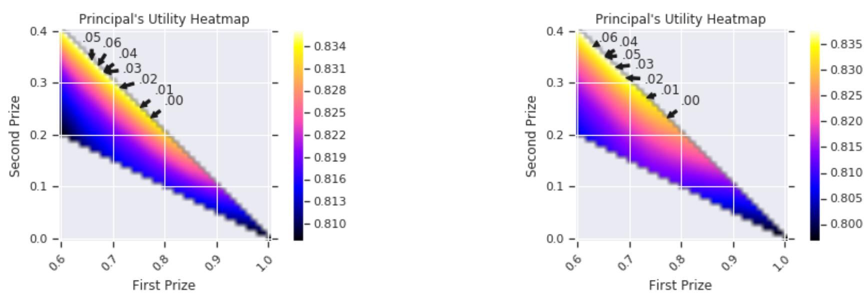

Figure 5: Heatmaps for the principal's utility in four bidder contests with realized bids drawn from an interval [bid$-d$,bid$+d$] either (L) uniformly or (R) according to Beta($\frac{1}{2}$, $\frac{1}{2}$), $d = 0.06$. Plotted on top of the heatmap are arrows to the optimal designs for different values of d annotated on the map.

We now investigate contests with more participants, showing how the performance noise affects the optimal design. Figure 6 shows the optimal design for $n = 10$ participants, under different performance noise levels. We only illustrate the top 3 prizes in a 3D plot (lower ranks typically get very little or no reward under the optimal design). Figure 6 shows that increasing the noise makes the optimal design spread the reward more evenly among the top ranks. Table 1 shows the same effect as a table, giving the optimal design and inequality in prize levels.

Finally, we investigate the limitations of our approach. Our framework may suggest sub-optimal design due to multiple issues. First, the simulation of how participants learn may not be an accurate model of their behavior. Second, the neural network may may have an approximation error in predicting the principal's utility. Third, the optimization procedure (Algorithm 1) may converge on a local rather than global optimum. Figure 7 illustrates the generalization error contrasting the principal's utility when running the FP simulation and when predicting it using the trained model on previously unobserved designs. We note that the neural network's predictions are slightly different from the simulation data, though they follow similar trends. Further, Figure 7 also marks the location of the optimized design suggested by Algorithm 2 with a star, showing how slight errors occur due to convergence to a local rather than global optimum (e.g. in the figure for the third rank prize).

4 Related Work and Conclusion

Crowdsourcing contests and all-pay auctions have received significant attention in the economic literature, including recently published surveys and books focusing on the topic (Vojnović, 2015; Dechenaux et al., 2015). We propose a neural approach to designing crowdsourcing contests. Earlier work has carried out equilibrium analysis for several restricted models of all-pay auctions (Milgrom

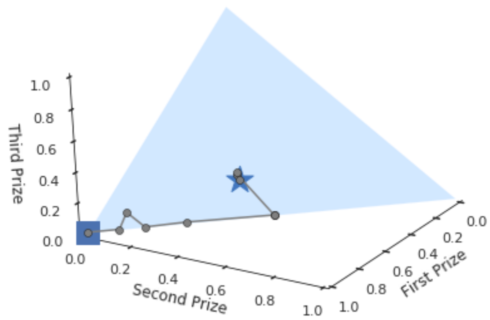

Figure 6: First three prizes of the optimized ten bidder contest plotted on top a simplex for reference. Each point denotes the optimal design for a different noise level. The square marks zero noise with the trajectory ending at the star with realized bids drawn uniformly from bid ± 0.06.

w	w_0^*	w_1^*	w_2^*	Gini
0.000	1.000	0.000	0.000	0.900
0.002	0.916	0.084	0.000	0.883
0.004	0.850	0.075	0.075	0.855
0.006	0.848	0.152	0.000	0.870
0.008	0.733	0.260	0.004	0.844
0.010	0.500	0.500	0.000	0.800
0.020	0.500	0.500	0.000	0.800
0.030	0.500	0.500	0.000	0.800
0.040	0.471	0.325	0.204	0.753
0.050	0.461	0.319	0.209	0.746
0.060	0.443	0.319	0.152	0.724

Table 1: First three prizes of the optimal ten bidder auction given by Algorithm 2. Prizes given to fifth and higher are all zero. Width of the uniform distribution around bidder's bid is specified in w column; Gini index of each design is in the last column.

