601.433/633 Introduction to Algorithms **Lecturer:** Michael Dinitz
Topic: Algorithmic Game Theory **Date:** 12/2/21

26.1 Introduction

Today we're going to spend some time discussing game theory and algorithms. There are a lot of different ways and places that algorithms and game theory intersect – we're only going to discuss a few of them. There is actually an incredibly active research area in theoretical CS called "algorithmic game theory", but we're only going to scratch the surface and discuss some of the more classical results. I don't work too much in AGT (although I have a few related papers), but it's a very important part of modern theoretical computer science, so I'm teaching a course on AGT next semester. You can think of this lecture as a preview / advertisement for that course.

26.2 Two-player zero-sum

One of the oldest settings in game theory are two-player zero-sum games. As an illustrative example, consider a penalty kick in soccer. As most of you probably know, penalty kicks are from such a close range that the goalie has to guess which direction the kick will be before it has actually happened. If (s)he guesses the correct direction (the direction where the kicker kicks it) then (s)he can block it, but otherwise the kick goes in. We can model this using the following matrix game, where the kicker is the row player and the goalie was the column player:

	Left	Right
Left	(0,0)	(1,-1)
Right	(1, -1)	(0,0)

In other words, if the shooter kicks it left/right and the goalie dives in the opposite direction then there's a goal: the shooter gets 1 and the goalie gets -1. If they go in the same direction, though, there is no goal and no player gets any payoff.

In general, there is a matrix M with n rows and m columns, and each entry is of the form $(x, -x)$. Such an entry at position (i, j) means that if the row player chooses action i and the column player chooses action j then the row player receives payoff x and the column player receives payoff $-x$. Saying that the game is "zero-sum" just means that in every row the sum of the two payoffs is 0 (note that x could be negative, so the column player isn't necessarily restricted to negative payoffs, but we'll usually think of x as being positive). Sometimes we will just say that $M \in \mathbb{R}^{n \times m}$ and only put the row player's payoffs in the matrix, since that uniquely determines the payoffs of the column player.

Given such a game, what should a player (either row or column) do? A natural thing to try for is a (randomized) strategy that maximizes the expected payoff even when the opposing player does the best they can against it. In other words, we could like to maximize (over the choice of all of our randomized strategies) the minimum (over all possible strategies for our opponent) of our expected payoff. Intuitively, this is the strategy we should play if our opponent knows us well – if we play anything else then they will have some opposing strategy where we do worse. Such a strategy is called a *minimax* strategy.

Slightly more formally, if we are the row player we want to find values $p_1, p_2, \ldots, p_n$ such that these values are a probability distribution over the rows (each p_i is at least 0 and $\sum_{i \in [n]} p_i = 1$) and we maximize $\min_{j \in [m]} \sum_{i \in [n]} p_i M(i,j)$. Let V be this value, i.e.,

$$V = \max_{\text{probability distributions } p} \min_{j \in [m]} \sum_{i \in [n]} p_i M(i,j).$$

This means that no matter what the column player does, the expected payoff of the row player is at least V (if the row player plays any distribution over the columns it does no better than the single best column, which itself would give value at least V to the row player).

Interestingly, note that we can actually compute the minimax value and strategy using linear programming! This is pretty straightforward from the definition:

$$\begin{array}{lll} \max & V & \\ \text{subject to} & \sum_{i=1}^{n} p_i = 1 & \\ & \sum_{i=1}^{n} p_i M(i,j) \geq V & \forall j \in [m] \\ & 0 \leq p_i \leq 1 & \forall i \in [n] \end{array}$$

So in a two-player, zero-sum game, each player can actually compute its optimal strategy (the column player can write a similar LP).

So what about the above example? It's pretty clear that both players actually have the same minimax strategy: choose between left and right with probability 1/2 each. Then the shooter has expected payoff of 1/2 and the goalie has expected payoff of $-1/2$.

Let's change the example slightly to have a goalie who is weaker on the left: if the goalie dives left and the shooter also kicks it there, there is still a 1/2 probability of the ball getting through. This gives us the following game:

	Left	Right
Left	$(\frac{1}{2}, -\frac{1}{2})$	(1,-1)
Right	(1, -1)	(0,0)

In this game the minimax strategy for the shooter is $(2/3, 1/3)$, which guarantees expected gain of at least 2/3 no matter what the goalie does. The minimax strategy for the goalie is also $(2/3, 1/3)$, which guarantees expected loss of at most 2/3 no matter what the shooter does.

Theorem 26.2.1 (Minimax Theorem (von Neumann)) *Every 2-player zero-sum game has a unique value V such that the minimax strategy for the row player guarantees expected gain of at least V, and the minimax strategy for the column player also guarantees expected loss of at most V.*

This value V is known as the *value* of the game. This theorem is somewhat counterintuitive. For example, it implies that if both players are optimal, it doesn't hurt to just publish your strategy. Your opponent can't take advantage of it.

