

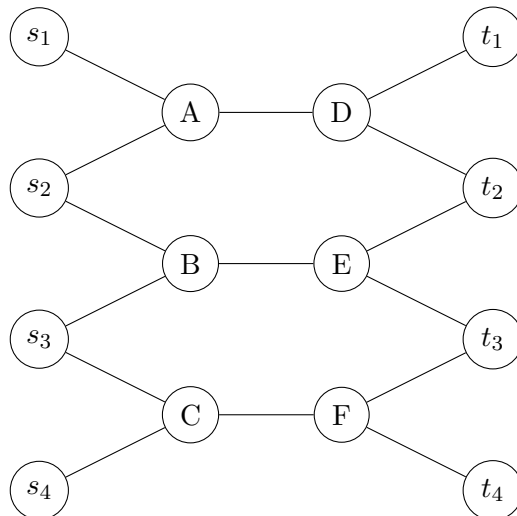
1. Shapely Network Design:

In this game we have a graph, and each player is represented by a specific pair of vertices, a starting vertex and an ending vertex and they effectively want to create some path from their starting vertices to the end vertices.

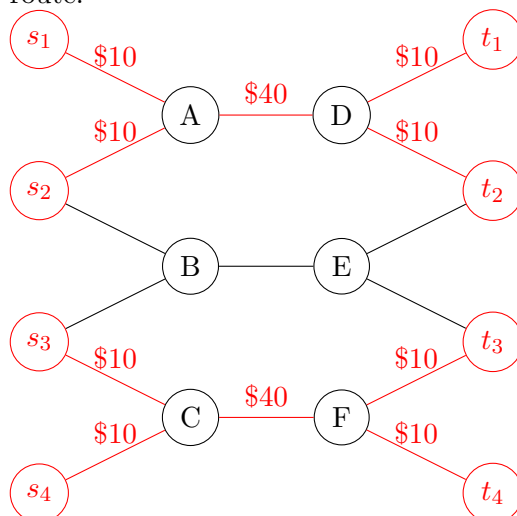
- (a) n players, who are aware of a graphical network G where each edge has some perceived cost on it.

In order to use an edge e with cost C_e , all players on that edge divide the cost evenly. Each player has a "source" vertex s and a target vertex t . They want to create a path from s to t , and they want to minimize their cost.

- (b) Let's look at example. Consider the graph below, with players s_1, s_2, s_3 , and s_4 . In this graph, the middle column of edges costs \$40, and the two outside columns cost \$10.



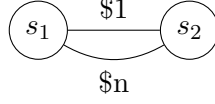
- (c) One possible route would be if s_1, s_2 took the top route and s_3, s_4 took the bottom route:



This gives us a total cost of $10 \times 4 + 40 + 10 \times 4 + 40 = 160$

2. Now let us consider a new scenario where two vertices, s and t , have two paths between them, one that costs 1, the other costs n .

- (a) $PoA \geq$ the number of players n .



- (b) If everyone takes the top path, solution cost is 1.
If everyone take the bottom path, this is still a NE and the total cost is n .
- (c) $PoA \leq o(N)$
- (d) $PoS = \Omega \log(n)$ (HW)

Definition 8.1 (Potential Functions). A potential function is a function that maps every possible state to a real number. An exact potential function encodes some ability for us to relate each players utilities/costs in varying solutions to the potential value in those solutions.

3. Shapely Network Design – Potential Functions:

Our Potential function is Φ :

$$\Phi(s) = \sum_{edges} c_e \cdot \sum_{i=1}^{k_e} \frac{1}{i}$$

Considering our scenario where k players go to the left where the edge's cost is 1 and $n - k$ go to the right where the edge's cost is n , our potential function becomes:

$$\Phi(k) = 1(1 + \frac{1}{2} + \dots + \frac{1}{n-k})$$

Using this potential function, let us look closer at this problem:

- (a) If one of the k left players went to the right, what is the change in their costs?

Their cost changes from $\frac{1}{k} \rightarrow \frac{n}{n-k+1}$

- (b) What is the difference in potential for these two "game states"?

$$\Phi(s) - \Phi(s') = \frac{n}{n-k+1} - \frac{1}{k}$$

Definition 8.2 (Exact Potential Functions). A potential function is a function that maps every possible state to a real number. An exact potential function encodes some ability for us to relate each players utilities/costs in varying solutions to the total potential value in those solutions.

A potential function Φ is an exact potential function if for 2 different game states where only 1 player p_i has changed their strategy, if $\Phi(s) - \Phi(s') = v_i(s') - v_i(s)$ where v_i is utilities to the i th player

Now let us look back at SND, in particular the example used to show PoA , with our potential function

$$\Phi(s) = \sum_{edges} c_e \cdot \sum_{i=1}^{k_e} \frac{1}{i}$$

2

Between our two states s and s' , only a path is changing from p to p' . Hence we can restrict our attention to two sets of edges:

A which represents the ones that are in p but not p'

B which represents the ones that are not in p but are in p'

For edges in A, our costs are dropping by $\frac{c_e}{k_e}$

For edges in B, our costs are increasing by $\frac{c_e}{k_e+1}$

A potential game is one that has exact potential function. For every potential game, there is always at least one pure NE, one that corresponds to the minimization of the potential function. Hence, for every finite potential game, each player playing their best response to everyone else, eventually converges to a NE.

Theorem 8.3. *Imagine we have a potential game with potential Φ , and we have two constants $A, B \geq 0$ such that:*

$$\frac{\text{cost}(s)}{A} \leq \Phi(s) \leq B \text{cost}(s)$$

Then we know that $\text{PoS} \leq A \cdot B$

Let s be the state that minimizes the potential Φ , let s^* be the optimal solution

$$\frac{\text{cost}(s)}{A} \leq \Phi(s) \text{ and } \Phi(s) \leq \Phi(s^*)$$

$$\Phi(s^*) \leq B \text{cost}(s^*)$$

$$\frac{\text{cost}(s)}{A} \leq B \text{cost}(s^*)$$

$$\frac{\text{cost}(s)}{\text{cost}(s^*)} \leq AB \Rightarrow \text{PoS} \leq AB$$

Applying this for SND, we had exact potential $\Phi(s) = \sum_{\text{edges}} c_e \sum_{i=1}^{k_e} \frac{1}{i} \approx O(\log(n))$

We want to find A, B so that

$$\frac{\text{cost}(s)}{A} \leq \Phi(s) \leq B \text{cost}(s)$$

From our above formula, we know that if $k_e = 0$, then the second summation sums to 0, and if $k_e \geq 1$, then the second sum is greater or equal to 1. This part of this summation is $O(\log)$, so we shall call this $H_{k_e} \leq H_n$. Hence we can pick $A = 1$ as $\text{cost}(s) \leq \Phi(s)$. And we can pick $B = H_n = O(\log)$, so

$$\text{PoS} = O(\log n)$$