Algorithmic Game Theory

Meher Mankikar

6 May 2021

1 Introduction

In this chapter, we will be talking about Algorithmic Game Theory which is formed at the intersection of computer science and traditional game theory. This area of study attempts to understand and design algorithms in strategic environments. While Game Theory is the foundations of mathematic economics, there are also many applications in computer science as well. There are Games in AI where we model "rational agents" and their interactions. There are also Games in Algorithms. Several Game Theory problems have interesting algorithmic status that can be looked at (some are in NP but not known to be NP-complete). Most recently, there has been a lot of research in Games, the Internet, and E-commerce. This research considers the internet to be a huge experiment in the interactions between agents (both human and automated) and asks: How do we set up the rules of this game to harness "socially optimal: results? These are just a few, but there are many more applications of Game Theory in computer science, economics, and beyond.

2 Games and Equilibria

Definition 1: **Game Theory** is the study of mathematical models of strategic interaction among rational decision makers.

Definition 2: A **Normal Form Game** is defined as follows

- Set of players $N = \{1, \cdots, n\}$.
- Strategy set S
- For each $i \in N$, a utility function $u_i : S^n \to \mathbb{R}$. If each $j \in N$ players strategy $s_j \in S$, the utility of player i is $u_i(s_1, \cdots, s_n)$.

So in a normal form game, we have a set of players that goes from player 1 to player n. We then assume WLOG that we have the same strategy set of all players. Note that this is not always the case in games, but it is pretty straightforward to convert between the scenarios. We then have a utility function for each player. The utility function takes in a vector of strategies with one for each player and outputs the utility of player i if every other player plays their corresponding strategy.

Definition 3: **Nash Equilibrium** is a vector of strategies $S = (s_1, \cdots, s_n) \in S^n$ such that for all $i \in N$ and $s_i^t \in S$, $u_i(S) \geq u_i(s_1, \cdots, s_{i-1}, s_i', s_{i+1}, \cdots, s_n)$.

Phew, that was a lot. More intuitively, we can think about the Nash Equilibrium as a state in which each player's strategy is the best response to the strategies of the other players. (Think: if a player knew the moves of all other players before hand, what move would he make?) In a Nash Equilibrium, there is no incentive for either player to deviate from the current situation. In other words, making any linear change to their move would not increase their payoff given that the other player's move stays the same. We will see that the Nash Equilibrium is not always the state that is the best for both players (provides the highest payoff).

Example: We can see an example by looking at the game Rock-Paper-Scissors.

	Rock	Paper	Scissors
Rock	0,0	-1,1	1,-1
Paper	1,-1	0,0	-1,1
Scissors	-1,1	1,-1	0,0

Using this table, one player's choice is represented by a row and the other by a column. The numbers in each cell are the payoff for the row and column player for each possible outcome of the game.

One important idea that can be seen through this game is the idea of *randomized* strategies. If both players randomize their choices uniformly, then there is no incentive for either player to change their move. In other words, a pair of probability distributions like this is a Nash Equilibrium.

Theorem 1: The **Nash Theorem** states that every bimatrix game has a Nash Equilibrium.

Proof This proof is super long and confusing and not really important for the rest of this chapter so we will skip it. You are welcome.

However, it is important to note that there is no known polynomial-time algorithm for computing the Nash equilibrium of a given game. The problem of finding a Nash equilibrium is "PPAD-hard" (between P and NP).

2.1 Exercise: Nash Equilibrium

Exercise 1 One example of a common game that is studied in Game Theory is the Prisoner's Dilemma. In this game, there are two prisoners. Each has the option to cooperate with their partner criminal or defect (hand in their partner). The game can be represented in a table like the following:

	Cooperate	Defect
Cooperate	-1,-1	-9,0
Defect	0,-9	-6,-6

What is the dominant strategy? How many Nash Equilibria are there? What are they?

Answer
The dominant strategy for both players is to defect. Note that it is always better to defect than to cooperate no matter what the other player does. To see this, look at the table. If the column player is cooperating, if we defect we get a payout of 0 and if we cooperate we get a payout of -1. If the column player is defecting, if we defect we get a payout of -6 and if we cooperate we get a payout of -9. So in both cases, the payout is higher if we defect. Note that this will result in both players defecting, although this is not the best strategy for the players. There is 1 Nash Equilibria: both players defect.

2.2 Exercise: Nash Equilibrium 2

Exercise 2 Here is another example of a game. In this game, the professor is the rows. He can decide whether to make an effort when preparing his lecture for today or slack off and not prepare. The columns represent a student who can either listen or sleep during the lecture. The cells represent the payoff for the professor and the student.

