L06 Potential and Congestion Games

CS 295 Introduction to Algorithmic Game Theory

Ioannis Panageas

Potential Games

Definition (Potential Games). *A normal form game is specified by*

- *set of n players* $[n] = \{1, ..., n\}$
- *For each player i a set of strategies/actions S_i and a utility $u_i : \times_{j=1}^{n} S_j \to \mathbb{R}$ denoting the payoff of i.*
- *set of strategy profiles* $S = S_1 \times ... \times S_n$.
- *There exists a potential function $\Phi : S \to \mathbb{R}$ so that for all agents i and s_i, s_i'*

$$\Phi(s_i, s_{-i}) - \Phi(s_i', s_{-i}) = u_i(s_i, s_{-i}) - u_i(s_i', s_{-i}).$$

Potential Games

Definition (Potential Games). *A normal form game is specified by*

- *set of* n *players* $[n] = \{1, ..., n\}$
- *For each player* i *a set of strategies/actions* S_i *and a utility* $u_i : \times_{j=1}^{n} S_j \to \mathbb{R}$ *denoting the payoff of* i.
- *set of strategy profiles* $S = S_1 \times ... \times S_n$.
- *There exists a potential function* $\Phi : S \to \mathbb{R}$ *so that for all agents* i *and* s_i, s_i'

$$\Phi(s_i, s_{-i}) - \Phi(s_i', s_{-i}) = u_i(s_i, s_{-i}) - u_i(s_i', s_{-i}).$$

Example (Battle of sexes).

5, 2	-1,-2
-5, -4	1,4

$\Rightarrow$

4	0
-6	2

Potential Games

Definition (Potential Games). *A normal form game is specified by*

- *set of n pl*
- *For each p*
  $u_i : \times_{j=1}^{n} S$
- *set of strat*
- *There exists a potential function $\Phi : S \to \mathbb{R}$ so that for all agents i and s_i, s_i'*

$$\Phi(s_i, s_{-i}) - \Phi(s_i', s_{-i}) = u_i(s_i, s_{-i}) - u_i(s_i', s_{-i}).$$

Weighted Potential Games:

$$\Phi(s_i, s_{-i}) - \Phi(s_i', s_{-i}) = w_i \cdot \left(u_i(s_i, s_{-i}) - u_i(s_i', s_{-i})\right),$$

where $w_i > 0$.

Example (Battle of sexes).

5, 2	-1,-2
-5, -4	1,4

$\Longrightarrow$

4	0
-6	2

Potential Games

Question: What is interesting about these games?

Answer: A pure Nash equilibrium always exists!

Lemma. *Let G be a potential game. It has a pure Nash equilibrium.*

Potential Games

Question: What is interesting about these games?

Answer: A pure Nash equilibrium always exists!

Lemma. *Let G be a potential game. It has a pure Nash equilibrium.*

Proof. Let s^* a pure strategy profile that minimizes Φ. Then s^* is a Nash equilibrium.

Potential Games

Question: What is interesting about these games?

Answer: A pure Nash equilibrium always exists!

Lemma. *Let G be a potential game. It has a pure Nash equilibrium.*

Proof. Let s^* a pure strategy profile that minimizes Φ. Then s^* is a Nash equilibrium. Assuming not, there is an agent i and strategy s_i' so that

$$u_i(s_i^*, s_{-i}^*) < u_i(s_i', s_{-i}^*).$$

Potential Games

Question: What is interesting about these games?

Answer: A pure Nash equilibrium always exists!

Lemma. *Let G be a potential game. It has a pure Nash equilibrium.*

Proof. Let s^* a pure strategy profile that minimizes Φ. Then s^* is a Nash equilibrium. Assuming not, there is an agent i and strategy s_i' so that

$$u_i(s_i^*, s_{-i}^*) < u_i(s_i', s_{-i}^*).$$

$$\Phi(s_i^*, s_{-i}^*) - \Phi(s_i', s_{-i}^*) = u_i(s_i^*, s_{-i}^*) - u_i(s_i', s_{-i}^*) < 0.$$

Contradiction!

