Lecture 21: Price of Anarchy

Lecturer: *Sahil Singla* Last updated: March 22, 2022

1 Nonatomic Routing Games

A non-atomic routing game has the following setup:

- A directed graph, G.
- Uncountably many, infinitesimally small players.
- For each player, a source node s and a destination $t \in G$. Denote by r_{st} the total mass of players who want to travel from s to t.
- Each edge $e \in G$ has a *cost function*, $c_e(\cdot)$. $c_e(x)$ denotes the time it takes when x units of traffic want to traverse edge e.

For any s, t such that $r_{st} > 0$, we'll assume there exists at least one path from s to t. Let P_{st} denote the set of all such paths, and we'll let f_p denote the mass of players who choose to take path p to get from s to t. Finally, we can now define the game:

- Every player chooses a path from their source to their destination.
- Let $f_e = \sum_{p|e \in p} f_p$, this is the total mass of traffic using edge e.
- The cost of taking edge e is $c_e(f_e)$.
- The delay for a player taking path p is $\sum_{e \in P} c_e(f_e)$.
- The total delay is $\sum_e f_e c_e(f_e)$ (note that this equals $\sum_p f_p \sum_{e \in p} c_e(f_e)$).

We say that a flow is *optimal* for $(G, \vec{r}, c)$ if it has minimal total delay over all flows that send r_{st} flow from s to t (for all s, t). We say that a flow is in *equilibrium* if no player can gain from deviating. Note that because players are non-atomic, they can switch paths without affecting the cost of that path. So a flow is in equilibrium iff:

$$\forall s, t, \text{ and } p, p' \in P_{st}, f_p > 0 \quad \Longrightarrow \quad \sum_{e \in p} c_e(f_e) \leq \sum_{e \in p'} c_e(f_e). \tag{1}$$

Above, note that if the condition is violated, then there is some player currently choosing to take path p even though p' has lower cost. Let's look at some examples:

Example 1: Pigou's Example There are two nodes, s and t, with two edges between them: one has $c(x) = 1$ for all x, the other has $c(x) = x$ for all x. $r_{st} = 1$.

- What's the optimal flow for this example? We can write an optimization: minimize $(1 - x) \cdot 1 + x \cdot x = x^2 - x + 1$ (if we send x flow along the bottom edge and $1 - x$ along the first edge, this is the total cost). This is minimized at $x = 1/2$ for a total delay of $3/4$.
- What is an equilibrium flow for this example? Send no flow along the top edge, and all traffic along the bottom edge. The total delay is 1.
- Are there any other equilibria? No. If < 1 flow uses the bottom edge, then the cost of the bottom edge is < 1, but there is positive flow on the top edge which costs 1.

The *Price of Anarchy* of an example denotes the ratio between the cost of the *worst possible equilibrium* and the *optimal outcome*. So by bullet two above, the Price of Anarchy of Pigou's example is at least $4/3$. By bullet three, there are no other equilibria, so in fact the price of anarchy is exactly $4/3$.

Theorem 1. *[Roughgarden and Tardos 01] For any $G, \vec{r}$, and c such that for all e, $c_e(x) = ax + b$ (for some real constants $a, b \geq 0$), the price of anarchy is at most $4/3$.*

2 Analyzing the Worst Case for Pigou Example

In this section we show that the worst-possible price of anarchy for the Pigou example with linear costs is $4/3$. In the next section we will show that Pigou example are actually the worst possible graphs, so price of anarchy for linear cost functions is always at most $4/3$ for any graph.

Below, a *Pigou Example* is a two-node, two-edge graph where r units of traffic want to go from s to t, the top edge has a constant cost equal to $c(r_{st})$, and the bottom edge has cost function $c(x)$.

Note that, for a given c, r, it's an equilibrium to send all r units of traffic along the bottom edge, which induces total cost $r \cdot c(r)$. The optimal flow sends some y along the bottom edge and $r - y$ along the top edge, which induces total cost $rc(r) - yc(r) + yc(y)$.

The price of anarchy for a class $\mathcal{C}$ of cost functions over the Pigou example is defined as:

$$\alpha(\mathcal{C}) := \sup_{c \in \mathcal{C}, r \geq 0, y \geq 0} \frac{r \cdot c(r)}{(r - y) \cdot c(r) + y \cdot c(y)}. \tag{2}$$

Notice that in the above definition y could potentially be greater than r. This makes little sense intuitively but will be crucial in our application in the next section.

Lemma 2. *For the class $\mathcal{C}$ of affine cost functions $c_e(x) = ax + b$ (for some real constants $a, b \geq 0$), we have $\alpha(\mathcal{C}) = 4/3$.*

Proof. Consider any cost function $c \in \mathcal{C}$ where $c(x) = ax + b$ for some a, b. We first argue that if $y > r$ then the RHS of (2) is at most 1. This is because

$$\frac{r \cdot c(r)}{(r-y) \cdot c(r) + y \cdot c(y)} \leq \frac{r \cdot c(r)}{(r-y) \cdot c(r) + y \cdot c(r)} \leq 1,$$

where the first inequality uses that $c(x)$ is monotone and the denominator is non-negative.

