Simulation of All-Pay Auctions with Convex Costs

Avichal Goel, Akshaj Kadaveru, Elizabeth Zou

May 16, 2019

1 Introduction

Crowdsourcing, or the practice of obtaining information or input into a task or project by enlisting the services of a large number of people, either paid or unpaid, typically via the Internet, is an effective way of acquiring content and continues to grow in popularity today. One example of crowdsourcing is Wikipedia, the online encyclopedia and the most popular wiki on the planet – all the information on Wikipedia pages are contributed by the community of writers, editors, and administrators. Another example is Stack Overflow, where developers ask questions and post answers, motivated by the “reputation points” attached to their accounts. And of course, websites based on ratings that Internet users provide, such as Yelp and Rate My Professors, all depend on crowdsourcing.

As crowdsourcing has become such a significant aspect of modern society, the search for near optimal mechanisms for crowdsourcing has also become increasingly important. We take on the approach that generalizes crowdsourcing models as all-pay auctions, and examine past work that investigates approximately optimal mechanisms in settings where bidders' utility functions are convex. We also describe experiments that analyze how well these mechanisms perform in a behavioral game theory model, simplified with symmetry and discretization.

2 Theoretical Model

In typical auction models, each bidder i has private value v_i, and submits a bid b_i. In an all-pay auction, where each bidder is charged their bid regardless of whether or not they receive an allocation, the utility of a bidder i can be described by $v_i x_i - b_i$, where $x_i \in \{0,1\}$ is the indicator variable of whether or not bidder i receives the item being auctioned.

In the crowd-sourcing model, a bidder i's utility is described by $x_i - c(e_i)$, where e_i is the amount of effort the bidder puts in, x_i is the prize they receive, and $c(e_i)$ is the bidder's cost of exerting effort e_i. To account for convex costs, we focus on models where a bidder's perceived payment is $c(e_i) = e_i^d$, for some exponent $d \geq 2$. For simplicity, let each bidder have a fixed private value $v_i \in [0,1]$ that represents their skill. We then define $q_i = v_i e_i$ to be the quality of a submission from a bidder with skill v_i and effort e_i. So in other words, skill represents “unit quality per unit effort".

Notice that for a specific bidder i, maximizing $x_i - c(e_i)$ is equivalent to maximizing $v_i^d(x_i - e_i^d) = v_i^d x_i - q_i^d$. When $d = 1$, this becomes $v_i x_i - q_i$. This is exactly the same equation as the all-pay auction. Thus, crowd-sourcing can be thought of as an all-pay auction where bidders' private values are their skills, and their bids are the qualities of their submissions.

In a real-world situation, it is hard to objectively judge the “quality" of a submission to a crowd-sourcing contest in a measurable way. In other words, a mechanism which allocates proportionally by values or quality is infeasible. Thus, it makes sense to rank the submissions and award prizes according to a bidder's rank, rather than a number corresponding to the quality of their submission. In this paper, we will focus on prize allocations of this type.

In the following section, we review literature on mechanisms for all-pay auctions with performance guarantees with respect to the optimal BIC mechanism.

3 Literature Review

In *Simple vs Optimal Mechanisms in Auctions with Convex Payments*[1], Greenwald et al. [2018] were motivated to investigate mechanisms for all-pay auctions, as this is essentially equivalent to deterministically charging each bidder a payment solely based on his bid (i.e. independent of other bidder's valuations), which is optimal for revenue maximization. Additionally, Greenwald et al. [2018] examined allocation rules that depend on a bidder's relative ranking, instead of the absolute value of his bid, because these allocation rules can be applied to settings with unknown payment functions (i.e., when we do not know the exponent d in the cost function $c_i(p_i) = p_i^d$). It is easy to see that these allocation rules also allow for simple mechanisms that generalize well, as they do not require knowledge of the distribution of submission qualities we will receive (i.e., prior-free).

Specifically, if prizes were allocated uniformly among the bidders whose submission qualities fall in the top half, then bidders would be incentivized to produce submissions achieving at least above the median. This would achieve a similar effect to allocating the prizes uniformly to all bidders above the median of the quality distribution, with the advantage that we do not need to know the quality distribution.

