Inefficiency of equilibria, and potential games

Computational game theory
Spring 2009

Michal Feldman

Inefficiency of equilibria

- Outcome of rational behavior might be inefficient
- How to measure inefficiency?
  - E.g., prisoner's dilemma

3,3	0,5
5,0	1,1

- Define an objective function
  - Social welfare (= sum of players' payoffs): utilitarian
  - Maximize $\min_i u_i$ (egalitarian)
  - …

Inefficiency of equilibria

- To measure inefficiency we need to specify:
  - Objective function
  - Definition of approximately optimal
  - Definition of an equilibrium
  - If multiple equilibria exist, which one do we consider?

Common measures

- Price of anarchy (poa)=cost of worst NE / cost of OPT
- Price of stability (pos)=cost of best NE / cost of OPT
  - Note: poa, pos ≥ 1 (by definition)
- Approximation ratio: Measures price of limited computational resources
- Competitive ratio: Measures price of not knowing future
- Price of anarchy: Measures price of lack of coordination

Price of anarchy

Prisoner's dilemma

3,3	0,5
5,0	1,1

- Example: in prisoner's dilemma, poa = pos = 3
  - But can be as large as desired
- Wish to find games in which pos or poa are bounded
  - NE "approximates" OPT
  - Might explain Internet efficiency.
- Suppose we define poa and pos w.r.t. NE in pure strategies
  - we first need to prove existence of pure NE

Max-cut game

- Given undirected graph G = (V,E)
- players are nodes v in V
- An edge (u,v) means u "hates" v (and vice versa)
- Strategy of node i: $s_i \in \{\text{Black},\text{White}\}$
- Utility of node i: num of neighbors with different color
- <u>Lemma</u>: for every graph G, corresponding game has a pure NE

Proof 1

- Claim: OPT of max-cut defines a NE
- Proof:
  - Define strategies of players by cut (i.e., one side is Black, other side is White)
  - Suppose a player i wishes to switch strategies: i's benefit from switching = improvement in value of the cut
  - Contradicting optimality of cut

Proof 2

- Algorithm greedy-find-cut (GFC):
  - Start with arbitrary partition of nodes into two sets
  - If exists node with more neighbors in own side, move it to other side (repeat until no such node exists)
- Claim 1: GFC provides 2-approx. to max-cut, and runs in polynomial time
- Proof:
  - Poly time: GFC terminates within at most |E| steps (since every step improves the value of the solution in at least 1, and |E| is a trivial upper bound to solution)
  - 2-approx.: Each node ends up with more neighbors in other side than in own side, so at least |E|/2 edges are in cut (since #edges in cut > #edges not in cut)

Proof 2 (cont'd)

- Claim 2: cut obtained by GFC defines a NE
- Proof: obvious, as each player stops only if his strategy is the best response to the other players' strategies
- Conclusion: max-cut game admits a NE in pure strategies

Weighted max-cut

- Question: what about the weighted version of max-cut?
  - Does the greedy algorithm terminate?
  - If so, does it result in a Nash equilibrium?
  - Does it provide approximation to OPT?
  - Does it terminate in polynomial time?

Potential games

- Definition: a game is an ordinal potential game if there exists a function ϕ:S1x…xSn → R s.t. $\forall i,s_i,s_{-i},y_i$, $c_i(s_i,s_{-i}) > c_i(y_i,s_{-i})$ IFF $\phi(s_i,s_{-i}) > \phi(y_i,s_{-i})$
- Def: G is an exact potential game if $c_i(s_i,s_{-i}) - c_i(y_i,s_{-i}) = \phi(s_i,s_{-i}) - \phi(y_i,s_{-i})$
- Def: G is a weighted potential game if $c_i(s_i,s_{-i}) - c_i(y_i,s_{-i}) = w_i(\phi(s_i,s_{-i}) - \phi(y_i,s_{-i}))$
- Example: max-cut is an exact potential game, where ϕ is defined as the cut size
  - Unfortunately, ϕ is not always so natural

Finite Improvement Property

- A path in S is a sequence of states s.t. between every consecutive pair of states there is only one deviator
- A path is an improvement path w.r.t. G if each deviator has a strict advantage

$$c_i(s^k) < c_i(s^{k-1})$$

- G has the FIP if every improvement path is finite
- Clearly if G has the FIP then G has at least one pure equilibrium
  - Every improvement path terminates in an equilibrium point

Generalized Ordinal Potential Function

- Definition: a game is a generalized ordinal potential game if there exists a function ϕ: S1x…xSn → R s.t. $\forall i, s_i, s_{-i}, y_i$,
  $c_i(s_i,s_{-i}) > c_i(y_i,s_{-i})$ implies $\phi(s_i,s_{-i}) > \phi(y_i,s_{-i})$
- Lemma: Let G be a finite game. Then, G has the FIP iff G has a generalized ordinal potential function.

