Computational Game Theory	**Fall Semester, 2012**
Lecture 4: March 14	
Lecturer: Amos Fiat	*Scribe: Gil Tamir and Orit Moskovich*

4.1 Potential Games

4.1.1 Definitions of Potential Games

Definition A game is an *ordinal potential game* if there exists

$\phi : S_1 \times ... \times S_n \to \mathbb{R}\ s.t\ \forall i, s_i, s_{-i}, s'_i . c_i(s_i, s_{-i}) > c_i(s'_i, s_{-i}) \Longleftrightarrow \phi(s_i, s_{-i}) > \phi(s'_i, s_{-i})$

where S_i is player $i's$ strategy.

Definition A game is an *exact potential game* if

$$c_i(s_i, s_{-i}) - c_i(s'_i, s_{-i}) = \phi(s_i, s_{-i}) - \phi(s'_i, s_{-i})$$

Definition A game is a *weighted potential game* if

$$c_i(s_i, s_{-i}) - c_i(s'_i, s_{-i}) = \frac{\phi(s_i, s_{-i}) - \phi(s'_i, s_{-i})}{w_i}$$

Lemma 4.1 *A game is a potential game* $\Longleftrightarrow$ *local improvements always terminate*

Proof Let us define a direct graph ($Improvements\ Graph$) *G=(V,E)* according to the game as follows:

$$V = \{s | s \in S_1 \times ... \times S_n\}$$

$$E = \left\{ \overrightarrow{(s', s)} | \begin{array}{c} s, s' \in V,\ s'\ differs\ from\ s\ only\ in\ the\ strategy \\ of\ a\ single\ player\ i,\ c_i(s_i, s_{-i}) > c_i(s'_i, s_{-i}) \end{array} \right\}$$

We'll prove the following claim, and the lemma will derive.

Claim 4.2 *A potential function exists* $\Longleftrightarrow$ *G does not contain cycles.*

Proof

- Assume G contains a cycle $(s_1, ..., s_k)$ and assume by contradiction that a potential function ϕ does exist. Thus, according to the definition of potential game, and the construction of the graph,

$$(s_1, s_2), ..., (s_{k-1}, s_k), (s_k, s_1) \in E \Longrightarrow \phi(s_1) > \phi(s_k) > \phi(s_{k-1}) > ... > \phi(s_1)$$

- Assume G does not contain any cycle. Let us define a potential function ϕ : for every sink $\hat{s}$ (a node without outgoing edges), $\phi(\hat{s}) = 0$. For every node with an outgoing edge towards one of the sinks, define it's potential to *1*. In general, on the i^{th} step, every undefined node s which points to a node defined in the previous steps, define $\phi(s) = i$. Since G contains no cycles, and due to the construction of the graph, it's easy to see that ϕ is a legal potential function.

4.1.2 Examples for Potential Games

In the following examples we'll describe a few games in which the table representing the game also represents the corresponding improvements graph.

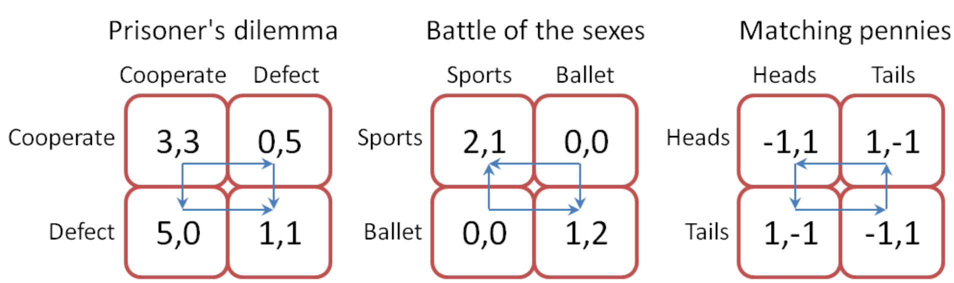

Figure 4.1: Potential games

In the prisoner's dilemma game, it's easy to see that the right bottom square is a Nash Equilibrium, and in fact a sink in the improvements graph.

