3.1 Equilibrium Flows and Social Cost Optimization for Selfish Routing

Recall the set-up for a selfish routing problem on a directed network $G = (V, E)$. We are given a collection of distinct source-sink pairs $(s_i, t_i)$ with associated demands $r_i$, along with edge-specific non-negative delay functions $d_e : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$. A valid flow on this network is an assignment of non-negative values $f_P \geq 0$ to the paths of $G$ such that

$$\sum_{P: s_i \rightarrow t_i} f_P = r_i \quad \text{for all } i,$$

summing over all $s_i \rightarrow t_i$ paths $P$. Further, we define the flow over an edge $e$ to be

$$f(e) = \sum_{P \ni e} f_P,$$

and the delay through a path $P$ to be

$$\sum_{e \in P} d_e(f(e)).$$

We say that a flow $f$ is a (Nash) equilibrium flow if $f_P > 0$ for an $s_i \rightarrow t_i$ path $P$ only when it is a minimum delay $s_i \rightarrow t_i$ path. That is, for each other $s_i \rightarrow t_i$ path $Q$, we have

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

Finally, we define the social cost of a flow $f$ as

$$SC(f) = \sum_e f(e)d_e(f(e)),$$

i.e. the total delay experienced by the routed agents. For a moment, we relax our definition of social cost to

$$SC(f) = \sum_e c(e)(f(e)),$$

allowing edge-specific costs $c_e : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$ which are continuous, differentiable, and monotonically increasing. Let’s examine this function from an optimization perspective.

**Proposition 3.1** If a flow $f$ minimizes social cost as defined above, then, for each path $P$ with $f_P > 0$, we have

$$\sum_{e \in P} c'_e(f(e)) \leq \sum_{e \in Q} c'_e(f(e))$$

for all paths $Q$ between the same $s_i \rightarrow t_i$ pair.
Proof: Suppose for the sake of contradiction that \( f_P > 0 \) and

\[
\sum_{e \in P} c'_e(f(e)) > \sum_{e \in Q} c'_e(f(e))
\]

for such a path \( Q \). Then, we would be able to move a small amount of traffic \( \delta > 0 \) from \( P \) to \( Q \) and decrease the social cost. Indeed, for each \( e \in Q \setminus P \),

\[
c_e(f(e) + \delta) = c_e(f(e)) + \delta c'_e(f(e)) + o(\delta)
\]

and, for each \( e \in P \setminus Q \),

\[
c_e(f(e) - \delta) = c_e(f(e)) - \delta c'_e(f(e)) + o(\delta),
\]

as \( \delta \to 0 \). Thus, our adjustment to \( f \) would result in the following negative change in social cost,

\[
\delta \left( \sum_{e \in Q \setminus P} c'_e(f(e)) - \sum_{e \in P \setminus Q} c'_e(f(e)) \right) + o(\delta)
\]

\[
= \delta \left( \sum_{e \in Q} c'_e(f(e)) - \sum_{e \in P} c'_e(f(e)) \right) + o(\delta) < 0,
\]

for sufficiently small \( \delta > 0 \). We have contradicted our initial hypothesis, proving the proposition.

Claim 3.2 The converse is true, assuming that each \( c_e \) is convex.

Proof: In this case, we are dealing with a convex program, since \( SC \) is convex as a sum of convex functions and our constraints are linear. Thus, any local minimum of \( SC \) is a global minimum. It turns out that the relevant “local moves” to assess local optimality are of the previous form, routing an infinitesimal amount of flow from a path with positive flow to another between the same \( s_i \to t_i \) pair. By the same logic as before, all moves of this form will increase social cost, so flows satisfying our condition are indeed local (and thus global) minimizers of social cost.

Now, let’s examine this convexity assumption with the standard costs \( c_e(x) = xd_e(x) \). Differentiating, we find that

\[
c'_e(x) = d_e(x) + xd'_e(x),
\]

which intuitively makes sense, as \( d_e(x) \) is a “selfish” term, quantifying the delay experienced by an individual if they switch to a path using edge \( e \), and \( xd'_e(x) \) multiplies the current traffic times the extra delay per unit traffic to get an “externality” term quantifying the additional delay experienced by others. A real function is convex if and only if its derivative is monotonically increasing, so our convexity assumption is rather mild, only requiring that \( d'_e(x) \) not decrease so fast as to counteract multiplication by \( x \). Under this assumption, we have a nice corollary.

Corollary 3.3 If each \( c_e \) is convex, then a global minimizer of social cost can be computed in polynomial time.

Since we are minimizing a convex objective function over a convex set, this problem is indeed computationally tractable.\(^1\)

---

\(^1\)Recall that \( g(x) = o(x) \) as \( x \to 0 \) if \( g(x)/x \to 0 \) as \( x \to 0 \). We can further assume that the remaining terms are \( O(\delta^2) \) with some mild assumptions, but these aren’t necessary.

\(^2\)Alternatively, one can derive the condition of Proposition 3.1 from the Karush–Kuhn–Tucker (KKT) conditions.

\(^3\)To be a bit more precise, we should make a change of variables and consider the flow at each edge associated with each \( s_i \to t_i \) pair (to avoid dealing with exponentially many variables). Then, this problem can be tackled (up to some approximation error) in polynomial time using, say, the ellipsoid method.
**Corollary 3.4** If each $d_e$ is continuous and monotonically increasing, then a Nash equilibrium for the selfish routing problem can be computed in polynomial time.

**Proof:** We can reduce this to the previous case by introducing costs

$$\tau_e(x) = \int_0^x d_e(y)dy.$$  

These new costs are convex, since each $d_e$ is monotonically increasing. Thus, our new objective $\sum_e \tau_e(f(e))$ is also convex, and we can find a minimizing flow in polynomial time. Its optimizer must satisfy the condition of Proposition 3.1 which, by design, is identical to that for Nash equilibrium. \[\square\]

Next class, we will work towards the following theorem.

**Theorem 3.5** Let $D$ be a class of possible delay functions containing all non-negative constants, such that each $d \in D$ is monotonically increasing, continuous, and continuously differentiable, with $x \mapsto xd(x)$ convex. Then for any input to the routing problem using delay functions from $D$ (i.e. any graph $G$, input-output pairs $s_i, t_i$, demands $r_i$, and assignments $e \mapsto d_e \in D$), the price of anarchy (PoA) for this instance is no greater than that for the following two edge graph $G_0$,

![Graph](image)

where the constant and delay function $d$ are chosen from $D$ to maximize PoA. That is,

$$\max_{f \text{ Nash}, f^* \text{ on } G} \frac{SC_G(f)}{SC_G(f^*)} \leq \max_{f \text{ Nash}, f^* \text{ on } G_0} \frac{SC_{G_0}(f)}{SC_{G_0}(f^*)},$$

(Identifying the graphs with their instances).