Chapter 3

Simulating the Price of Anarchy

This chapter constructs a novel simulation of first-price single-payer auctions to demonstrate that the ratio of actual social welfare to optimal social welfare is within the bounds given by the the price of anarchy for this format. First, the construction of this simulation is discussed. Next, we demonstrate the results of running the simulation on bidders using known Bayes-Nash equilibrium strategies and an arbitrary bidding strategy. Finally, we demonstrate that these bounds also hold for no-regret learning agents in a fixed action variant of the first-price auction.

3.1 A Simulated Auction Environment

To simulate sequential first-price auctions, we construct a program in python that allows us to create an arbitrary number of bidders, each with valuation distributions of our choice who simultaneously bid on an item being auctioned for as many sequential auctions, or rounds, as we choose. Each round represents an auction for a new, but similar item where the valuation distributions for each bidder remains the same. At the beginning of each round, every bidder draws a new valuation from their distribution. The bidders then each simultaneously submit a bid to the auction and the winner is determined by the highest bid where ties are broken with equal prob-

ability among those with the same bid. After the winner is selected, they are given utility $u(v, b) = v - b$, the difference between their valuation and bid that round. At each round the total social welfare is v_i, the valuation of the winning bidder as per definition 3. The optimal social welfare each round then is the highest valuation of any bidder. These values are summed across auctions to get the total social welfare for this sequence of auctions and to see what the optimal social welfare would have been. This is laid out in the pseudo-code version of the sequential auction below where we use the super script t to denote which round each variable is from[1].

Algorithm 1: Sequential First-Price Single-Item Auction

Initialize SW, OW, and POA to zero

for $t = 1, \ldots, T$ **do**

Each bidder draws their valuation v_i^t from their distribution F_i;

Each bidder uses their strategy s_i^t to submit a bid;

The highest bidder is assgined the object and they pay their bid, if tie, a winner is choosen randomly among them;

Each player has their utilites updated according to if they won the object;

if *player i wins the auction* **then**
$SW \leftarrow SW + v_i^t$

if *player j has the highest valuation* **then**
$OW \leftarrow OW + v_j^t$

$POA \leftarrow \frac{SW}{OW}$

3.1.1 Simulating Bayes-Nash Equilibria

Using the simulation outlined above, we first demonstrate that bidders using known Bayes-Nash equilibrium strategies have an average price of anarchy greater than 0.66. First, we simulate the case of two bidders each drawing their valuation from the

[1]The "$\leftarrow$" symbol used in the algorithm means assignment of value. For example $x \leftarrow 1$ is the variable x is assigned a value of 1. This reduces the ambiguity of using the "=" symbol which could be also be a statement or proposition in pseudo-code.

uniform distribution $[0, 1]$. We are using the hard coded strategies that $s(v_i^t) = \frac{v_i^t}{2}$ for each bidder which was shown to be the unique Bayes-Nash Equilibrium in chapter 2. The results are shown in table 3.1 below, where the cumulative price of anarchy is given up to the specified round specified [2].

Round	POA
1	1.0000
10	1.0000
100	1.0000
1,000	1.0000
10,000	1.0000
100,000	1.0000

Table 3.1: Price of anarchy in two player symmetric auction

As can be seen this example is somewhat silly to simulate since this equilibrium is fully efficient. As each player bids half their valuation, the winner is always the player with the highest valuation. Hence the actual social welfare is always the same as the optimal social welfare:

$$\frac{SW(\mathbf{s}(\mathbf{v}); \mathbf{v})}{\text{OPT}(\mathbf{v})} = 1.$$

We now move to simulate the more interesting example of two bidders with asymmetric distributions. We let bidder one chose their valuation from the uniform distribution over the interval $[0, 1]$ and bidder two chooses their valuation from from the uniform distribution on the interval $[0, 2]$. As stated in chapter 2, the unique Bayes-Nash equilibrium for this auction is:

[2]The POA is calculated as per algorithm 1.

