Simulating Auctions

A Thesis

Presented to

The Division of History and Social Sciences

Reed College

In Partial Fulfillment

of the Requirements for the Degree

Bachelor of Arts

Robert S. Irvin

May 2020

Approved for the Division
(Economics and Mathematics FIXME)

Jeffery Parker   David Perkinson

Acknowledgements

The cat, Pinoe.

Preface

This is an example of a thesis setup to use the reed thesis document class.

List of Abbreviations

You can always change the way your abbreviations are formatted. Play around with it yourself, use tables, or come to CUS if you'd like to change the way it looks. You can also completely remove this chapter if you have no need for a list of abbreviations. Here is an example of what this could look like:

AI	Artificial Intelligence
MAS	Multi-Agent System
MDP	Markov Decision Processes
ML	Machine Learning

Contents

List of Tables

List of Figures

Abstract

The preface pretty much says it all.

Dedication

You can have a dedication here if you wish.

Introduction

This thesis will lay out the literature on the price of anarchy bounds for single item, first price auctions and then will construct a simulation to demonstrate these bounds for no-regret agents. Chapter one hopes to be a broad overview of the field to give context for this discussion covering price of anarchy, auctions, and simulation. Chapter two will go through the mathematical theory in a more rigorous way laying out what sort of behavior we expect and why for no-regret agents in these auctions as well as how the price of anarchy for first price, single payer auctions were derived. Chapter three will be the empirical results of a simulation which hopefully accord with what the theory says.

Note to the readers of this draft of chapter 1: the in-text citations generated by Latex are formatted incorrectly/in the wrong style for this draft and I will talk to CUS to try and get this fixed. These will look like (Author1 & Author2 (year)) which is incorrect.

Chapter 1

Computational Economics and Auctions

Within the last three decades, computational economics has been on the rise. This branch of economic research encompasses two major ideas. One is that the increasing power of computers can help solve and understand classical economic problems through increasingly more complex simulations and numerical analysis. Two, that the mathematical methods developed in the field of theoretical computer science can be used to gain better understanding of existing models in terms of their algorithmic and computational complexity properties. We aim to take elements from both of these frameworks to better understand the social welfare of auctions at equilibrium.

1.1 Auctions, Equilibria, and Anarchy

It is perhaps obvious why economists would be inserted in studying auctions. Auctions are one of the most basic market structures that have roots going back to at least the ancient Greeks and still exist today in places such as art auctions and Ebay (Mochon & Saez (2015)). But in the past 15 years, economists have been joined by computer scientists who are increasingly interested in the strategic interactions of

agents within this setting. Christos Papadimitriou said in a 2015 lecture at the Simons Institute that it was the advent of the internet, an artifact out of their control, that turned theoretical computer science into a "physical science." Now computer scientists had to "approach the internet with the same humility that economists approach the market..." He went on to say that "it also turned us [computer science] into a social science. It was obviously about people and incentives. Without understanding this, you cannot understand the internet" (Papadimitriou (2015)). It is within this framework that computer scientists first began to study auctions as they existed on the internet. At first, this meant trying to understand the auctions present there such as Ebay and Googles sponsored search auctions. But now, they are moving beyond that and taking the mathematical tools of theoretical computer science and applying them as a lens to understand and explain the world beyond the scope of the internet. Here specifically we are looking at the field algorithmic game theory, the study of the algorithms and complexity of strategic interactions. For economists, this can be thought of as a new toolbox for unpacking and understanding the models and structures that already dominate the field. For example, in the case of a Walrasian auctioneer who calculates the clearing prices of a combinatorial auction, their problem was shown to be NP-complete, a complexity class usually called "intractable" due to the time it takes to solve these problems (the best algorithms here are generally guess and check, which gets out of hand for large inputs i.e. possible combinations) (Papadimitriou (2015))[1]. Using the lens of computational complexity is one way of assessing what assumptions we are making about the computational power of our rational agents.

---

[1]NP is the class of problems that a given solution can be checked in polynomial time, i.e. $O(n^k)$ operations where n is the size of the input and k is any positive integer. NP-complete means that all problems in NP reduce to solving this problem. This is when you will usually hear "taking more time to compute then the age of the universe" etc...

1.1.1   The Price of Anarchy

Another idea that has come out of algorithmic game theory is the price of anarchy (POA), a mathematical way of showing the difference between the social welfare in the optimal case and in the worst case equilibrium for a game. More formally, the price of anarchy for a game is the ratio of the minimum equilibrium social welfare in a game over the best possible social welfare of the game. An illustrative example of the price of anarchy can be seen in selfish routing games such as the one pictured below in figure 1.1 that comes from Tim Roughgarden's 2016 book "20 Lectures on Algorithmic Game Theory". In this game, each player chooses which edges to take
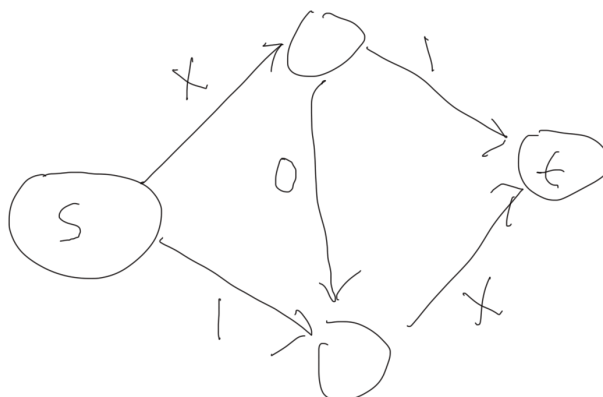
Figure 1.1: Routing game from s to t. Strategic interaction will make everyone worse off.

on the directed graph from the source s to reach the terminal t and try to do it in the least time, total weight of edges taken, as possible. For the edges labeled x, x is equal to the proportion of players who take the route.[2] For example if 50% of the players take that edge then $x = 0.5$. This can be seen as analogous to traffic when driving a car, the more people take a road, the slower the traffic goes and the longer it takes to get somewhere. Knowing this, each player must choose which path to take to get to their destination as quickly as possible. As you can quickly verify, the best solution

---

[2]These figures are slightly off. The edges should have weights $f(x) = x$ or $f(x) = 1$ giving us the cost of using the edge as a function of the proportion of people who are using that edge. This will be fixed when I draw these graphs properly for the final draft and not in MS paint.

for society, minimizing the total driving time for all, is when half of the drivers take the top route, and half of the drivers take the bottom route taking in total 1.5 for each driver.
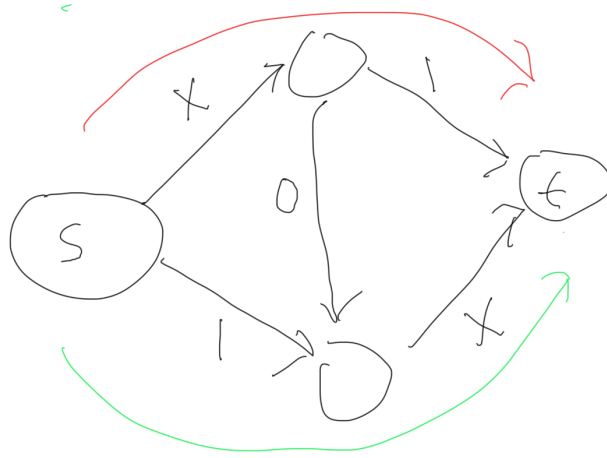
Figure 1.2: Socially optimal route: half taking top, half taking bottom

However, this is not an equilibrium. The drivers taking the top route have incentive to instead take the middle edge to try and lower their total time traveled. In fact, the Nash equilibrium will end up with all of the drivers taking the top x, going through the middle, and then the bottom x. Only then will no players have reason to deviate. These actions to individually and strategically try to decrease their cost end up producing a worse outcome for all players and society as a whole (the social welfare)[3].

Now, it will take time 2 for each player to reach the terminal. Thus, under strategic interaction we see that the equilibrium is sub-optimal for society, and in this case all players as well. In this case, the price of anarchy can be computed as

$$POA = 3/4$$

---

[3] A careful observer will note that if the edge with cost 0 were not there, this would not be a problem. This is called Braess's Paradox where having this extra edge counter intuitively leads to worse outcomes. (Roughgarden (2016))
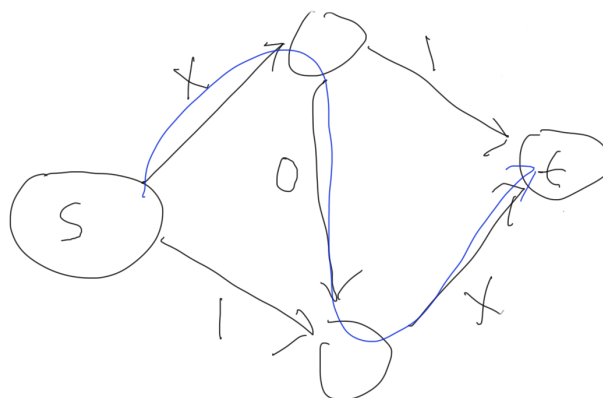
Figure 1.3: Route under strategic interaction

. Interestingly, bounds for the price of anarchy can be found for the class of game as a whole regardless of its individual construction. The price of anarchy bound for these selfish routing games is know to be 3/4, meaning that our example is as bad as the price of anarchy could get for any selfish routing game.

