601.436/636 Algorithmic Game Theory **Lecturer:** Michael Dinitz
Topic: Nonatomic Routing Games **Date:** 2/24/22

10.1 Introduction

Today we're going to focus on routing games, which are historically the first major class of games where the Price of Anarchy was studied.

10.2 Setup

In a nonatomic routing game we have the following setup (note that it's a little different from most of the games we've been talking about because there are essentially an infinite number of players).

- Directed (multi)graph $G = (V, E)$
- Source $s \in V$ and sink $t \in V$. We could have multiple source/sink pairs (commodities) and everything we do will still be true, but let's keep it simple today with one commodity. See Exercise 11.5 in Roughgarden's book.
- A *rate* (or amount) r of traffic. On Tuesday we said that this was 1, but today we're going to generalize to $r > 0$.
- A cost function $c_e : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ for all $e \in E$. We'll assume that these cost functions are nonnegative, continuous, and nondecreasing. Most of our focus today will be on the affine case, where $c_e(x) = ax + b$ with $a, b \geq 0$.
- Let $\mathcal{P}$ be the set of $s \to t$ paths in G.
- A *flow* f is $\{f_P\}_{P \in \mathcal{P}}$ such that $f_P \geq 0$ for all $P \in \mathcal{P}$ and $\sum_{P \in \mathcal{P}} f_P = r$.
- Given a flow f, for each $e \in E$ let $f_e = \sum_{P \in \mathcal{P} : e \in P} f_P$.
- The cost of using a path $P \in \mathcal{P}$ is $c_P(f) = \sum_{e \in P} c_e(f_e)$
- The total cost of a flow (the social cost) is

$$\begin{aligned} C(f) &= \sum_{P \in \mathcal{P}} f_P c_P(f) \\ &= \sum_{P \in \mathcal{P}} f_P \sum_{e \in P} c_e(f_e) = \sum_{e \in E} \sum_{P \in \mathcal{P} : e \in P} f_P c_e(f_e) = \sum_{e \in E} c_e(f_e) \sum_{P \in \mathcal{P} : e \in P} f_P \\ &= \sum_{e \in E} c_e(f_e) f_e \end{aligned}$$

- A flow f is an *equilibrium flow* (i.e., a Pure Nash but written this way for historical reasons and because there are an infinite number of players) if $c_P(f) \leq c_{P'}(f)$ for all $P, P' \in \mathcal{P}$ with $f_P > 0$ (note that $f_{P'}$ could be 0). Note that this means that all paths with nonzero flow have the exact same cost, and all other paths have at least that cost.

We'll use equilibrium flows as our notion of Nash, and will want to bound the Price of Anarchy (ratio between the cost of the worst equilibrium flow and the optimal flow).

It should be intuitive that equilibrium flows exist, since (intuitively) we can just inject a differentially small amount of flow using whatever the current lowest-cost path is, which will infinitesimally raise the cost of that path, and repeat. But of course there's a lot of annoyingness when dealing with infinitesimals, so proving this is a bit tricky. I'm not going to do the details, but they're all in Section 18.3.1 of the NRTV book.

Theorem 10.2.1. *There is at least one equilibrium flow, and every equilibrium flow has the same total cost.*

Proof Sketch. This can be done via the potential method (i.e., this is a potential game except there are an infinite number of players). We'll use the potential function

$$\Phi(f) = \sum_{e \in E} \int_0^{f_e} c_e(x)\, dx.$$

Note that this is basically the same potential function that we used in Lecture 5 to argue that atomic routing games have pure Nash equilibria, just with an integral instead of another sum (since there are now an infinite number of players). Now we can do a more complicated version of our previous potential argument to argue that any global minimum of Φ is an equilibrium flow (and thus one exists). Because each c_e is nondecreasing, it's not too hard to show that Φ is actually convex, and so for any two global minima the chord between them consists entirely of global minima. Combining this with the individual convexity of each integral in the sum actually implies that if f, f' are two different equilibrium flows, then $c_e(f_e) = c_e(f'_e)$ for all $e \in E$. □

10.3 Pigou

Let's remember (and generalize a bit) Pigou's example, which we discussed last class. In the version we did last time, there were two nodes s and t with a top edge and a bottom edge from s to t, with the top edge having cost function $c_{top}(x) = 1$ and the bottom edge having cost function $c_{bottom}(x) = x$. We had $r = 1$, and showed that the Price of Anarchy in this example was 4/3. We then generalized this a bit to have the bottom edge have cost function $c_{bottom}(x) = x^p$ for larger values of p, and saw that as p gets larger the price of anarchy gets larger.