	Listen	Sleep
Make Effort	$10^6, 10^6$	-10,0
Slack off	0, -10	0,0

Is there a dominant strategy here? How many Nash Equilibria are there? What are they?

Answer
There is no dominant strategy in this case. The best move for the student depends on what the professor does and the best move for the professor depends on what the student does. There are 2 Nash Equilibria: Make Effort and Listen, Slack Off and Sleep.

3 Going Once, Going Twice, SOLD to the giraffe in the back!

In this section, we will talk about the basics of Mechanism Design, namely single-item auctions. These auctions all have the following set-up. A seller has a single good and there are some n strategic bidders who may be interested in buying the item.

We now need to model the desires of a bidder. We will say that bidder i has a valuation v_i which represents the maximum willingness to pay for the item they are selling. So the goal of bidder i is to acquire the item for as little as possible, but it must be less than v_i. Also note that the valuation of each bidder is private (neither the seller or any other bidder knows it).

We will also assume a quasi-linear utility model. This means that if a bidder loses the auction, his utility is 0. If he wins at price p, his utility is $v_i - p$.

3.1 Sealed-Bid Auctions

We will now focus on a certain type of auction called a sealed bid auction. There are three steps to this kind of auction:

1. Each bidder i writes their bid b_i and places it in a sealed envelope to give to the auctioneer
2. The auctioneer decides who wins (gets the good).
3. The auctioneer decides the price that the winner pays for the good.

We will now see different ways that steps 2 and 3 can be implemented.

Definition 4: Most of the time, in practice is to choose the winner as the bidder with the highest bid and ask them to pay their own bid. This is called a **first-price auction**.

However, theoretically, first-price auctions are difficult to reason about both for a bidder and for a seller. For the bidder, it is hard to strategize about how to bid. For the seller, it is hard to predict the price that the good will sell for.

Definition 5: A different auction format that is much easier to reason about theoretically is called a **second-price** or **Vickrey auction**. In this format, the winner does not pay how much they bid. Instead, the winning bidder pays the price that the second highest bidder offered plus some small increment.

Theorem 2 In a second-price auction, every bidder has a dominant strategy. This strategy is to set their bid b_i equal to their private valuation v_i. This strategy maximizes the utility of bidder i no matter what the other bidders do.

Proof

Fix an arbitrary player i and its valuation v_i, and the bids $\vec{b}_{\text{-}i}$ of all the other players (note that $\vec{b}_{\text{-}i}$ is the vector of bids for all bidders excluding bidder i). We want to show that bidder i's utility is maximized when we set $b_i = v_i$ (Remember v_i is the valuation which is constant but each bidder can set their b_i to be whatever they want).

Let $B = max_{j \neq i} b_j$ be the highest bid by some other bidder. Since this is a second-price auction, even though there are an infinite number of possibilities for the bid of i, only distinct outcomes can result. If $b_i < B$, then i loses and gets utility 0. If $b_i \geq B$, then i wins at price B and gets utility $v_i - B$.

We now consider the two possible cases. If $v_i < B$, the highest utility that bidder i can get is $max\{0, v_i - B\} = 0$ which is achieved by bidding truthfully and losing. Second, if $v_i \geq B$, the highest utility bidder i can get is $max\{0, v_i - B\} = v_i - B$ which it achieves by bidding truthfully and winning. $\square$

Importance: Note that this claim shows that second-price auctions are easy to participate in and reason about because you don't need to reason about the other bidders (how many there are, their valuations, or whether they bid truthfully).

3.2 Exercise: Another Important Theorem

Exercise 3 Prove that in a second-price auction, every truthtelling bidder is guaranteed non-negative utility.

Answer

If you lose, your utility is 0. If bidder i wins, their utility is $v_i - p$ where p is the second highest bid. Since i is the winnder (the highest bidder) and their bid is their true valuation, $p \leq v_i$ and so $v_i - p \geq 0$. $\square$

3.3 Exercise: Practice with Vickrey Auctions

Exercise 4 You are part of an advertising firm and are bidding for 2 slots for your ads n the front page of a newspaper. You submit bids (b_1, b_2) where b_1 is the bid for slot 1 and b_2 is the bid for slot 2 given that you got slot 1. Your valuation on some day (v_1, v_2) where $v_1 > v_2 > 0$. You start with \$10000 and the auction is repeated for 1000 days. On each day you are given the total money you have left, the value (v_1, v_2), and the number of bidders in the auction. Under a Vickrey auction, the winner of k slots pays the sum of the k highest rejected bids by other bidders. For your values (v_1, v_2), is keeping your bids equal to your values a dominant strategy? Why or why not?