Potential Games

Algorithm (Greedy).

1. Initialize $s^{(0)}$ arbitrarily.
2. **Loop**
3. **Find** agent i, s'_i so that $u_i(s'_i, s^{(t)}_{-i}) > u_i(s^{(t)})$
4. **Set** $s^{(t+1)} = (s'_i, s^{(t+1)}_{-i})$.
5. $t = t + 1$
6. **If** no agent exists **STOP**

Lemma. *The algorithm above reaches a pure Nash equilibrium.*

Potential Games

Algorithm (Greedy).

1. Initialize $s^{(0)}$ arbitrarily.
2. **Loop**
3. **Find** agent i, s_i' so that $u_i(s_i', s_{-i}^{(t)}) > u_i(s^{(t)})$
4. **Set** $s^{(t+1)} = (s_i', s_{-i}^{(t+1)})$.
5. $t = t + 1$
6. **If** no agent exists **STOP**

Lemma. *The algorithm above reaches a pure Nash equilibrium.*

Proof. Construct a directed graph with $|S|$ vertices and an edge from $s \to s'$ if strategy profiles s, s' differ in one agent only, say i and $u_i(s') > u_i(s)$.

- The graph has no cycles.

Potential Games

Algorithm (Greedy).

1. Initialize $s^{(0)}$ arbitrarily.
2. **Loop**
3. **Find** agent i, s_i' so that $u_i(s_i', s_{-i}^{(t)}) > u_i(s^{(t)})$
4. **Set** $s^{(t+1)} = (s_i', s_{-i}^{(t+1)})$.
5. $t = t + 1$
6. **If** no agent exists **STOP**

Lemma. *The algorithm above reaches a pure Nash equilibrium.*

Proof. Construct a directed graph with $|S|$ vertices and an edge from $s \to s'$ if strategy profiles s, s' differ in one agent only, say i and $u_i(s') > u_i(s)$.

- The graph has no cycles.
- The algorithm reaches a sink vertex (no outgoing edges).

Congestion Games

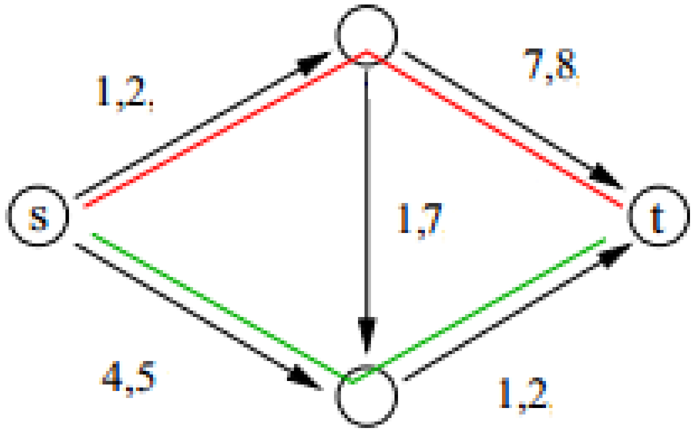
A congestion game is defined by:

- n set of players.
- E set of edges/facilities/ bins.
- $S_i \subset 2^E$ the set of strategies of player i.
- $c_e : \{1, ..., n\} \to \mathbb{R}^+$ cost function of edge e.

For any $s = (s_1, ..., s_n)$

- $l_e(s)$ number of players (load) that use edge e.
- $c_i(s) = \sum_{e \in s_i} c_e(l_e)$ the cost function of player i.

Congestion Games



For this game:

$n = \{1, 2\}$ (red, green)
E are the edges of the network.
S_i is all $s - t$ paths.
c_e on edges.