So from now we assume $y \leq r$. After substituting $c(x) = ax + b$, the RHS of (2) simplifies to:

$$\frac{ar^2 + br}{ar^2 + br - yar - yb + yb + ay^2} = \frac{ar^2 + br}{ar^2 - yar + ay^2 + br} = 1 + \frac{yar - ay^2}{ar^2 - yar + ay^2 + br}.$$

Now let's make a quick observation: if we want to make this as big as possible (to find the worst possible example), we want to set $b = 0$, because increasing b only increases the denominator (and not the numerator). Once we set $b = 0$, note that the a cancels in both the numerator and denominator, and we can take some derivatives to see that $y = r/2$ maximizes the term. Plugging back in we get $1 + \frac{r^2/2 - r^2/4}{r^2 - r^2/2 + r^2/4} = 4/3$.

So we already have an example with PoA 4/3, and the above proof shows that we can never do worse with linear cost functions. □

3 Why does Pigou Example give the worst graph?

We prove that the Pigou example gives us a bound on the price of anarchy of all graphs.

Proposition 3. *For all classes of cost functions $\mathcal{C}$ that contains all constant functions, the example with the worst possible price of anarchy, over* all graphs*, and* all routing demands *with cost functions in $\mathcal{C}$ is a Pigou example.*

It says that actually we don't need to reason about arbitrary graphs, etc. to find the worst possible setting, we only need to consider Pigou examples, which are much easier to reason about. The proof of this proposition will need the following lemma which intuitively says that if we fix the costs of edges to be given by an equilibrium flow, then every other flow that meets the demand pays a cost at least that of the equilibrium flow.

Lemma 4 (Variational Inequality)**.** *Let f, g be two flows for demand $\vec{r}$. Moreover, let f be an equilibrium flow. Then*

$$\sum_e f_e \cdot c_e(f_e) \leq \sum_e g_e \cdot c_e(f_e) \qquad \Longleftrightarrow \qquad \sum_e (g_e - f_e) \cdot c_e(f_e) \geq 0.$$

Proof. Consider demand r_{st} from s to t. Suppose L_{st} is the cost of the shortest path from s to t after we fix edge costs to be $c_e(f_e)$. We know from (1) that every path p from s to t (i.e., $p \in P_{st}$) with $f_p > 0$ has cost $c_p(f) = L_{st}$, and every other path $p \in P_{st}$ has to pay at least L_{st}. Since g to also sends r_{st} flow from s to t (i.e., $\sum_{p\in P_{st}} f_p = \sum_{p\in P_{st}} g_p = r_{st}$), we get

$$\begin{aligned}\sum_e g_e \cdot c_e(f_e) &= \sum_{p\in P_{st}} g_p \cdot c_p(f) \geq \sum_{p\in P_{st}} g_p \cdot L_{st} = r_{st} \cdot L_{st} \\ &= \sum_{p\in P_{st}} f_p \cdot L_{st} = \sum_{p\in P_{st}} f_p \cdot c_p(f) = \sum_e f_e \cdot c_e(f_e),\end{aligned}$$

which completes the proof. □

Now we prove Proposition 3.

Proof of Proposition 3. For a given a graph G, costs c, and demand r, let f^* denote the optimal flow and let f be an equilibrium flow. Now (2) for $r = f_e$ and $y = f_e^*$ tells us that

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

After rearranging, we can use this to lower bound the cost of the optimal flow as:

$$\sum_e f_e^* \cdot c_e(f_e^*) \quad \geq \quad \sum_e \left(\frac{f_e \cdot c_e(f_e)}{\alpha(\mathcal{C})} - (f_e - f_e^*) \cdot c_e(f_e)\right) \quad \geq \quad \sum_e \frac{f_e \cdot c_e(f_e)}{\alpha(\mathcal{C})},$$

where the last inequality uses Lemma 4 for $g = f^*$. □

Finally, putting Lemma 2 and Proposition 3 together proves Theorem 1.

Note that the above also provides a bound on how bad Braess' paradox can be: Let G be the original graph (before adding edges) and G' be the graph after some edges are added. Let $c_{eq}(G)$ denotes the cost at the worst possible equilibrium for graph G, and $c_{opt}(G)$ denote the cost of optimally routing. Then by Theorem 1, we know that $c_{eq}(G') \leq 4/3 \cdot c_{opt}(G')$. We also know that $c_{opt}(G') \leq c_{opt}(G)$ (this is simply because G' has additional edges that we're free to ignore if we want). Finally, we have $c_{opt}(G) \leq c_{eq}(G)$ simply because c_{opt} is the cost of the best solution, and c_{eq} is the cost of some solution. So chaining everything together we get that $c_{eq}(G') \leq 4/3 \cdot c_{opt}(G') \leq 4/3 \cdot c_{opt}(G) \leq 4/3 \cdot c_{eq}(G)$, meaning that the equilibrium can't get worse by more than a factor of 4/3 by adding edges.