$$s_1(v_1) = \frac{4}{3v_1}\left(1 - \sqrt{1 - \frac{3v_1^2}{4}}\right)$$
$$s_2(v_2) = \frac{4}{3v_2}\left(\sqrt{1 + \frac{3v_2^2}{4}} - 1\right)$$

As stated in chapter 2, this auction is not fully efficient as the bidder who is drawing their valuation from the smaller distribution has to shade their bid less (bid higher relative to their valuation) in this equilibrium. We simulate this sequentially 100,000 times and get the following results as shown in table 3.2:

Round	POA
1	1.0000
10	1.0000
1,000	0.9919
10,000	0.9924
100,000	0.9935

Table 3.2: Price of anarchy in two player asymmetric auction

While this auction is not fully efficient, it is highly efficient. The price of anarchy in this auction never drops below 0.99. The main reason this happens is that while it is possible for the bidder drawing from the smaller distribution to win even if they have the smaller valuation, this only occurs when their two valuations are relatively close. This means that the social welfare lost in this case is not much even if it is not fully efficient. In both of these auction we see that they are well above the 0.66 lower bound guaranteed for all first-price single-item auction formats.

3.1.2 Minimal Intelligence Bidders

Before going on to simulate the auction using no-regret bidders, we first move to establish a baseline for how well each auction setting (based on number of bidders and distribution choice) performs with agents that are not learning. To do this we construct agents that bid randomly between zero and their valuation every round i.e. each players bid is chosen uniformly from the distribution $[0, v^t]$. We call these agents minimally intelligent since they are not overbidding, but they are also clearly using a nonsensical strategy for an auction. One should note that while this is a bad strategy, it is possible to formulate much worse strategies for our bidders from both utility maximization and efficiency standpoints [3]. This rather represents agents who are incapable of learning or doing their homework and thus randomly select from all options. The results for simulating sequential first price auctions with 2, 10, and 100 agents using this strategy are shown in the table 3.3 below.

Round	2 Agent POA	10 Agent POA	100 Agent POA
1	0.9000	0.8410	0.9859
10	0.9312	0.8987	0.9419
1,000	0.9279	0.8812	0.9480
10,000	0.9150	0.8880	0.9467
100,000	0.9164	0.8887	0.9470

Table 3.3: Price of anarchy in two player asymmetric auction

Table 3.3 shows how the price of anarchy evolves through the rounds as the number of rounds goes to 100,000. Since the agents are not learning anything and the bids are randomly sampled each time, these efficiency results should approximately converge to the expected value as $T \to \infty$. To better illustrate the relationship between price of anarchy in each of these sequential auctions and the number of bidders, we graph the above simulation now run on 2 to 100 bidders. Each of these simulations is run

[3] It's rather fun to think up such strategies! For example, if each bidder shaded little when they had a low draw and shaded a lot when they had a high valuation, this can lead to highly inefficient outcomes

100 times, and the min, max, and average POA are graphed below in figure 3.1.