In their paper, Greenwald et al. [2018] considered an all-pay auction that allocated prizes uniformly to the top $n/4$ bidders, where n is the number of bidders. They proved that if the cost of production is of the form $c_i(p_i) = p_i^d$ for an exponent $d \geq 2$, values are drawn from a monotone hazard rate distribution that has a continuous cumulative distribution function supported by $[0, \overline{v}]$, and $n \geq 32\log(16\overline{v}/\kappa)$ (where κ is the median of the value distribution), then the unique Bayes-Nash equilibrium is when bidder i submits a bid (or in our case, quality)

$$b_i(v_i) = c_i^{-1}\left(v_i\hat{x}_i(v_i) - \int_0^{v_i} \hat{x}_i(z_i)dz_i\right),$$

where $\hat{x_i}$ is defined as the interim allocation where

$$\hat{x_i}(v_i) = \frac{4}{n}\Pr\left(v_i \text{ is among } \frac{n}{4} \text{ highest values } | v_i\right) = \frac{4}{n}\Pr\left(\sum_{j\neq i} 1_{v_j \geq v_i} \leq \frac{n}{4} - 1 | v_i\right).$$

Greenwald et al. [2018] then proved that in this Bayes-Nash Equilibrium, the revenue APX satisfies

$$\frac{APX}{OPT} \geq \frac{1}{16},$$

where OPT is the optimal revenue.

To further investigate their theoretical findings, Greenwald et al. [2018] tested their $n/4$ auction with randomly drawn monotone hazard rate distributions through simulations. The results of these experiments showed that the above mechanism achieves ≈ 70% of the optimal revenue on average, which is much better than the 7% they guaranteed.

We now introduce our own experiments to see if a similar auction with a convex cost function will perform well on a behavioral game theory model.

4 Experimental Model

In our experiments, we extend the mechanisms investigated by Greenwald et al. [2018] to an alternative type of behavioral game theory model, rather than constraining to Nash.

In this model, we assume bidders are categorized into different "layers of rationality", with $p_1, p_2, p_3, \dots$ bidders in layers $1, 2, 3, \dots$ respectively. Bidders bid as following:

- All bidders in layer 1 bid their true value.
- All bidders in layer 2 bid optimally, on the assumption that other bidders follow the bidding rule of layer 1.
- All bidders in layer 3 bid optimally, on the assumption that p_1 bidders follow the bidding rule of layer 1, and p_2 bidders follow the bidding rule of layer 2.
- So on and so forth.

Our motivation for a behavioral game theory model comes from the reality that not all bidders in the real world act with the same level of rationality, and that some bidders may have more information than others. Thus, we believe behavioral game theory models generalize better to real world examples.

Additionally, the typical behavioral game theory model involves each bidder knowing the total number of other bidders. For example, bidders of type 3 would typically assume that $\frac{p_1}{p_1+p_2} \cdot N$ bidders are of type 1 and $\frac{p_2}{p_1+p_2} \cdot N$ bidders are of type 2, where N is the total number of bidders. In real world auctions, this value is not usually known. We decided to assume the bidders knew only the number of bidders in previous layers, rather than the total number of bidders.

In the next section, we explain in further detail how we implement this.

5 Code

The link to our GitHub repository is `https://github.com/Aviously/Auction-Simulation`. Our code is composed of two classes: a Strategy class, which encodes our behavioral game theory model, and a Simulation class, which implements our simulations. We also have a code file with our tests.

5.1 Strategy Class

In the Strategy class, we determine each bidder's optimal action for all possible values. That is, we would like to calculate a function $bestBid(l,v)$ that returns the best action for a bidder on layer l with value v. Let V denote the set of possible values and B denote the set of possible bids.

In order to assist our calculations, we define the following functions:

- $beat(b,l)$ – the probability that a bidder in layer l will bid more than b. More precisely:

$$beat(b,l) = \frac{1}{|V|} \sum_{v \in V} \mathbb{1}\{bestBid(l,v) > b\}$$

- $dp(b,i,j)$ – the probability that among the first i bidders, exactly j bid more than b. We can define this recursively as:

$$dp(b,i,j) = dp(b,i-1,j-1) \cdot beat(b,l) + dp(b,i-1,j) \cdot (1 - beat(b,l))$$

  Intuitively, if the i^{th} person's bid was greater than b, we want the probability that exactly $j-1$ other people before the i^{th} person had also bid higher than b, which is what $dp(bid,i-1,j-1)$ holds. On the other hand, if the i^{th} person bid lower, we need j (instead of $j+1$) people from before to have bid higher.