In the battle of the sexes game there are two sinks, both of them are representing a pure Nash Equilibrium.

And in the matching pennies game, there is no pure Nash Equilibrium in this game (there is a mixed NE, shown in lecture no.???). This result can also be explained by detecting the cycle in the graph.

4.1.3 Properties of Potential Games

1. Admit a pure strategy Nash Equilibrium
2. Best response dynamics converge to Nash Equilibrium- Start with some strategy profile. While exists a player who can switch to a better response, do so. Terminate when no player can improve, and reach a NE.
3. Price of stability is bounded (proved later on)

Theorem 4.3 *Every potential game admits a pure NE*

Proof Let s be a pure profile minimizing ϕ. We'll prove that s is a NE. Suppose by contradiction that s is not a NE. Therefore, player i can improve by deviating to a new profile s' $s.t$ $c_i(s') - c_i(s) < 0 \rightarrow \phi(s') - \phi(s) < 0$. Thus, $\phi(s') < \phi(s)$, contradicting that s minimizes ϕ.

4.1.4 Identical Machines Load Balancing

Game definition

Consider the following load balancing game:

- M identical machines
- N jobs
- $(w_i)_{i \in N}$ implies the weight of each job
- For each machine, we define the load on the machine as the sum of the jobs working on that machine: define $J = \{\, i \mid job\, i\, is\, on\, machine\, j \,\}$, than $L_j = \sum_{i \in J} w_i$

- The cost function of each player, denoted by c_i, is defined as the load of the machine on which job i is working.

Weighted potential game

We'll now define a potential function: $\phi = \frac{1}{2}\sum_{j=1}^{M} L_j^2$

W.L.O.G, let's examine the case where job j moves from it's original machine M_2 to another machine M_3 :

$$\Delta(\phi) = \phi(a') - \phi(a) = \frac{1}{2}\left((L_3 + w_j)^2 - L_3^2 - \left(L_2^2 - (L_2 - w_j)^2\right)\right)$$

$$= L_3^2 + 2w_j L_3 + w_j^2 - L_3^2 - L_2^2 + L_2^2 - 2w_j L_2 + \frac{w_j^2}{2}$$

$$= w_j(L_3 - L_2) + w_j^2 = w_j(L_3 + w_j - L_2) \overset{*}{=} w_j \Delta(C_j)$$

(*) The last equation is derived from the fact that $L_3 + w_j - L_2 = \Delta(C_j) < 0$ (since otherwise, job j will be better off staying on machine M_2)

Thus, the game we described is a weighted potential game.

4.1.5 Unrelated Machines

Let's look at a slightly different game:

Game definition

- M related machines
- N jobs
- $(w_{i,j})_{i \in N,\, j \in M}$ implies the weight of job i when is working in machine j
- For each machine, we define the load on the machine as the sum of the jobs working on that machine: define $J = \{\, i \mid \textit{job } i \textit{ is on machine } j \,\}$, than $L_j = \sum_{i \in J} w_{i,j}$

- The cost function of each player, denoted by c_i, is defined as the load of the machine on which job i is working.

Ordinal potential game

We'll now show that this game is an ordinal potential game, by defining an appropriate potential function: $\phi = \sum_{j=1}^{M} 4^{L_j}$
Again, we'll examine the case where a single job moves from it's original machine M_1 to another machine M_2:

$$\Delta(\phi) = \phi(a') - \phi(a) = 4^{L_1 - w_{i,1}} + 4^{L_2 + w_{i,2}} - 4^{L_1} - 4^{L_2}$$

