Algorithmic Game Theory March 23, 2016

Lecture 4

Lecturer: Michal Feldman *Scribe: Nadav Bar, Omri Koshorek, Noam Mor, Michael Rotman*

1 Coalitions

In the previous lecture, we have defined a Strong Nash Equilibrium as an equilibrium in which no coalition can improve its situation by deviating from it. We now define the Strong Price of Anarchy as the cost of the best Strong NE relative to the optimal cost.

Definition 1 $\text{SPoA} = \frac{\text{Cost of best Strong NE}}{\text{OPT}}$

Theorem 2 *For any fair cost-sharing game:*

$$\text{SPoA} = \theta\left(\log n\right) \tag{1}$$

Observation 3 *We've already seen that, for a cost-sharing game:*

$$\begin{aligned} \text{PoA} &= n && (2)\\ \text{PoS} &= \log n && (3) \end{aligned}$$

We'll soon see that the SPoA $\leq \log n$.

Let $P = (P_1, \ldots, P_k)$ be a profile in strong equilibrium in a fair cost-sharing game, and let $P^* = ({P^*}_1, \ldots, {P^*}_k)$ be the optimal profile in the same game.

Let $A_j = \{1, \ldots, j\}$ the coalition of the players 1 through j. Since P is a strong equilibrium, it is not worth it for any of the coalitions A_j to deviate from P. Therefore there has to be a player in each of these coalitions that does not benefit from the transition. Without loss of generality, we assume that player j does not benefit from the transition of A_j from P to P^*.

Now we use the following two facts:

- The cost of a player j when everyone plays profile P is at least player j's cost after A_j has transitioned to profile P^* - this because of our assumption that player j does not benefit from the transition.

- Player j's cost when A_j plays P^* is at least their cost when A_j plays P^* *and no other player exists*. This because adding players to a fair cost-sharing game can only lower the cost for everyone.

We write the equations and then sum everything up.

$$\begin{array}{ccccc}
c_k(P) & \leq & c_k\left(P^*_{A_k}, P_{-A_k}\right) & \leq & c_k\left(P^*_{A_k}\right) \\
c_{k-1}(P) & \leq & c_{k-1}\left(P^*_{A_{k-1}}, P_{-A_{k-1}}\right) & \leq & c_{k-1}\left(P^*_{A_{k-1}}\right) \\
\vdots & \leq & \vdots & \leq & \vdots \\
c_1(P) & \leq & c_1\left(P^*_{A_1}, P_{-A_1}\right) & \leq & c_1\left(P^*_{A_1}\right) \\
\hline
\text{cost}(P) & & \leq & & \sum_{i=1}^{k} c_i\left(P^*_{A_i}\right)
\end{array} \tag{4}$$

We need to derive an upper bound for $\frac{\text{cost}(P)}{\text{cost}(P^*)}$. Reminder: fair cost-sharing games are potential games with the following potential function:

$$\boxed{\phi(P) = \sum_{e \in E} \sum_{i=1}^{p_e} \frac{\gamma_e}{i}} \tag{5}$$

The following important observation relates $c_i\left(P^*_{A_i}\right)$ to the potential function:

$$c_i\left(P^*_{A_i}\right) = \sum_{e \in P^*_i} \frac{\gamma_e}{P^i_e} = \phi\left(P^*_{A_i}\right) - \phi\left(P^*_{A_{i-1}}\right) \tag{6}$$

where P^i_e is the number of players from the group A_i that use the edge e as part of the profile P^*. We now have:

$$\begin{aligned}
\text{cost}(P) &\leq \sum_{i=1}^{k} c_i\left(P^*_{A_i}\right) && (7)\\
&= \sum_{i=1}^{k} \left[\phi\left(P^*_{A_i}\right) - \phi\left(P^*_{A_{i-1}}\right)\right] && (8)\\
&= \phi(P^*) - \phi\left(P^*_{A_0}\right) && (9)\\
&= \phi(P^*) && (10)\\
&\leq H_k \text{cost}(P^*) && (11)
\end{aligned}$$

(8) is a telescopic sum, and $\phi\left(P^*_{A_0}\right)$ is 0, since there are no players in A_0. The inequality in (11) is from last week's lemma.

Therefore we get:

$$\frac{\text{cost}\,(P)}{\text{cost}\,(P^*)} \leq H_k \tag{12}$$

2 Dynamics and Equilibrium

Up until now our discussion only included static equilibriums. Here is what we have so far:

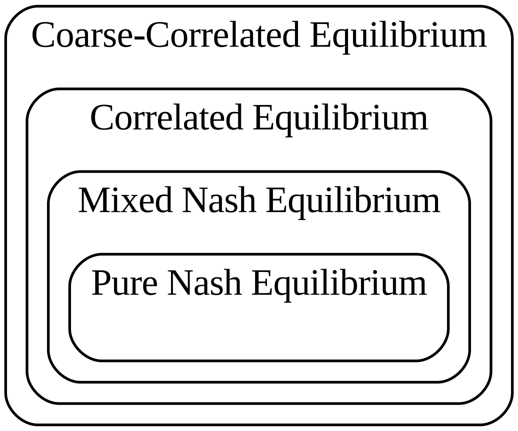

Figure 1: Equilibrium Hierarchy

We now move on to discuss dynamics. We will be interested in dynamics that converge quickly (polynomially) to an equilibrium.

