CS 6840 Algorithmic Game Theory January 29, 2020

Lecture 4: Price of Anarchy for Selfish Routing

Instructor: Eva Tardos *Scribe: Evan Hansen, Jiazhen Tan*

1 Recap

For the definition of the problem, see the past lecture notes. Recall that we required the delay functions d_e to be continuous and monotonically increasing, and, although most of these results still hold, we will also assume they are continuously differentiable and that $x \mapsto xd_e(x)$ is always convex.

Under these assumptions, we showed that

Theorem 1. *If the condition*

$$\sum_{e \in P} d_e(f(e)) \leq \sum_{e \in Q} d_e(f(e)) \tag{1}$$

holds for all P, Q with $f_P \neq 0$ where P and Q are both $s_i \rightsquigarrow t_i$ paths for every s_i, t_i then the flow f is a Nash Equilibrium.

Definition 1. We defined the social cost of a given flow f as

$$SC(f) = \sum_e f(e)d_e(f(e)). \tag{2}$$

We would like to know when a given flow is optimal; we also showed

Theorem 2. *If*

$$\sum_{e \in P} d_e(f(e)) + f(e)d'_e(f(e)) \leq \sum_{e \in Q} d_e(f(e)) + f(e)d'_e(f(e)) \tag{3}$$

holds with the same qualifications on P, Q as in Theorem 1 then the flow is optimal.

Recall also

Definition 2. The price of anarchy is defined as

$$\max \frac{SC(f)}{SC(f^*)} \tag{4}$$

where the maximum is taken over all Nash equilibrium flows f and feasible flows f^*.

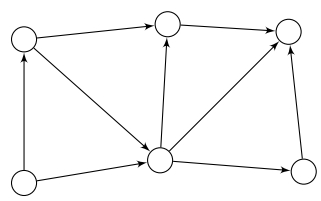
2 Argument

Our goal is to be able to compare the social cost of a Nash equilibrium solution to the optimal (minimal) social cost possible. To do that, we are interested in bounding the PoA (Price of Anarchy). What we will prove is

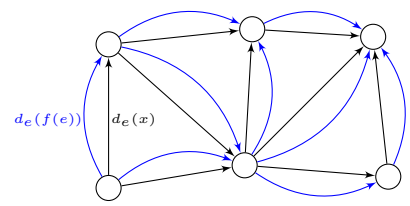
Theorem 3. *For any class of delay functions $\mathcal{D}$ satisfying the assumptions we have made for delay functions (continuous and monotone increasing) plus the condition that all constant functions are in the class $\mathcal{D}$, the maximum price of anarchy achievable in any network is achieved in a Pigou-like network. A Pigou-like network is one with only two nodes, two edges, and where one edge has a constant function as its delay function.*

For the sake of interpretability, we prove this only in the case where every function in $\mathcal{D}$ is also assumed to be continuously differentiable and where for any $d \in \mathcal{D}$, $x \mapsto xd(x)$ is a concave function.

Proof. We first consider an arbitrary network, G, which has associated s_i, t_i pairs, $r_i > 0$ demands, and d_e delay functions.



An example of G



Create $\bar{G}$ by adding edges $\bar{e}$

Fix a Nash equilibrium flow f. Create a new graph $\overline{G}$, which is the same as G, but for every edge $e = (u, v)$ we add a new edge $\overline{e} = (u, v)$ which has associated delay function $d_{\overline{e}}(x) = d_e(f(e))$. It is important to note that this is a constant function since we fixed f at the beginning of the analysis.

Also let f^* denote the optimal flow in the original network G.

Question 1. *How does the optimum change, and what is the Nash flow $\overline{f}$ in $\overline{G}$?*

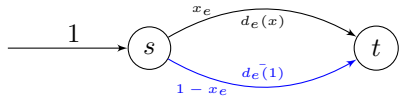
The Nash flow is still f. To see this, look at the equation in Theorem 1. Since all new edges that were added have equal delay (at flow f) to the edge that was already between those two nodes, there is no way that this condition can become violated if it was not already so under f. So $\overline{f} = f$.

Further, the optimum flow can only improve. This is easy to see because f^* is still a feasible flow in $\overline{G}$.

Importantly, this gives us the important conclusion that the Price of Anarchy in G is no greater than that in $\overline{G}$.

Question 2. *Can we get a good description of the optimal flow $\overline{f^*}$ in $\overline{G}$?*

Example 1. *Consider a Pigou-like network G with just two nodes s, t and one edge $e = (s, t)$ with delay function $d_e(x) = x$. Suppose the demand from $s \to t$ is 1. The Nash flow f is the only possible flow which sends 1 unit of traffic across e. Then form $\overline{G}$.*



Because the delay under f on e was 1 we form a new edge $\overline{e} = (s, t)$ with delay function $= x \mapsto 1$. If we want to optimize this, we use single-variable calculus. Suppose that x_e is the flow that we decide to send

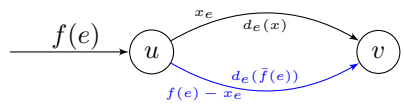
on e. Then the social cost will be

$$x_e d_e(x_e) + (1 - x_e) d_e(1) = x_e^2 - x_e + 1. \tag{5}$$

We differentiate and set equal to zero, getting

$$2x - 1 = 0 \implies x = 1/2. \tag{6}$$

This example is not particularly useful by itself, but illustrates an important point. For any edge $e = (u, v)$ in our (presumably much more complex) graph G, the subgraph of $\overline{G}$ consisting of just the vertices u, v and edges $e, \overline{e}$ is nearly the same as the example.