Figure 7: Utility for the ten bidder contest vs the top four prizes with realized bids drawn uniformly from bid $\pm .06$. Note plots are shown on different scales and for different slices of the design space.

& Weber, 1982; Baye et al., 1993; 1996; Krishna & Morgan, 1997; Amegashie, 2006; Siegel, 2009), including the impact of risk-aversion (Fibich et al., 2006; Gao et al., 2012). Empirical evaluation of how people actually bid in such settings has revealed significant discrepancies with the predictions of the equilibrium based analysis (Rapoport & Amaldoss, 2004; Gneezy & Smorodinsky, 2006; Liu et al., 2014). Such empirical work suggests that people tend to employ simple learning heuristics (Rapoport & Amaldoss, 2004; Dechenaux et al., 2015).

We examine a principal deciding on a rank reward allocation, in order to maximize its utility. This broadly falls in the field of *mechanism design* or *auction design* (Myerson, 1981; Nisan & Ronen, 2001), a subfield in economics, seeking to decide the "rules of the game" so as to achieve desired outcomes. Typically, auctions are designed *manually* by economists seeking to maximize revenue (Myerson, 1981; Bulow & Roberts, 1989; Roth, 2002). In contrast, we *automate* the process, similarly to the recent field of *automated mechanism design* (Conitzer & Sandholm, 2002; Sandholm, 2003; Conitzer & Sandholm, 2004; Hajiaghayi et al., 2007; Guo & Conitzer, 2010).

We use machine learning to search the space of designs, akin to recent deep-learning mechanism design frameworks proposed for other auction types (Dütting et al., 2017; Feng et al., 2018; Manisha et al., 2018; Tacchetti et al., 2019). In contrast to these, we leverage agent learning of the auction (Mizuta & Steiglitz, 2000; Vorobeychik & Wellman, 2008). Learning agents are increasingly capable of solving complex problems; using such capable agents for mechanism design holds the promise of optimizing the design of mechanisms in more complex settings than previously possible.

Our empirical analysis shows the promise of automated mechanism design based on deep learning. However, our technique has several limitations, such as the dependence on a good model for the learning of agents, and errors introduced by inaccurate function approximation and converging on local optima. Several questions are open for future research. Can our methods generalize well to other mechanism design domains such as other types of auctions? What are good models of agent learning in other strategic settings? Do such models do a good job in characterizing the bidding behavior of *human* participants? Finally, can better methods be devised to optimize over designs?

REFERENCES

J Atsu Amegashie. A contest success function with a tractable noise parameter. *Public Choice*, 126 (1-2):135–144, 2006.

Simon P Anderson, Jacob K Goeree, and Charles A Holt. Rent seeking with bounded rationality: An analysis of the all-pay auction. *Journal of Political Economy*, 106(4):828–853, 1998.

Nikolay Archak and Arun Sundararajan. Optimal design of crowdsourcing contests. *ICIS 2009 proceedings*, pp. 200, 2009.

Michael R Baye, Dan Kovenock, and Casper G De Vries. Rigging the lobbying process: an application of the all-pay auction. *The american economic review*, 83(1):289–294, 1993.

Michael R Baye, Dan Kovenock, and Casper G De Vries. The all-pay auction with complete information. *Economic Theory*, 8(2):291–305, 1996.

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

Robert M Bell and Yehuda Koren. Lessons from the netflix prize challenge. *SiGKDD Explorations*, 9(2):75–79, 2007.

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

Lucian Bu, Robert Babu, Bart De Schutter, et al. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2):156–172, 2008.

Jeremy Bulow and John Roberts. The simple economics of optimal auctions. *Journal of Political Economy*, 97(5):1060–1090, 1989.

Shuchi Chawla, Jason D Hartline, and Balasubramanian Sivan. Optimal crowdsourcing contests. *Games and Economic Behavior*, 2015.

Caroline Claus and Craig Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. *AAAI/IAAI*, 1998(746-752):2, 1998.

Chen Cohen and Aner Sela. Allocation of prizes in asymmetric all-pay auctions. *European Journal of Political Economy*, 24(1):123–132, 2008.

Vincent Conitzer and Tuomas Sandholm. Complexity of mechanism design. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, pp. 103–110. Morgan Kaufmann Publishers Inc., 2002.

