601.436/636 Algorithmic Game Theory **Lecturer:** Michael Dinitz
Topic: Introduction to Inefficiency of Equilibria **Date:** 2/25/20
Scribe: Michael Dinitz

9.1 Introduction

Today we're going to start the second part of the course: studying the (in)efficiency of equilibrium. In other words, we're going to leave aside for now the question of how a game arrived at an equilibrium, and instead study the *quality* of equilibria. This class is really a brief introduction, so I'm going to be a bit handwavy, and we'll dive more into details in the next few weeks.

9.1.1 Definitions

Let's think about a cost minimization game with n players $[n]$, strategy sets $S_1, S_2, \ldots, S_n$ with $S = S_1 \times S_2 \times \cdots \times S_n$, and a cost function $C_i : S \to \mathbb{R}$ for each $i \in [n]$. We want to study how close to optimal the equilibria of this game are, but before we can do that we need to consider a few modeling questions

1. What is the objective function we use to compare solutions? In order to define the "best" outcome, we need some objective function $f : S \to \mathbb{R}$
2. What kind of equilibria are we going to study? Pure/mixed Nash? Correlated? Coarse correlated?
3. If there are multiple equilibria, which one(s) do we consider?

There are no right or wrong answers to these questions – they just give us different questions to consider, and what's most important depends on the context. There are some common choices, though.

For the objective (question 1), the most common choice is "social welfare" (for utility maximization) or "social cost" (for cost-minimization): $f(s) = \sum_{i=1}^n C_i(s)$. But sometimes we'll care about things like fairness: $f(s) = \max_{i=1}^n C_i(s)$. And sometimes there are even weirder ones: a lot of my work has focused on a particularly weird setting of $f(s) = |\{i \in [n] : C_i(s) < 0\}|$.

For questions 2 and 3, different choices give different measures. There are a few that we're going to tend to focus on. Let M denote the set of Nash equilibria, and let CC denote the set of coarse correlated equilibria. Let $OPT = \min_{s \in S} f(s)$.

Definition 9.1.1 *The* Price of Anarchy *is the ratio of the worst Nash to OPT:*

$$\frac{\max_{\sigma \in M} \mathbf{E}_{s \sim \sigma}[f(s)]}{OPT}$$

Definition 9.1.2 *The* Price of Stability *is the ratio of the best Nash to OPT:*

$$\frac{\min_{\sigma \in M} \mathbf{E}_{s \sim \sigma}[f(s)]}{OPT}$$

Definition 9.1.3 *The* Price of Total Anarchy *is the ratio of the worse CCE to OPT:*

$$\frac{\max_{\sigma \in CCE} \mathbf{E}_{s\sim\sigma}[f(s)]}{OPT}$$

Note that these values are always at least 1, and the price of stability is at most the price of anarchy which is at most the price of total anarchy. For utility maximization games we can change the definitions accordingly (worst is now min, best is max) and get values that are always less than 1, and (if we want) we can put OPT in the numerator to get values larger than 1 again.

The price of anarchy has received the most attention, so we'll tend to focus on it, but we'll talk about all of them.

9.2 Warmup: Prisoner's Dilemma

Remember the prisoner's dilemma game:

	confess	silent
confess	(4,4)	(1,5)
silent	(5,1)	(2, 2)

Let's consider the social cost objective. Then OPT is when both players are silent, giving total cost of $2+2=4$. On the other hand, the only Nash equilibrium is the pure equilibrium where both players confess, giving total cost of 8. Thus the price of anarchy equals the price of stability equals $8/4=2$. This might not seem that bad, but consider what happens if we have the same "structure" but different numbers:

	confess	silent
confess	(α,α)	$(1,\alpha+1)$
silent	$(\alpha+1,1)$	(2, 2)

Let $\alpha > 2$. Then OPT is the same and it's still true that the only Nash equilibrium is for both players to confess. But this Nash has total cost 2α, and thus the price of anarchy = price of stability $= (2\alpha)/4 = \alpha/2$. Thus by making α arbitrarily large, the price of anarchy gets arbitrarily bad!