Remark: Defined by Rosenthal in 1973. Capture atomic routing **games!**

Congestion Games are Potential Games

Theorem (Rosenthal 73′)**.** *Congestion Games are potential games.*

Proof. We need to come up with a potential function!

Congestion Games are Potential Games

Theorem (Rosenthal 73′). *Congestion Games are potential games.*

Proof. We need to come up with a potential function!

Consider the function $\Phi(s) = \sum_{e \in E} \sum_{j=1}^{l_e(s)} c_e(j)$.

Congestion Games are Potential Games

Theorem (Rosenthal 73′). *Congestion Games are potential games.*

Proof. We need to come up with a potential function!

Consider the function $\Phi(s) = \sum_{e \in E} \sum_{j=1}^{l_e(s)} c_e(j)$.
Fix agent i strategy s, s', where s, s' differ on agent's i strategy.

- $\Phi(s) = \sum_{e \in s \cap s'} \sum_{j=1}^{l_e(s)} c_e(j) + \sum_{e \in s \setminus s'} \sum_{j=1}^{l_e(s)} c_e(j) + \sum_{e \in s' \setminus s} \sum_{j=1}^{l_e(s)} c_e(j)$.
- $\Phi(s') = \sum_{e \in s \cap s'} \sum_{j=1}^{l_e(s')} c_e(j) + \sum_{e \in s' \setminus s} \sum_{j=1}^{l_e(s')} c_e(j) + \sum_{e \in s \setminus s'} \sum_{j=1}^{l_e(s')} c_e(j)$.

Congestion Games are Potential Games

Theorem (Rosenthal 73'). *Congestion Games are potential games.*

Proof. We need to come up with a potential function!

Consider the function $\Phi(s) = \sum_{e \in E} \sum_{j=1}^{l_e(s)} c_e(j)$.
Fix agent i strategy s, s', where s, s' differ on agent's i strategy.

- $\Phi(s) = \sum_{e \in s \cap s'} \sum_{j=1}^{l_e(s)} c_e(j) + \sum_{e \in s \setminus s'} \sum_{j=1}^{l_e(s)} c_e(j) + \sum_{e \in s' \setminus s} \sum_{j=1}^{l_e(s)} c_e(j)$.
- $\Phi(s') = \sum_{e \in s \cap s'} \sum_{j=1}^{l_e(s')} c_e(j) + \sum_{e \in s \setminus s'} \sum_{j=1}^{l_e(s')} c_e(j) + \sum_{e \in s' \setminus s} \sum_{j=1}^{l_e(s')} c_e(j)$.

Same load $\qquad l_e(s) = l_e(s') + 1 \qquad l_e(s) = l_e(s') - 1$

Congestion Games are Potential Games

Theorem (Rosenthal 73'). *Congestion Games are potential games.*

$$\Phi(s) - \Phi(s') = \sum_{e \in s \setminus s'} c_e(l_e(s)) - \sum_{e \in s' \setminus s} c_e(l_e(s'))$$

- $\Phi(s) = \sum_{e \in s \cap s'} \sum_{j=1}^{l_e(s)} c_e(j) + \sum_{e \in s \setminus s'} \sum_{j=1}^{l_e(s)} c_e(j) + \sum_{e \in s' \setminus s} \sum_{j=1}^{l_e(s)} c_e(j).$
- $\Phi(s') = \sum_{e \in s \cap s'} \sum_{j=1}^{l_e(s')} c_e(j) + \sum_{e \in s \setminus s'} \sum_{j=1}^{l_e(s')} c_e(j) + \sum_{e \in s' \setminus s} \sum_{j=1}^{l_e(s')} c_e(j).$

Same load $\qquad l_e(s) = l_e(s') + 1 \qquad l_e(s) = l_e(s') - 1$

Congestion Games are Potential Games

Proof cont. $\boxed{\Phi(s) - \Phi(s') = \sum_{e \in s \setminus s'} c_e(l_e(s)) - \sum_{e \in s' \setminus s} c_e(l_e(s'))}$

- $u_i(s) = \sum_{e \in s \cap s'} c_e(l_e(s)) + \sum_{e \in s \setminus s'} c_e(l_e(s))$.
- $u_i(s') = \sum_{e \in s \cap s'} c_e(l_e(s')) + \sum_{e \in s' \setminus s} c_e(l_e(s'))$.