1.1.2 Auctions

This framework of the price of anarchy is now being applied to understand auctions by researchers in the algorithmic game theory field. Where for selfish routing games, the price of anarchy has been shown to have the tight bound above, the price of anarchy for single price auctions only has approximate upper and lower bounds set on it, and is not known at all for double auctions. Before exploring this further, we should set up the mathematical framework of auctions that is necessary to compute such bounds.

An **auction** is a market mechanism, operating under specific rules that determines to whom one or more items will be awarded and at what price. For a bidder in an auction, the **value** v_i is how much the bidder values the item. This is sometimes called a private valuation as this value is generally unknown by the other participants

in the auction. The **bid**, b_i, is the offer that bidder i submits for for an item. **Sincere bidding** is when $v_i = b_i$, **underbidding** is when $v_i > b_i$, and **overbidding** is when $v_i < b_i$. The highest bid made by any bidder is denoted b^*, and is the winning bid (if multiple bids equal the winning bid, then some tie-breaking rule must be used). The **selling price**, p^* is the final price that the bidder actually pays for the item (which depending on the auction type need not equal b^*). Under the **first-price** rule, the bid submitted by the winner is equal to the selling price. Before the auction begins, each bidder knows their personal or private value for the item. An auction consists of a set of bidders, $I = (1, 2, ..., N)$ and a seller. After the auction, the bidder i wins the item if their bid is higher than the bid placed by any other bidder k, $b_i > max_{k \neq b_k}$. In a single unit auction, the **income of the bidder** i is equal to their value of the item:

$$\Gamma_i^* = v_i$$

and the **surplus of the bidder** i is equal to the difference between income and price paid

$$\Pi_i^* = \Gamma_i^* - p^*.$$

If a bid placed by a bidder is less than the winning bid, they do not win anything and their income and surplus are both zero. The **seller's revenue** in a single unit auction is equal to the price paid by the winning bidder

$$R^* = p^*.$$

There are multiple different pricing rules in an auction that determine who gets allocated the item. In a **First-Price** auction the winning bidder pays the amount of their bid, which is the highest bid of the auction: $p^* = b^*$. Also called **pay-what-you-bid** (PWYB). In a **Second-price** auction, the winning bidder pays an amount that is equal to the second highest bid for the awarded item Vickrey (1961)

[4]. People tend to bid lower than their private valuation in first price auctions since if they bid that value, their profit is zero. Thus, the first price rule the item could be awarded to someone who values the item less than other bidders. Second price auctions encourages bidder to bid their true values (as they will gain positive profits if they win no matter the second highest bid). This encourages an efficient allocation of items (Mochon & Saez (2015)). For the moment we will confine ourselves to first price auctions as this is where most of the strong results in POA analysis of auctions currently are.

Before discussing the known bounds on the price of anarchy for first price, single payer auctions, it is worth understanding how the bids might lead to a non-optimal outcome or what we mean by that. Due to only knowing their own valuation of the good, each bidder must act under uncertainty as to how much they should bid to beat out the unknown valuations of the other bidders. However, if they bid their exact private valuation then they will get a surplus of zero, or zero utility. In order for them to get some utility for the item they must be paying less then the exact amount they value the item. How much each bidder should bid less than their valuation, or "shade", their bid is determined by how much they think that the other party values the item. To capture this interaction auctions are represented as Bayesian games where each bidder is drawing their bids from distributions known to the other player (that need not be the same). If one player knows that the other is drawing from a distribution with a smaller mean than they are (i.e. probably doesn't value the item as much), the Bayes-Nash equilibrium will have them shade their bid less and this other person will shade their bid more. This can lead to the person who values the item less winning the auction and creating less social welfare (the summed surplus of all bidders and the seller).

Syrgkanis and Tardos proved in 2013 that the price of anarchy in first price, single

---

[4]The second price, sealed bid auction is also known as a Vickrey auction after the economist who invented it

item auctions is at least $1 - \frac{1}{e} \approx 0.63$. The exact upper bound on the price of anarchy for single payer, first price auctions remains unknown in the general case. This bound is true regardless of how many bidders there are or what distributions they are drawing their bids from.

This moves closer to answering the question for what is the price of anarchy at equilibrium, but these results do not pay attention to how the players arrive at these equilibria. In real world applications, we expect that players might play repeatedly in the same auction and learn as they play rather than come in with pre-computed strategies. This is especially true for when computing the equilibrium is computationally hard and the stakes of each individual auction is small. Given these observations, it is natural to ask questions about how the efficiency results carry over to adaptive game environments. The model for learning agents that is commonly used in the field is **no-regret learning**. An algorithm for a player satisfies the no-regret condition if, in the limit as the number of times the game is played goes to infinity, the average reward of the algorithm is at least as good as the average reward for the best fixed action in hindsight (assuming the sequence of actions for the other players remains unchanged)[5]. If each player incorporates this kind of learning algorithm, then it has been shown that these can converge to a larger class of equilibrium called correlated equilibrium where each player conditions their response on the expected action of the other player. Luckily, the previous theorem has been extended so that we know that the price of anarchy for the correlated equilibria of first price auctions are also at least 0.63 (Roughgarden et al. (2017))

With all of this set up, we now state our goal: to simulate no-regret learning algorithms for agents in a first price, single payer auction to see how well the price of anarchy holds. We also want to compare this with other learning algorithms that

---

[5]That is to say that the algorithm will converge to having a loss no worse than any fixed strategy we would have rather picked in hindsight as the limit goes to infinity. A more precise definition, example algorithms, and uses will be shown in chapter 2 to clarify what "regret" is and how this converges to zero

aren't no regret to see if this framework of using these algorithms is appropriate for making generalizations about equilibrium under learning.

1.2 Simulated Agents and Simulated Economies

The use of computers in economics goes back all of the way to general purpose computers being invented in the 1940's. Wassily Leontif used a computer to invert a 39 x 39 matrix to help solve his input output model. Since then, computers use in economics has exploded. With computers, economists are able to solve bigger matrices, do Monte-Carlo simulations, create multinomial probit models, and use full information maximum likelihood estimation (Backhouse & Cherrier (2016)). While one branch of computational economics is focused on creating stronger and stronger calculators to facilitate empirical research, another branch has focused on creating simulated economies that allow economists to construct a blended version of theory and research within a computer program. Within these simulated economies, theories can be coded into the simulation which, when run, can allow the researcher to conduct experiments that might not be practical to conduct in the real world.

One kind of a simulation that can be run is called an agent-based model (ABM), a simulated system of autonomous decision makers (agents). These models are able to generate complex behavior even if only simple assumptions are made about the behavior of the coded agents. That is, these agents interacting with each other in complex ways are able to produce emergent phenomena in the macro structure of the system.

For example, in the 1970's Thomas Schilling created a computer simulation to try and understand how and why self segregated neighborhoods formed. He coded a virtual environment where agents were given a simple preference, they are only happy if they are not the minority in their neighborhood and will keep moving otherwise.

This simple model can illustrate the main tenets of agent based modeling. First, we have agents who are representative of people in the real world. Their preferences to this respect are simple and easy to understand where they are only "happy" if half of their closest neighbors are the same as them. If they are not happy they will move somewhere else arbitrarily. These preferences can be represented as the short procedure, or algorithm, shown below where S is just the space they live in.

1. Draw a random location in S

2. If happy at new location, move there

3. Else, go to step 1

In this case, we get to choose what that environment looks like and like Shelling we can just say that it is a one by one unit square and we can say that their neighbors are the ten closest people to them (in Euclidean distance) on that square. When you run this simulation with green and orange dots representing the types of people you get the following behavior cycling through each of our 250 agents with the above procedure until every agent is happy.
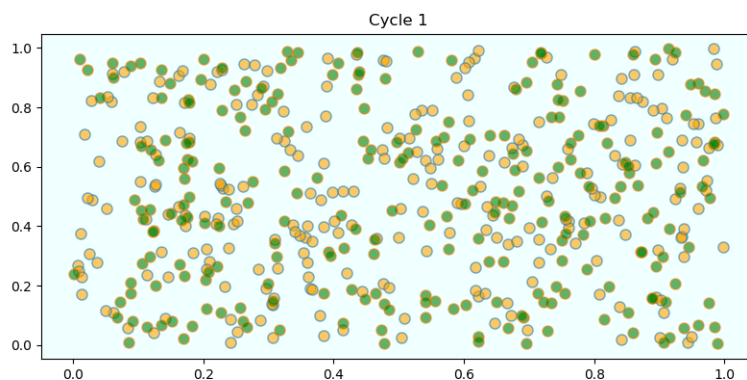
Figure 1.4: Schelling's Segregation Model: Cycle 1

