Algorithmic Game Theory September 22, 2008

Lecture 3: Network Design

Lecturer: Sergei Vassilvitskii *Scribe:Etienne Vouga*

1 Overview

We will examine the network design problem, which features

- A directed graph $G = (V, E)$.
- A number of players. Each player i wants to go from some source vertex s_i to a sink vertex t_i.
- A cost c_e for each edge e, for which the edge can be bought. Several players can split the cost of an edge. Once an edge is bought, all players can use it.

A valid network is then any set of purchases that allows every player to travel from his source to his sink by travelling only along bought edges. If only one player wishes to cross a certain edge, naturally that player pays the full cost of the edge. If several players wish to cross the same edge, however, such as the middle edge in Figure (1), we must specify how they are to split the cost. For now we impose a fair cost sharing scheme, where all players who cross an edge split its cost evenly. Later we will look at an alternative scheme.

2 Optimal Solution

Given a graph, sources and sinks, and edge costs as described above, one can ask for a subset $T \subseteq E$ of edges such that

- Every source s_i is connected (using the edges in T) to its sink t_i, and
- T minimizes the total cost $C(T) = \sum_{e \in T} c_e$.

Such a T is called the *optimal solution*. Finding T is equivalent to the *Steiner tree problem*, which is known to be NP-complete.

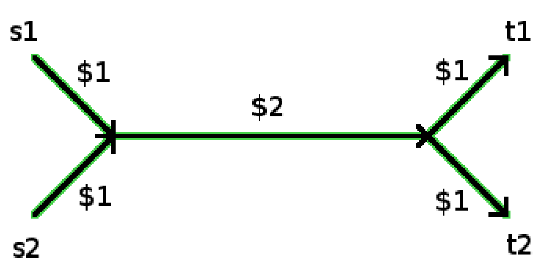

Figure 1: A sample two-player network. The optimal and Nash solutions (green) are identical.

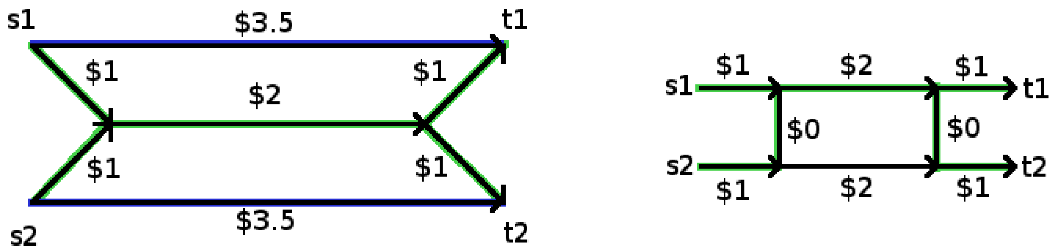

Figure 2: Left: An example network where the optimal solution is a Nash solution (green), but a second Nash solution (blue) exists as well. Right: An example where the Nash solutions are optimal.

3 Selfish Solution

Alternatively, we could let each player individually choose a path p_i from s_i to t_i. For each edge e, we then let k_e be the number of people using e, so that the total cost charged to the i-th player for his path is

$$C_i(p_i) = \sum_{e \in p_i} \frac{c_e}{k_e}.$$

A solution $\hat{T} \subset E = \bigcup_i p_i$ is then *selfish* or *Nash* if, for any alternative path $\tilde{p}_i$ for player i, $C_i(\tilde{p}_i) \geq C_i(p_i)$. As illustrated in Figure (2), the optimal solution may or may not be Nash, and a Nash solution is not necessary unique.

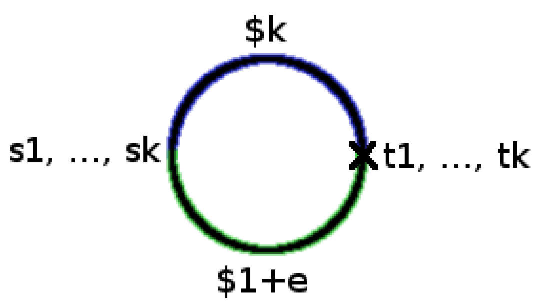

Figure 3: A network for k players where the price of stability is a big improvement over the price of anarchy. The optimal solution (which is also a Nash solution) has cost $1 + e$, for any small $e > 0$. (green). The worst Nash solution has cost k (blue).