Same load

Congestion Games are Potential Games

Proof cont. $\boxed{\Phi(s) - \Phi(s') = \sum_{e \in s \setminus s'} c_e(l_e(s)) - \sum_{e \in s' \setminus s} c_e(l_e(s'))}$

- $u_i(s) = \sum_{e \in s \cap s'} c_e(l_e(s)) + \sum_{e \in s \setminus s'} c_e(l_e(s))$.
- $u_i(s') = \sum_{e \in s \cap s'} c_e(l_e(s')) + \sum_{e \in s' \setminus s} c_e(l_e(s'))$.

Same load

$$\boxed{\Rightarrow u_i(s) - u_i(s') = \sum_{e \in s \setminus s'} c_e(l_e(s)) - \sum_{e \in s' \setminus s} c_e(l_e(s'))}$$

We conclude that $\Phi(s) - \Phi(s') = u_i(s) - u_i(s')$.

Congestion Games are Potential Games

Proof cont. $\boxed{\Phi(s) - \Phi(s') = \sum_{e \in s \setminus s'} c_e(l_e(s)) - \sum_{e \in s' \setminus s} c_e(l_e(s'))}$

- $u_i(s) = \sum_{e \in s \cap s'} c_e(l_e(s)) + \sum_{e \in s \setminus s'} c_e(l_e(s))$.
- $u_i(s') = \sum_{e \in s \cap s'} c_e(l_e(s')) + \sum_{e \in s' \setminus s} c_e(l_e(s'))$.

Same load

$$\boxed{\Rightarrow u_i(s) - u_i(s') = \sum_{e \in s \setminus s'} c_e(l_e(s)) - \sum_{e \in s' \setminus s} c_e(l_e(s'))}$$

We conclude that $\Phi(s) - \Phi(s') = u_i(s) - u_i(s')$.

Remark: Monderer and Shapley showed that potential games can be reduced to congestion games!

An Algorithm for symmetric network congestion games

Assumption: All players have the same endpoints S and T (and thus they all have the same set of paths/strategies).

Basic idea: Min-cost flow reduction

An Algorithm for symmetric network congestion games

Assumption: All players have the same endpoints S and T (and thus they all have the same set of paths/strategies).

Basic idea: Min-cost flow reduction

Definitio... *and a sink t we w*...

- *Each*... $a(u,v)$.

Min-cost flow via LP!

$$\min \sum_{e:(u,v)} f(u,v) \cdot a(u,v)$$

s.t $f(u,v) \leq c(u,v)$ for all edges (u,v) capacity cosntraints

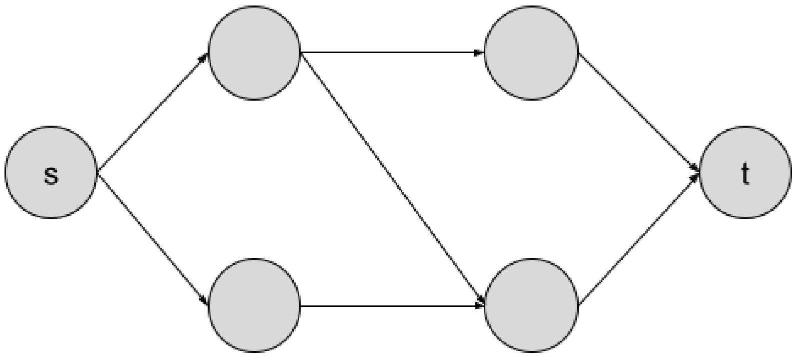
$f(u,v) = -f(v,u)$ for all edges (u,v)