As can be seen, in cycle 1 (Figure 1.4), the agents are well distributed among each other, but as they move in cycles 2-5, they become progressively more segregated. After 5 cycles all agents are happy and the simulation terminates. With very few
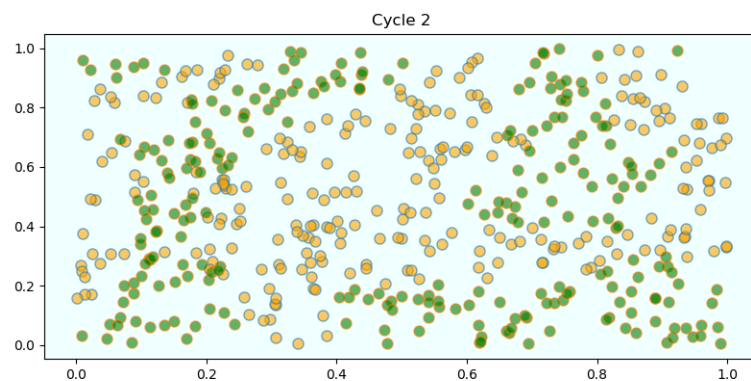
Figure 1.5: Schelling's Segregation Model: Cycle 2
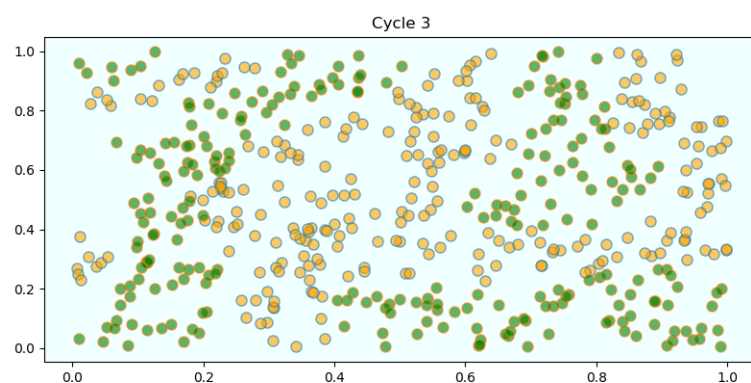
Figure 1.6: Schelling's Segregation Model: Cycle 3
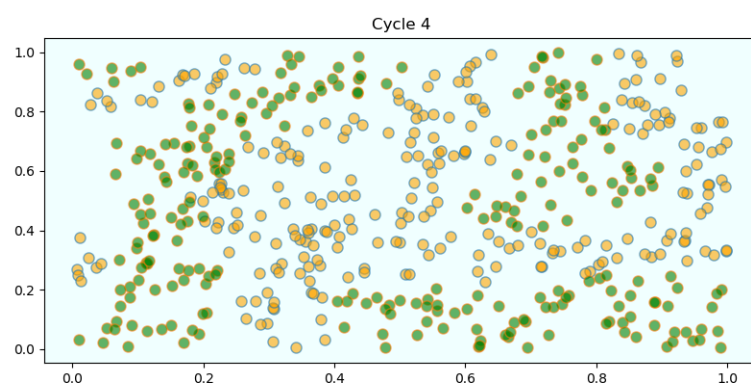
Figure 1.7: Schelling's Segregation Model: Cycle 4

assumptions about the agents' preferences, we can see the resulting emergent behavior of segregated neighborhoods in the system as a whole. Not only that, but the agents
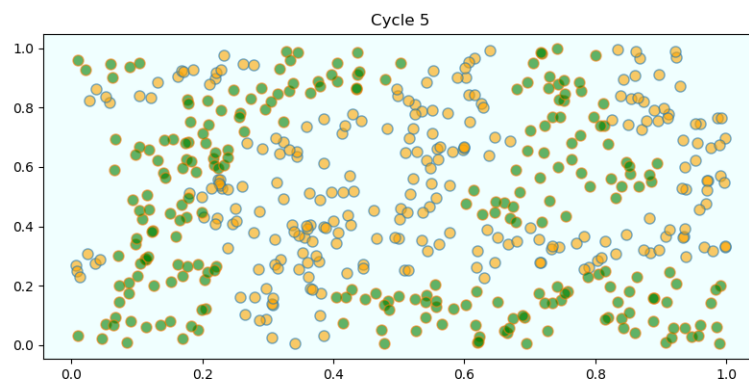
Figure 1.8: Schelling's Segregation Model: Cycle 5

naturally move to an equilibrium as they adjust their behavior to what their neighbors are doing (Sargent & Stachurski (2019)).

Agent based modeling has been used to build and understand much more complicated systems then the example illustrated above. The Santa Fe Institute in New Mexico is one of the main proponents of agent based modeling releasing a manifesto supporting using it to understand the "complexity" of economics from the ground up (Backhouse & Cherrier, 2016). They built the Santa Fe Artificial Stock Market in the 1990's to try and simulate the behavior of agents on the stock market and how they adapt their adapting trading strategies effects the outcome of the market. This is one of the first examples of agents learning and adapting to their environments as part of the model (LeBaron (2002)). In this model agents used a genetic algorithm to adapt their trading strategy at each period by modifying a string (for example 00011100) where each bit in the string told the agent to use a certain behavior or not. Those trading strategies that did well were coded to survive longer where the agents with worse strategies would randomly modify their own or take a more successful agents strategy (Arthur (1992)). This was supposed to be representative of the learning of traders on the stock market so that the insights taken from running these agents in simulation could be applied to learn something about the real world.

The authors of Santa Fe Stock Market paper at the time suggested that this was

one of many algorithms that could be used to stand in for human behavior saying that reinforcement learning or deep learning could also be used to stand in for human intelligence[6] (CITATION NEEDED, This was in Shareen Joshi's thesis and in one of their papers, but I am having trouble tracking down the quote. Will use her thesis to find original source or remove reference). Recent research in the field as well as in the field of multi-agent systems, a similar branch of computer science suggest that this is not the case. Because these simulations have agents competing in non-stationary environments that are changing from the perspective of any individual agent every time some other agent changes their behavior, the choice of algorithm dramatically changes how the system behaves (Rejeb & Guessoum (2005), Shoham & Leyton-Brown (2008)). Further, research that has compared the behavior of human agents in strategic settings to that of these algorithms have found that there is no general algorithm that best approximates human learning(Tesfatsion (2002)). From situation to situation, different algorithms more appropriately behave like humans. This is a problem only assuming that you want your agents to in some way represent human behavior, we might simply want our agents to represent the "rational choice" in any given situation. These algorithms aren't necessarily doing that either. As Holland and Miller say in their 1991 paper "Artificial Adaptive Agents in Economic Theory", "Usually there is only one way to be fully rational, but there are many ways to be less rational." The way they suggest to get around this is to try and build models that have robust behavior across algorithm choice. This is probably not true for their own artificial stock market as it is not even robust across the choice of parameters to program the market. The literature of game theory provides a nice solution to this. The no-regret algorithms they use have the simple property of doing better in the long run than fixing their strategy randomly in the beginning. This is a simple learning

---

[6]Reinforcement learning is when an algorithm plays a game repeatedly and updates its beliefs about what actions will lead to the best payoff. Deep learning uses deep neural networks to try and estimate the best outcomes in a fashion similar to regression.

requirement that means that this kind of algorithm should be more robust across representing human behavior. It also allows us to combine our simulated models and analytical models in a nice way as we can mathematically describe the processes of such algorithms behavior and we can code agents to use algorithms that have that behavior. This allows the learning process to be less of a black box and instead to be as simple and thus generalizable as possible.

Moving forward this thesis aims to better understand the price of anarchy in auctions by simulating auctions using these no-regret learning algorithms[7] and seeing how they behave compared to the socially optimal equilibrium. These simulations will try to answer two main questions, what equilibrium do these algorithms converge to (if they converge at all) and what is the calculated price of anarchy in these systems compared to the bounds that theory tells us should exist. Using this framework to explore auctions we will start with a single-item first price auction as it is the best understood theoretically and is the easiest to code. We will then move to simultaneous first price auctions for complementary goods as this is also relatively well understood in terms of the price of anarchy. Finally, time permitting, we will discuss the open question of the price of anarchy in double auctions and discuss how one might go about proving bounds for it.

---

[7]these will be discussed in chapter 2, but no-regret learning is a type of reinforcement learning that have certain convergence properties

Chapter 2

Price of Anarchy in First Price Auctions

This chapter lays out the mathematical framework for modeling auctions as Bayesian games of incomplete information, formally defines the price of anarchy in auctions, and shows the bounds on the price of anarchy for first price, single payer auctions.

2.1 Auctions as Bayesian Game

Auctions are typically modeled as Bayesian games also known as games with incomplete information. While it is obvious why an auction consisting of the strategic interaction of bidders could be modeled as a game, how it should be modeled requires some thought. After all, while each bidder knows their own valuation of the item being sold, unless for some reason (and against their own interest) the other players announced their valuation of the item before the auction began our bidder will not know how the other players value the item. In fact, it is persicly this lack of information that makes auctions interesting! If all players came into an auction knowing the valuation of other players, for first price, single item auctions the bidder with the highest value would always bid slightly higher than the next highest valuation and

win the item (or bid their own valuation if there is a tie for first). Rather it is the uncertainty that players face about the valuation of other players that make things interesting as players must guess how much they should shade their bid (as again no player will ever bid above their valuation) based off of what they know about other players.

What can we say that bidders know about the valuations of other bidders? Certainly they do not know nothing as we all have reasonable expectations about what some item is worth to others. No one will value a candy bar at a million dollars. But a collector of candy bars might value it at a higher value than an ordinary person who just want to eat the candy bar. For each, could say that this bidder has a probability distribution from which they are drawing their valuation from that is know to the other bidders. For example we might expect normal people valuation of candy bars to be a normal distribution centered at \$1, but the collectors value might be a leplacian distribution (allowing for more black swan events) centered at \$5. This sort of strategic interaction where the players know the distributions that parameters of the game are drawn from are typically called games of incomplete information, or Bayesian games.

