1 Nash Equilibrium

Given $n$ players, each player $i$ has a set of actions $S_i$. Cost function for each player $i$ is defined as follows:

$$c_i : S_i \rightarrow \mathbb{R}$$

Nash-Equilibrium is a state when none of the players can increase his utility by a one-sided deviation, if all others remain in their previous choice.

Let’s assume that we have a social cost function, which we wish to minimize and let’s assume that $OPT = \min_{S}(cost(S))$

**Definition 1** Price of Anarchy - the relation between the value of the cost function at a worst-case Nash Equilibrium and the optimal value $OPT$. It represents how bad the situation is because of manipulative and strategy-based players.

**Definition 2** Price of Stability - the relation between the value of the cost function at a best-case Nash Equilibrium and the optimal value $OPT$.

There are two sources to lack of efficiency:
1. The players are selfish and strategy-based.
2. Lack of coordination - Nash equilibrium speaks of one-sided deviations, but there could be coalitional deviations, when a group of players agree to change their choices and by that increase each player’s utility within the coalition. We’ll analyze the Price of Anarchy (PoA) and the Price of Stability (PoS) in cases when coordination between players is possible.

**Definition 3** Strong Equilibrium - a state when no group of players can coordinate the actions of players in it, in a way which will increase every player’s utility in the group. Note that every Strong Equilibrium is necessarily a Nash Equilibrium.

For example, while the Prisoner’s Dilemma has a Nash Equilibrium (in a state when both players choose not to cooperate), it does not have a Strong Equilibrium.
Definition 4  Strong Price of Anarchy (SPoA) - the relation between the value of the cost function in a worst-case Strong Equilibrium and the optimal value -OPT.

Note that Strong Equilibrium ⊆ Nash Equilibrium, and therefore SPoA ≤ PoA

Definition 5  k-Strong Equilibrium - a state in which no coalition with size at most k, can improve its utility by changing its choice of actions (of the players in it).

Note: Nash Equilibrium = 1 - Strong Equilibrium.

Also, $n - SE \subseteq (n - 1) - SE \subseteq ... \subseteq 2 - SE \subseteq 1 - SE = NE$.

2 Job Scheduling

Job Scheduling is a large family of problems, which have the property of always having a Strong Equilibrium. We'll discuss the Unrelated Machines Model. Let's assume there are $m$ machines $M_1...M_m$. There are $n$ jobs waiting to be performed on some machine and the weight function of job $j$ on machine $M_i$ is $w_i(j)$.

Each player's strategy (each job represents a player) is to decide on which machine to run. The cost for every player is the total load of the machine it runs on. If $S_i$ is the strategy of job $j$ (the machine it chose to run on), the load of $M_i$ is defined as

$$L_i(S) = \sum_{S_j=M_i} w_i(j)$$

There is no meaning to the order of the jobs running on a machine. The target is to minimize the makespan of the machine which is the last to finish, and the value of the target function is defined as $OPT = \min_S \text{makespan}(S)$ We know that the Nash Equilibrium in mixed-strategies will always exist. However, this is not known for pure strategies.

Definition 6  Potential Game - a game in which a Pure Nash Equilibrium always exists, but not always a Strong one. The Job Scheduling game is an example of a potential game.

<table>
<thead>
<tr>
<th>job</th>
<th>$M_1$</th>
<th>$M_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$j_1$</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>$j_2$</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>$j_3$</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>

Claim 7  Every game with unrelated machines has a Strong Equilibrium.
**Definition 8**  
Lexicographic Sort - if $l_i$ is the load of machine $M_i$, the vector 

$$(l_1 \ldots l_m) < (l'_1 \ldots l'_m)$$

if until a certain index $i$ the loads are equal, and at index $i$

$$l_i < l'_i.$$

A configuration $S$ is said to be less than $S'$ if the sorted vector of $S$ is less than that of $S'$.