Now, to show that this game is an ordinal potential game, we need to show that

$$\Delta(C_i) < 0 \iff \Delta(\phi) < 0$$

Indeed,
(1) $\Delta(C_i) < 0 \iff L_2 + w_{i,2} - L_1 < 0 \iff L_2 + w_{i,2} < L_1$
(2) $L_1 - w_{i,1} < L_1$(assuming $w_{i,1} > 0$)
From *(1)* + *(2)* we conclude that

$$\Delta(\phi) = 4^{L_1 - w_{i,1}} + 4^{L_2 + w_{i,2}} - 4^{L_1} - 4^{L_2} < 4^{L_1} + 4^{L_1} - 4^{L_1} - 4^{L_2} = 4^{L_1} - 4^{L_2} < 0$$

Assuming that $L_1 < L_2$ (otherwise, player i has no incentive to move from M_1 to M_2).
Thus, we can conclude that this game is an ordinal potential game.

4.1.6 Network Games

Network Construction

Definition *Multicast routing* Given a directed graph $G = (V, E)$ with edge costs $c_e > 0$, a source node s, and k agents located at terminal nodes $t_1, ..., t_k$.

Agent j must construct a path P_j from node s to its terminal t_j.

Definition *Routing* Given a directed graph $G = (V, E)$ with edge costs $c_e > 0$, and k agents seeking to connect s_i, t_i pairs, agent j must construct a path P_j from node s_j to its terminal t_j.

Definition *Fair share* If x agents use edge e, they each pay $\frac{c_e}{x}$.

Multicast Routing Example: Shapely Price Sharing (fair cost sharing)

In the following example (see fig. 4.2) there are two agents A_1, A_2, seeking to construct paths P_1, P_2 from s to t_1, t_2, respectively. At first, both agents choose the outer paths. A_1 has no incentive to switch paths, since moving to the middle path will cost him more (the cost of the outer path is 4 as appose to the middle path which costs $5 + 1 = 6$. On the other hand, A_2 does have an incentive to move, since the middle path costs less that to outer one. But once this happens, A_1 will want to move to the middle path, since now it costs $\frac{5}{2} + 1$ (using the fare share cost). Finally, both agents using the middle path is a NE.

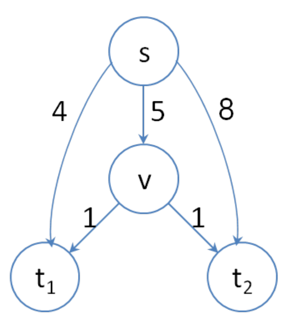

Figure 4.2: Shapely Price Sharing

4.1.7 Social Optimum

First, recall the definitions from lecture no.2:

$$Price\ of\ Anarchy\ (PoA) = \frac{cost\ of\ worst\ NE}{cost\ of\ OPT}$$

$$Price\ of\ Stability\ (PoS) = \frac{cost\ of\ best\ NE}{cost\ of\ OPT}$$

Now, define the *social optimum* to be the strategy that minimizes total costs of all agents.

Observation: In general, there can be many NEs. Even when it is unique, it does not necessarily equal the social optimum.

Examples

Let's examine two games. In each game will see the best and worst NE, and the social optimum.

First Example

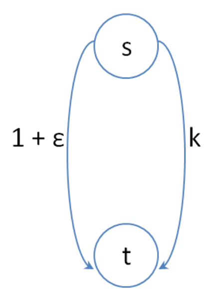

Figure 4.3: single source and destination

The first example (fig. 4.3) describes a game in which agents are willing to route from node s to node t.

- *Social optimum* $= 1 + \epsilon$ (consider the case when all agents route via the left edge, the total cost is $1 + \epsilon$)
- *Best NE* $= 1 + \epsilon$ (the social optimum is actually a NE)
- *Worst NE* $= k$ (consider the case when all k agents route via the right edge)

$\implies PoS = 1,\ PoA = k$

Second Example

The second example (fig. 4.4) describes a game where there are two agents, one routing from s to t_1 and the other from s to t_2

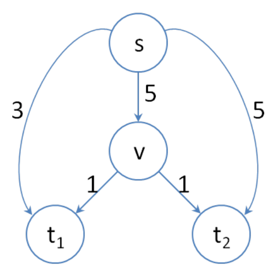

Figure 4.4: single source and multiple destinations

- *Social optimum* $= 7$ (consider the case when both agents route via the middle path, with Shapley cost sharing)
- *Unique NE* $= 8$ (both agents use the outer paths)

$\implies PoS = PoA = \frac{8}{7}$

Lemma 4.4 $PoA \leq k$

Proof Let N be the worst NE, and suppose by contradiction that $c(N) > k \cdot OPT$. Then, there exists a player i $s.t$ $c_i(N) > OPT$. But in this case, player i can deviate to OPT (by paying OPT alone), contradicting that N in a NE.