2.1.1   Bayesian Games

Bayesian games of incomplete information are games in which one or more of the players don't have "full knowledge" of the game that is being played. Introduced by John C. Harsanyi in 1967, rather than players knowing every parameter of the game situation such as utility functions, possible strategies, and information held by other players, each player knows a probability distribution from which these will be drawn. In his paper, Harsanyi says that this type of game can be thought of as a normal game, where "nature" goes first drawing from these probability distributions and assigning values before play begins without the players knowing which specific variation of the

game they are playing (TODO CITE Harsanyi). Importantly, each player does know the probability distributions from which each value is selected. Formally such games are defined as follows,

Definition 1. A game with incomplete information, also known as a Bayesian game where there are I players, $G = (\Theta, S, P, u)$ consists of:

1. A set $\Theta = \Theta_1 \times \ldots \times \Theta_I$, where Θ_i is the finite set of possible types for player i.

2. A set $S = S_1 \times \ldots \times S_I$, where S_i is the set of possible strategies for player i

3. A joint probability distribution $p(\theta_1, \ldots, \theta_I)$ over types. For finite type space, assume that $p(\theta_i) > 0$ for all $\theta_i \in \Theta_i$

4. Payoff functions $u_i : S \times \Theta \to \mathbb{R}$

(TODO CITE Levin)

Using this definition to model our auction the types of players will consist of the publically known distribution from which they are drawing their valuation. That is, our auction will consist of bidders who know their own valuation of the item being bidded on, and the distribution from which each of the other players is drawing their own valuations. In a first price, single item auction if player i wins with bid b_i, we define their payoff the winner to be $u_i = v_i - b_i$, the difference between their valuation and their bid and the losers all get $u_i = 0$ since they did not receive the item. Now, a strategy for a player is a function $s_i \in S_i$ that maps a valuation v_i in support of Θ_i to a bid $s_i(v_i)$.

Now, we move to the idea of equilibrium in this system. In games of complete information the central equilibrium concept is usually a Nash equilibrium, the set of strategies for all players in which each individual player cannot increase their utility by deviating from their strategy given the strategy of all the other players. This concept is now updated to give us a Bayes-Nash equilibrium where we must also

incorporate the distributions for which players are drawing from. A strategy profile constitutes a Bayes-Nash equilibrium if for every player i and every valuation v_i that the player might have, the player chooses a bid $s_i(v_i)$ that maximizes her conditional expected utility where the expectation is over the valuations of the other players, conditioned on a bidder i's valuation being v_i. We express that as an equation below.

Definition 2. A Bayesian strategy profile $(f_1, \ldots, f_I)$ is a Bayesian Nash Equilibrium if for all i, θ_i, and s_i :

$$\sum_{\theta_{-i} \in \Theta_{-i}} u_i(f_i(\theta_i), f_{-i}(\theta_{-i}), \theta_i, \theta_{-i}) p(\theta_{-i} \mid \theta_i) \geq \sum_{\theta_{-i} \in \Theta_{-i}} u_i(s_i, f_{-i}(\theta_{-i}), \theta_i, \theta_{-i}) p(\theta_{-i} \mid \theta_i)$$

Chapter 3

Literature Review

This "chapter" is my attempt to get words on a page in the process of writing the literature review. It is also an attempt to structure my thought process on what I have read and what questions I want to answer. In general, the proceeding sections would like to answer the following questions:

- Why Multi-Agent Simulations?

- How do the agents model people, what are the assumptions?

- How does learning affect these simulations?

- Does the non-stationarity of multi-agent systems make learning techniques invalid as models of human behavior?

Because this is a cursory document it will move back and forth between detailed notes on a single paper and a more structured literature review where I feel stronger on the material.

3.1   Why use Agent Based Modeling?

Tries to answer :

- What questions do Agent Based Models (ABM) try to answer?

- What are their strengths?

- What are their shortcomings?

- Examples and results?

3.1.1   The Logic of Agent Based Simulations

From "Agent-based modeling: Methods and techniques for simulating human systems" Bonabeau (2002):

Agent-based modeling (ABM) is a technique that models a system as a computer simulation of autonomous decision making entities that we call agents. The agents behavior and decision making process should in some way represent the behavior of the system they are supposed to model. Models are able to produce complex behavior from even relatively simple assumptions about agents that may be insightful for understanding the system they are trying to simulate. "Sophisticated ABM sometimes incorporates neural networks, evolutionary algorithms, or other learning techniques to allow realistic learning and adaptation." The main idea of ABM's is "describing a system from the prospective of its constituent units."

"The benefits of ABM over other modeling techniques can be captured in three statements: (i) ABM captures emergent phenomena; (ii) ABM provides a natural description of a system; and (iii) ABM is flexible." The main benefit from this list being that ABM captures emergent phenomena that result from the interaction of individual agents. Example given is Traffic Jam that is result of car agents, but the traffic jam moves in the opposite direction of the cars and the traffic jam is itself an emergent behavior from the Car agents. One reason to use and ABM model is if there may be emergent phenomena; " when (i) Individual behavior is nonlinear and can

be characterized by thresholds, if-then rules, or nonlinear coupling. (ii) Individual behavior exhibits memory, path dependence, and hysteresis, non-markovian behavior, or temporal correlations including learning and adaptation. (iii) Agent interactions are heterogeneous and can generate network effects. (iv) Averages will not work. Aggergate DEQ's tent to smooth out fluctuations, not ABM, which is important because under certain conditions, fluctuations can be amplified."

ABM is natural for explaining a system of behavioral entities such as traffic jams, stock markets, voters, or the inter-workings of an organization. Modeling Markets: Markets, and stock markets specifically have their resulting dynamics come from the interaction of many agents. This can be understood using ABM. NASDAQ used a an ABM to make decision about mechanism changes where there are many agents who learn via reinforcement learning and neural networks with the goal not being to emulate human behavior, but all possible behavior in regards to the change in mechanism. Found that decreasing tick-size actually increased ask-bid spread and made price discovery harder for agents. Another use is modeling auctions. EBAY lets people build agents to automate the bidding process. **important to thesis below:** "ultimately, transactions among economic software agents will constitute an essential and perhaps even dominant portion of the world economy." People at IBM have been exploring impact of shopbots on market dynamics with simulated shopbot economies of buyers and sellers. Economists have studied phenomena such as price dispersion by using ABM models where the goal is to design economic software agents. "In particular, they have been examining agent economies in which (i)search costs are nonlinear; (ii) some portion of the buyer population makes no use of search mechanisms; and (iii) shopbots are economically motivated strategically pricing their services so as to maximize their own profits." These stratagies of ABM techniques can also be applied to studying many agents playing an economic game. Author calls this "Game Theory without the theory." and says "Game theory is a great framework, but

game theorists suffer from self-imposed constrains: being able to prove their theorems puts severe limitations to what is possible" in particular he argues it is impossible to model realistic games within their framework.

The author describes the following issues with ABM. Like all models (in social sciences especially) the model must serve a purpose and cannot be a general-purpose model. **important:** "Another issue has to do with the very nature of the systems one is modeling with ABM in social sciences: they most often involve human agents, with potentially irrational behavior, subjective choices, and complex psychology–in other words, soft factors, difficult to quantify, calibrate, and sometimes justify." Should only use ABM to justify qualitative results (not quantitative) due to the varying degrees of accuracy input into the simulation. Bonabeau (2002)

How this applies to Thesis: This goes in depth on the "why" of Agent based modeling, specifically justifying it for bottom up modeling in the cases where that may be the best way to model something in the social sciences. The really interesting parts to me was about markets which highlights more or less exactly what I want to do, looking at markets composed of heterogeneous agents that learn via methods in AI. Some of these models are not even trying to emulate human behavior but just study the dynamics of competitive computer agents in a market setting (or how to build the best agents). The real issue I seem to be focusing in on is, what results can we get from such AI agent models when we know that different learning algorithms behave differently within the context of a non-stationary learning environment. As noted in my notes from a different section, when multiple agents are interacting and learning together they inherently create a non-stationary environment (constantly changing). This constant changing means that only under very specific circumstances and with certain algorithms can it be proved the agents will converge to an (optimal) equilibrium. Can a neural net be said to approximate a humans learning? Probably not,

but many simulations use it anyway. Does changing the composition of agents of each learning algorithm change the resulting dynamic in a meaningful way? If so, like I suspect, what can one even learn from such a simulation that generalizes at all to markets of agents artificial or otherwise?

Artificial Adaptive Agents in Economic Theory, by John H. Holland and John H. Miller.

Recent (as of 1991) advances in machine learning offer the possibility of creating artificial adaptive agents (AAA) that can be tested in a wide variety of environments. Creates a complex adaptive system. Subjects that economists study can be classified as complex adaptive systems with networks of interacting agents and dynamic aggregate behavior. Agents in this system are adaptive if they are given some performance measure such as a utility function that they try to maximize over time. AAA models don't require that agents perform technically demanding computations and derivations to optimize their behavior such as classical models imply.