4 Price of Stability

Since the Nash solution is not unique, we refine our definition of the price of anarchy, introduced in the previous lecture:

$$\text{Price of Anarchy} = \frac{C(\text{worst Nash solution})}{C(\text{optimal solution})}.$$

For example, the price of anarchy for the left network in Figure (2) is $\frac{7}{6}$.

Given that all players will act selfishly, an outside agent could nevertheless notice that there are multiple Nash solutions, and direct the players to implement that solution; it would then be to no player's advantage to deviate form that solution. The *price of stability* measures how close this directed selfish solution is to optimal:

$$\text{Price of Stability} = \frac{C(\text{best Nash solution})}{C(\text{optimal solution})}.$$

Figure (3) gives an example network where the price of stability is a significant improvement over the price of anarchy.

4.1 Bounding the Price of Stability

We will now prove an upper bound on the price of stability, using the technique of potential functions.

Given a choice of paths $S = \{p_i\}$ we associate to each edge e a *potential*

$$\begin{aligned}\phi_e(S) &= c_e \left(1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{k_e}\right) \\ &= c_e H_{k_e},\end{aligned}$$

where H_{k_e} is the k_e-th harmonic number. The potential $\Phi(S)$ of the system is then the sum of all potentials $\sum_{e \in E} \phi_e(S)$.

Lemma 4.1 *Suppose player i changes his chosen path from p_i to p_i'. Then if $S' = \{p_1, \ldots, p_i', \ldots, p_n\}$,*

$$C_i(p_i') - C_i(p_i) = \Phi(S') - \Phi(S).$$

Proof. Let's categorize the edges of E.

1. Edges e in both p_i and p_i'.
2. Edges $e \in p_i, e \notin p_i'$.
3. Edges $e \notin p_i, e \in p_i'$.
4. Edges in neither p_i nor p_i'.

Edges of type 1 or 4 have no effect on C_i. For each edge of type 2, player i receives a refund $\sum_{e \in 2} \frac{c_e}{k_e}$, and he pays a fee $\sum_{e \in 3} \frac{c_e}{k_e+1}$ for each edge of type 3. The total change in cost is thus

$$C_i(p_i') - C_i(p_i) = \sum_{e \in 3} \frac{c_e}{k_e + 1} - \sum_{e \in 2} \frac{c_e}{k_e}. \tag{1}$$

The change in potential is just the sum of the changes in the individual edge potentials,

$$\Phi(S') - \Phi(S) = \sum_e \left(\phi_e(S') - \phi_e(S)\right).$$

The potentials of edges of type 1 or 4 do not change, so they contribute nothing to the above sum. For the other two types,

$$\sum_{e \in 2} \left(\phi_e(S') - \phi_e(S)\right) = \sum_{e \in 2} \left(c_e H_{k_e - 1} - c_e H_{k_e}\right) = -\sum_{e \in 2} \frac{c_e}{k_e}$$
$$\sum_{e \in 3} \left(\phi_e(S') - \phi_e(S)\right) = \sum_{e \in 3} \left(c_e H_{k_e + 1} - c_e H_{k_e}\right) = \sum_{e \in 3} \frac{c_e}{k_e + 1},$$

the same two terms as in (1), so

$$C_i(p_i') - C_i(p_i) = \Phi(S') - \Phi(S).$$

□

Corollary 4.2 *Let S^n be a set of paths that minimize Φ. Then S^n is a Nash solution.*

Proof. Suppose, for contradiction, that it's not. Then there is some player i who can deviate from path p_i to p_i' to cheapen his cost, $C_i(p_i') - C_i(p_i) < 0$. But then

$$\Phi(S') - \Phi(S^n) < 0$$

$$\Phi(S') < \Phi(S^n)$$

and S^n is not a minimum, a contradiction. □

Now for a network with k players let S^n be the minimizer of Φ, S the best Nash solution, and S^* the optimum solution. Then

- $C(S) \le C(S^n)$ by the corollary.
- $C(S^n) \le \Phi(S^n)$ since for each edge, $c_e \le c_e \left(1 + \frac{1}{2} + \ldots + \frac{1}{k_e}\right) = \phi_e$.
- $\Phi(S^n) \le \Phi(S^*)$ since S^n minimizes Φ.
- $\Phi(S^*) \le C(S^*) H_k$ since for each edge, $\phi_e = c_e H_{k_e} \le c_e H_k$.

Chaining together inequalities,

$$C(S) \le C(S^*) H_k$$

$$\text{Price of Stability} = \frac{C(S)}{C(S^*)} \le H_k \approx \log k.$$

In fact this bound is tight: Figure (4) shows a worst-case network where the optimal solution has cost $1 + \epsilon$ for $\epsilon \to 0$, and the only Nash solution has cost $1 + \frac{1}{2} + \ldots + \frac{1}{k} = H_k$.