2.1 Best Response Dynamics

Definition 4 *Best-Response dynamics (BRD).*

As long as the current profile, S, is not a pure equilibrium:

- *Find a player i and a profitable deviation s'_i - a strategy s'_i such that $c_i(S) > c_i(s'_i, S_{-i})$.*
- *Switch from the profile S to the profile (s'_i, S_{-i}).*

Observation 5 *If BRD dynamic stops, it stops at a Nash Equilibrium.*

Proof: BRD continues as long as there is still a player i that can improve their situation by changing their strategy alone. Therefore, if BRD stopped, it means that no player can improve their situation by deviating alone from S - that S is a NE. ■

There are two variants of this dynamic. In *better response dynamics* we choose any profitable s'_i. In *best response dynamics*, we choose a s'_i that is the best response to S_{-i}. We don't address this distinction now, but it may come into effect later.

Fact 6 *There can be games that have a Pure Nash Equilibrium, but for which BRD will never converge.*

Reminder: potential games are games such that there is a function ϕ that for every profile S, and all players i, and all deviations $s'_i \in S_i$,

$$\phi\left(s_i', s_{-i}\right) - \phi\left(s\right) = c_i\left(s_i', s_{-i}\right) - c_i\left(s\right) \tag{13}$$

Let us now examine BRD's convergence in potential games. Since BRD always chooses a favourable deviation for player i, it follows that the value of c_i is strictly decreasing during BRD. From the equation above, a decrease in c_i must also decrease ϕ. Therefore the value of ϕ is strictly decreasing during BRD. Since our games are finite, ϕ must have a minimum, and therefore BRD will always converge in potential games.

Fact 7 *Finding a mixed NE is PPAD-complete. Finding a pure NE in potential games is PLS-complete.*

2.2 No Regret Dynamics

Assume that we have only one player. That player certain player has n possible actions they can do. A is actions set.

At any given time $t = 1, \ldots, T$:

1. The player chooses a mixed strategy p^t (a distribution over A).
2. The adversary chooses a cost function $c^t : A \to [0, 1]$.
3. An action a^t is chosen according to the distribution p^t, and the cost to the player is $c^t(a^t)$. The cost function c^t is known to the player.

Given the cost functions c^t, the best cost of the algorithm is:

$$\text{best cost} = \sum_{t=1}^{T} \min_{a \in A} c^t\left(a\right) \tag{14}$$

We would like to find a strategy for the player that minimizes the difference between the the actual cost of the dynamic and the best cost, under the assumption that our adversary wants to increase the player's cost.

Observation 8 *The difference between the best cost and the actual cost in the No-Regret dynamic can be very large.*

Example 1 *In this example, there are only two possible actions:* $|A| = 2$*, and the adversary chooses* c^t *from these two options:*

$$c^t(a) = \left\{ \begin{array}{c} \begin{pmatrix} 1 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 1 \end{pmatrix} \end{array} \right\} \tag{15}$$

At each timestamp, the adversary gives a cost of 1 *to the action for which the player gave the highest probability, and for the other a cost of* 0*.*

If the player plays strategy 1 with probably p *and strategy 2 with probability* $1 - p$*, their expected cost is* $\max(p, 1-p)$*. Therefore the best course of action for the player is to play each of the strategies with probability 0.5. The player's expected cost after* T *rounds is therefore* $\frac{T}{2}$*. The best cost for the player is* 0*.*

The difference between the expected cost and the best cost in this example is linear with T, which is very bad. This means we won't be able to obtain a good upper bound. We look then for a different metric for success.

Definition 9 *The regret of a sequence of actions* $a^1, \ldots, a^T$ *against a constant action* $a \in A$ *is:*

$$\frac{1}{T}\left[\sum_{t=1}^{T}\left(c^t\left(a^t\right) - c^t(a)\right)\right] \tag{16}$$

Definition 10 (No Regret Algorithm) *Let* $\mathcal{A}$ *be an online decision-making algorithm.*

- *An adversary chooses a function* $c^t : A \to [0,1]$*.*
- *The algorithm* $\mathcal{A}$ *has no regret if for every adversary and for every action* a

$$\text{Regret} \underset{T\to\infty}{\to} 0 \tag{17}$$

Observation 11 *Any deterministic algorithm will achieve poor results in this benchmark.*

Example 2 *Assume there are* n *possible actions, with* $T \gg n$*. Assume that the player plays a pure strategy* a^t *in day* t*. Then the adversary will choose:*

$$c^t(a) = \begin{cases} 1 & a = a^t \\ 0 & a \neq a^t \end{cases} \tag{18}$$

After T rounds, there has to be an action a whose total cost is at most $\frac{T}{n}$ - otherwise the total cost exceeds T (pigeonhole principle). The regret against the action a is then:

$$\frac{1}{T}\left[T-\frac{T}{n}\right]\underset{T\to\infty}{\not\to}0 \tag{19}$$