We're going to generalize this in a few ways. First, consider some cost function c, and some rate r (not necessarily 1). We're going to change the Pigou network so that the bottom edge has cost function c, and the top edge has cost function which is the fixed cost $c(r)$. So if we use the cost function $c(x) = x^p$ with $r = 1$, then this is exactly our old Pigou example since $c(1) = 1$.

Let's analyze the Price of Anarchy for this generalized Pigou network. Since c is nondecreasing, it's always a dominant strategy to use the bottom edge – its cost is always at most $c(r)$, the cost of the top edge. So in any equilibrium flow, the total cost is $r \cdot c(r)$. On the other hand, a non-equilibrium flow could send some amount x of flow along the bottom edge, in which case it would have cost

$x \cdot c(x) + (r - x)c(r)$. Thus the price of anarchy is the worst case of this over all x, i.e.,

$$\sup_{0 \le x \le r} \left(\frac{r \cdot c(r)}{x \cdot c(x) + (r - x) \cdot c(r)} \right)$$

As we discussed last class, if $r = 1$ and $c(x) = x$ then this supremum is $4/3$ and is achieved by setting $x = 1/2$. To make this gap as big as possible, we can also take supremum over r.

We're going to slightly change this to allow all $x \ge 0$, not just $x \le r$. This doesn't change anything – since c is nondecreasing, the denominator is always minimized at some $0 \le x \le r$. And it will make our life much easier later.

Now suppose we want to find the worst Pigou network for some *class* of cost function $\mathcal{C}$, e.g., the affine cost functions, the quadratic cost functions, etc. Then in addition to taking the supremum over x and r, we also get to take the supremum over cost functions from this class. This gives the following definition.

Definition 10.3.1. *Let $\mathcal{C}$ be a class of cost functions. The* Pigou bound *for $\mathcal{C}$ is*

$$\alpha(\mathcal{C}) = \sup_{c \in \mathcal{C}} \sup_{r \ge 0} \sup_{x \ge 0} \left(\frac{r \cdot c(r)}{x \cdot c(x) + (r - x) \cdot c(r)} \right)$$

We're not going to do it, but it's not *that* hard to actually figure out the Pigou bound for different classes $\mathcal{C}$. Roughgarden's book has it for some simple examples. For affine functions it is exactly $4/3$, and in general for degree p polynomials it is approximately $p/\ln p$.

The amazing thing, which we're going to prove next, is that this is actually the worst case. More formally, we're going to prove the following theorem.

Theorem 10.3.2. *Let $\mathcal{C}$ be a class of cost functions. Then any nonatomic routing game with cost functions in $\mathcal{C}$ has Price of Anarchy at most $\alpha(\mathcal{C})$.*

Proof. Let f^* denote the optimal flow and let f be an equilibrium flow. So we want to show that $\frac{C(f)}{C(f^*)} \le \alpha(\mathcal{C})$.

For any edge $e \in E$, we can "instantiate" the Pigou function with $c = c_e$ and $r = f_e$ and $x = f_e^*$. That is, we can pick these particular values, and since the Pigou bound is defined as the supremum over all possible values, the Pigou bound is at least as large as what we get with these choices. (This is why we wanted to allow for $x \ge r$). So

$$\alpha(\mathcal{C}) \ge \frac{r \cdot c(r)}{x \cdot c(x) + (r - x) \cdot c(r)} = \frac{f_e \cdot c_e(f_e)}{f_e^* \cdot c_e(f_e^*) + (f_e - f_e^*) \cdot c_e(f_e)}$$

Why did we do this? Because at least some of these terms look like things we care about: the numerator is the cost to the equilibrium flow of e, and the first term of the denominator is the cost to the optimal flow of e. The second term of the denominator is a little confusing, and is something we're going to have to deal with.