Vincent Conitzer and Tuomas Sandholm. Self-interested automated mechanism design and implications for optimal combinatorial auctions. In *Proceedings of the 5th ACM Conference on Electronic Commerce*, pp. 132–141. ACM, 2004.

Emmanuel Dechenaux, Dan Kovenock, and Roman M Sheremeta. A survey of experimental research on contests, all-pay auctions and tournaments. *Experimental Economics*, 18(4):609–669, 2015.

Dominic DiPalantino and Milan Vojnovic. Crowdsourcing and all-pay auctions. In *Proceedings of the 10th ACM conference on Electronic commerce*, pp. 119–128. ACM, 2009.

Paul Dütting, Zhe Feng, Harikrishna Narasimhan, and David C Parkes. Optimal auctions through deep learning. *arXiv preprint arXiv:1706.03459*, 2017.

Zhe Feng, Harikrishna Narasimhan, and David C Parkes. Deep learning for revenue-optimal auctions with budgets. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 354–362. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

Gadi Fibich, Arieh Gavious, and Aner Sela. All-pay auctions with risk-averse players. *International Journal of Game Theory*, 34(4):583–599, 2006.

Drew Fudenberg and David M Kreps. Learning mixed equilibria. *Games and Economic Behavior*, 5(3):320–367, 1993.

Xi Alice Gao, Yoram Bachrach, Peter Key, and Thore Graepel. Quality expectation-variance tradeoffs in crowdsourcing contests. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

Uri Gneezy and Rann Smorodinsky. All-pay auctionsan experimental study. *Journal of Economic Behavior & Organization*, 61(2):255–275, 2006.

Mingyu Guo and Vincent Conitzer. Computationally feasible automated mechanism design: General approach and case studies. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

Mohammad Taghi Hajiaghayi, Robert Kleinberg, and Tuomas Sandholm. Automated online mechanism design and prophet inequalities. In *Twenty-First AAAI Conference on Artificial Intelligence*, volume 7, pp. 58–65, 2007.

John Joseph Horton and Lydia B Chilton. The labor economics of paid crowdsourcing. In *Proceedings of the 11th ACM conference on Electronic commerce*, pp. 209–218. ACM, 2010.

Junling Hu, Michael P Wellman, et al. Multiagent reinforcement learning: theoretical framework and an algorithm. In *ICML*, volume 98, pp. 242–250. Citeseer, 1998.

Amir Jafari, Amy Greenwald, David Gondek, and Gunes Ercal. On no-regret learning, fictitious play, and nash equilibrium. In *ICML*, volume 1, pp. 226–233, 2001.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Vijay Krishna and John Morgan. An analysis of the war of attrition and the all-pay auction. *journal of economic theory*, 72(2):343–362, 1997.

Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pp. 157–163. Elsevier, 1994.

Tracy Xiao Liu, Jiang Yang, Lada A Adamic, and Yan Chen. Crowdsourcing with all-pay auctions: A field experiment on taskcn. *Management Science*, 60(8):2020–2037, 2014.

Padala Manisha, CV Jawahar, and Sujit Gujar. Learning optimal redistribution mechanisms through neural networks. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 345–353. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

Paul R Milgrom and Robert J Weber. A theory of auctions and competitive bidding. *Econometrica: Journal of the Econometric Society*, pp. 1089–1122, 1982.

Hideyuki Mizuta and Ken Steiglitz. Agent-based simulation of dynamic online auctions. In *2000 Winter Simulation Conference Proceedings (Cat. No. 00CH37165)*, volume 2, pp. 1772–1777. IEEE, 2000.

Roger B Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1):58–73, 1981.

Vishnuteja Nanduri and Tapas K Das. A reinforcement learning model to assess market power under auction-based energy pricing. *IEEE transactions on Power Systems*, 22(1):85–95, 2007.

Arkadii Semenovich Nemirovsky and David Borisovich Yudin. Problem complexity and method efficiency in optimization. 1983.

Noam Nisan and Amir Ronen. Algorithmic Mechanism Design. *Games and Economic Behavior*, 35(1-2):166–196, 2001.

Christos H Papadimitriou and Tim Roughgarden. Computing correlated equilibria in multi-player games. *Journal of the ACM (JACM)*, 55(3):14, 2008.

Amnon Rapoport and Wilfred Amaldoss. Mixed-strategy play in single-stage first-price all-pay auctions with symmetric players. *Journal of Economic Behavior & Organization*, 54(4):585–607, 2004.

