

Lecture 2: February 6

Lecturer: Alex Fabrikant

Scribe: Rong Ge & Sushant Sachdeva

Disclaimer: These notes have not been subjected to the usual scrutiny applied to formal publications. They may be distributed outside this class only with the permission of the Instructor.

2.1 Selfish Source Routing

2.1.1 Definition and Example

A Selfish Source Routing game (called “Selfish Routing” in the literature) is a game played by n player on a graph G (See Figure 2.1). Each player i has a starting vertex s_i and a destination vertex t_i , and their strategies are choosing any path in graph G that connects s_i and t_i . Each edge e has a cost function $c_e(k_e)$, where k_e is a number of players that chooses this edge. The cost of a player i is the sum of edge costs along the path it chooses.

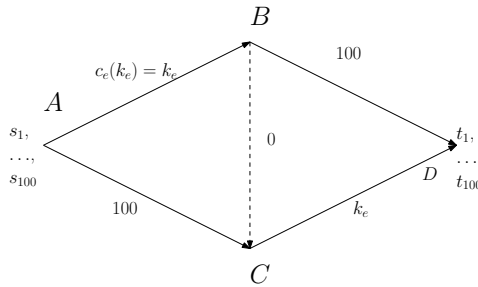


Figure 2.1: A Selfish Routing Game

The player set of the game is $N = \{1, 2, \dots, n\}$, strategy sets are $S_i = \{P \mid P \text{ is a path from } s_i \text{ to } t_i\}$. When players choose strategy $S = (P_1, P_2, \dots, P_n)$, $k_e(S) = |\{P_i \mid e \in P_i\}|$, $c_i(S) = \sum_{e \in P_i} c_e(k_e(S))$.

In this section we assume the edge cost functions are all affine functions. That is, $c_e = a_e k_e + b_e$. We further assume that a_e and b_e are non negative because the intuition of this game comes from route selecting. k_e is the number of people using the edge, and cost reflects the time needed to get through the edge, which should be nonnegative and nondecreasing when k_e increases.

In the example shown in Figure 2.1, the game has 100 players $\{1, 2, \dots, 100\}$. All players share the same starting vertex (A) and destination vertex (D). Their strategy sets are $\{\{AB, BD\}, \{AC, CD\}\}$, so that they can choose between the upper path (ABD) and the lower path (ACD). The cost functions are written near the edge, $c_{AB} = k_{AB}$, $c_{BD} = 100$, $c_{AC} = 100$, $c_{CD} = k_{CD}$. It's easy to see that there's a pure Nash Equilibrium in this game: 50 players choose the upper path and the other 50 players choose the lower path. The cost for each player $c_i = \sum_{e \in P_i} c_e(k_e) = 150$. If any player changes his choice, then his cost will become $c_i = 151$, which is worse. The social cost of this Nash Equilibrium (in this game social cost is the sum of individual costs) is 15000, and it is easy to see that this is optimal.

Now consider the edge BC is added to the graph and its cost $c_{BC} = 0$. The previous strategy is no longer a Nash Equilibrium because the players choosing the upper path will have incentive to switch to the

path $ABCD$. After they all switched, the players choosing the lower path will also have incentive to switch to the path $ABCD$ because segment ABC is shorter than edge AC . So in the new pure Nash Equilibrium, all players choose the path $ABCD$, and their costs are $c_i = 200$, which is worse than the cost before edge BC is added.

The optimal social cost of the new game is achieved by following the Nash Equilibrium of the previous game (before BC is added), the price of anarchy of this game is $\frac{20000}{15000} = \frac{4}{3}$.

2.1.2 Upper Bound on Price of Anarchy

In this subsection we prove that selfish routing game with affine costs has an upper bound on price of anarchy.

Proof:

Assume G is a routing game that has the worst price of anarchy, let $f = (P_1, P_2, \dots, P_n)$ is a pure Nash Equilibrium, $f^* = (P_1^*, P_2^*, \dots, P_n^*)$ is the Social Optimal.

By properties of pure Nash Equilibrium, when player i switch his path from P_i to P_i^* , his cost cannot increase, so

$$\sum_{e \in P_i} (a_e k_e + b_e) \leq \sum_{e \in P_i^*} (a_e (k_e + 1) + b_e) \quad (2.1)$$

Now we bound the social cost of f ,

$$\begin{aligned} C(f) &= \sum_{i=1}^n \sum_{e \in P_i} (a_e k_e + b_e) \\ &\leq \sum_{i=1}^n \sum_{e \in P_i^*} (a_e (k_e + 1) + b_e) \\ &\leq \sum_{i=1}^n \sum_{e \in P_i^*} (a_e (k_e + k_e^*) + b_e) \\ &= \sum_{e \in E} (a_e (k_e + k_e^*) + b_e) k_e^* \\ &= \sum_{e \in E} (a_e k_e^* + b_e) k_e^* + \sum_{e \in E} a_e k_e k_e^* \end{aligned}$$