II Current techniques. AAA can be created using algorithms such as classifier systems, genetic algorithms, neural networks, and reinforcement learning. Wondering how much to outcomes of a system are tied to the learning algorithms used confronts the rationality postulates. "Usually there is only one way to be fully rational, but there are many ways to be less rational." When building theory off of AAA, want robust behavior across algorithm choice. Holland & Miller (1991)

Agent-Based Computational Economics: Growing Economies from the Bottom Up, by Leigh Tesfastsion 2002 Market economies are complex adaptive systems consisting of a large number of adaptive agents. Local interactions between these agents give rise to macro behavior of the markets which then feedback into the nature

of the local interactions. Agent based models give a way to deal with the dual feedback mechanisms between micro and macro economics. Economists have used ABMs to model complex emergent phenomenons such as inductive learning, imperfect competition, endogenous trade network formation, and co-evolution of individual behaviors and economic institutions. Author calls this agent-based computational economics (ACE) (everyone seems to have a different work/ acronym for these). Can think of these models as laboratories to study economies under controlled experimentation. One aspect they study is emergent behaviors and structures in the economy such as they have formed in real life, without top down planning. Another aspect studied with these is mechanism design, what are the effects of certain rules or regulations on the emergent behavior (think of the NASDAQ example in Bonabeau). While we think of these models being purely representative, the advent of the internet and shopbots is blurring these lines as 'real' economies are developing around artificial agents. ACE researches have used many algorithms to represent the learning process of computational agents. Earliest example was genetic algorithms, but now there is also Q-learning, classifier systems. These learning algorithms were initially developed by computer scientists for optimal performance. This is appropriate for automated economic processes, but might not be for other social simulations. It might even be appropriate for the agents to jointly learn (shared learning / strategy formulation) to improve market conditions. To model humans and real-world economies, the learning algorithm will actually need to incorporate salient characteristics of actual human making. Might have neighborhoods of agents independently co-evolve their strategies. Aware of this, researchers are doing more systematic evaluations of each learning algorithm, and for example Dawid (citation needed) showed that in dynamic multi-agent economies learning via genetic algorithms, the precise parameter configuration can have a large impact on the long run behavior of the economy. The learning study by Rust et al. showed that of thirty computational trading algorithms submitted to

a double-auction tournament at the Santa Fe Institute, the winner was a very simple sniping algorithm that would bid at the last minute beating out algorithms that were at the cutting edge of AI. Gode and Sunder (citation needed) did research into continuous double-actions with computational agents found that the efficacy of this market type were derived largely from its structure and not from the learning of the agents (Good Mechanism Design!). Market efficiency levels close to 100 percent were obtained with agents who submitted bids and asks at random. An open question, is how to model the minds of computational agents. Should they only be viewed as logic machines or as controllers for embodied activity? For automated markets, there is no reason that the agents created have to mimic humans, but for modeling real markets, it is crucial to model humans for it to have predictive power. Observed evidence suggests no algorithm performs best in all circumstances, and no algorithm best replicates human behavior. Tesfatsion (2002)

Agent-Based Computational Economics: A Constructive Approach To Economic Theory by Leigh Tesfatsion. Published 2005.

This paper goes through an extended example of an agent based simulation of a Walrasian Auctioneer market to show the benefits and drawbacks of the agent based approach. Not finished reading yet, quite long. Tesfatsion (2005)

The Complexity of Exchange by Robert Axtell Computational complexity of two classes of market mechanisms are compared: the Walrasian market where prices are determined by a central auctioneer and decentralized markets with concurrent exchange between coalitions of agents. An important aspect of agent based models of markets is the price formation process. Through interaction, the prices *emerge* whether through a market maker, or not. Talks about how agent based models don't really try to model equilibrium in the economy, but complexity. Have ecologies of

strategies among agents that when tipped in one direction rapidly evolve to a new quasi-equilibrium.

The complexity of Walrasian Exchange is in NP (Nondeterministic Polynomial time), and is computationally infeasible to be a mechanism for an actual market. (This result is cited)

Complexity of k-lateral exchange equillibria is in P (polynomial time) This result is proved and is the bulk of the paper. Shows that distributed interaction models are easier to compute and represent a better way of being a mechanism for determining prices. Axtell (2005)

3.1.2   Game Theory and Computation

Computer Science and Game Theory by Yoav Shoham (same author as Multi Agent Systems book) Game theory, while initially developed within the discipline of economics has within the beginning part of the 20th century become more and more relevant to computer science. It is integral to artificial intelligence, e-commerce, networking, and other areas of computer science (including cryptography, where Adam Groce uses it). The internet is the main reason that it has become so central as it deals with multiple agents interacting, each with their own information and interests. It is interesting to note that both game theory and computer science grew up at the same time and in the same place in particular with John Von Neumann at Princeton in the 1950s. The area started with exploring the algorithmic complexity of playing and solving games, but also dealt with multi-person operations research and bounded rationality. In modern research, the area of multi-agent learning has received much attention and is called interactive learning in the game theory literature. This is important particularly in the field of AI which is trying to move from learning in a single agent context to learning in a multi-agent environment. When multiple agents learn

concurrently, one cannot distinguish between leaning and teaching. In this environment, the 'optimal learning' is no longer well defined. Most game theorists assume agents with an infinite ability to reason. 'Bounded Rationality' however, is an area of research that involves games played by automata. In this context, the author questions if the idea of equilibrium as central in game theory needs to be reconsidered in light of competitions involving agents in markets that performed better when not considering this concept. Most of the effort that goes into making a good, competitive AI goes into thinking about making a good heuristic function, and searching and planning; not reasoning about the opponents behavior. Shoham (2008)

3.1.3 Example: Schelling's Segregation Model

Schelling's Segregation model is an example of a ABM designed in 1969 by Thomas C. Shelling to try and explain racial segregation. Example of how agent preferences can lead to interesting aggregate structure. This model is presented in the lecture notes of Thomas Sargent and I recreated the model in python. The model is as follows:

There are two kinds of agents, orange and green of which there are 250 of each type living in on a unit square. The agents each have a location, a point on the square, (x, y), where $0 < x, y, < 1$. The agents are *happy* if half of its neighbors, the ten closest agents in euclidean distance, are of the same type. If the agent is not *happy* then the agent is said to be *unhappy*. Initially, the agents are randomly assigned a point on the unit square i.e. each is initially assigned a point drawn from a bi-variate uniform distribution $S = (0, 1)^2$. Cycling though each agent, each agent is given a choice to stay or to move. If the agent is happy, then they stay otherwise they move according to the following procedure:

1. Draw a random location in S
2. If happy at new location, move there

3. Else, go to step 1

Continue this cycle, moving through the agents until everyone is happy and does not wish to move.

The results are shown in the following figures:
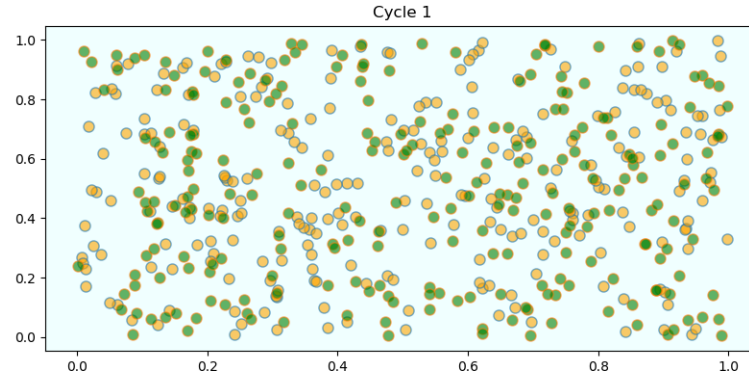
Figure 3.1: Schelling's Segregation Model: Cycle 1
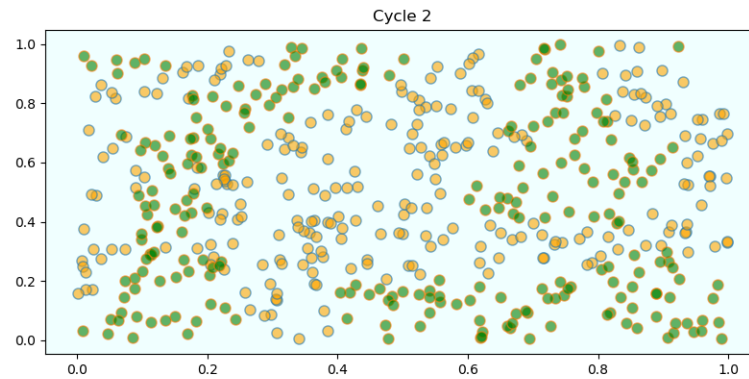
Figure 3.2: Schelling's Segregation Model: Cycle 2

As can be seen, in cycle 1 (Figure 3.1), the agents are well distributed among each other, but as they move in cycles 2-5, they become progressively more segregated. After 5 cycles all agents are happy and the simulation terminates. This example highlights the benefits of using an ABM. With very few assumptions about the agents preferences (wanting half or more of their neighbors to be the same type as them), we

Figure 3.3: Schelling's Segregation Model: Cycle 3

Figure 3.4: Schelling's Segregation Model: Cycle 4

Figure 3.5: Schelling's Segregation Model: Cycle 5

can see the resulting emergent behavior of segregated neighborhoods in the system as a whole.