9.3 Nonatomic Routing

Let's go back to another example from the first lecture: nonatomic routing. We talked about atomic routing when we talked about potential games, but now we're going to go back to the nonatomic case. Recall that in this setting we have an infinite (or finite but extremely large) number of players each sending infinitesimal traffic from some node s to some other node t in a directed (multi)graph. The delay along each edge e is some function $c_e(x)$, where x is the fraction of the flow that uses e. The strategies are all $s \to t$ paths, and the cost to a player of some strategy profile (i.e., an $s \to t$ flow of size 1) is the delay that they incur along their chosen path. Slightly more formally, given a flow f (distribution over the set $\mathcal{P}$ of $s \to t$ paths) with f_P being the flow along path P, we can let $f_e = \sum_{P \in \mathcal{P}: e \in P}$ be the flow using edge e and $c_P(f) = \sum_{e \in P} c_e(f_e)$ be the cost of using path P.

Let's use as our objective function the average player cost $\sum_{P \in \mathcal{P}} c_P(f) \cdot f_P$, and see happens at equilibrium. The following example is known as Pigou's example: there are two vertices s, t, and two parallel edges e_{top} and e_{bottom} from s to t. The top edge has $c_{e_{top}}(x) = 1$ and the bottom edge has $c_{e_{bottom}}(x) = x$. Clearly the only Nash equilibrium is when every player uses the bottom edge, i.e., flow 1 is sent along the bottom edge and 0 along the top edge. This has average player cost of 1.

But what is OPT? Suppose we sent α flow on the bottom edge and $1 - \alpha$ flow on the top edge. Then the average player cost is

$$(1-\alpha) \cdot 1 + \alpha \cdot \alpha = 1 - \alpha + \alpha^2.$$

This is minimized when $\alpha = 1/2$ to give an average player cost of 3/4. Thus the Price of Anarchy (and the Price of Stability) in this game is $1/(3/4) = 4/3$. In fact, this turns out to be the worst case: we're going to prove the following theorem on Thursday.

Theorem 9.3.1 *If all edge cost functions are affine, then the Price of Anarchy of any nonatomic routing game is at most* 4/3.

However, the restriction to affine is necessary. If the bottom edge has cost x^p for some large p, then the only Nash still has cost 1 but if we send ϵ flow on the top edge and $(1-\epsilon)$ on the bottom edge we get cost

$$\epsilon \cdot 1 + (1-\epsilon)(1-\epsilon)^p = \epsilon + (1-\epsilon)^{p+1}.$$

As p goes to infinity, this goes to ϵ. So the Price of Anarchy is unbounded as p goes to infinity (but is bounded for any particular p, as we'll also see next class).

9.4 Network Creation Games

Let's look at a different type of games on graphs. Instead of routing, it will be network formation (although this is in some ways similar to atomic routing – the main difference is the cost function). Here we have the following setup.

- a directed (multi)graph $G = (V, E)$, with cost function $c : E \to \mathbb{R}^+$ (so each edge has a fixed cost, rather than the previous cost function).
- k players, where player i has some source s_i and sink t_i and strategy set $S_i = \{s_i \to t_i \text{ paths}\}$.
- Given strategy profile $f = (P_1, P_2, \ldots, P_k) \in S_1 \times S_2 \times \cdots \times S_k = S$, the cost to player i is

  $$C_i(f) = \sum_{e \in P_i} \frac{c(e)}{f_e}$$

  where $f_e = |\{i : e \in P_i\}|$. In other words, the players who use an edge split the cost of that edge.

The global cost function we will use is the total cost of the created network, which is also the social cost:

$$cost(f) = \sum_{e \in \cup_{i=1}^{k} P_i} c(e)$$

For today, to make things simple, let's only look at pure Nash equilibria and analyze the *pure* prices of anarchy and stability.