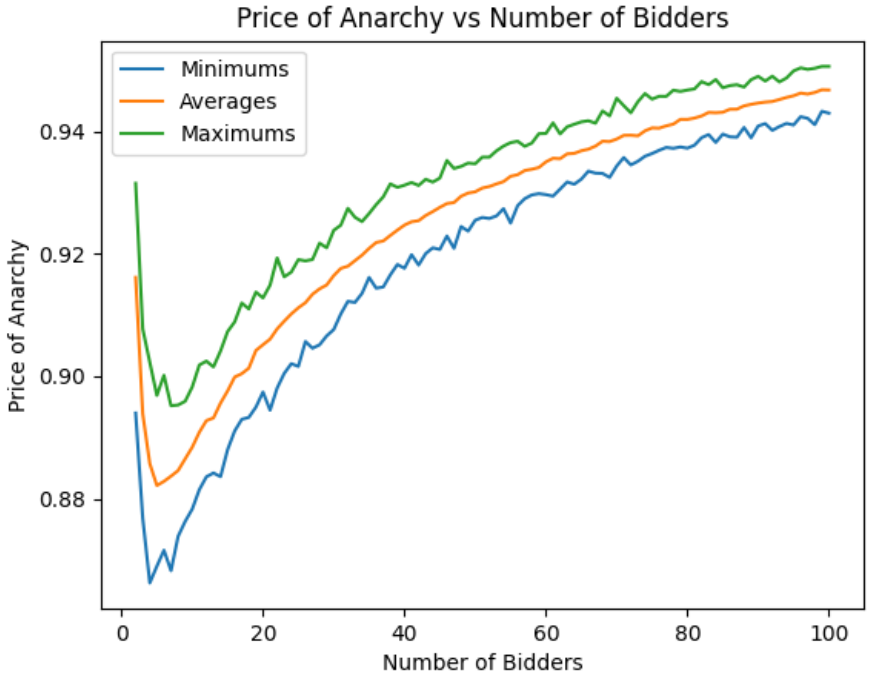

Figure 3.1: POA for 2 to 100 symmetric bidders

We see three things, one is that when all bidders are drawing from the same uniform distribution the market is incredibly efficient regardless of how smart the bidders are when choosing their strategy. That is, even when their strategies are arbitrary, this auction still performs reasonably well. The second thing to note is that in this setting, as the number of bidders increases the social welfare also increases. This makes sense as we would expect the probability of a reasonably high valuation winning to increase as there are more valuations per round. The third thing we notice, and most surprising of all, is that the increase in the price of anarchy as the number of players increases is not monotonic. At lower level of bidders, the efficiency actually decreases as we add more bidders. The minimum POA here is especially low.

We now conduct a similar simulation but in the case of asymmetric bidders. In this case half of the bidders are drawing uniformly from $[0, 1]$ and half from $[0, 2]$. The results are shown in table 3.4.

Round	2 Agent POA	10 Agent POA	100 Agent POA
1	0.1.000	1.0000	0.9726
10	0.9115	0.8178	0.9331
1,000	0.9054	0.8658	0.9310
10,000	0.9092	0.8630	0.9297
100,000	0.9112	0.8640	0.9304

Table 3.4: Price of anarchy in two player asymmetric auction

Again we see that the market is quite efficient in this case regardless of how smart the bidders are. We again also see that it seems to converge to optimal (i.e. to 1) as the number of bidders increases but non-monotonically. We simulate this for 2 to 100 bidders again 10 times at each number of bidders, where on odd numbers we allow the extra bidder to have distribution $[0, 1]$. This is shown is figure 3.2.