**Proof:**  
We’ll prove that the profile, which gives us the lexicographic minimum, is in Strong Equilibrium. Let’s assume that $S$ is the lexicographic minimum, but it is not Strong. Let $C$ be the minimal coalition, which can stray to $S'$ from $S$, so that each of the players within the coalition improves his state. The claim is that those machines exactly which get chosen by $C$, are the set of machines which are used after the stray. We’ll prove that  
1. In the lexicographic minimum, each machine which receives a job, also loses a job. That is true because otherwise, there will be a contradiction to Nash Equilibrium.  
2. Each machine, which loses a job, also receives a job. $C$ is minimal, but if a certain job would benefit by exiting from $C$, and the coalition would benefit as well, without having any job enter it, it’ll be a contradiction to the minimality of $C$, because initially it would be better for $C$ to exist without the exited job, which only added value to its load. There is a set of machines, so that from each one of them there was a job that exited and a job which came in, and $C$ benefited. But that is a contradiction to the lexicographic minimum, because it means that the coalition moved to a profile, which is lexicographically lower. ■

**Example:**

<table>
<thead>
<tr>
<th>job</th>
<th>$M_1$</th>
<th>$M_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$j_1$</td>
<td>$\epsilon$</td>
<td>1</td>
</tr>
<tr>
<td>$j_2$</td>
<td>1</td>
<td>$\epsilon$</td>
</tr>
</tbody>
</table>

The social optimum will be of course, when $j_1$ will run on $M_1$ and $j_2$ on $M_2$, and that will be a Nash Equilibrium as well a Strong Equilibrium, with a makespan of $\epsilon$. However, there is another Nash Equilibrium when $j_1$ runs on $M_2$ and $j_2$ runs on $M_1$, in which case it will not be a Strong Equilibrium, and the makespan will be 1. SPoA = 1.

**Theorem 9**

For each game with $m$ unrelated machines and $n$ jobs it is true that $\text{SPoA} \leq m$. That bound is tight.

**Proof:**

It is true that $L_1(S) \leq \text{OPT}$. Also, $\forall i \ L_i(S) - L_{i-1}(S) \leq \text{OPT}$. Therefore, $L_m(S) \leq m \cdot \text{OPT}$. ■

Note: In case of unrelated machines it is true that 

$SPoA = m$

12-3
In case of identical machines it is true that

\[ SPoA \geq \frac{2}{1 + \frac{1}{m}} \]

**Theorem 10**

\[ k - SPoA \geq \frac{n}{2k} \text{ (for } m = 2) \]

Example: Given 2 machines and 2 jobs, let’s look at the following input:

<table>
<thead>
<tr>
<th>jobs</th>
<th>M_1</th>
<th>M_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>j_1</td>
<td>2k</td>
<td>n-1+k+e</td>
</tr>
<tr>
<td>j_2</td>
<td>1</td>
<td>\frac{1}{n-1}</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>j_n</td>
<td>1</td>
<td>\frac{1}{n-1}</td>
</tr>
</tbody>
</table>

In this case, the optimal configuration will be when \( j_1 \) will run on machine \( M_1 \) and jobs \( j_2 \) ... \( j_n \) will run on \( M_2 \).

\[ k - SPoA \geq \frac{n-1+k+e}{2k} \geq \frac{n}{2k} \]

Note: Except for the case when \( k=1 \), \( k\)-SPoA can be bounded.

### 3 Network Formation Game

In this game the players are represented by vertexes in a directed connected graph and each player’s strategy is to decide to what other vertexes to connect (create an edge). The cost function of each player is a sum of: edges cost -

\[ B_S(V) = \alpha \cdot |S_V| \]

and the sum of the distances from the vertex to the rest. On one hand, every vertex will want to be a central vertex, but will not be willing to pay the edges cost. \( \alpha \) is the cost of each vertex.

\[ dist_S(V) = \sum_{w \in V} \delta_S(v, w) \]

The social cost function is simply

\[ \sum_{v \in V} C_v(S) \]
<table>
<thead>
<tr>
<th></th>
<th>$\alpha \leq 1$</th>
<th>$1 \leq \alpha \leq 2$</th>
<th>$\alpha \geq 2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$OPT$</td>
<td>Clique</td>
<td>Clique</td>
<td>Star</td>
</tr>
<tr>
<td>$NE$</td>
<td>Clique (unique)</td>
<td>Many NE, Clique not in NE, Star $\in$ NE</td>
<td>Many NE, Star $\in$ NE</td>
</tr>
</tbody>
</table>

Note: There always exists a Nash Equilibrium in pure strategies! The only parameter here was $\alpha$.

**Theorem 11** Generally, we showed that for every $1 \leq \alpha \leq 2$ and $n \geq 7$ there is no 3-SE. However, for any other $\alpha$ there always exists a Strong Equilibrium and also it is true that $SPoA \leq 2$ for any $\alpha \geq 2$.