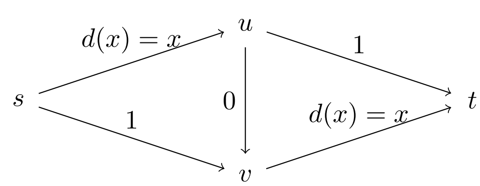
CS 6840 Algorithmic Game Theory January 24, 2020

Lecture 2: Selfish Routing

Instructor: Eva Tardos *Scribe: Yue Guo, Tegan Ellen Wilson*

*Note: This lecture comes from Tim Roughgarden's book, "Twenty Lectures on Algorithmic Game Theory", lecture 11. He provides an alternate proof for a portion of today's lecture that makes fewer assumptions.

Recall the Braess' paradox introduced in the last lecture:



Assuming people selfishly want to optimize for their own total delay, then with the 0 delay edge (u, v) people have a total delay of 2, while without the edge, people have a delay of 1.5.

The Braess' paradox is an instance of a *routing game*. A routing game is defined by a directed graph $G = (V, E)$, a set of source-sink pairs $s_i, t_i \in V$ with demand r_i between them, and a delay function $d_e(x)$ for every edge $e \in E$ which denotes the delay on an edge $e \in E$ when the traffic on the edge is x. Here we make several assumptions on the delay functions:

- x is always a positive real number
- $d_e(x)$ is monotonously increasing
- $d_e(x)$ is continuous
- $d_e(x)$ is differentiable
- $d_e(x)$ is convex

We make 2 assumptions for convenience: that d_e is differentiable and convex. However, they are not necessary–Tim Roughgarden's lecture does not make these assumptions. Additionally, one could make the assumption that the delay functions are all strictly increasing.

We assume that the several source-sink pairs are all distinct, that for any $i \neq j$, either $s_i \neq s_j$ or $t_i \neq t_j$. Each pair has a demand value $r_i \geq 0$, which indicates the amount of traffic which need to flow from s_i to t_i.

Defining a solution: For each path P from s_i to t_i for some i, we use f_P to denote the flow of traffic using this path from s_i to t_i. We say a flow is a feasible solution if for all i,

$$r_i = \sum_{P: s_i \to t_i} f_P$$

*Note that because the number of total $s_i \to t_i$ paths in a graph is exponential, this definition is not usable for computation. Thus, we define an alternate definition:

Consider the traffic goes through an edge e:

$$f(e) = \sum_{P:e\in P} f_P$$

Instead of keeping track of how much flow goes along each path, we instead only keep track of how much flow goes along each edge in the network. However, if we have multiple source-sink pairs, we cannot distinguish if we are truly flowing enough traffic from each source to their respective sinks, as the definition "forgets" where traffic came from.

Objective functions: 1) We could define the objective function selfishly, to selfishly minimize the total delay for each person. In other words, no single person wants to switch to another path as switching only makes its own delay worse. We formalize the objective as the following condition:

$$\sum_{e\in P} d_e(f(e)) \leq \sum_{e\in Q} d_e(f(e)), \quad \forall s_i \to t_i \text{ path } Q$$

If true, we call this a Nash equilibrium.[1]

Claim 1. *If the condition 1) fails, then there exists some $\epsilon > 0$, such that ϵ flow can improve its delay by switching to another path Q.*

Proof. Suppose that there exists some path Q from s_i to t_i such that $\sum_{e\in P} d_e(f(e)) - \sum_{e\in Q} d_e(f(e)) = \delta > 0$. When a tiny piece of flow switches from path P to path Q, the total delay on path Q will go up a little bit. Since $d_e(x)$ on every edge is continuous and differentiable, we can find a small enough $\epsilon > 0$ such that $\sum_{e\in Q} d_e(f(e+\epsilon)) - \sum_{e\in Q} d_e(f(e)) < \delta$. By switching to path Q, this ϵ flow improves its delay as $\sum_{e\in Q} d_e(f(e+\epsilon)) < \sum_{e\in P} d_e(f(e))$. ■

The converse of the claim also holds:

Claim 2. *If anyone wants to switch to another path Q, then the condition must be violated.*

Proof. Because of monotonic increasing. If there exists a path Q that ϵ flow can switch to and get better delay, then because delay functions are all monotonic increasing, it must be the case that the delay on path Q was already less than the delay on path P, since delay can only increase when the ϵ flow switches to path Q. ■

Questions. Till now we are considering the stable solution in which everyone only cares about whether the solution is optimal for itself or not. What will happen if people optimize the overall delay? Actually we are going to compare the selfish solution with the "globally optimal" solution. To discuss the solution globally, we need first find a way to describe the social welfare. Here are some possible definitions (very negotiable):

1. Total delay of everyone, $\sum_P f_P \sum_{e\in P} d_e(f(e))$. This is often called the *social cost* of the flow, or $SC(f)$. This is the most common definition, but may not always be "fair" in some sense. (A very tiny fraction may have to go to the moon and back)
2. Maximum delay over everyone. It might not be good enough either, as the maximum delay (e.g. someone wants to travel from New York to California) might "cover" some more common cases with lower delay (e.g. someone wants to go from the Cornell campus to its home in Ithaca).

[1] Sort of... It's more an extension of Nash equilibrium to a continuous instance, and works for the purposes of this lecture.

There are more alternative definitions to be explored in our homework. Here we are going to use the social cost (SC) function to measure how good the solution is globally. As mentioned before, the number of paths might be exponentially large which makes the calculation of social cost infeasible, we transform it into an equivalent definition:

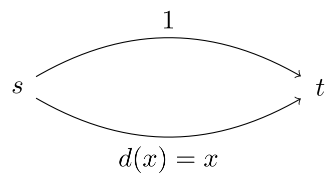
$$SC(f) = \sum_P f_P \sum_{e \in P} d_e(f(e)) = \sum_e d_e(f(e)) \sum_{P: e \in P} f_P = \sum_e f(e) d_e(f(e))$$

We would like to bound the ratio of the Nash Equilibrium solution to the solution with lowest social cost, i.e., measure how the efficiency of the whole system will downgrade when everyone is performing selfish.

Definition 1 (Price of Anarchy)**.** *Let f denote the flow in some Nash Equilibrium, and f^* denote the flow that minimizes the social cost, we define the* price of anarchy *(PoA) as*

$$\max_{f, f^*} \frac{\sum_e f(e) d_e(f(e))}{\sum_e f^*(e) d_e(f^*(e))}$$

Ex: due to Pigeau



Everyone will take the $d(x) = x$ path, which leads to a social cost of 1. But the solution with optimal social cost sends half the people along each path, and gives social cost of $3/4$. This gives the same ratio as Braess' paradox!

Here comes the goal theorem:

Theorem 1. *For any routing game with delay functions that are monotone increasing, continuous, differentiable, and convex, the Price of Anarchy is the worst case in a network like Pigeau's: one edge has constant delay, and the other has a delay function that is some function of x.*

Here we are assuming the function f be differentiable and convex for convenience. To see a proof without these assumptions, check Tim Roughgarden's book, lecture 11.