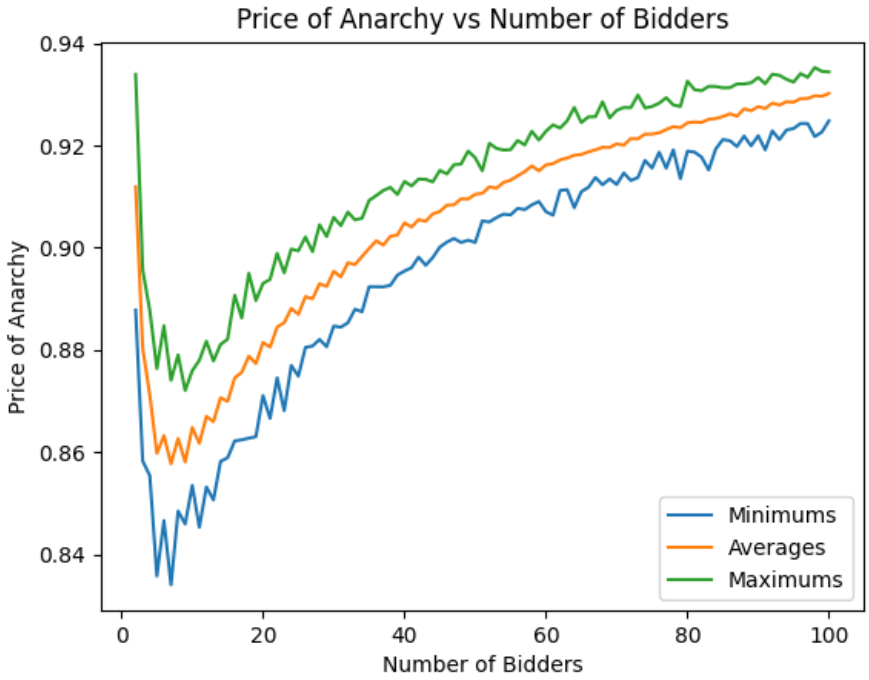

Figure 3.2: POA for 2 to 100 asymmetric bidders

Note that the baseline POA for minimal-intelligence agent auctions changes as we change the distributions. While the format seems quite efficient and well above the price of anarchy bounds (which again are only guaranteed for equilibria which this

is not) even with agents following these minimally intelligent strategies, we have no guarantee that there is not some way to construct this that these agents wouldn't do much worse. Given that it seems as the number of agents increases, the efficiency also seems to increase, one would expect that this would need to be done by picking more interesting distributions for the bidders to choose from and only having two bidders.

3.2 Simulating No-Regret Bidders

With the results from the Bayes-Nash equilibrium demonstrated and an efficiency baseline established, we move on simulating the bounds of coarse correlated equilibria in first-price auctions. Again, this is the equilibria we expect auctions to converge to if each agent is using a no-regret learning algorithm. To use these algorithms we do however have to make a concession to the environment we are simulating: we must now simulate an auction where the bidders are only allowed a finite number of actions.

3.2.1 Multiplicative Weights Algorithm

The no-regret learning algorithm we will use to train our bidders is called the multiplicative weights algorithm. It has been shown to satisfy the no-regret property in papers such as TODO and TODO but our implementation of it comes from Roughgarden (2016) and thus if each of our bidders use it our auction should converge to a coarse correlated equilibrium.

Before giving the algorithm, a few words are probably necessary to understand where it comes from and what it is trying to do. First, this algorithm is what is called an *online* algorithm. That is an algorithm that takes its inputs sequentially as it goes rather than getting all of its inputs up front. Next, this algorithm and many other algorithms for players learning in games are based around the player only

having a fixed number of actions they can choose from. For each round in the game, the player gets outside advice from "experts" who recommend to the player what to do at each round and the player picks among them to decide what to do. For us, these experts are our strategies that will map a valuation to a bid. At each time step t, the player picks the action to play and then after that, some adversary picks the utilities to assign for each action that could have been taken. This is a stronger condition then we will need as the adversary in a first price auction is the cumulative action of the other players, where the highest bid determines which strategies (if any) the player could have taken and won. However, in the general case this algorithm has been shown to be no-regret in the face of an adversary directly picking the utilities the learning agent receives.

Algorithm 2: Multiplicative Weights (MW) Algorithm

Initialize $w^1(a) = 1$ for every $a \in A$

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

Use distribution $p^t = \frac{w^t}{\sum_{a \in A} w^t(a)}$ over actions to pick $a \in A$ and output a.

Given the utility vector u^t, for every action $a \in A$ use the formula

$w^{t+1}(a) = w^t(a) \cdot (1 - \eta u^t(a))$ to update its weight.

The logic of this algorithm is simple. At each time step we see how well each of the possible actions performed and increase the weight, or probability of selecting that action in the future. This increase is done proportionally to how well the action did as determined and *learning rate*, η chosen before starting the procedure. As demonstrated in Roughgarden and Blum and Mansour (2007), the MW algorithm is no-regret if $\eta = \sqrt{(\ln n)/T}$ where n is the number of actions that this agent can choose from, and T is the number of rounds that will be played (yes this assumes that the player know that up front).