Chapter 4

Auctions and Markets

This chapter is about auctions, and artificial simulations in auctions. It will specifically cover what I am reading with reference to the dou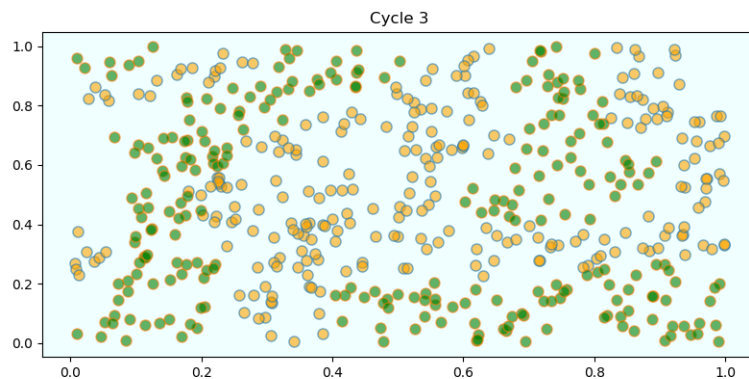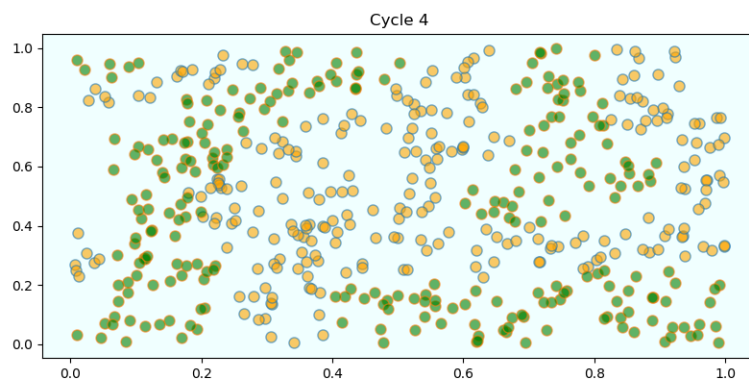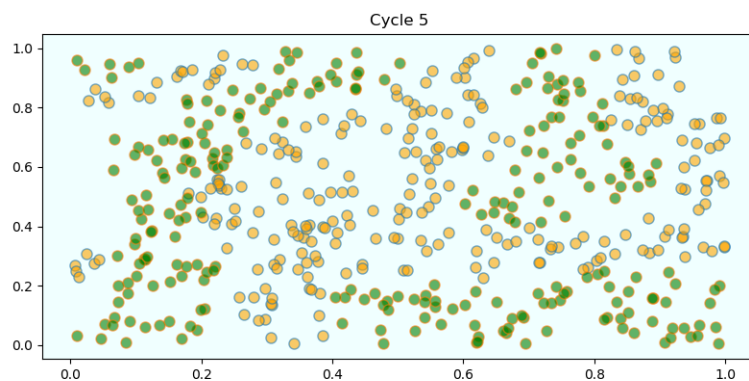ble auction simulation I hope to make. This is separated from the literature review for organizational purposes as I collect my notes, but will be moved there in the future.

4.0.1 Basic Notation and Ideas

From "Understanding Auctions" by Asuncion Mochon and Yago Saez. (TODO: Add citation manually)

An **auction** is a market mechanism, operating under specific rules, that determines to whom one or more items will be awarded and at what price. They are important to environments in which it is difficult to set a market price. For a bidder in an auction, the **value**, v_i is the maximum price that bidder i is willing to pay for an item. The **bid**, b_i is the offer that bidder i submits for an item. **Sincere bidding** is when $v_i = b_i$, **underbidding** is when $v_i > b_i$, and **overbidding** is when $v_i < b_i$. The highest bid made by any bidder is denoted b^*, and is the winning bid (if multiple bids equal the winning bid, then some tie-breaking rule must be used). The **selling price**, p^* is the final price that the bidder actually pays for the item (which depending on

the auction type need not equal b^*). Under the **first-price** rule, the bid submitted by the winner is equal to the selling price. Before the auction begins, each bidder know their personal or private value for the item. The term **winner's curse** refers to the idea that the bidder who wins may have paid to much for the item in the sense that no one else was willing to pay near that much for the item. That is, in a first price auction the winner paid extra the difference between their bid and the second highest bid. In some experimental work by Thaler, people tend to respond to the risk of overpaying by decreasing bids with respect to their values.

An auction consists of a set of bidders, $I = (1, 2, ..., N)$ and a seller. After the auction, the bidder i wins the item if their bid is higher than the bid placed by any other bidder k $b_i > max_{k \neq b_k}$. In a single unit auction, the **income of the bidder** i is equal to their value of the item:

$$\Gamma_i^* = v_i$$

and the **surplus of the bidder** i is equal to the difference between income and cost

$$\Pi_i^* = \Gamma_i^* - p^*$$

. If a bid placed by a bidder is less than the winning bid, they do not win anything and their income and surplus are both zero. The **sellers revenue** in a single unit auction is equal to the price paid by the winning bidder

$$R^* = p^*.$$

Pricing rules in an auction. **First-Price**: the winning bidder pays the amount of their bid, which is the highest bid of the auction: $p^* = b^*$. Also called **pay-what-you-bid** (PWYB). **Second-price**: the winning bidder pays an amount that is equal

to the second highest bid for the awarded item. First price auctions will make people tend to bid lower than their private valuation since if they bid that value their profit is zero. Under this rule the item could be awarded to someone who value the item less. Second price auctions encourages bidder to bid their true values (as they will gain positive profits if they win no matter the second highest bid). This encourages an efficient allocation of items.

Optimal Auctions are those that maximize the expected revenue for the seller. **Efficient Auctions** are those that the items are awarded to the bidders who most value them. These goals are often conflicting.

Double Auctions The main characteristic of a double auction is that multiple buyers and sellers interact, making it a two sided auction. In a double auction, buyers submit bids for the amount they are willing to pay, and sellers make offers, asks, for the price they are willing to sell. A synchronized continuous double auction is composed of r rounds, which are divided into s periods. In each period, there are a fixed number of t steps which alternate steps in which buy and sell orders are placed. Each step t begins with a bid/ask step in which, simultaneously, the buyers indicate the prices that they are willing to pay per unit, and the sellers indicate the prices at which they are willing to sell.

Notes for "The Double Auction Market: Institutions, Theories, and Evidence. Edited by Daniel Friedman and John Rut"

Ch. 1 "The Double Auction Market Institution: A survey" by Daniel Friedman

The continuous double auction allows traders to make offers to buy or sell and accept other traders' offers at any moment during a trading period. For past 100 years NY, London, and Chicago exchanges all operate under some form of double auction

rules. Laboratory experiments show that double auctions continuously produce very efficient prices and allocations, more so than theory would suggest.

Notes from "Bidding Strategies in Agent-Based Continuous Double Auctions" by Huiye Ma and Ho-fung Leung

In continuous double auctions (CDA), buyers submit increasingly higher bids and sellers submit increasingly lower asks at any moment during a trading period and transactions occur when the highest bid is at least as high as the lowest ask. Auctions in general consist of two components: protocols and strategies. The former is the rules each agent must abide by in the auction and the latter is the way that the agent tries to achieve their means within that protocol.

Basic CDA Mechanisms

Is a marketplace where there are agents selling goods (sellers) and agents buying goods (buyers). The sellers and buyers trade homogeneous goods. The current lowest ask is called the outstanding ask, oa. The current highest bid is called the outstanding bid, ob. A valid bid is a bid higher than the current ob and similarly a valid ask is any ask lower than current oa. Any invalid bid/ask is ignored by the market. Each buyer and seller has an acceptable price range $[]P_{ll}, P_{ul}$ for a CDA. P_{ll} is the lowest acceptable price and P_{ul} is the highest acceptable price. This range is formed on experience. Each buyer and seller has a reservation price that can be thought of as their valuation of the good. Buying a good for more then that or selling for less then that would cause the buyer/seller to lose profit. When the ob is higher than or equal to the oa, the agents who submitted those bids and asks will make a transaction. When there is a transaction or a specified time has passed, the round is terminated. A new round can then begin. Only one unit of good is transacted per round. The demand is the total number of goods that buyers want to buy and the supply is the

total number of goods that sellers want to sell.

At this point the author lists an algorithm (in pseudo code) for running the market. The author then goes on to give basic trading strategies and their pseudo code for zero intelligence traders and the like.

references to check:

A Bannall: Autonomous adaptive agents for single seller sealed bid auctions.

R. Das: Agent-human interactions in the continuous double auction.

S. Gjerstad: Price formation in double auctions

D.K. Gode: Allocative Efficiency of markets with zero-intelligence traders.

M. He A fuzzy-logic based bidding strategy for autonomous agents in continuous double auctions.

C. Preist: Economic Dynamics of agents in multiple auctions.

C. Preist: Adaptive agents in a persistent shout double auction.

V. Smith: an experimental study of competitive market behavior.

M.P. Wellman: Empirical game-theoretic analysis of the tac market games.

M.P. Wellman: Designing the market game for a trading agent competition.