Alvin E Roth. The economist as engineer: Game theory, experimentation, and computation as tools for design economics. *Econometrica*, 70(4):1341–1378, 2002.

Tuomas Sandholm. Automated mechanism design: A new application area for search algorithms. In *International Conference on Principles and Practice of Constraint Programming*, pp. 19–36. Springer, 2003.

Jeff S Shamma and Gürdal Arslan. Dynamic fictitious play, dynamic gradient play, and distributed convergence to nash equilibria. *IEEE Transactions on Automatic Control*, 50(3):312–327, 2005.

Yoav Shoham, Rob Powers, and Trond Grenager. Multi-agent reinforcement learning: a critical survey. *Web manuscript*, 2003.

Ron Siegel. All-pay contests. *Econometrica*, 77(1):71–92, 2009.

Dana Sisak. Multiple-prize contests - the optimal allocation of prizes. *Journal of Economic Surveys*, 23(1):82–114, 2009.

Andrea Tacchetti, DJ Strouse, Marta Garnelo, Thore Graepel, and Yoram Bachrach. A neural architecture for designing truthful and efficient auctions. *arXiv preprint arXiv:1907.05181*, 2019.

Milan Vojnović. *Contest theory: Incentive mechanisms and ranking methods*. Cambridge University Press, 2015.

Yevgeniy Vorobeychik and Michael P Wellman. Stochastic search methods for nash equilibrium approximation in simulation-based games. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2*, pp. 1055–1062. International Foundation for Autonomous Agents and Multiagent Systems, 2008.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

Erfu Yang and Dongbing Gu. Multiagent reinforcement learning for multi-robot systems: A survey. Technical report, tech. rep, 2004.

Haichao Zheng, Dahui Li, and Wenhua Hou. Task design, motivation, and participation in crowdsourcing contests. *International Journal of Electronic Commerce*, 15(4):57–88, 2011.

A Simulations Using Fictitious Play and Independent Multi-Agent Reinforcement Learning

The discussion in the main text has focused no applying Fictitious Play (FP) (Brown, 1951; Fudenberg & Kreps, 1993) for the simulation phase. As we discussed, an alternative is applying independent multi-agent reinforcement learning (Littman, 1994; Hu et al., 1998; Bu et al., 2008). We now provide an empirical analysis of the choice regarding the simulation phase.

We first note that FP operates by applying a simple learning rule: each agent assumes the opponents play a stationary (mixed) strategy. In each each round, every player chooses the best response to the empirical frequency of play of their opponents. The key parameter determining the computational cost of the simulation is this the number of such simulation rounds. Figure 8 investigates the impact of the number of rounds on the learned bidding strategies (distribution over bid levels), contrasting it with the Nash equilibrium for the game. It shows that same results as Figure 2 in the main text, but for varying number of FP rounds and discretization granularities for bid levels.

Figure 8: This figure compares the bidder CDFs. The top left plot shows the CDFs for each of five different auction designs assuming the bidder plays the Nash distribution. The first prize is listed in the legend; the second prize equals one minus the first; no prize is given to the third bidder. The remaining plots compare the Nash CDF with the CDF learned by fictitious self-play for different first prize amounts. Figures (b-d) also show the effect of training iterations and discretization granularity on the final CDF. The arrow in Figure (c) highlights a common trend where the CDF converges to Nash from above as a finer discretization is introduced. In other words, coarser discretizations lead to under bidding and, in turn, underestimates of the auctioneer utility. Figure (d) shows that increasing training iterations reduces error, but in a less structured manner than bid granularity.

Figure 8 indicates that the number of FP rounds and the granularity of discretization of bid levels have an impact on the learned bidding strategies. However, the results are somewhat robust to the choice of these parameters, yielding relatively similar bidding strategies under many settings (and in most settings the results are qualitatively very similar to the Nash equilibrium).

We now turn to investigate using multi-agent reinforcement learning (MARL) for the simulation phase. Independent MARL based methods have recently become a very popular means of modeling agent behavior in multi-agent environments. Such approaches have the advantage of being simple and relatively easy to implement. We have used independent REINFORCE agents (Williams, 1992), and investigate the bidding strategies learned by the agents. The bidding strategies learned using MARL are shown in Figure 9, and are contrasted with both the Nash equilibrium bidding strategies and the bidding strategies learned via FP (similarly to Figure 2 and Figure 8). These results show that for large first prizes (where the first prize is 0.8 or higher), all three methods yield a relatively similar distribution. However, there is a significant deviation for lower top prizes, where the RL distribution has a step function curve.

