Computing Equilibrium

Lecture 4.1

Support enumeration algorithm for NASH for 2 players.

Computing a PSNE Given a game $\Gamma = \langle N, (S_i)_{i \in N}, (u_i)_{i \in N} \rangle$ find a PSNE.

→ Observe that checking if a given pure strategy profile is a PSNE takes polynomial time.

Let all players have s strategies, and there are n players.

The number of strategy profiles is s^n.

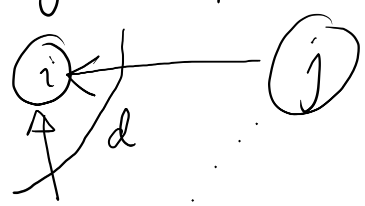
The input consists of $n \cdot s^n$ utility values (numbers).

$O(n \cdot s)$ time is required to check if a pure strategy profile is a PSNE.

We can find a PSNE (if it exists) in $O(n \cdot s^{n+1})$ time.

Important Classes of Succinct Games

1. <u>Graphical games:</u> We have a directed graph G on the set N of players. The utility of a player i depends only on the players who have a directed edge to i, including i. If the in-degree of G is at most d, then $(n \cdot s^{d+1})$ numbers are needed to represent the game.



2. <u>Sparse game</u>: In all but a few strategy profiles give non-zero utility.

3. <u>Symmetric game</u>: All players are identical. The utility of a player depends on the strategy played by the player and the number of players playing each strategy.

4. <u>Anonymous game</u>: The utility still depends <u>only</u> on the number of players playing each strategy.

$$n \cdot s \binom{n-1+s-1}{s-1} = n \cdot s \cdot \binom{n+s-2}{s-1}$$ numbers in input.

For symmetric game: $s \cdot \binom{n+s-1}{s-1}$.

5. <u>Network congestion game:</u> We have a graph G. Each player i has a source s_i and destination t_i. The load $l(e)$ of each edge is the number of players using that edge. Each edge e has a non-decreasing cost function $c_e : \mathbb{R} \to \mathbb{R}$. The strategy set of player i is the set of paths from s_i to t_i in G.

Let $(P_i)_{i \in N}$ be a strategy profile.

$$u_i\left((P_i)_{i \in N} \right) = - \sum_{e \in P_i} c_e(\ell(e))$$