Note: this bound is tight, considering the first example, where this bound is achieved.

4.1.8 Network Games

Multicast Routing: Fair Share

Definition The *harmonic* function is defined as $H(n) := \sum_{i=1}^{n} \frac{1}{i}$

Theorem 4.5 *multicast shapely price routing is an exact Potential game*

Proof We define for the game an appropriate *exact potential* function Φ

First Attempt Define $\Phi(S) := \sum_{i=1}^{k} c_i(S)$.

In order for Φ to be an *exact potential* function, when an agent changes his strategy, the change in Φ must be identical to the change in that agent's cost. Φ defined to be the overall cost of the agents does not satisfy this (fig. 4.5). Agent 1 switches sides, and his cost is lowered, yet since a new edge is added to the chosen edges, the overall cost is increased.

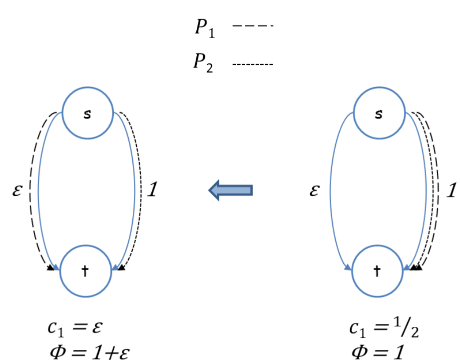

Figure 4.5: Potential games

Second Attempt Let $(P_1, ..., P_k)$ be the selected paths of the k agents, and x_e be the number of selected paths that use a given edge e. Define $\Phi(P) := \sum_{e \in E} c_e \cdot H(x_e)$.

To prove Φ is an *exact potential* function, consider agent j changing his selected path from P_j to P'_j. The net increase in agent j's cost is:

$$\sum_{e \in P'_j \setminus P_j} \frac{c_e}{x_e + 1} - \sum_{e \in P_j \setminus P'_j} \frac{c_e}{x_e}$$

Φ increases by:

$$\sum_{e \in P'_j \setminus P_j} c_e \cdot [H(x_e + 1) - H(x_e)] = \sum_{e \in P'_j \setminus P_j} \frac{c_e}{x_e + 1}$$

Φ decreases by:

$$\sum_{e \in P_j \setminus P'_j} c_e \cdot [H(x_e) - H(x_e - 1)] = \sum_{e \in P_j \setminus P'_j} \frac{c_e}{x_e}$$

We see the change in Φ is identical to the change in cost for player j, thus Φ is an *exact potential* function for the game. □

Claim 4.6 *Let $(P_1, ..., P_k)$ be the selected paths of the k agents, $C(P_1, ..., P_k)$ be the overall cost of these paths, and Φ be the exact potential function defined in the previous theorem. For any such selected paths the following holds:*

$$C(P_1, ..., P_k) \leq \Phi(P_1, ..., P_k) \leq H(k) \cdot C(P_1, ..., P_k)$$

Proof Let x_e denote the number of selected paths that use a given edge e, and E^+ denote the set of edges used by at least one selected path. By the definitions of Φ and $H(n)$ we have:

$$C(P_1, ..., P_k) = \sum_{e \in E^+} c_e \leq \sum_{e \in E^+} c_e \cdot H(x_e) = \phi(P_1, ..., P_k)$$