This algorithm fulfills our purpose of simulating agents learning in auctions, but

in some sense it is unsatisfying that the learning algorithm requires a finite set of actions that the bidder does not even get to choose. It would be more interesting if we were able to give our agents some reasonable algorithm that allowed them to formulate their own strategies or mappings between their valuation and bid rather than choosing from a pre-made set. This would introduce a whole host of other problems from the reasonableness of expecting agents to implement such algorithms to the ability to prove that such algorithms converge to an equilibria. Part of the beauty of using multiplicative weights is that it is simple and has nice mathematical properties. Using more interesting learning techniques these properties might no longer hold and then we would have to wonder what our simulation is really showing[4]?

3.2.2 Uniform Distribution Simulations

The first simulation we run is a repeat of the symmetric auction setting, but now using agents learning with the multiplicative weights algorithm. We create 101 strategies that bidders can choose from to shade their bid, from bidding zero percent of their valuation with one percent increases up to bidding 100 percent of their valuation. that is $S = \{0, 0.01 \cdot v_i, 0.02 \cdot v_i, \ldots, 0.99 \cdot v_i, v_i\}$. First, we run the simulation with two symmetric bidders each choosing their valuation from the uniform distribution over $[0, 1]$. The results are shown below in table 3.5

Round	POA
1	1.0000
10	0.9517
1,000	0.9129
10,000	0.9578
100,000	0.9947

Table 3.5: Price of anarchy in two player asymmetric auction with no-regret learning

[4]This thesis grew out of an interest in doing just that, throwing "smarter" algorithms such as neural networks and machine learning into existing multi-agent simulations. It becomes hard to tell what the point of such simulations are when the dynamics might just be properties of the interaction of the specific algorithms used and not of the system itself.

Here we can see that after 100,000 rounds the price of anarchy converges to 0.99 and near perfect efficiency as the agents learn how to play the game. It's important to point out here that the above table is not an average, but simply one run of the simulation. Since each time we run the simulation it is possible for the bidders to learn to converge to a new equilibrium, the POA values for each simulation can be different. Averages don't make sense in this context as we are more concerned with the lowest possible POA that the system converges too. We now repeat this using 2 through 100 agents, each symmetric and drawing from a uniform $[0, 1]$ as above. We run 100 simulations with each number of agents, each for 100,000 rounds[5]. This gives us the following results as shown in figure 3.3

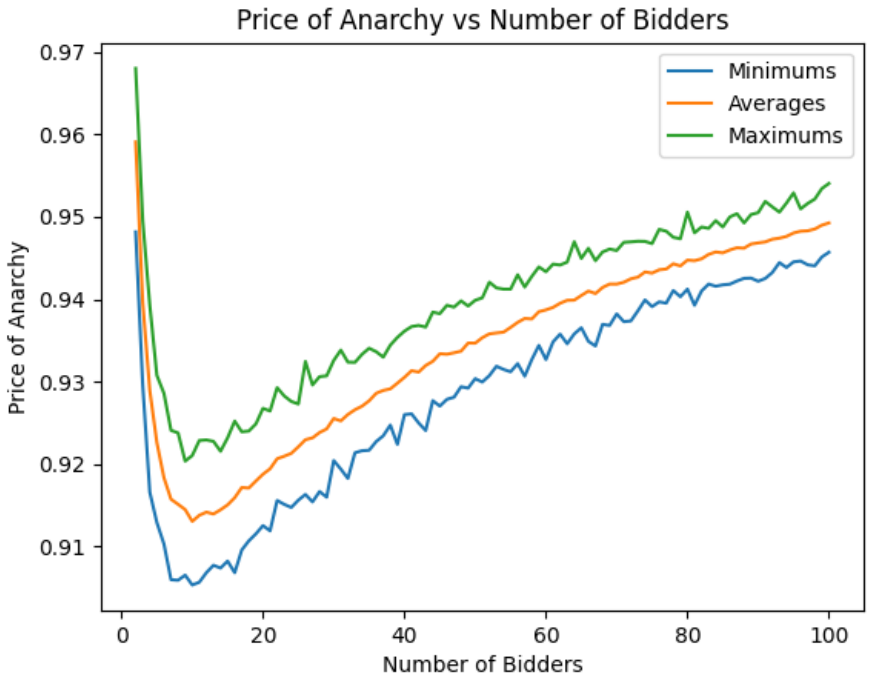

Figure 3.3: POA for 2 to 100 symmetric bidders

We can see here that our efficiency in the minimum case with our learning agents is about as good as the best-case with the random guessing agents. It makes since