One way we could try to find the optimum in $\overline{G}$ is by performing local optimization from the Nash flow f, only changing the flow like we did in the example: pushing some flow from the original edge onto the new edge. To do this we determine an amount of flow x_e to place on e, and push the $f(e) - x_e$ amount of flow we are removing onto $\overline{e}$.

If we were to actually carry this out, we would be optimizing a formula similar to Equation 5 for each edge, namely

$$x_e d_e(x_e) + (f(e) - x_e) d_e(f(e)). \tag{7}$$

Because we required d_e to be continuously differentiable, our solution must have the derivative of this expression equal to 0.

This gives the property

$$d_e(x_e) + x_e d_e'(x_e) = d_e(f(e)) \tag{8}$$

which will be satisfied after this local optimization.

Now look again at the conditions for optimality and Nash equilibrium, copied here for convenience.

Condition for optimality

$$\sum_{e \in P} d_e(f(e)) + f(e) d_e'(f(e)) \leq \sum_{e \in Q} d_e(f(e)) + f(e) d_e'(f(e)) \tag{9}$$

Condition for Nash Equilibrium

$$\sum_{e \in P} d_e(f(e)) \leq \sum_{e \in Q} d_e(f(e)) \tag{10}$$

We will be showing that this local optimization gets the optimal flow $\overline{f^*}$, so we begin using that notation to distinguish it from f.

We already know that the bottom condition holds for every e in the original graph (with the flow f).

For any edge e that was in the original graph G, we can use Equation 8 to show that its term (LHS below) in a summand in (9) satisfy

$$d_e(\overline{f^*}(e)) + \overline{f^*}(e) d_e'(\overline{f^*}(e)) = d_e(f(e)).$$

For each of the edges of the form $\overline{e}$, its delay is fixed at $d_e(f(e))$, and its derivative is 0. Hence its contribution to a term in the summand will be $d_e(f(e))$.

So now suppose that $\overline{f^*}_P \neq 0$ and is an $s_i \rightsquigarrow t_i$ path.

Then the following are equivalent

$$\begin{aligned} \sum_{e \in P} d_e(\overline{f^*}(e)) + \overline{f^*}(e) d'_e(\overline{f^*}(e)) &\leq \sum_{e \in Q} d_e(\overline{f^*}(e)) + \overline{f^*}(e) d'_e(\overline{f^*}(e)) \\ \sum_{e \in P} d_e(f(e)) &\leq \sum_{e \in Q} d_e(f(e)). \end{aligned} \tag{11}$$

But the second line is the same as Equation 10, which we already knew to hold. Hence we have answered Question 2: perform this kind of local optimization starting at the Nash Equilibrium f and you will get an optimal flow.

Now you might think we just swept something under the rug: Equation 10 only contains paths in G, while 11 contains paths in $\overline{G}$, including edges of the form $\overline{e}$, which are not in Equation 10. But this is not consequential, because for a path P which contains an edge $\overline{e}$, the value of the sum is the same if we replace P with P', where in P' we have replaced every edge of the form $\overline{e}$ with its counterpart $\overline{e}$. Then we can apply Equation 10 and it is valid.

Question 3. *How does this prove Theorem 3?*

We claim the following arithmetical fact is true without proof:

Lemma 1. *For all $x, y, z, v \in \mathbb{R}^{\geq 0}$, $z, v \neq 0$,*

$$\frac{x+y}{z+v} \leq \max\left(\frac{x}{z}, \frac{y}{v}\right). \tag{12}$$

Then each pair of nodes u, v and edges $e, \overline{e}$ we have been considering for local optimization is a Pigou like graph. Its Nash Equilibrium, when considered as its own network with the source-sink pair u, v, will be to put all the flow on e, the same as its flow was in the Nash Equilibrium f in G. Its optimum is the same as the local optimization that we considered above, and hence is the same as it was in the global case, as we showed above. Then simply apply Lemma 1 iteratively to prove the theorem.

$$\begin{aligned} PoA &= \frac{\sum_e f(e) d_e(f(e))}{\sum_e \left(x_e d_e(x_e) + (f(e) - x_e) d_e(f(e))\right)} \\ &\leq \max_e \frac{f(e) d_e(f(e))}{x_e d_e(x_e) + (f(e) - x_e) d_e(f(e))} \end{aligned} \tag{13}$$

where here the PoA is of $\overline{G}$. The theorem holds for G because we already concluded earlier that the PoA in $\overline{G}$ is at least as large as it is in G.

■