Multiplicative Weight (MW) Algorithm

1. $w^1(a)=1,\ \forall a\in A$
2. $\forall t=1,\ldots,T$, $p^t=\frac{w^t}{\Sigma^t}$, $\Sigma^t=\sum_{a\in A}w^t(a)$

Weight Update:

$$w^{t+1}(a)=w^t(a)(1-\epsilon)^{c^t(a)} \tag{20}$$

We now present an analysis of the algorithm. We will compare the cost of the algorithm, and the cost of the best single action a^*, to Σ^T. Let $\mathrm{OPT}=\sum_{t=1}^{T}c^t(a^*)$, the cost of the best action.

- A lower bound on Σ^T is $\Sigma^T\geq(1-\epsilon)^{\mathrm{OPT}}$

  Proof:

  $$\Sigma^T=\sum_{a\in A}w^T(a)\geq w^t(a^*)=w^1(a^*)\prod_{t=1}^{T}(1-\epsilon)^{c^t(a^*)}=(1-\epsilon)^{\sum_{t=1}^{T}c^t(a^*)}=(1-\epsilon)^{\mathrm{OPT}} \tag{21}$$

  ■

- The expectation value of the algorithm at time t is

  $$\mathrm{ALG}^t=\sum_{a\in A}p^t(a)c^t(a)=\sum_{a\in A}\frac{w^t(a)}{\Sigma^t}c^t(a) \tag{22}$$

  Recall that

  $$\Sigma^{t+1}=\sum_{a\in A}w^{t+1}(a)=\sum_{a\in A}w^t(a)(1-\epsilon)^{c^t(a)}\leq\sum_{a\in A}w^t(a)\left(1-\epsilon c^t(a)\right) \tag{23}$$

  The above is true for $\epsilon\in\left[0,\frac{1}{2}\right]$ and $c^t(a)\in[0,1]$. The RHS is

  $$\sum_{a\in A}w^t(a)\left(1-\epsilon c^t(a)\right)=\Sigma^t\left(1-\epsilon\mathrm{ALG}^t\right) \tag{24}$$

  We obtained,

  $$\Sigma^{t+1}\leq\Sigma^t\left(1-\epsilon\mathrm{ALG}^t\right) \tag{25}$$

We can now relate between the two bounds

$$(1-\epsilon)^{\mathrm{OPT}} \leq \Sigma^T \leq \Sigma^1 \prod_{t=1}^{T} \left(1-\epsilon \mathrm{ALG}^t\right) \tag{26}$$

Taking ln from both sides we obtain

$$\mathrm{OPT} \ln(1-\epsilon) \leq \ln n + \sum_{t=1}^{T} \ln\left(1-\epsilon \mathrm{ALG}^t\right) \tag{27}$$

The Taylor series expansion for ln around 0 is: $\ln(1-\epsilon) = -\epsilon - \frac{1}{2}\epsilon^2 - \cdots \leq -\epsilon$. Therefore we have:

$$\mathrm{OPT}\left(-\epsilon - \frac{1}{2}\epsilon^2\right) \leq \mathrm{OPT}\left(-\epsilon-\epsilon^2\right) \leq \mathrm{\ln n} - \sum_{\mathrm{t=1}}^{\mathrm{T}} \epsilon \mathrm{ALG^t} \tag{28}$$

Rearranging the terms and recalling that OPT is bound from above by T, we get:

$$\mathrm{ALG} = \sum_{t=1}^{T} \mathrm{ALG}^t \leq \mathrm{OPT} + \epsilon \mathrm{OPT} + \frac{\ln n}{\epsilon} \leq \mathrm{OPT} + \epsilon T + \frac{\ln n}{\epsilon} \tag{29}$$

We want to minimize the $T\epsilon + \frac{\ln n}{\epsilon}$ term, so we choose $\epsilon = \sqrt{\frac{\ln n}{T}}$. This gives:

$$\mathrm{ALG} \leq \mathrm{OPT} + 2\sqrt{T \ln n} \tag{30}$$

By definition, the regret is:

$$\mathrm{Regret} = \frac{1}{T}(\mathrm{ALG} - \mathrm{OPT}) \leq 2\sqrt{\frac{\ln n}{T}} \tag{31}$$

One can easily see that as T goes to infinity, the regret goes to 0. This shows that the Multiplicative Weight algorithm is a No-Regret algorithm.

Observation 12 *The number of iterations required to get* Regret $= \alpha$ *is* $\frac{\log n}{\alpha^2}$.

Proof:

$$\mathrm{Regret} \leq 2\sqrt{\frac{\ln n}{T}} = \alpha \Rightarrow T = 4\frac{\log n}{\alpha^2} \tag{32}$$

■

The Multiplicative Weight algorithm converges to α Regret in a number of steps quadratic in $\frac{1}{\alpha}$, and as we have seen, is a No-Regret algorithm.