We can now rearrange this equation to get

$$f_e^* \cdot c_e(f_e^*) + (f_e - f_e^*) \cdot c_e(f_e) \geq \frac{1}{\alpha(\mathcal{C})} f_e c_e(f_e).$$

Summing this inequality over all $e \in E$, we get

$$\sum_{e \in E} (f_e^* \cdot c_e(f_e^*) + (f_e - f_e^*) \cdot c_e(f_e)) \geq \frac{1}{\alpha(\mathcal{C})} \sum_{e \in E} f_e c_e(f_e)$$
$$\implies C(f^*) + \sum_{e \in E} (f_e - f_e^*) c_e(f_e) \geq \frac{1}{\alpha(\mathcal{C})} C(f).$$

This is almost want we want! If that weird sum term weren't there, then this would imply the theorem. So to finish the proof, we just want to show that $\sum_{e \in E} (f_e - f_e^*) c_e(f_e) \leq 0$, or equivalently that $C(f) - \sum_{e \in E} f_e^* c_e(f_e) \leq 0$.

To do this, note that since f is an equilibrium flow, every path with nonzero flow has the exact same cost. Let L be this cost. Then $C(f) = \sum_{P \in \mathcal{P}} f_P c_P(f) = L \sum_{P \in \mathcal{P}} f_P = rL$. Now let's handle the second term:

$$\sum_{e \in E} f_e^* c_e(f_e) = \sum_{e \in E} c_e(f_e) \sum_{P \in \mathcal{P}: e \in P} f_P^* = \sum_{P \in \mathcal{P}} \sum_{e \in P} f_P^* c_e(f_e) = \sum_{P \in \mathcal{P}} f_P^* \sum_{e \in P} c_e(f_e) = \sum_{P \in \mathcal{P}} f_P^* c_P(f)$$
$$\geq L \sum_{P \in \mathcal{P}} f_P^* = rL.$$

Thus $C(f) - \sum_{e \in E} f_e^* c_e(f_e) \leq 0$, so we're done! □

10.4 Application: Network Overprovisioning

This proof has some nonobvious but neat implications beyond just the theorem statement itself. As an example, let's talk about something called *overprovisioning*. Network operators typically overprovision their networks (add more capacity than they think should actually be needed), since empirically networks tend to perform significantly better if they are slightly overprovisioned. There are many explanations for this, but there's actually a neat connection to the price of anarchy.

In a typical network scenario, each edge usually has some *capacity* u_e. Consider the cost function

$$c_e(x) = \begin{cases} \frac{1}{u_e - x} & \text{if } x < u_e \\ \infty & \text{if } x \geq u_e \end{cases}$$

I'm not going to go into this, but it turns out that this is the "expected delay" in something called an M/M/1 queue, which is a simple model that is commonly used to model networks. So this is a reasonable cost function.

We say that a network is *β-overprovisioned* if $f_e \leq (1 - \beta) u_e$ for all $e \in E$ in an equilibrium flow. In other words, at equilibrium, every link has β fraction of its capacity spare. If you think about how we proved the main theorem, we only ever applied the cost functions to the equilibrium flow and the optimal flow, so as long the network is β-overprovisioned we only ever apply them when there's spare capacity (this can be made formal – see Chapter 12.1 of Roughgarden's book). Then we can use the main theorem to analyze the Pigou bound for these cost functions under the guarantee that there's β spare capacity. If you do this, you get that the Price of Anarchy is bounded by

$$\frac{1}{2}\left(1+\sqrt{\frac{1}{\beta}}\right)$$

This is pretty good! For example, if $\beta = 0.1$ (so we overprovision by 10%), the Price of Anarchy is bounded by 2.1. So in overprovisioned networks, selfish routing doesn't hurt us very much.