- $r(b,l)$ – the expected payout of bidding b with respect to the information of a player in layer l. If we let p_i denote the prize for i^{th} place and let idx equal the position of the first bidder in layer l, we can write:
$$r(b,l) = \sum_i p_i \cdot dp(b, idx, i)$$
In our simulation, we scale this expectation to account for bidders in earlier layers only knowing a fraction of the total bidders.

- $u(v,b,p)$ – the utilility for a bidder with value v, bid b, and prize allocation p. In our simulation, we fix some constant d and use:
$$u(v,b,p) = v^d p - b^d$$

Now, we process the layers in increasing order, and for each layer l, break the problem down into the following steps:

- Calculate the $bestBid(l,v)$ for all $v \in V$. Based on the functions defined above, we see that
$$bestBid(l,v) = \operatorname*{argmax}_{b \in B} u(v,b,r(b,l))$$
The only expeption is the first layer of bidders, in which case their bid is simply a random $b \in B$.

- Based on the calculated $bestBid$ values, update $beat(b,l)$ for all $b \in B$.

- Using the new $beat$ values, update $dp(b,i,j)$ for all $b \in B$, $i \in [s_l, s_{l+1})$, and $j \in [0, s_{l+1})$, where s_l is the index of the first person in layer l. Essentially, we would like to update the values for all bidders in layer l, and the number of bidders that could exceed a given bid up to this point is at most s_{l+1}.

Once we repeat the above process for each layer, we'll know the optimal bid for any bidder given their layer on value.

5.2 Simulation Class

Each instance of the Simulation class is initialized with the following:

- The number of simulation trials to run
- An objective function to evaluate the simulation with
- The number of discrete values each bidder's value can take on
- The number of discrete values each bidder's bid can take on
- An array containing the number of bidders in each layer of rationality
- An array containing the prize allocations, sorted by bid rank (e.g. $[1/2, 1/2, 0, 0, 0, 0, 0, 0]$)

We have functions for each of the following:

- Generate random values from a uniform distribution for each bidder.
- Return the rank of bidder i given all the bids.
- Return the allocation to bidder i using the allocation rule defined in the instance of the Strategy class used in the simulation.

- Simulate an auction, which generates random values for each bidder, casts a bid for each bidder in each layer of rationality, calculates the “quality” of each bid, then returns a numerical evaluation of the simulation using a given objective.
- Simulate n auctions, which runs n simulations and calculates the average numerical evaluation.

We also evaluate using three different objectives to pass into instances of the Simulation class:

1. Maximizing the average quality over all bids.
2. Maximizing the average quality over the top quarter of bids.
3. Maximizing participation, which we define to be the number of non-zero bidders.

6 Experimental Results

We ran a simulation for each of the following variables, where we hold everything else constant and change that variable:

- The total prize pool
- The number n of top bidders who receive prizes, with the prizes being uniform among these n bidders
- The number n of top bidders who receive prizes, with half of the prizes being x and half being $2x$, for some x, among these n bidders

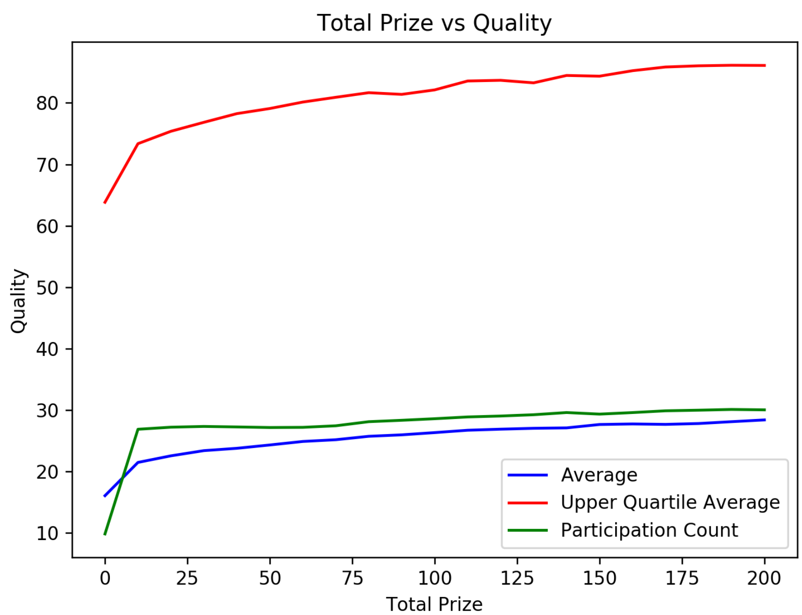
The default parameters we used were:

- 1000 total trials
- 50 discrete values for each value
- 200 discrete values for each bid
- 60 total bidders: 6 layers of rationality, with 10 bidders in each layer
- 100 as the total prize
- 1/15 of the total prize awarded to each of the top 15 out of 60 bidders

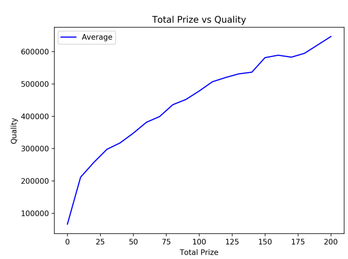
In all our simulations, we used $d = 4$ as the exponent in the cost function $c_i(p_i) = p_i^d$.

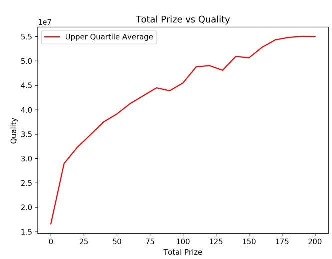
6.1 Total Prize Pool

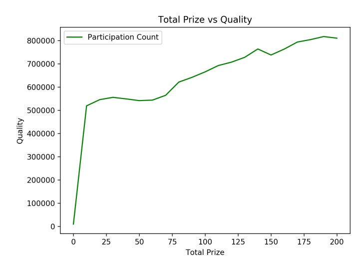
Keeping all other parameters as their default values, we changed the prize pool from 0 to 200 in increments of 10.



While one might originally expect this graph to be linear in the prize, since the cost of exerting effort e_i is e_i^4, it makes sense that these graphs are sublinear. Looking at total prize versus the fourth power of quality gives graphs that look linear:





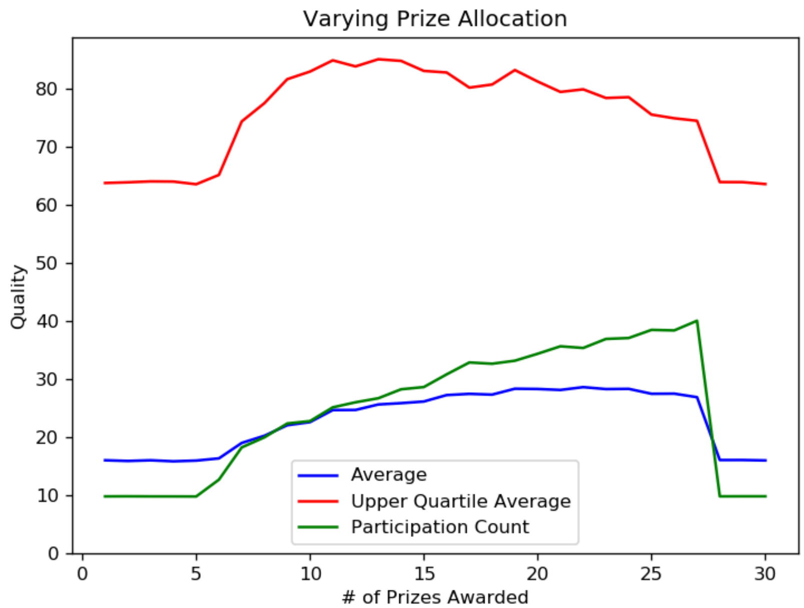


This means that increasing the prize pool has a mostly linear impact on effort, which results in sublinear impacts on the average quality and upper-quartile quality.

Interestingly here, the participation graph also looks linear when taken to the fourth power. This could just be a result of not enough data points, or perhaps (participation)^4 also increases linearly with the total prize.

6.2 Number of Bidders Who Receive Prizes

Keeping all other parameters as the defaults, including fixing the total prize at 100, we changed the prize array to allocate equally to the top k bidders, for each $1 \le k \le 30$.