Consider a simple example: two nodes s and t with all k players having source s and sink t, a top edge from s to t with cost k, and a bottom edge from s to t with cost $1+\epsilon$. The OPT is clearly $1+\epsilon$: we can just buy the bottom edge. Every player choosing the bottom edge is also a Nash, since if any player switched their cost would go from $\frac{1+\epsilon}{k}$ up to $k/1 = k$. Thus the pure price of stability is 1. But if all players choose the top edge then each one has cost $k/k = 1$, while if any one of them switched they would have cost $(1+\epsilon)/1 = 1+\epsilon$. Thus all players using the top edge is also a Nash equilibrium, and the global cost of this solution is k. Hence the pure price of anarchy is only $k/(1+\epsilon)$.

A natural question from this example is whether the price of stability is always 1 for these network creation games. Unfortunately, the answer is no. Consider the following example. There are k players, each one of which has their own source s_i. There is an edge from each s_i to a global sink t, with the cost of this edge being $1/i$. There is one other vertex v, and there is an edge of cost 0 from each s_i to v. Finally, there is an edge of cost $(1+\epsilon)$ from v to t.

OPT in this game is for every player to go from their source to v (along a zero-cost edge) and then to t, for a total cost of $1+\epsilon$. But in this solution player k is paying $(1+\epsilon)/k$, while if they switch to the edge (s_k, t) they would pay $1/k$. Once they've switched, the same logic will hold for player $k-1$, etc. So it's not too hard to see that the unique Nash is when every player uses their direct edge, for a total cost of $\sum_{i=1}^{k} \frac{1}{i} = \Theta(\ln k)$. So the price of stability is only $O(\ln k)$. In a few lectures we'll prove that this is the worst case:

Theorem 9.4.1 *The price of stability in this kind of network creation game is at most* $O(\ln k)$*.*

9.5 Scheduling Games

Let's consider a simple non-graph game based on scheduling jobs on machines.

- There are n players $[n]$ and also machines $[n]$ (so the same number of jobs as machines).
- Each player has strategy set equal to the machines $[n]$
- Jobs want to be processed quickly, so the cost to a job is the number of jobs that select the same machine: $C_i(s) = |\{j : s_j = s_i\}|$
- Our global objective is an extreme notion of fairness: the maximum load, aka the *makespan*:

$$f(s) = \max_{k \in [n]} (|\{j : s_j = k\}|)$$

The optimal solution is obvious: each job should get their own machine, so $OPT = 1$. This is clearly also a (pure) Nash. Moreover, these bijections are the only pure Nashes: if there are two jobs that choose the same machine then there must be some machine with no jobs on it, so any player on a machine with load at least 2 would have incentive to switch. Thus any pure Nash is a bijection between jobs and machines, and thus the pure price of anarchy is 1.

This is not true when we allow mixed Nashes, though. Consider the following set of mixed strategies: every player (job) chooses a strategy (machine) uniformly at random. I claim that this is a Nash equilibrium. To see this, consider some player i. Under the current distribution, the expected cost to player i is

$$\begin{aligned}\mathbf{E}[C_i(s)] &= \sum_{j=1}^{n} \mathbf{Pr}[s_i = j]\,\mathbf{E}[C_i(s)|s_i = j] = \frac{1}{n}\sum_{j=1}^{n}\left(1 + \sum_{k \neq i} \mathbf{Pr}[s_k = j]\right) \\ &= 1 + \frac{1}{n}\sum_{j=1}^{n}\sum_{k \neq i}\frac{1}{n} = 1 + \frac{1}{n}\sum_{j=1}^{n}\frac{n-1}{n} = 1 + \frac{n-1}{n}\end{aligned}$$

But if player i deviated to action j (recall that we only need to worry about pure strategy deviations), its expected cost would be

$$1 + \sum_{k \neq i} \mathbf{Pr}[s_k = j] = 1 + \frac{n-1}{n}$$

Hence no player has incentive to deviate, so this is a Nash equilibrium.

An important classical result (known as "balls-in-bins") is a tight bound the expected maximum load under this distribution, i.e., the expected global cost under this distribution, of $\Theta(\frac{\log n}{\log \log n})$. Note that this is expectation of the maximum load (which is what we care about), *not* the maximum over machine of the expected load (which is only 1).

Thus the price of anarchy (allowing mixed Nash) is no better than $\Omega(\frac{\log n}{\log \log n})$.