The results in Figure 9 indicates that the model for agent learning may have a significant impact on the assumed bidding strategies (and hence on the choice of a design). Ultimately, we feel like this is a modelling decision on the user's side. In other words, in order to choose a good design, one must first determine what is a reasonable model of the learning behavior of agents. In the case of crowdsourcing contests (all-pay auctions), using the FP learning rule yields results that

Figure 9: This figure compares the bidder CDFs. The top left plot shows the CDFs for each of five different auction designs assuming the bidder plays the Nash distribution. The first prize is listed in the legend; the second prize equals one minus the first; no prize is given to the third bidder. The remaining plots compare the Nash CDF with the CDFs learned by fictitious self-play and REINFORCE respectively using their optimal hyperparameter configurations (iterations, bid levels, learning rate, batch size). Both REINFORCE and fictitious self-play agree closely with the Nash equilibrium when the first prize ≥ 0.8. For smaller prizes, there is a larger discrepancy. We see that FP is consistently closer to the Nash equilibrium than REINFORCE.

are more similar to those used in traditional Nash equilibrium based analysis. In contrast, if one believes agents are more likely to be reinforcement learners, an alternative bidding strategy is a likely outcome. One possible choice is a conservative approach, where one only considers design where there is a consensus between simulation learning rules (e.g. FP or MARL). In this case, one may opt for a large top prize, as in this case, the different models for agent learning behaviour agree with each other.

B Design Process Architecture and Hyeprparameters

To aid reproducability, we now provide full details regarding our network architecture and hyperparameters.

For our neural network m_θ, we have used a simple feedforward network with 2 hidden layers, 256 neurons per layer, and `ReLU` nonlinearities. We trained the network for $10,000$ iterations using the Adam optimizer Kingma & Ba (2014) with a learning_rate=1e-3, using with default $\beta_1 = 0.9$, $\beta_2 = 0.999$ and mini-batches of size 51 for the 3 bidder auction and 1000 for the 10 bidder auction.

Design Optimization (`E-EMA`): We initialized designs such that the first prize was given 0.9 and all remaining marginals were given a constant $z_{i>1} = \epsilon$. We performed $100,000$ iterations of `E-EMA` with a learning rate of 0.1 for the 3 bidder auction and $200,000$ iterations with a learning rate of 0.001 for the 10 bidder auctions.

Principal Utility Fuction: We set $a = 50$ and $b = 0.1$ (see Figure 10).

Fictitious Play: We ran fictitious play for $100,000$ iterations with a discretization of 1001 effort levels for the bid interval $[0, 1]$. We were searching for a symmetric equilibrium so all bidders played using the same bid distributions, i.e. using fictitious self-play.

C Analytic Results on Noiseless Auctions

We now very briefly discuss how one can solve for the closed form bidding strategies in crowdsourcing contests. A more detailed discussion of this can be found in contest theory textbooks (Vojnović, 2015) and in various papers on all-pay auctions (Milgrom & Weber, 1982; Baye et al., 1993; 1996;

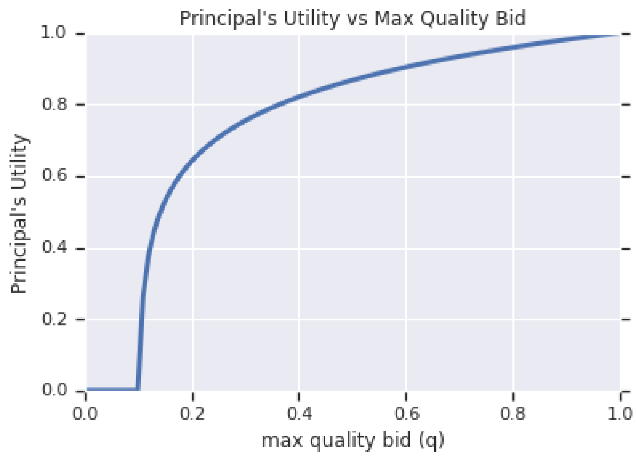

Figure 10: The principal's utility function: $\max(\log(a(q-b)), 0)$.