Answer

Yes, bidding equal to your valuations is a dominant strategy. Suppose a bidder bids

equal to their valuation. Then there are three cases: they win no slot, 1 slot, or 2 slots.
1) Win no slot: If they increase their bids, they could win a slow. But the price is greater than their value so the payoff would be negative. So bidding equal to valuation is a dominant strategy.
2) Win 1 slot: If they change their bids to win 2 slows, then the payoff would go down as the price for the second slot would be higher. If they change their bid to win no slot, the payoff also goes down, so bidding equal to valuation is dominant strategy.
3) Win 2 slots: Increasing bid does not impact payoff. Decreasing bids may cause them to lose one or both slots and decrease payoff. So bidding equal to valuation is the dominant strategy.

3.4 Awesome Auctions

We now turn to a category of auctions known as awesome actions. Awesome actions can be defined by the following 3 properties:

- **Strong incentive guarantees** The auction is dominant-strategy incentive-compatible (DSIC).
- **Strong performance guarantees** If bidders report truthfully, then the auction maximizes the social surplus $\sum_{i=1}^{n} v_i x_i$ where x_i is 1 if i wins and 0 if i loses.
- **Computational efficiency** The auction can be implemented in polynomial time.

Theorem 4 The Vickrey auction is awesome.

Importance
Rather than seeing a proof of this fact, we will discuss what this theorem means for theory about Vickrey auctions.

Firstly, from the perspective of a bidder, the DSIC property (guarantees truthful reporting is a dominant strategy and there is never negative utility) makes it easy to choose a bid. From the perspective of a seller, the DSIC property makes it easy to reason about the outcome of the action.

However, the DSIC property on its own is not enough. The surpus-maximization property states that even though valuations of each bidder were unknown to the auctioneer, the auction will always successfully identify the bidder with the highest valuation. (This is really cool)

The final property is evidently beneficial as we have seen many times in this course already. To have any practical use, we need to be able to run an auction in a reasonable amount of time.

4 Check Your Understanding Questions

1. What is the definition of a normal form game?
2. What is a Nash Equilibrium? Does one always exist for a bimatrix game?
3. Give an example of a game and a corresponding Nash Equilibrium.
4. What are the 3 steps of a sealed bid auction?
5. What are the differences between a first-price and second-price auction?
6. Why are second-price auctions easier to study theoretically?
7. What are the three properties of awesome auctions?
8. Why is computational efficiency important?

5 Answer Key

1. Game has set N of players 1 to n, a strategy set S, and a utility function for each player that depends on the strategies of all other players.
2. A state in which each player's strategy is the best response to the strategies of the other players. Yes.
3. Rock-Paper-Scissors. Uniformly random plays.
4. 1) Each bidder secretly gives their bid to the auctioneer. 2) The auctioneer deciders who wins. 3) The auctioneer decides what price the winner pays.
5. A first-price auction is where the winner pays their own bid. In a second-price auction, the winner pays the price of the second highest bid plus some small amount.
6. Second price auctions are easy to reason about because there is a dominant strategy and there is always non-negative utility.
7. Strong incentive guarantees, strong performance guarantees, can be solved in poly time.
8. Being able to solve problems in polynomial time means that there is a reasonable algorithm that can be used in real world scenarios (and won't take eons to complete).

6 Acknowledgements

15-251 Staff. Chapter 23: Track C. Diderot.one, www.diderot.one/courses/92/books/357/chapter/5187?toc=book.

Etessami, Kousha. Algorithmic Game Theory and Applications: Lecture 1: What Is Game Theory? Inf.ed.ac.uk, www.inf.ed.ac.uk/teaching/courses/agta/lec1e.pdf.

Mehta, Vaibhav. 15-381 Spring 06 Assignment 4: Game Theory and Auctions. Cs.cmu.edu, 8 Mar. 6AD, www.cs.cmu.edu/afs/cs.cmu.edu/academic/class/15381-s06/www/hw4/hw4_sol.pdf.

Roughgarden, Tim. CS364A: Algorithmic Game Theory: Lecture #1: Introduction and Examples. Timroughgarden.org, 2013, timroughgarden.org/f13/l/l1.pdf.

Roughgarden, Tim. CS364A: Algorithmic Game Theory: Lecture #2: Mechanism Design Basics. Timroughgarden.org, 2013, http://timroughgarden.org/f13/l/l2.pdf.

Algorithmic Game Theory. Wikipedia, Wikimedia Foundation, 20 Apr. 2021, en.wikipedia.org/wiki/Algorithmic_game_theory.