Finite Improvement Property

- Lemma: every finite OP game has the FIP.
- Is the converse true?

G'' =

	A	B
A	(2,0)	(0,0)
B	(1,1)	(2,0)

- G'' has the FIP ((A,B) is an equilibrium)
- any OPF must satisfies the following impossible relations:

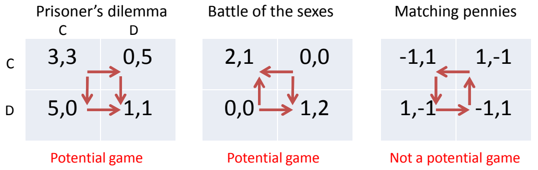
  P(A,A) > P(B,A) > P(B,B) > P(A,B) = P(A,A)

Properties of (all kinds of) potential games

- Admit a Nash equilibrium
- Best-response dynamics converge to NE
- Price of stability is bounded

Examples

→ direction of local improvement



- Example: find a potential function ϕ for prisoner's dilemma:
  - $\phi(C,C)=x$
  - $\phi(C,D)=x+2$
  - $\phi(D,C)=x+2$
  - $\phi(D,D)=x+3$
- Any substitution of x satisfying $\phi(C,D) = \phi(D,C) = \phi(C,C)+2 = \phi(D,D)-1$ is a potential function for prisoner's dilemma

Best-response dynamics converge to a NE

- Best-response dynamics:
  - Start with any strategy profile
  - If a player is not best-responding, switch that player's strategy to a better response (must decrease potential)
  - Terminate when no player can improve (thus a NE)

- Alas, no guarantee on the convergence rate

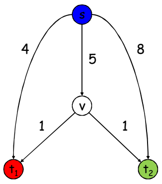
Existence of a pure NE

- Theorem: every potential game admits a pure NE
- Proof: we show that the profile minimizing ϕ is a NE
  - Let s be profile minimizing ϕ
  - Suppose by contradiction it is not a NE, so i can improve by deviating to a new profile s'
  - $\phi(s') - \phi(s) = c_i(s') - c_i(s) < 0$
  - Thus, $\phi(s') < \phi(s)$, contradicting s minimizes ϕ

- More generally, the set of pure-strategy Nash equilibria is exactly the set of local minima of the potential function
  - Local minimum = no player can improve the potential function by herself

Multicast Routing

1	2	1 pays	2 pays
outer	outer	4	8
outer	middle	4	5 + 1
middle	outer	5 + 1	8
middle	middle	5/2 + 1	5/2 + 1



Multicast Routing

- Multicast routing: Given a directed graph $G = (V, E)$ with edge costs $c_e \geq 0$, a source node s, and k agents located at terminal nodes $t_1, \ldots, t_k$. Agent j must construct a path P_j from node s to its terminal t_j.
- Fair share: If x agents use edge e, they each pay c_e / x.

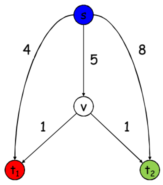
Slides on cost sharing based on slides by Kevin Wayne.
Copyright @ 2005 Pearson-Addison Wesley.
All rights reserved.

Recall price of anarchy and stability

- Price of anarchy (poa)=cost of worst NE / cost of OPT
- Price of stability (pos)=cost of best NE / cost of OPT

Nash Equilibrium

- Example:
  - Two agents start with outer paths.
  - Agent 1 has no incentive to switch paths (since 4 < 5 + 1), but agent 2 does (since 8 > 5 + 1).
  - Once this happens, agent 1 prefers middle path (since 4 > 5/2 + 1).
  - Both agents using middle path is a Nash equilibrium.

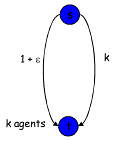


Price of anarchy

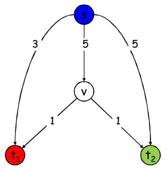
- Claim: poa ≤ k
- Proof:
  - Let s be the worst NE
  - Suppose by contradiction c(s) > k OPT
  - Then, there exists a player i s.t. $c_i(s)$ > OPT
  - But i can deviate to OPT (by paying OPT alone), contradicting s is a NE
- Note: bound is tight (lower bound in prev. slide)

Socially Optimum

- Social optimum: Minimizes total cost to all agent.
- Observation: In general, there can be many Nash equilibria. Even when its unique, it does not necessarily equal the social optimum.



