Algorithmic Game Theory
Final Exam

Juan Pablo Royo Sales
Universitat Politècnica de Catalunya

December 18, 2020

1 Exercise 1

A Network Congestion Game (NCG) is **symmetric** when $s_i = s$ and $t_i = t$, for $i \in N$ according to the definition that we have seen in class. Basically all the players, have the same source and destination endpoint s and t and all have the same path strategies.

The algorithm to find Pure Nash Equilibrium (PNE) in SNCG in *poly-time* is doing a reduction to Min-Cost Flow Problem (MINCOST) in order to find the Optimum of $\varphi(s)$.

Algorithm 1: Compute MINCOST reduction of SNCG

Input : $G = (V, E, s, t)$, d_e delay function
Output: MINCOST representation of the problem
for *each* $e \in E$ **do**
  for 1 *up to* d_e **do**
    $E \leftarrow E \cup \{e'_i\}$ such that $d_{e'_i} = 1$;
  $E \leftarrow E \setminus \{e\}$;
return N;

Basically the reduction is replacing each edge e with the amount of the delay function for e d_e of new edges but this time each of this edges with delay function equal to 1.

The MINCOST of the new network N minimizes $\varphi(s)$.

Since the algorithm is giving also a Local Optimum, then s is a PNE.

2 Exercise 2

As we have seen in class the **Social Cost** is the sum of all players

$$C(s) = \sum_{u \in V} c_u(s) = \alpha|E| + \sum_{u,v \in V} d_G(u,v) \tag{1}$$

Every pair of vertices that is not *connected* its distances is ≥ 2, there is a lower bound for the social cost:

$$C(s) \geq \alpha|E| + 2|E| + 2(n(n-1) - 2|E|) \tag{2a}$$
$$= 2n(n-1) + (\alpha - 2)|E| \tag{2b}$$

2.1 (a)

In this case when $\alpha < 1$ the social Optimum is achieved when $|E|$ is maximum. So, every vertex is connected with each other. It is a complete graph. Any Nash Equilibrium (NE) should be of diameter 1 which implies that the complete graph is the only NE.

2.2 (b)

In the case $1 \leq \alpha < 2$, the social Optimum is achieved by a complete graph, but not NE. Any NE is for diameter at most 2. So, the social cost is equal to 2b and it is the worst cost when $|E|$ is minimum, which is $n - 1$ for a connected graph. Therefore the worst NE is the S_n Star.

2.3 (c)

As we have seen in class when $\alpha > 2$ we need to minimize the number of edges $|E|$ in order to reduce the cost, but at the same time the graph should be connected. Therefore only trees with diameter 2 have optimal cost. One example that has NE, with diameter 2 and minimum number of edges is a **Cycle Graph** with 4 vertex.

2.4 (d)

Since when $\alpha > n^2$ no player has any incentive to buy an edge, there is going to be the minimum amount possible of edges at the same time with the graph

connected, that by definition are **Spanning Trees** as we have seen in class. All are NE because no one has incentive to change. The PoA is giving by:

$$PoA = \frac{c(T_n)}{c(S_n)} \tag{3a}$$
$$\leq \frac{\alpha(n-1) + (n-1)(n-1)}{\alpha(n-1) + 1 + 2n(n-1)} \tag{3b}$$
$$= O(1) \tag{3c}$$

3 Exercise 3

3.1 (a)

In order to show that it is a Simple Game (SG), we need to show that the valuation function is **monotone**.

We assume in this game that

$$v(X) = \begin{cases} 1 \text{ } \textit{if } X \textit{ is a dominant set of } G \\ 0 \text{ } \textit{otherwise} \end{cases}$$

And on the other hand we know that A Game is monotone if $v(C) \leq v(D)$ for any C, D such that $C \subseteq D$.

Lets take any $C, D \in G$ such that $C \subseteq D$ and lets consider the following cases:

- If D is a dominant set, any subset of vertices on D, lets call C such that $C \subseteq D$, have 2 possibilities: either
  - C is also dominant because we remove some border vertex which adjacency vertices are connected to other $v \in C$, so $v(C) = v(D) = 1$
  - C is not dominant because selected subset is not a resulting dominant set, in which case $v(C) = 0 < v(D)$
- If D is non dominant then any subset is also non dominant so $v(C) = v(D) = 0$.

Then we can conclude that $v(C) \leq v(D)$, for any $C, D \in G, C \subseteq D$. So this is **monotone**.

Therefore it is a SG.

3.2 (b)

Base on the theorem that states *A simple game has non-empty core if and only if it has a veto player*, we can check if there is a veto player for some *dominant game* in order to see if it has a non-empty core. An example of that is a S_n star in which the center vertex c belongs to any coalition that contains c but it cannot be remove from any of those coalition because if we remove it it is not a *dominant set*.

3.3 (c)