

# Routing games

The game is given by a directed graph  $G = (V, E)$ , a set of edge cost functions  $(c_e)_{e \in E}$ , and a finite number of players, indexed by  $k \in \{1, \dots, K\}$ .

- The cost function of an edge  $e$  is a function  $c_e : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ . It determines the cost of the edge given the total flow on that edge.
- A player  $k$  is given by: a source node  $s_k \in V$ , a destination node  $d_k \in V$ , and a total flow  $F_k$  (i.e. the total mass of traffic that this player is allocating). The action set of the player is the paths that connect  $s_k$  to  $d_k$ , denoted by  $\mathcal{P}_k$ .
- At iteration, each player  $k$  chooses a flow distribution  $f_k \in \mathbb{R}_+^{\mathcal{P}_k}$ , such that  $\sum_{p \in \mathcal{P}_k} f_{k,p} = F_k$ . The flow distributions of all players determine the edge flows as follows: for an edge  $e$ , the edge flow is

$$\phi_e = \sum_k \sum_{p \in \mathcal{P}_k : e \in p} f_{k,p}$$

Another way to write this is

$$\phi = \sum_k M^k f_k$$

where  $M^k$  is an incidence matrix for player  $k$ , defined as follows:  $M^k \in \mathbb{R}^{E \times \mathcal{P}_k}$ , such that

$$M_{e,p}^k = \begin{cases} 1 & \text{if } e \in p \\ 0 & \text{otherwise} \end{cases}$$

Once we have the edge flows, we can compute the edge costs simply by applying the edge cost functions. Let  $y$  be the vector in  $\mathbb{R}^E$  defined by

$$y_e = c_e(\phi_e)$$

Then we compute the path costs by summing the edge costs along the path. So for all  $k$ , and all  $p \in \mathcal{P}_k$ , the path cost is

$$\ell_p^k = \sum_{e \in p} y_e$$

so the path costs for player  $k$  can be written simply in terms of the incidence matrix

$$\ell^k = (M^k)^T y$$

To summarize, when we construct the graph, we need: the node set  $V$ , the edge set  $E$ , the edge cost functions  $c_e, e \in E$ , and the player description  $(s_k, d_k, F_k)$  for each  $k$ . From this, we can compute the paths  $\mathcal{P}_k$  and compute the incidence matrices  $M_k$ .

Then, at each iteration, each player chooses a path flow distribution  $f_k$ , which we use to compute:

1. the edge flows  $\phi$
2. the edge costs  $y$
3. the path costs  $\ell_k$  for each  $k$

Once this is done we reveal to each player  $k$  the path cost vector  $\ell_k$  and we start the next round.

# 1 Learning dynamics

We will consider learning dynamics of the form

$$x_k^{(t+1)} = \min_{x \in \Delta^{\mathcal{P}_k}} \langle x_k, \ell_k^{(t)} \rangle + \frac{1}{\eta_t^k} D_{\psi_k}(x_k, x_k^{(t)})$$

This update is an iteration of the mirror descent algorithm (Nemirovski and Yudin, 1983). Here, we have

- $x$  is the normalized flow distribution.  $\Delta^{\mathcal{P}_k}$  is the simplex of distributions over  $\mathcal{P}_k$ .
- $D_\psi$  is the Bregman divergence associated to the distance generating function  $\psi$ , which needs to be strongly convex.  $D_\psi$  is defined as

$$D_\psi(x, y) = \psi(x) - \psi(y) - \langle \nabla \psi(y), x - y \rangle$$

By convexity of  $\psi$ ,  $D_\psi$  is always non negative, and  $D_\psi(x, y) = 0 \Leftrightarrow x = y$ .

- $\eta_t$  is a learning rate, and can be thought of as a generalized step size. The smaller  $\eta_t$ , the closer  $x^{(t+1)}$  will be to  $x^{(t)}$  (since we put more weight on the Bregman divergence term)

One can prove that if  $\eta_t^k$  satisfies certain assumptions, the joint state of the system is guaranteed to converge to equilibrium. We would like to

1. Verify that the system indeed converges to equilibrium (or close to an equilibrium)
2. Estimate the player dynamics (i.e. their distance generating function and their learning rates). This could allow us to predict what the future flow distributions may be.

## 1.1 Solution in the KL divergence case

When  $D_\psi$  is the KL divergence,  $D_\psi(x, y) = \sum_i x_i \ln \frac{x_i}{y_i}$ , then there is a closed-form solution of the above minimization problem. The solution is given by

$$(x_k^{(t+1)})_i = \frac{e^{-\eta_t^k (\ell_k^{(t)})_i}}{\sum_j (x_k^{(t)})_j e^{-\eta_t^k (\ell_k^{(t)})_j}}$$

## 1.2 Estimating the player dynamics

We assume that each player has their own distance generating function  $\psi_k$  and their own sequence of learning rates  $\eta_t^k$ . We want to estimate these parameters. For now, let us assume that  $\psi_k$  is given, and we want to estimate  $\eta_k$ . At iteration  $t$ , we know the flow distribution  $x_k^{(t)}$ , the vector of losses  $\ell_k^{(t)}$ , so we can compute, for a given  $\eta$ , the solution to the minimization problem

$$\hat{x}^{(t+1)}(\eta) = \arg \min_{x_k \in \Delta^{\mathcal{P}_k}} \langle x_k, \ell_k^{(t)} \rangle + \frac{1}{\eta} D_{\psi_k}(x_k, x_k^{(t)})$$

Then we can argue that a good value of  $\eta$  is the one that minimizes the distance between the distribution predicted by the model,  $\hat{x}_k^{(t+1)}(\eta)$ , and the actual flow distribution  $x_k^{(t+1)}$ . So we would like to minimize

$$\hat{\eta}_t^k = \arg \min_{\eta \geq 0} D_{\psi_k}(\hat{x}_k^{(t+1)}(\eta), x_k^{(t+1)})$$

I believe that this problem is convex for any  $\psi$ . We should check that this is true.

Once we solve this first question, we can think about estimating the distance generating function. To simplify, we can assume that the distance generating function is a linear combination of a finite set of known distance generating functions. So we can write

$$\psi_k(x, y) = \langle \theta_k, \Psi(x, y) \rangle$$

where  $\Psi$  is a vector of known functions  $\Psi(x, y) = (\psi_1(x, y), \dots, \psi_M(x, y))^T$ .