Here, the first term is exactly the social cost of optimal solution, and we apply Cauchy-Schwartz on the second term,

$$\begin{aligned} C(f) &\leq \sum_{e \in E} (a_e k_e^* + b_e) k_e^* + \sum_{e \in E} a_e k_e k_e^* \\ &= C(OPT) + \sum_{e \in E} \sqrt{a_e} k_e \cdot \sqrt{a_e} k_e^* \\ &\leq C(OPT) + \sqrt{\sum_{e \in E} a_e k_e^2} \sqrt{\sum_{e \in E} a_e k_e^{*2}} \\ &\leq C(OPT) + \sqrt{C(f)} \sqrt{C(OPT)} \end{aligned}$$

So we get

$$C(f) \leq C(f^*) + \sqrt{C(f)C(f^*)} \quad (2.2)$$

Assume price of anarchy is p , that is $\frac{C(f)}{C^*(f)} = p$, apply this to 2.2

$$p - 1 \leq \sqrt{p}$$

Solve for p , the result is $p \leq \frac{3+\sqrt{5}}{2}$. ■

2.2 Congestion Games

The above potential argument generalizes to a large family of games called *Congestion Games*. Consider the following general game where we have:

- N - Number of players
- E - A set of resources that the players will 'share'
- $\{\mathcal{E}_i \subseteq E\}_i$ - The set of strategies for each player
- $\forall e \in E, c_e : \mathbf{N} \rightarrow \mathbf{N}$ - A function for each edge that would map the load on that edge to the cost for that edge (payoff would be $-c_e$).
- $\forall i \in [N], c_i(\vec{s}) = \sum_{e \in s_i} c_e(k_e)$ - The cost for each player being the sum of all the costs for the edges included in the player's strategy.

For each edge, we define the following potential function, $\psi_e = \sum_{i=1}^{k_e} c_e(i)$ and the overall potential function to be $\Psi = \sum_{e \in E} \psi_e$. Now, it is easy to see that when a player, say j decides to change his strategy, from s_j to s'_j . The change in the potential function would be exactly the same as the change in the cost for player j i.e. $\Psi(s_{-j}, s'_j) - \Psi(s) = c_j(s_{-j}, s'_j) - c_j(s)$.

Since Ψ is defined over a finite domain, it has a minimum. At the absolute minimum, the value of Ψ cannot decrease if any player chooses to unilaterally change his/her strategy. Thus, using the above fact, no change in strategy for a single player would lead to a lower cost for the player. Thus, the minimum is a pure Nash Equilibrium!

Note that we needed only the fact that it was a local minimum, so there could be many pure Nash Equilibria. Also, the construction of the potential function relied crucially on the fact that the cost of an edge depended only on the load on the edge and not the individual players using that edge.

FACT : It may seem feasible that some games do not have the above structure but possibly still have a potential function as above, that permits you to deduce the existence of a pure Nash Equilibrium, but a result (citation????) states that any game that has a potential function with the property that the change in the potential function is exactly equal to the change in the cost for the player, must be a *Congestion game* with the above structure.

2.3 Another selfish network routing game

In this section, we consider a slightly different selfish network routing game. We are given a single source and a destination vertex, with several parallel edges between them. There are N players. Each player i has a traffic w_i that it wishes to route, all through one single edge. Each edge is associated with an *edge latency*,

l_e . So the strategy set for each player $\mathcal{E}_i = E$. The cost for each player is the product of edge latency of the edge that it picks and the total traffic through that edge.

$$c_i(s) = l_{e_i} \sum_{j:e_i=e_j} w_j$$

Since this is not a congestion game, we can't use a potential argument analogous to the above to prove that this game has a pure Nash Equilibrium. Nevertheless, we will show that this game does have one. Corresponding to each strategy profile, we associate a vector $L = (L_e)_{e \in E}$. Given a strategy profile, we calculate $l_e = l_e \sum_{i:e_i=e} w_i$ and sort these in descending order to create the vector L . We sort the L 's lexicographically.

Now, we claim that the first element in lexicographic order is a pure Nash Equilibrium. To see that, let us consider what happens when the strategy profile corresponds to the top element in the lexicographic order. For concreteness, assume that the element is 40, 10, 10, 5. Now, a player will clearly not consider moving to an edge that already has higher or equal load, so the only choices to consider are the edges that have a lower load than the current edge e that the player is using. If such a move actually leads to a decrease in the cost for the player, the final cost on the new edge e' should be lower than the cost with e , i.e. $l_e > l'_{e'}$. This would mean that L' comes before L in lexicographic ordering, which is a contradiction and hence the top element corresponds to a pure Nash Equilibrium.