$\sum_w f(u,w) = 0 \;\; \forall u \neq s,t$ flow conservation

$\sum_w f(s,w) = d$ and $\sum_w f(w,t) = d$

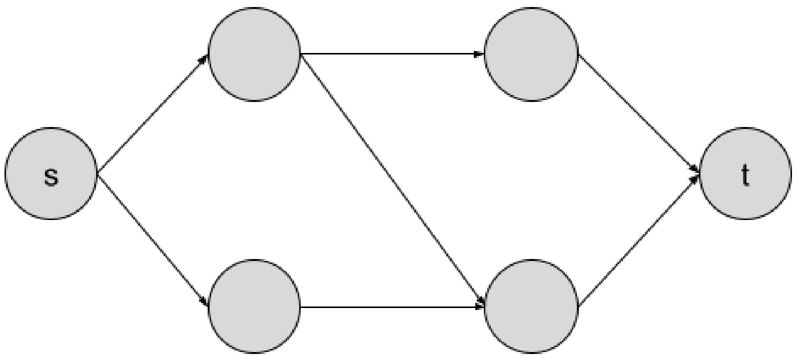
An Algorithm for symmetric network congestion games; the reduction

Initial graph in the Congestion Game.

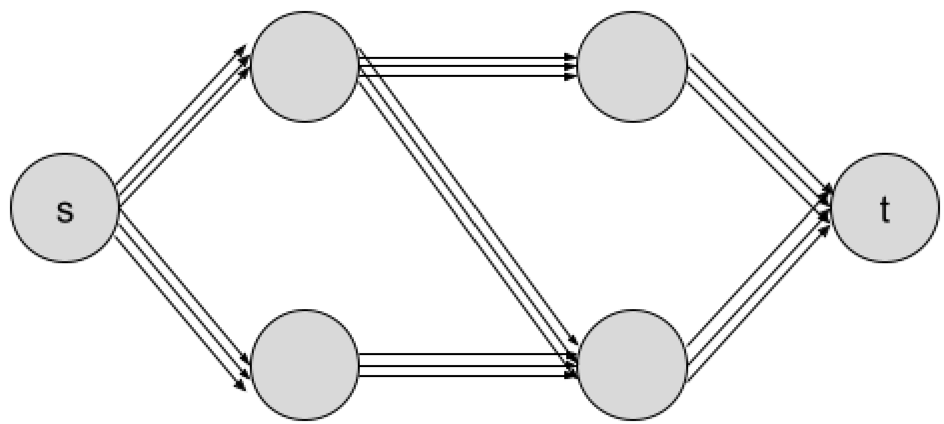


An Algorithm for symmetric network congestion games; the reduction

Initial graph in the Congestion Game.

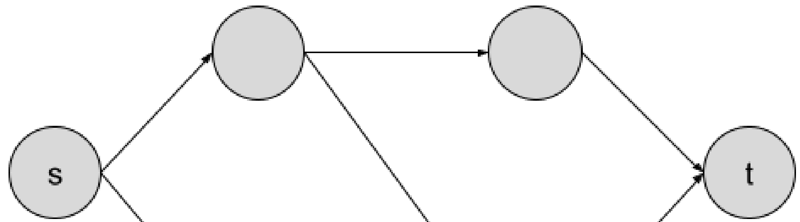


Create another graph with same vertices and for each edge $e := (u, v)$ add n parallel edges of capacity one and costs in increasing order $c_e(1), \ldots, c_e(n)$



An Algorithm for symmetric network congestion games; the reduction

Initial graph in the Congestion Game.



The min-cost flow minimizes the potential Φ! HW2

Create another graph with same vertices and for each edge e $:= (u, v)$ add n parallel edges of capacity one and costs in increasing order $c_e(1), \ldots, c_e(n)$