Since the number of players is k, we have:

$$\phi(P_1, ..., P_k) \leq \sum_{e \in E^+} c_e \cdot H(k) = H(k) \cdot \sum_{e \in E^+} c_e = H(k) \cdot C(P_1, ..., P_k)$$

□

Theorem 4.7 *For all multicast shapely price routing games, there is a Nash equilibrium for which the overall cost of all agents exceeds that of the social optimum by at most a factor of H(k). That is, the price of stability for all multicast shapely price routing is $\leq$ H(k), which converges to log(k) as k tends to infinity.*

Proof Let $P^* := (P_1^*, ..., P_k^*)$ denote a selection of paths providing the socially optimal overall cost. Consider the run of the *best response dynamics* algorithm starting from P^*.

As we've shown previously, the algorithm must converge to a Nash equilibrium, which we denote $P := (P_1, ..., P_k)$. Applying the lower bound of the previous claim to P, we have:

$$C(P_1, ..., P_k) \leq \Phi(P_1, ..., P_k)$$

Since in every step of the algorithm the changing agent's cost is lowered, so is the value of Φ, and we have:

$$\Phi(P_1, ..., P_k) \leq \Phi(P_1^*, ..., P_k^*)$$

Applying the upper bound of the previous claim to P^*, we have:

$$\Phi(P_1^*, ..., P_k^*) \leq H(k) \cdot C(P_1^*, ..., P_k^*)$$

Hence, assuming a non-zero socially optimum cost, we have

$$PoS \leq \frac{C(P)}{C(P^*)} \leq H(k)$$

□

We will now see this upper bound is tight. That is, no better upper bound exists that holds for all multicast shapely price routing games.

Claim 4.8 *There exists a multicast shapely price routing game for which the price of stability is* $\frac{H(k)}{(1+\epsilon)}$

Proof Consider the game illustrated in fig. 4.6.

The social optimum is achieved by every agent taking the right path, for an overall cost of $(1 + \epsilon)$.

The unique Nash equilibrium is achieved by every agent taking the corresponding top path, for an overall cost of $H(k)$. □

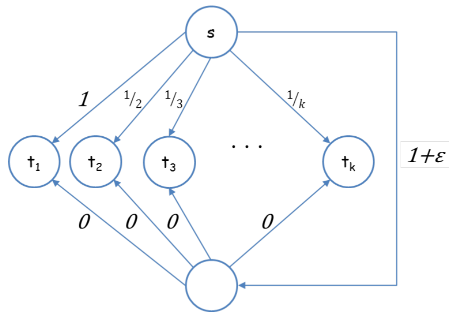

Figure 4.6: Upper bound tightness

Since the claim above is true for every $\epsilon > 0$, the $H(k)$ bound is tight.

Undirected Multicast Routing: Fair Share

Bounding the PoS of multicast shapely price routing games when the edges are undirected is currently a major open research problem.

• Is the $H(k)$ upper bound still tight? The case used to prove the directed case does not carry over (fig. 4.7), where the social optimum with cost $\frac{1}{k}$ is a Nash equilibrium.

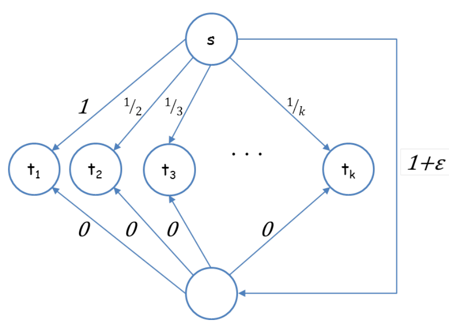

Figure 4.7: Upper bound tightness

• For the subset of games where there is an agent in every node and a single source for all agents, an upper bound of $O(loglog(k)$ is known.

4.2 Congestion Games (Rosenthal 73)

Definition A congestion game is defined as follows:

• A set R of resources.

• A set of k agents.

• For each agent i with with a set of strategies $A_i \subseteq \mathcal{P}(R)$, representing all the subsets of resources which would meet that agent's needs.