[5]This is a relatively small number, but necessarily chosen for the sake of computation time with 1,000 agents

that the efficiency would increase as we would expect people with higher valuations to win more often as people learn to minimize their regret. It is theoretically possible in some games that social welfare could be better off with everyone choosing arbitrary strategies then under learning agents, but we see in this game, and with our learning agents that is not the case.

Finally, we simulate the case of having two bidders with asymmetric valuation distributions where one draws uniformly from $[0, 1]$ and the other draws from $[0, 2]$. We can see the result of simulating this 100,000 times in table 3.6 below.

Round	POA
1	1.0000
10	0.8901
1,000	0.9314
10,000	0.9770
100,000	0.9961

Table 3.6: Price of anarchy in two player asymmetric auction with no-regret learning

Here we see the agents converge to a very good efficiency similar to the behavior we saw when using the Bayes-Nash equilibrium for this setting and better than in the two bidder minimum-intelligence case. Now again we conduct this simulation with 2 to 100 asymmetric bidders. For each number of bidders we simulate this 100 times, when there are an odd number of bidders, there is an extra $[0, 1]$ bidder. Each sequential auction consists of 100,000 rounds. The results are shown in figure 3.4

Figure 3.4 is similar to the figures we have seen for every other auction. We see very high efficiency for two or three bidders, and then see a dramatic decrease as it goes to 9. We then see the numbers slowly climb back towards being fully efficient in the best, worst, and average cases.

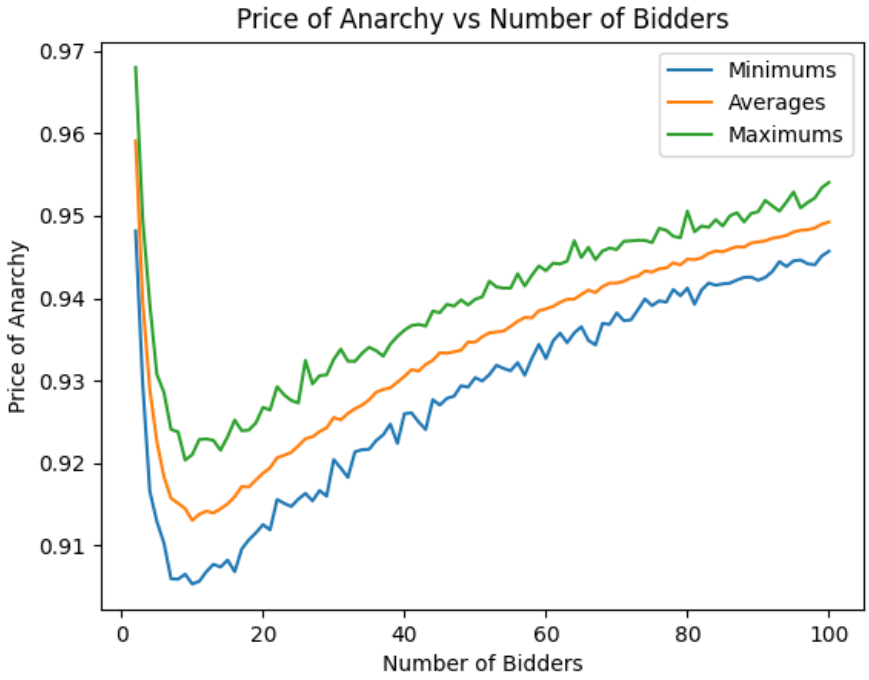

Figure 3.4: POA for 2 to 100 asymmetric bidders