4 Resource Augmentation Bounds

In the rest of lecture, we'll prove a related result on what's called *resource augmentation*. Here, we'll ask questions like "what's better, routing flow f meeting demand $\vec{r}$ in equilibrium, or routing flow g meeting demand $2\vec{r}$ optimally?" You can alternatively view this as making the following choice as a network administrator to improve traffic flow:

- Route flow centrally.
- Upgrade all links so that $c_{new}(x) = c_{old}(x/2)/2$ (mathematically this is the same as dividing all demanded flow by 2).

It turns out that the second option is *always* better for any example, even when the costs are not affine functions (but are non-decreasing)!

Lemma 5. *Consider any equilibrium flow f in network G with costs c with $\vec{r}$ demanded flow. Also, consider the optimal flow g for the same G, c, but with $2\vec{r}$ demanded flow. Then*

$$\sum_e f_e \cdot c_e(f_e) \leq \sum_e g_e \cdot c_e(g_e).$$

Proof. Consider any equilibrium flow f in the former case, and let f_e denote the flow sent along edge e. Consider any optimal flow g in the second case, and let g_e denote the flow sent along edge e. Here's the high-level idea: some edges will have less flow than in g than they did in f. These edges might be super cheap (even zero cost), and this flow might get routed basically for free. However, these edges together cannot possibly carry very much flow (in particular, they can't carry more than the original $\vec{r}$). The remaining flow must go on edges that are at least as expensive in g as they are in f, and therefore the total cost is at least what it was in f. Now to make it formal:

We'll consider an intermediate graph, G', where the the cost functions are $c'_e(x) = c_e(f_e)$, and the demand is still $2\vec{r}$. Observe that the optimal flow in G' has cost at least twice $\sum_e f_e c_e(f_e)$. Why? This is using the variational inequality (Lemma 4) as $g/2$ exactly meets demand $\vec{r}$:

$$\sum_e f_e \cdot c_e(f_e) \leq \sum_e g_e/2 \cdot c_e(f_e).$$

Okay, but the problem is that we used these make-believe edge costs, and we have no idea how they compare to the real edge costs. So we need to show that actually g doesn't do much better when we substitute the real costs for these make-believe costs.

Now, we want to try and show that switching from the "fake costs" $c'_e(x)$ to the real costs $c_e(x)$ doesn't save us more than L_{st} for any s-t path. There are two kinds of edges to consider: those for which $g_e \geq f_e$, and those for which $g_e < f_e$. In G', we estimate the cost of g as:

$$\begin{aligned} 2\sum_e f_e c_e(f_e) \leq \sum_{e,f_e} g_e c_e(f_e) \quad &= \quad \sum_{e,f_e \leq g_e} g_e c_e(f_e) + \sum_{e,f_e > g_e} g_e c_e(f_e) \\ &\leq \sum_{e,f_e \leq g_e} g_e c_e(g_e) + \sum_e f_e c_e(f_e) \\ &\leq \sum_{e,f_e \leq g_e} g_e c_e(g_e) + \sum_e f_e c_e(f_e) + \sum_{e,f_e > g_e} g_e c_e(g_e). \end{aligned}$$

The first inequality is exactly what we proved in the first paragraph: that when using the "fake costs" $c_e(f_e)$ instead of $c_e(g_e)$, the cost of g is at least twice the cost of f. The second inequality makes two observations. The first is that when $f_e \leq g_e$, $c_e(f_e) \leq c_e(g_e)$ because $c_e(\cdot)$ is monotone non-decreasing. The second is that whenever $g_e < f_e$, we certainly have $g_e c_e(f_e) \leq f_e c_e(f_e)$. The last inequality only adds another non-negative sum where we sum over all e (instead of just e such that $f_e > g_e$).

Finally, we can conclude the theorem statement immediately from rearranging the first and last inequality to get:

$$\sum_e f_e c_e(f_e) \leq \sum_e g_e c_e(g_e).$$

Again, the high-level takeaway from the proof is the following: there are some edges where $g_e \geq f_e$. On these edges, it's at least as expensive to send flow in g as it is in f, which means g is expensive. On the other hand, there might be some edges where $g_e < f_e$, and these edges might be very cheap. But because these edges have so little flow on them, they can't actually be routing very much of the total flow. □

The high-level takeaway from this entire section is that price of anarchy bounds can be used to guide design and not just prove approximation guarantees.

Notes

This lecture is based on COS 445 notes from Princeton University [2].

References

[1] Algorithmic Game Theory. Nisan, Roughgarden, Tardos, Vazirani (eds.), Cambridge University Press 2007.

[2] https://www.cs.princeton.edu/∼smattw/Teaching/cos445sp19.htm