Krishna & Morgan, 1997; Siegel, 2009; DiPalantino & Vojnovic, 2009; Horton & Chilton, 2010; Zheng et al., 2011). [6]

We are interested in finding the symmetric Nash equilibrium for an all-pay auction, as discussed in Section 1.1. In a symmetric Nash equilibrium, all bidders use the same bidding strategy σ, which is simply a distribution over the bid levels. In a symmetric Nash equilibrium, no bidder i wants to unilaterally deviate from σ to an alternative bidding strategy $\tilde{\sigma}_i$.

We write the CDF of a bidding strategy as $B(b)$, and attempt to identify the symmetric Nash equilibrium. First note that this equilibrium strategy is atomless. If it weren't, agents bidding at the atom could achieve non-infinitesimal increases in their expected prize money by increasing their bid infinitesimally so as to outperform all other bids at the atom, therefore B would not be Nash.

The expected prize money from bidding b when all bidders are following the bidding strategy B is given by:

$$\sum_{j=1}^{n} w_j G_j(B(b)), \text{ where } G_j(z) = \binom{n-1}{j-1} z^{n-j}(1-z)^{j-1}$$

Each term of the sum is simply the value of the j^{th} prize w_j times the probability $G_j(B(b))$ that a bid of percentile $B(b)$ achieves rank j against a set of $n-1$ independent bids drawn from B.

Proposition: The symmetric Nash equilibrium results in an expected value of 0 to all participants.

Proof: $B(0) = 0$ and B is continuous because B is atomless.

We write the expected utility when bidding b against opponents bidding according to B as $s(b; B)$

Choose $\delta > 0$. The value $s(b; B)$ of bids $b < B^{-1}(\delta)$ is bounded by the expected prize money under those bids, i.e.

$$s(b; B) \leq \sum_{j=1}^{n} w_j G_j(B(b)) \leq \sum_{j=1}^{n} w_j G_j(\delta)$$

Since $G_j(\delta)$ tends to 0 as δ tends to 0, for any $\epsilon > 0$, $\exists \delta > 0$ s.t. bids $b \leq B^{-1}(\delta)$ have an expected value $s(b; B) \leq \epsilon$. Furthermore, because $\delta > 0$, some such bids are in the support of B. Therefore

[6] Some prior work, such as Amegashie (2006), has made progress analytically for specific noise models, but not for the models considered in this work.

there are bids in the Nash with value arbitrarily close to 0. Therefore no bid $\tilde{b}$ can have $s(\tilde{b}; B) > 0$, since this would imply that there were bids that outperformed bids in the support of the Nash.

Finally note that a bid of 0 cannot win a prize, but also incurs no cost, so has a value of 0. Therefore the value to bidders of the Nash must also be at least 0. $\square$

The proposition tells us that the symmetric Nash equilibrium $B(b)$ satisfies:

$$s(b; B) = \sum_{j=1}^{n} w_j G_j(B(b)) - b = 0 \tag{1}$$

This equation is a polynomial of order $n-1$ in $B(b)$ for each value of b. Polynomials of up to order 4 can be solved analytically, therefore the CDF of the symmetric Nash can be expressed analytically for auctions with 5 or fewer bidders.

For any number of bidders, we can easily express the inverse-CDF using equation 1 as follows:

$$\sum_{j=1}^{n} w_j G_j(B(b)) - b = 0$$

$$b = \sum_{j=1}^{n} w_j G_j(B(b))$$

$$B^{-1}(y) = \sum_{j=1}^{n} w_j G_j(y)$$

This allows sampling directly from the Nash equilibrium bid distribution in the noiseless setting, but relies on the fact that the probability of winning with a bid of b depends on B only through the value of $B(b)$, which is not true in a noisy auction.

A final result of interest to this work concerns settings for which the optimal design in the noiseless auction is to award the entire prize budget to the first place prize:

Theorem: If the principal's utility function $u : \mathbb{R}_+ \mapsto \mathbb{R}$ is strictly increasing, continuously differentiable and its inverse u^{-1} is log-concave, then $\mathbb{E}[u(\max b_i)]$ under the symmetric Nash equilibrium is maximized by allocating the entire prize budget to the first prize.

A proof can be found in Vojnović (2015).

Note however, that the inverse of $\log(a(x-b))$ is nowhere log-concave for $b > 0$. Therefore the utility function considered in this work is not covered by this theorem. Indeed, we often found superior designs that awarded prizes to multiple places.