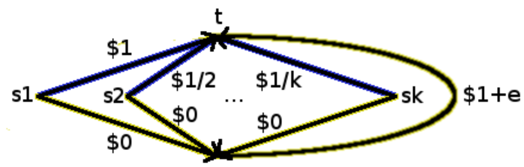

Figure 4: A worst-case network for k players. In the Nash solution (blue), the i-th player pays $\frac{1}{i}$, whereas the total cost of the optimal solution (yellow) is just $1+e$, for $e \to 0$.

5 Potential Games

The method of potential functions used here can also be used to analyze other games. The key is the existence of a magical potential Φ that directly relates the overall potential of the system to the benefit to individual players, as described in the theorem. Games with such a Φ are called *potential games*. As with network design, the minimizer of a potential games's Φ is a Nash solution. Moreover, suppose there is some lower bound $\epsilon > 0$ such that, if player i can deviate from his current strategy to decreases his cost, his cost does so by at least ϵ (this condition is easy to show for finite games.) Then we have an algorithm for finding a Nash solution: start with any solution S. If no player can deviate to a cheaper strategy, S is Nash. Otherwise, let a player deviate, and check again.

Each deviation decreases Φ by at least ϵ, and Φ is bounded below since, at best, every player pays 0 cost, so this algorithm is guaranteed to terminate and find a Nash solution. Unfortunately, does so may take a while.

Consider again the specific case of the network design problem, and suppose all edge costs are integral with $C = \max_e c_e$ be the cost of the most expensive edge (So $c_e \in \{0, 1, 2, \ldots, C\}$.). Then we know, for any starting solution S, $0 \leq \Phi(S) \leq C\,|E|\,H_k$. Furthermore it's easy to lower bound the change in potential at every step. In the worst case, at number of players sharing one edge will drop from k to $k-1$, thereby reducing ϕ_e by at least $\frac{1}{k}$. Therefore after $O(C|E|k \log k)$ best responses we must have found a Nash Equilibrium. (Otherwise the potential Φ would have become negative.)

6 Open Research Question

What if G is undirected? When we bounded the price of stability above by H_k, we never used that G was directed; however, the example in Figure (4) used to show tightness of this bound no longer works. What is the new tight bound? For simplicity, you can assume that all players share the same sink t. Even in this setting, no example networks have been found with price of stability greater than $\frac{12}{7}$.

7 Other Cost Shares

In all of the above, we charged all players equally for the edges they shared. We now look at alternative ways to split the cost.

Consider a cost sharing function $\xi_e(i, S_e)$ which, for edge e, player i, and set of players who share the edge S_e, gives the cost charged to player i for e. We require any cost-sharing scheme ξ_e to satisfy two common-sense conditions:

- Individually Rational: $\xi_e(i, S_e) = 0$ if $i \notin S_e$. In other words, players who do not use an edge do not have to help pay for it.
- Budget Balance: $\sum_{i \in S_e} \xi_e(i, S_e) = c_e$. The cost of the edge is paid in full, and the players as a whole are not overcharged.

For example, the fair cost sharing scheme we've been using, $\xi_e = \frac{c_e}{|S_e|}$, satisfies both properties.

Suppose all players have a common sink t. Then we can define the distance $d(s_i, t)$ to be the sum of the costs of the edges in the cheapest path from s_i to t. We then order the players so that

$$d(s_1, t) \leq d(s_2, t) \leq \ldots \leq d(s_k, t)$$

and define a cost sharing function

$$\xi_e(i, S_e) = \begin{cases} c_e, & \text{if } i \text{ is the smallest index in } S_e \\ 0, & \text{otherwise.} \end{cases}$$

For this cost sharing function, finding Nash solutions is easy. The first player can ignore all others, and so will choose the shortest path from s_1 to t. For the second player, arriving at any point along p_1 is just as good as arriving at t, so he will build the shortest path from s_2 to p_1. Similarly, the i-th player will build the shortest path from s_i to $p_1 \cup p_2 \ldots \cup p_{i-1}$.

Finding the paths p_i is equivalent to finding a minimum spanning tree, the players essentially simulate Prim's algorithm. It is also known that

$$C(\text{minimum spanning tree}) \leq 2C(\text{Steiner tree}),$$

so for this choice of ξ_e,

$$\text{Price of Anarchy} \leq 2.$$