Simple Agents, Intelligent Markets by Karim Jamal

Economics usually focuses in on equilibrium achieved by rational agents who optimize their utility functions. Exploration to th extent at which equilibrium arises from the characteristics of the environment have been limited. Normally, representative agents in models have sophisticated computational abilities who solve for equilibrium. Plott and Sunder have shown that markets with uncertainty and asymmetrically distributed information disseminate information and converge near rational expectation equilibrium when populated with profit-motivated human traders. This paper shows that the same result can be achieved with minimally intelligent traders in a computer

simulation. Jamal et al. (2017)

Chapter 5

Agent Systems

This chapter lays out the background material of multi-agent systems–what they are, how they work, and why economists might be interested in them–that drives the rest of the thesis.

5.1   Agents

An **agent** is something that acts. This comes from the Latin word agere, to do. In general, anything might be considered an agent so long as it can perceive the environment and act upon it. A **rational agents** is something that acts to achieve the best outcome or with uncertainty, the best expected outcome in a given situation. For example we are agents, we have the ability to take an action and effect the world. We even might be considered rational since we are generally trying to navigate the world around us so as to try and lead to our preferred outcome. People are not the only agents, animals and machines are also agents capable of acting on the world around them.

5.1.1   In Economics

In economics many of the models are constructed with utility maximizing agents in line with Decision Theory and Rational Choice Theory. Because economics seeks to understand the outcomes of peoples interactions, and perhaps how these outcomes could be improved, if people can be modeled as utility maximizers, it is then possible to analytically solve for this solution and see what the results are.

The primary purpose of constructing agents in economics is to see how their interactions play out in an economic setting that can not be solved analytically.

5.1.2   In Artificial Intelligence

In artificial intelligence, the primary purpose of understanding agents is to try and create the most effective artificial agents possible. This discipline cares deeply about how good the agents they are able to make are able to solve problems, either general or specific. To do this, they use much of the theory of agents with a deep emphasis on how a constructed agent can be taught or learn from its environment to achieve its goals.

5.2   Formal Model of Agents

This section lays out a more formal definition of agents, specifically geared towards constructed or artificial agents such as we will be constructing. An **agent** is anything that can be viewed as perceiving its environment through sensors and acting upon its environment through actuators. A **precept** is a perceptual input to an agent, and a **precept sequence** is the complete history of everything the agent has ever perceived. The agent then has an **agent function** that maps these precepts to an action. In the case of an artificial agent this is done with an **agent program** that is the physical implementation of the agent function. In the case of an agent simulated

on a computer, this will be the computer code that decides what action to take. The agents actions will then cause the environment to go through a series of states and the agent will measure the outcome with some **performance measure**.

For our purposes we focus on what are called **utility based agents**, where the agent has a utility function as an internalized performance measure and is able to numerically assign values of "goodness" to the outcome from the agents perspective. A rational agent will always try to maximize average expected utility over its lifetime as explained by decision theory.

5.2.1   Task Environment

The environment that an agent exists in and their relationship to it is critical for understanding the system as a whole. For example the environment can fully observable with the agent's sensors able to see the entire environment, or only partially observable to the agent. There can also be multiple agents within the system, each trying to maximize their own performance measure. This is called a **multi-agent system** and it will be a key aspect of all of the simulations we construct. It is important to note that the agents in the multi-agent system share no internal state and must communicate (if they so wish to do so) through the environment. The environment can also be deterministic or stochastic, either depending entirely upon its current state and the agents actions or contains randomness.

5.2.2   Agents Who Learn

Learning is a critical part of being a rational agent. We expect that as agents interact with their environment, they should be able to adapt and change in order to continue to maximize their utility. The general model of an artificial agent who learns is as follows.

First, the agent reprieves its environment and sends that information to the per-

formance function that determines what it will do. There is now a critic that gives feedback for how the agent is doing based on aprori information and the observed state that sends feedback to a learning element that can change the behavior of the performance element. Based on what the learning element is trying improve, there is also a problem generator that suggests actions to the performance element so that different actions can be tested and evaluated. Russel & Norvig (2010)

Chapter 6

Reinforcement Learning

Reinforcement learning has become a popular way of teaching agents how to play games and interact with each other in Machine Learning (ML). Because the dynamics and behavior of a final simulation are determined by the learning algorithm used for its composing agents, we spend some time developing the ideas of reinforcement learning in both a single agent and multi-agent setting.

6.1 Single Agent Learning

Reinforcement learning is the area of machine learning where agents learn through repeated interactions with their dynamic environment. This section gives a brief overview of learning algorithms used by single agents with a focus on deep reinforcement learning. In general, reiforcment learning formalizes the interaction of agents and environment using a Markov decision process (MDP).

Definition: Markov Decision Process

A Markov decision process is 5-tuple $\langle \mathcal{S}, \mathcal{A}, R, T, \gamma \rangle$ where,

1. $\mathcal{S}$ represents a finite set of states

2. $\mathcal{A}$ represents a finite set of actions

3. T is a transition function $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0,1]$ that determines the probability of transitioning from any state $s \in \mathcal{S}$ to any state $s' \in \mathcal{S}$ given any possible action $a \in A$

4. R is the reward function $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$

5. $\gamma \in [0,1]$ is the discount factor that balances immediate and future rewards

6.1.1   Q-Learning

Q-Learning is an an algorithm that has been developed for single-agent, fully observable environments with discrete actions. Q learning involves creating a table of expected payoffs for each action a available at a state s denoted $\hat{Q}(s,a)$. Each time the agent transitions from a state s to a state s' taking an action a and recieving payoff r, the Q table is updated according to the following assignment:

$$\hat{Q}(s,a) \leftarrow \hat{Q}(s,a) + \alpha[(r + \gamma max_{a^1}\hat{Q}(s',a')) - \hat{Q}(s,a)]$$

where we pick $\alpha \in [0,1]$ which we call the learning rate.

6.1.2   Deep Reinforcement Learning

Using deep learning in the form of Neural Networks can be helpful for exploring large policy spaces efficiently. It also eliminates the need to hand design features to be optimized for the reinforcement learning process. Shoham & Leyton-Brown (2008)

Chapter 7

Tables and Graphics

7.1 Tables

The following section contains examples of tables, most of which have been commented out for brevity. (They will show up in the .tex document in red, but not at all in the .pdf). For more help in constructing a table (or anything else in this document), please see the LaTeX pages on the CUS site.

Table 7.1: Correlation of Inheritance Factors between Parents and Child

Factors	Correlation between Parents & Child	Inherited
Education	-0.49	Yes
Socio-Economic Status	0.28	Slight
Income	0.08	No
Family Size	0.19	Slight
Occupational Prestige	0.21	Slight

If you want to make a table that is longer than a page, you will want to use the longtable environment. Uncomment the table below to see an example, or see our online documentation.

Table 7.2: Chromium Hexacarbonyl Data Collected in 1998–1999

Chromium Hexacarbonyl			
State	Laser wavelength	Buffer gas	Ratio of $\frac{\text{Intensity at vapor pressure}}{\text{Intensity at 240 Torr}}$
$z^7P_4^\circ$	266 nm	Argon	1.5
$z^7P_2^\circ$	355 nm	Argon	0.57
$y^7P_3^\circ$	266 nm	Argon	1
$y^7P_3^\circ$	355 nm	Argon	0.14
$y^7P_2^\circ$	355 nm	Argon	0.14
$z^5P_3^\circ$	266 nm	Argon	1.2
$z^5P_3^\circ$	355 nm	Argon	0.04
$z^5P_3^\circ$	355 nm	Helium	0.02
$z^5P_2^\circ$	355 nm	Argon	0.07
$z^5P_1^\circ$	355 nm	Argon	0.05
$y^5P_3^\circ$	355 nm	Argon	0.05, 0.4
$y^5P_3^\circ$	355 nm	Helium	0.25
$z^5F_4^\circ$	266 nm	Argon	1.4
$z^5F_4^\circ$	355 nm	Argon	0.29
$z^5F_4^\circ$	355 nm	Helium	1.02
$z^5D_4^\circ$	355 nm	Argon	0.3
$z^5D_4^\circ$	355 nm	Helium	0.65
$y^5H_7^\circ$	266 nm	Argon	0.17
$y^5H_7^\circ$	355 nm	Argon	0.13

State	Laser wavelength	Buffer gas	Ratio of $\frac{\text{Intensity at vapor pressure}}{\text{Intensity at 240 Torr}}$
$y^5H_7^\circ$	355 nm	Helium	0.11
a^5D_3	266 nm	Argon	0.71
a^5D_2	266 nm	Argon	0.77
a^5D_2	355 nm	Argon	0.63
a^3D_3	355 nm	Argon	0.05
a^5S_2	266 nm	Argon	2
a^5S_2	355 nm	Argon	1.5
a^5G_6	355 nm	Argon	0.91
a^3G_4	355 nm	Argon	0.08
e^7D_5	355 nm	Helium	3.5
e^7D_3	355 nm	Helium	3
f^7D_5	355 nm	Helium	0.25
f^7D_5	355 nm	Argon	0.25
f^7D_4	355 nm	Argon	0.2
f^7D_4	355 nm	Helium	0.3
Propyl-ACT			
$z^7P_4^\circ$	355 nm	Argon	1.5
$z^7P_3^\circ$	355 nm	Argon	1.5
$z^7P_2^\circ$	355 nm	Argon	1.25
$z^7F_5^\circ$	355 nm	Argon	2.85
$y^7P_4^\circ$	355 nm	Argon	0.07
$y^7P_3^\circ$	355 nm	Argon	0.06
$z^5P_3^\circ$	355 nm	Argon	0.12
$z^5P_2^\circ$	355 nm	Argon	0.13
$z^5P_1^\circ$	355 nm	Argon	0.14