26.3 General games and Nash equilibria

In general games we can remove both of the restrictions: we allow more than 2 players (although 2 is often a useful number to consider), and payoffs don't need to add to 0. Among other things, this means that games no longer have a unique value. And instead of minimax strategies, we have the notion of a *Nash equilibrium.* Informally, a Nash equilibrium is a strategy for each player (these might be randomized strategies) such that no player has any incentive to deviate. Let's do a simple example: two people walking down the sidewalk, deciding which side to walk on.

	Left	Right
Left	(1,1)	(-1,-1)
Right	(-1, -1)	(1,1)

This game has three Nash equilibria: both people walk on the left, both people walk on the right, or both people decide random with probabilities $(1/2, 1/2)$. Note that the first two equilibria give both players payoff of 1, while the third equilibrium gives both players expected payoff of 0. Nevertheless, it is an equilibrium: if that's what both players are doing, then neither player has any incentive to change their strategy. To see this, consider the row player. Knowing that the column player is playing each column with probability 1/2, then if the row player uses probability p_L, p_R they will get expected payoff of $\frac{1}{2}(1 \cdot p_L - 1 \cdot p_R) + \frac{1}{2}(-1 \cdot p_L + 1 \cdot p_R) = 0$, and thus the row player has no incentive to deviate from $p_L = p_R = 0.5$.

26.3.1 Computing Nash

Nash proved that if the number of players is finite and the number of possible actions for each player is finite, then there is always at least one Nash equilibrium (this is not true if we restrict the players to have *deterministic* strategies, but is true if we allow randomized strategies). However, his proof is nonconstructive – it doesn't give an algorithm for actually finding an equilibrium. This hasn't seemed to bother economists and mathematicians, but it should bother us! After all, the whole point of an equilibrium from an economics point of view is that it's a good solution concept – the claim is that markets/systems will naturally end up at equilibrium. But if it's hard to compute an equilibrium, then how can we possibly expect a massive distributed system like a market to end up at equilibrium?

So we have the following algorithmic question: given a game, can we compute a Nash equilibrium? This is actually a bit tricky to formalize. Since an equilibrium always exists the decision question "does this game have an equilibrium" is certainly not NP-complete. And since equilibria don't come with a notion of "value" like minimax, we can't create a less trivial problem like "does this game have an equilibrium with large value". Instead, we have to use a different complexity class: PPAD. Without going into any details, PPAD is a class of problems in which the answer is always "yes", but where (we believe) it is hard to actually find a solution in general.

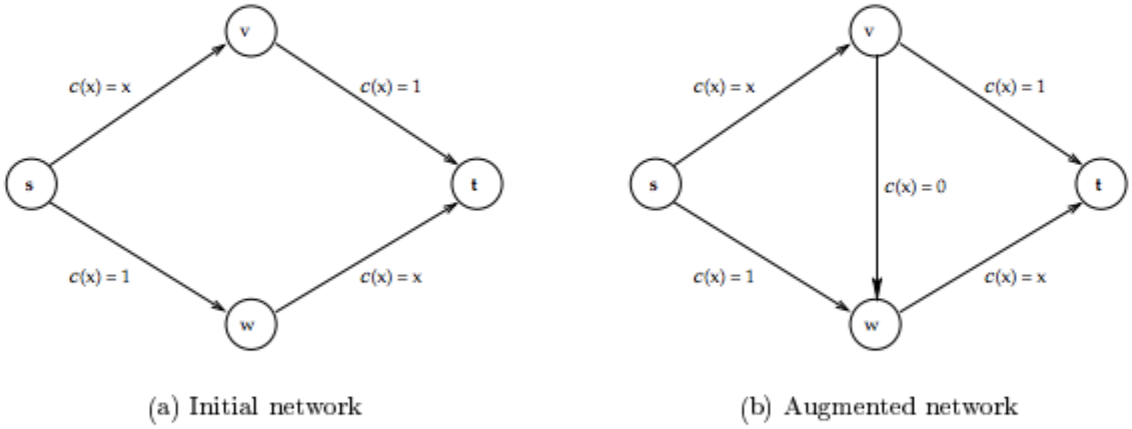
Theorem 26.3.1 (Daskalakis, Goldberg, Papadimitriou) *Computing a Nash equilibrium is PPAD-complete.*

From a computational point of view, this means that Nash equilibria are in fact not good solution concepts in general, since there's no real reason assume that a game will end up at such an equilibrium. Of course, for specific games/markets we might be able to prove that it is easy to compute a Nash equilibrium, or even that natural distributed algorithms (e.g., best-response dynamics) converge quickly to an equilibrium. But in general we don't believe that this is the case. Among other things, this has motivated other notions of equilibrium which are actually computable, such as correlated and coarse-correlated equilibria (which were defined previously by economists but were only recently shown to have natural distributed algorithms).

