1 Coalitional Games

In today’s lecture, we will look at coalitional games. In a coalitional model, we focus on what groups of players can achieve rather than on what individual players can do. A stability requirement for a coalitional game is that the outcome be immune to deviations by groups of players, i.e., no subset of players can unilaterally improve their outcome. A strong criterion is that all players in the group are strictly better off. A weaker criterion is that the outcome improves for at least one player in the group and makes it no worse for all others. Some applications of coalitional games are:

- Cost sharing for network design: Users benefit from being connected to a server. So they have to build up a broadcast tree. However, it costs to maintain the server/network and the question is how to share the costs.

- Queue management: Multiple users want to route traffic through a switch, which has a flow dependent delay (cost). The queueing delay cost has to be shared among the users. This is similar to the bandwidth sharing game which we looked at in an earlier lecture, except that the utility need not be a separable function and the cost sharing need not be proportional.

We now consider a simple version of a coalitional game, namely a coalitional game with transferable payoff (transferable payoff means that there is no restriction on how the total payoff may be divided among the members of a group).

1.1 Coalitional games with transferable payoff

We have a finite set $N$ of users and a function $v : 2^N \to \mathbb{R}^\geq 0$. So, $v$ assigns a non-negative number to every $S \subseteq N$. We can think of $v(S)$ as the total payoff available to the members of set $S$. A natural question that arises is how to share the value $v(N)$ among all the users so that there is no incentive to deviate. Before looking at rules for stable sharing, we consider some examples.

- An expedition of $n$ people discover treasure. It requires two people to carry out one piece of the treasure, in which case the value of the treasure is equally shared between the two. For a subset $S$, the value is given by

$$v(S) = \left\lfloor \frac{|S|}{2} \right\rfloor$$

Let’s see if there is way of sharing the value in a stable manner. If $|N| = 2$, then clearly $(1/2, 1/2)$ is a stable sharing. What happens when $|N| = 3$? Note that $(1/2, 1/2, 0)$ is not a stable sharing since the third person can offer, say the second person, a little more than half. Similarly $(1/3, 1/3, 1/3)$ is not a stable sharing since two players can each get a value of $1/2$ if they form a coalition. It can be shown there is no stable sharing now.
Majority vote: A subset \( S \) gets a value of 1 if it consists of a majority of the players and nothing otherwise. So \( v(S) \) is given by

\[
\begin{align*}
v(S) &= 1, \text{ if } |S| \geq |N|/2 \\
&= 0, \text{ otherwise }
\end{align*}
\]

It can be shown that there is no stable sharing when \( |N| \geq 3 \).

1.1.1 Rules for stable sharing

**The Core:** The core is a solution concept for coalitional games, analogous to Nash equilibria for non-cooperative games. For a coalitional game with transferable payoff, a cost sharing is in the core if no coalition can obtain a payoff which is better than the sum of the members’ current payoffs.

We note that a cost sharing \( x \) has \( x_i \geq 0 \forall i \in N \) and \( \sum_{i \in N} x_i = v(N) \). Then, the condition for a cost sharing \( x \) to be in the core is that \( \forall S \subseteq N, \sum_{i \in S} x_i \geq v(S) \). Equivalently, there is no set \( S \) and payoff vector \( y \) with \( \sum_{i \in S} y_i = v(S) \) for which \( y_i > x_i \forall i \in S \).

Frequently, the core is empty (as in the two examples that we considered). So we want to determine when the core is not empty. Observing that the core is characterized by a set of linear inequalities, we have the following result, referred to as the Bondareva-Shapley theorem.

**Theorem 1** A coalitional game with transferable payoff has a non-empty core iff \( \forall y \in \mathbb{R}^N, \text{ if } \sum_{S \subseteq N} y_S = 1 \forall i, \text{ then } \sum_S y_S v(S) \leq v(N) \).

**Proof.**

We use the following fact about linear programs:

\[ \exists x, x \geq 0, Ax \leq b \forall y \geq 0, \text{ if } y^T A = 0, \text{ then } y^T b \geq 0 \]

Here \( A \) is an \( m \times n \) matrix and \( x, b, y \) are \( n \times 1, m \times 1, m \times 1 \) vectors, respectively.

For a cost sharing \( x \) to be in the core, we need

\[
\begin{align*}
x_i &\geq 0 \forall i \in N, \sum_{i \in N} x_i = v(N) \text{ and } \sum_{i \in S} x_i \geq v(S) \forall S \subseteq N \\
&\text{Noting that } \sum_{i \in N} x_i = v(N) \text{ can be written as } \sum_{i \in N} x_i \leq v(N) \text{ and } \sum_{i \in N} x_i \geq v(N) \text{ and that } \text{the latter inequality can be omitted, the problem reduces to finding } x \text{ with }
\end{align*}
\]

\[
\begin{align*}
x_i &\geq 0 \forall i \in N, \sum_{i \in N} x_i \leq v(N), -\sum_{i \in S} x_i \leq -v(S) \forall S \subseteq N \\
&\text{Applying the above fact about linear programs, we get that the core is non-empty iff } \forall y \geq 0, \text{ if } \sum_{S \subseteq N} y_S = y_N \forall i, \text{ then } \sum_S y_S v(S) \leq y_N v(N). \text{ Dividing the inequalities by } y_N, \text{ we get the required result.}
\end{align*}
\]

We provide some intuition for the theorem. Note that for the core to be non-empty, it must be the case that for any partition of \( N \), say \( A_1, \ldots, A_k \), \( \sum v(A_i) \leq v(N) \). Otherwise, there is an incentive for a subset of users to break away. Now, consider the case when \( y_S \in \{0,1\} \). Then \( \sum_{S \subseteq N} y_S = 1 \forall i \in N \) implies that subsets \( S \) with \( y_S = 1 \) form a partition of \( N \). So \( \sum_S y_S v(S) \leq v(N) \).
\(v(N)\) is precisely the partition inequality. **When \(y\)'s are fractional, the meaning is less clear, but we can think of \(\sum S y_S v(S) \leq v(N)\) requiring the expected value of the “partition” not to exceed \(v(N)\).**

In the next lecture, we will look at solution concepts for cost sharing when the core is empty.