• For each resource $r \in R$, a cost function $c_r : \{1, ..., k\} \rightarrow \Re$.

For any $a := (a_i, a_{-i})$ selection of strategies by the k players, Let $n_r(a)$ be the number of players that chose r in their strategy, and the cost for each agent i is $\sum_{r \in a_i} c_r(n_r(a))$.

Example Multicast routing is a congestion game.

• The set resources is defined to be the set of edges

• For each agent i the set of strategies A_i is defined to be the set of possible paths connecting it's source to it's destination.

• For each edge r, with routing cost $rcost_r$, we define the cost function $c_r := \frac{rcost_r}{n_r}$.

In a later lecture, it will be shown that not only multicast routing, but all exact potential games are congestion games.

Theorem 4.9 *Every congestion game is an exact potential game*

Proof We define an *exact potential* function Φ:

$$\Phi(a) := \sum_{r \in R} \sum_{j=1}^{n_r(a)} c_r(j)$$

One interpretation of this definition is each agent selecting a resource ignores the effects of all later agents.

As an agent switches strategy, and starts using new resources R^+ and stops using previous resources R^-, the net increase in the agent's cost is:

$$\sum_{r\in R^+} c_r(n_r(a)+1) - \sum_{r\in R^-} c_r(n_r(a))$$

This is exactly the change in the value of Φ as well, thus it is an exact potential function. □

Example Consider an existing network of edges, with multiple sources $\{s_i\}$ and destinations $\{t_i\}$ among the nodes. Each edge e has a monotonic increasing cost function based on the number of agents selecting it.

By definition this is a congestion game.

Claim 4.10 *Finding a Nash equilibrium in the above congestion game reduces to solving a minimal cost maximum flow problem.*

Proof Finding a Nash equilibrium in an exact potential game (and hence in a congestion game) is equivalent to finding a local minimum of the potential function.

We construct a min cost max flow problem such that a minimal cost flow implies a set of strategies of the k players in the congestion game.

Remove all edges (u, v), and for all $i \in \{1, ..., k\}$, add an edge connecting u to v with capacity 1 and cost $c_{(u,v)}(i)$. Add a master source connected to all sources $\{s_i\}$ and a master destination connected to all destinations $\{t_i\}$, with each of these additional edges e having capacity 1 and cost 0.

The min cut (and hence the max flow) is k, as every two connected non-master nodes have overall connecting capacity k, and both master nodes have overall connecting capacity k. Since every edge has capacity 1, every flow implies a selected path for each agent (fig. 4.8)

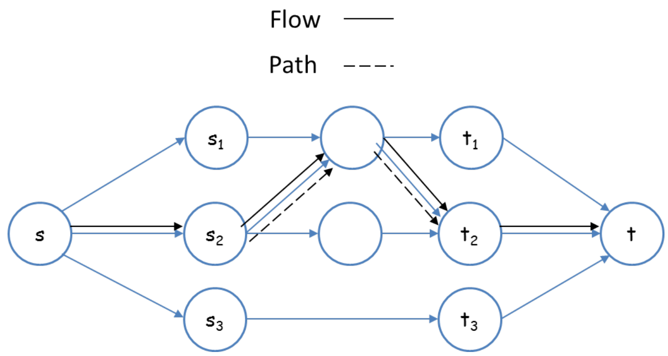

Figure 4.8: Upper bound tightness

In a minimal cost flow, parallel edges of the same capacity will be chosen in increasing order according to their cost. Since the cost functions are monotonic increasing, if l edges are selected between u and v, the contribution of (u, v) to the cost of the flow is $\sum_{j=1}^{l} c_r(j)$, and the overall cost of the flow is $\sum_{r\in R}\sum_{j=1}^{n_r(a)} c_r(j)$. This is exactly $\Phi(a)$, where Φ is the universal exact potential function previously defined, and a is the implied set of strategies. Since the the min flow cost is the global minimum, $\Phi(a)$ is the global minimum of the potential function, and a is a Nash equilibrium. $\square$