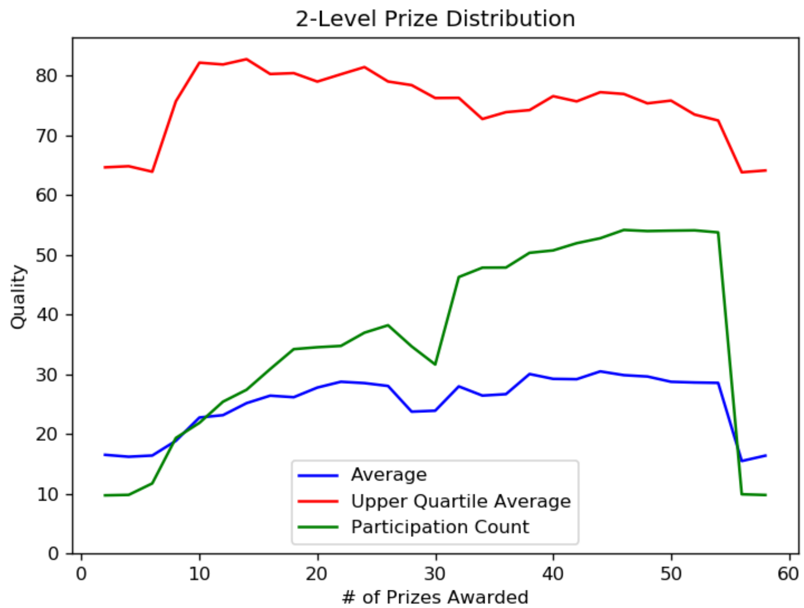


The upper quartile average is maximized at $k = 13$ (so each of the top 13 bidder receives 100/13 prize). This is pretty close to the choice by Greenwald et al. [2018] to allocate to the top $n/4$ bidders, where n is the number of bidders (in this case $n/4 = 15$).

Interestingly, though, the average bid (and thus the overall sum) as well as the participation count are both maximized later, at $k = 27$ for participation and $k = 22$ for overall average. So, while $k = 15$ is ideal for improving the average quality of the top bids, allocating smaller prizes at a higher quantity ($\sim 2/5$ of the total bidder count) is optimal for improving the total quality and participation.

6.3 Number of Bidders Who Receive Prizes, Across Two Tiers of Prizes

Again keeping all other parameters as defaults, and fixing the total prize at 100, we split the top n bidders into the top $n/2$ bidders and the next $n/2$ best bidders, and give a prize of $2x$ to each bidder in the first group and a prize of x to each bidder in the second group, such that the sum of all prizes is 100.



We find that the upper quartile average is maximized at $n = 12$, which corresponds to having the top 6 bidders be given a prize of $2x$ each, and the next 6 bidders be given a prize x each, where x is such that the total prize is 100 (in this case, $x = 100/18$). This is close to the optimal total number of bidders (13) from the previous part (with uniform prize allocation).

The participation count is maximized at $n = 46$, and the average is maximized at $n = 44$. It is interesting that the average and participation are both maximized at similar points, and yield graphs of similar shapes in both this section and the previous section; this indicates that the average bid is more of an indicator of participation count than it is the of the qualities of the top bids.

In this section, we discover that awarding prizes to around 2/3 of the bidders is ideal for maximizing the average bid and improving participation. This is different from the optimal fraction of bidders in the previous section ($\approx 2/5$).

7 Conclusion

In conclusion, we modeled crowdsourcing situations as all-pay auctions and reviewed past work on approximately optimal mechanisms where bidders' utility functions are convex. We also ran experiments under a behavioral game theory model and analyzed how well these mechanisms perform in simulations. From our results, we saw that the prize pool and participant effort is directly correlated under a linear relationship. Additionally, the effort exerted by a bidder with respect to the number of people receiving prizes exhibited an initial increase, followed by a decrease once too many bidders began to receive prizes; this is true in both scenarios with uniform prize allocations and two-tiers of prize allocations. In the future, we hope to experiment with more prize distributions, such as a linear decay based on rank, as well as different distributions for the number of bidders in each layer.

References

[1] Amy Greenwald, Takehiro Oyakawa, and Vasilis Syrgkanis. Simple vs optimal mechanisms in auctions with convex payments. `https://arxiv.org/pdf/1702.06062.pdf`, 2018.