State	Laser wavelength	Buffer gas	Ratio of $\frac{\text{Intensity at vapor pressure}}{\text{Intensity at 240 Torr}}$
Methyl-ACT			
$z^7P_4^\circ$	355 nm	Argon	1.6, 2.5
$z^7P_4^\circ$	355 nm	Helium	3
$z^7P_4^\circ$	266 nm	Argon	1.33
$z^7P_3^\circ$	355 nm	Argon	1.5
$z^7P_2^\circ$	355 nm	Argon	1.25, 1.3
$z^7F_5^\circ$	355 nm	Argon	3
$y^7P_4^\circ$	355 nm	Argon	0.07, 0.08
$y^7P_4^\circ$	355 nm	Helium	0.2
$y^7P_3^\circ$	266 nm	Argon	1.22
$y^7P_3^\circ$	355 nm	Argon	0.08
$y^7P_2^\circ$	355 nm	Argon	0.1
$z^5P_3^\circ$	266 nm	Argon	0.67
$z^5P_3^\circ$	355 nm	Argon	0.08, 0.17
$z^5P_3^\circ$	355 nm	Helium	0.12
$z^5P_2^\circ$	355 nm	Argon	0.13
$z^5P_1^\circ$	355 nm	Argon	0.09
$y^5H_7^\circ$	355 nm	Argon	0.06, 0.05
a^5D_3	266 nm	Argon	2.5
a^5D_2	266 nm	Argon	1.9
a^5D_2	355 nm	Argon	1.17
a^5S_2	266 nm	Argon	2.3
a^5S_2	355 nm	Argon	1.11
a^5G_6	355 nm	Argon	1.6
e^7D_5	355 nm	Argon	1

7.2 Figures

If your thesis has a lot of figures, LaTeX might behave better for you than that other word processor. One thing that may be annoying is the way it handles "floats" like tables and figures. LaTeX will try to find the best place to put your object based on the text around it and until you're really, truly done writing you should just leave it where it lies. There are some optional arguments to the figure and table environments to specify where you want it to appear; see the comments in the first figure.

If you need a graphic or tabular material to be part of the text, you can just put it inline. If you need it to appear in the list of figures or tables, it should be placed in the floating environment.

To get a figure from StatView, JMP, SPSS or other statistics program into a figure, you can print to pdf or save the image as a jpg or png. Precisely how you will do this depends on the program: you may need to copy-paste figures into Photoshop or other graphic program, then save in the appropriate format.

Below we have put a few examples of figures. For more help using graphics and the float environment, see our online documentation.
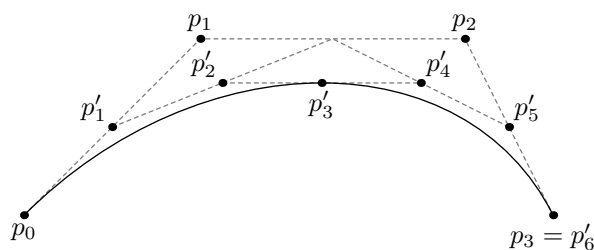
And this is how you add a figure with a graphic:

Figure 7.1: A Figure

7.3   More Figure Stuff

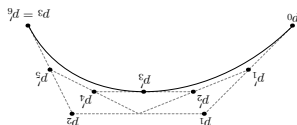You can also scale and rotate figures.

Figure 7.2: A Smaller Figure, Flipped Upside Down

7.4   Even More Figure Stuff

With some clever work you can crop a figure, which is handy if (for instance) your EPS or PDF is a little graphic on a whole sheet of paper. The viewport arguments are the lower-left and upper-right coordinates for the area you want to crop.
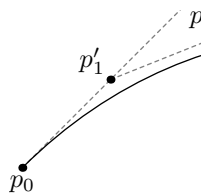
Figure 7.3: A Cropped Figure

7.4.1   Common Modifications

The following figure features the more popular changes thesis students want to their figures. This information is also on the web at `web.reed.edu/cis/help/latex/graphics.html`.
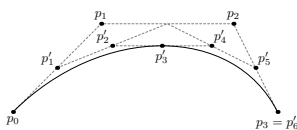
Figure 7.4: Subdivision of arc segments. You can see that $p_3 = p_6'$.

Conclusion

Here's a conclusion, demonstrating the use of all that manual incrementing and table of contents adding that has to happen if you use the starred form of the chapter command. The deal is, the chapter command in LaTeX does a lot of things: it increments the chapter counter, it resets the section counter to zero, it puts the name of the chapter into the table of contents and the running headers, and probably some other stuff.

So, if you remove all that stuff because you don't like it to say "Chapter 4: Conclusion", then you have to manually add all the things LaTeX would normally do for you. Maybe someday we'll write a new chapter macro that doesn't add "Chapter X" to the beginning of every chapter title.

4.1 More info

And here's some other random info: the first paragraph after a chapter title or section head *shouldn't be* indented, because indents are to tell the reader that you're starting a new paragraph. Since that's obvious after a chapter or section title, proper typesetting doesn't add an indent there.

Appendix A

The First Appendix

Appendix B

The Second Appendix, for Fun

Bibliography

Arthur, W. B. (1992). On Learning and Adaptation in the Economy. *Queen's University, Department of Economics, Working Papers*, (854). `http://ideas.repec.org/p/qed/wpaper/854.html{\%}5Cnpapers2://publication/uuid/821D9AA7-EB02-45D1-8CCC-AA817E38A09A`

Auchincloss, A. H., & Garcia, L. M. T. (2015). Brief introductory guide to agent-based modeling and an illustration from urban health research. *Cadernos de Saúde Pública*.

Axtell, R. (2005). The complexity of exchange. *Economic Journal*, *115*(504), 193–210.

Backhouse, R., & Cherrier, B. (2016). 'It's Computerization, Stupid!' The Spread of Computers and the Changing Roles of Theoretical and Applied Economics. *SSRN Electronic Journal*, (pp. 1–36).

Blume, L., Easley, D., Kleinberg, J., Kleinberg, R., & Tardos, É. (2015). Introduction to computer science and economic theory. *Journal of Economic Theory*, *156*(November), 1–13. `http://dx.doi.org/10.1016/j.jet.2014.11.002`

Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences of the United States of America*, *99*(SUPPL. 3), 7280–7287.

Boutilier, C., Shoham, Y., & Wellman, M. P. (1997). Economic principles of multi-agent systems. *Artificial Intelligence*, *94*(1-2), 1–6.

Harsanyi, J. C. (1967). Games with Incomplete Information Played by "Bayesian" Players, I–III Part I. The Basic Model. *Management Science*, *14*(3), 159–182.

Holland, J., & Miller, J. (1991). Artificial Adaptive Agents in Economic Theory. *American Economic Review*.

Jamal, K., Maier, M., & Sunder, S. (2017). Simple Agents, Intelligent Markets. *Computational Economics*, *49*(4), 653–675.

LeBaron, B. (2002). Building the Santa Fe artificial stock market. *Physica A*, (June).

MacAl, C. M., & North, M. J. (2010). Tutorial on agent-based modelling and simulation. *Journal of Simulation*, *4*(3), 151–162.

Mochon, A., & Saez, Y. (2015). *Understanding Auctions*. Springer International Publishing Switzerland.

Papadimitriou, C. (2015). On Algorithmic Game Theory I. `https://www.youtube.com/watch?v=ggi2PZbO0oQ{\&}t=814s`

Rejeb, L., & Guessoum, Z. (2005). Firms Adaptation in Dynamic Economic Systems. In *Artificial Economics*, (pp. 53–64).

Roughgarden, T. (????). Beyond Worst-Case Analysis I. `https://www.youtube.com/watch?v=thHt1lhLqJA`

Roughgarden, T. (2016). *Twenty lectures on algorithmic game theory*. Cambridge University Press.

Roughgarden, T., Syrgkanis, V., & Tardos, É. (2017). The price of anarchy in auctions. *Journal of Artificial Intelligence Research*, *59*, 59–101.

Russel, S. J., & Norvig, P. (2010). *Artificial Intelligence: A Modern Approach*. Upper Saddle River, New Jearsy: Prentice Hall, 3 ed.

Sargent, T. J., & Stachurski, J. (2019). Lectures in Quantitative Economics.

Shoham, Y. (2008). Computer science and game theory. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*.

Shoham, Y., & Leyton-Brown, K. (2008). Multiagent systems: Algorithmic, Game-Theoretic, and logical foundations. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*, *9780521899*, 1–483.

Tesfatsion, L. (2002). Agent-based computational economics: growing economies from the bottom up.

Tesfatsion, L. (2005). AGENT-BASED COMPUTATIONAL ECONOMICS : A CONSTRUCTIVE APPROACH TO ECONOMIC THEORY 18 December 2005. *Computational Economics*, *2*(October 2003), 831–880. `http://linkinghub.elsevier.com/retrieve/pii/S1574002105020162`

Vickrey, W. (1961). American Finance Association Counterspeculation , Auctions , and Competitive Sealed Tenders Author (s): William Vickrey Source : The Journal of Finance , Vol . 16 , No . 1 (Mar ., 1961), pp . 8-37 Published by : Wiley for the American Finance Associat. *16*(1), 8–37.