26.3.2 Braess's Paradox

Nash equilibria can behave somewhat strangely and counterintuitively. The most famous example of this is Braess's Paradox, which comes up in routing games. In a routing game (at least the simplest versions), we are given a graph, together with a source and destination. We assume there are a huge number of players (say $1/\epsilon$), each of which is trying to get from the source to the destination as quickly as possible. So each player is responsible for ϵ traffic, and its actions are the possible paths from the source to the destination. However, the length of an edge might be some function of the total traffic along the edge, rather than just a number.

Let's do an example. Consider the graph from part (a) of the following figure, where for each edge, the function $c(x)$ denotes the length of the edge as a function of the fraction of traffic using it.



(a) Initial network

(b) Augmented network

The only Nash equilibrium is for half of the players to choose the top path and half to choose the bottom path. If more than half choose the top or more than half choose the bottom, then they could obtain shorter travel time by switching. So at equilibrium, every player has travel time equal to 3/2.

But now suppose that the government decides to invest in a new road. This is such a great road that it takes almost no time to travel across it, no matter how many players use it. We might think that this can only make things better – clearly if the road is between the source of the destination then it's a huge help (travel time gets cut to 0), and if it's between any two other points then it still provides a ton of extra capacity. Unfortunately, this is not the case.

Consider what happens if we place it between the two non-source/destination nodes (part (b) of the previous figure). Suppose that α traffic uses the top path, β uses the bottom, and the remaining $1-\alpha-\beta$ use the zig-zag. Then players using the top path take $1+(\alpha+1-\alpha-\beta) = 2-\beta$ time, the bottom path takes $1+(\beta+1-\alpha-\beta) = 2-\alpha$ time, and the zig-zag path takes $\alpha+(1-\alpha-\beta)+\beta+(1-\alpha-\beta) = 2-\alpha-\beta$ time. Since α and β are at least 0, this means that the zig-zag path is always the fastest! So the only Nash equilibrium is where all players use the new zig-zag path, giving a travel time of 2 for each player!

This effect is called Braess's Paradox. Note that it is entirely due to game-theoretic behavior: if we could tell every player what to do then we could force them all to simply ignore the new road. But since players are selfish, adding this fancy new road actually decreased the quality of the system. Nowadays, people use the term "Braess's Paradox" to mean such a situation, even outside of routing games. For example, a few years ago I wrote a paper showing that in wireless networks, improving technology (e.g. improving the signal-to-noise ratio that we can decode, or allowing nodes to choose their broadcast power, or allowing fancy decoding techniques like interference cancellation) can actually result in worse Nash equilibria, and thus worse actual performance.

26.3.3 Price of Anarchy

Braess's paradox has been known for a while, but (like with the computational issues involving Nash) it didn't seem to bother economists too much. I'm not quite sure why this is, but maybe if you assume that games/markets are "natural" then there's not much point in comparing them to centralized solutions. But comparing to optimal solutions is exactly what we do all the time in theoretical CS! We did this, for example, with approximation and online algorithms.

This motivates the definition of the *price of anarchy*, which was introduced by Koutsoupias and Papadimitriou in 1999. For a given game, let OPT denote the "value" of the best solution, which is typically (although not always) the social welfare, i.e. the sum over all players of the value obtained by the player. So, for example, in the routing game we looked at above $OPT = 3/2$. For a fixed equilibrium s, let $W(s)$ denote the "value" of the equilibrium, which again might be defined differently for different games but (for example) in the routing game is equal to the average trip length when players use equilibrium s. So before the new road there was only one possible s and $W(s) = 3/2$, and after the new road there is still only one possible s but it is a different equilibrium and now $W(s) = 2$. Let $\mathcal{S}$ denote the set of all equilibria.

Definition 26.3.2 *The* price of anarchy *of a minimization game is* $\max_{s\in\mathcal{S}} W(s)/OPT$*, and the PoA of a maximization game is* $\min_{s\in\mathcal{S}} W(s)/OPT$.

In other words, the price of anarchy of a game is the ratio between the *worst* Nash equilibrium and the optimum value. We don't have time to go into it, but analyzing the price of anarchy of various games has been a popular area for the last 20 years or so. We now understand many classes of games quite well. For example, take routing games: it is known (thanks to Tim Roughgarden) that as long as edge lengths are a linear function of the traffic across them (as they were in our example) the Price of Anarchy is always at most 4/3. Recall that in our example, OPT was 3/2 but the only equilibrium had value 2, and hence the price of anarchy was $2/(3/2) = 4/3$. So in fact our simple example of Braess's paradox is the worst possible.