Social optimum = 1 + ε
Nash equilibrium A = 1 + ε
Nash equilibrium B = k



Social optimum = 7
Unique Nash equilibrium = 8

pos=1, poa=k

pos=poa=8/7

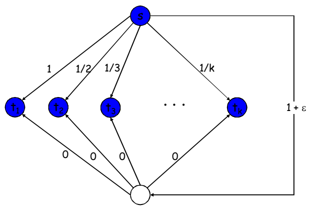
Price of Stability

- What is price of stability in multicast routing?
- Lower bound of log k:

Social optimum: Everyone
Takes bottom paths.

Unique Nash equilibrium:
Everyone takes top paths.

Price of stability:
$H(k) / (1 + \varepsilon)$.
↑
$1 + 1/2 + \ldots + 1/k$



- upper bound will follow..

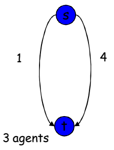
Finding a potential function

Attempt 1:

Let $\Phi(s) = \sum_{j=1}^{k} \text{cost}(t_j)$ be the potential function.

A problem: The potential might increase when some agent improve.

Example: When all 3 agents use the right path, each pays 4/3 and the potential (total cost) is 4.
After one agent moves to the left path the potential increases to 5.



Finding a potential function

Attempt 2:

Consider a set of paths $P_1, \ldots, P_k$.

- Let x_e denote the number of paths that use edge e.
- Let $\Phi(P_1, \ldots, P_k) = \sum_{e \in E} c_e \cdot H(x_e)$ be a potential function.
  H(0) = 0, ↑ $H(k) = \sum_{i=1}^{k} \frac{1}{i}$
- Consider agent j switching from path P_j to path P_j'.
- Change in agent j's cost:

$$\underbrace{\sum_{f \in P_j' - P_j} \frac{c_f}{x_f + 1}}_{\text{newly incurred cost}} < \underbrace{\sum_{e \in P_j - P_j'} \frac{c_e}{x_e}}_{\text{cost saved}}$$

Potential function

- Φ increases by

$$\sum_{f \in P_j' - P_j} c_f \left[H(x_f + 1) - H(x_f)\right] = \sum_{f \in P_j' - P_j} \frac{c_f}{x_f + 1}$$

- Φ decreases by

$$\sum_{e \in P_j - P_j'} c_e \left[H(x_e) - H(x_e - 1)\right] = \sum_{e \in P_j - P_j'} \frac{c_e}{x_e}$$

- Thus, net change in Φ is identical to net change in player j's cost

■

Bounding the Price of Stability

Claim: Let $C(P_1, \ldots, P_k)$ denote the total cost of selecting paths $P_1, \ldots, P_k$.

For any set of paths $P_1, \ldots, P_k$, we have

$$C(P_1, \ldots, P_k) \le \Phi(P_1, \ldots, P_k) \le H(k) \cdot C(P_1, \ldots, P_k)$$

Proof: Let x_e denote the number of paths containing edge e.

- Let E^+ denote set of edges that belong to at least one of the paths.

$$C(P_1, \ldots, P_k) = \sum_{e \in E^+} c_e \le \underbrace{\sum_{e \in E^+} c_e H(x_e)}_{\Phi(P_1, \ldots, P_k)} \le \sum_{e \in E^+} c_e H(k) = H(k)\ C(P_1, \ldots, P_k)$$

Bounding the Price of Stability

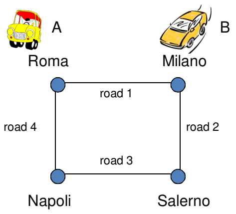
Theorem: There is a Nash equilibrium for which the total cost to all agents exceeds that of the social optimum by at most a factor of H(k) (i.e., price of stability ≤ H(k)).

Proof:

- Let $(P_1^*, \ldots, P_k^*)$ denote set of socially optimal paths.
- Run best-response dyn algorithm starting from P^*.
- Since Φ is monotone decreasing $\Phi(P_1, \ldots, P_k) \le \Phi(P_1^*, \ldots, P_k^*)$.

$$C(P_1, \ldots, P_k) \le \underset{\substack{\uparrow \\ \text{previous claim} \\ \text{applied to P}}}{\Phi(P_1, \ldots, P_k)} \le \Phi(P_1^*, \ldots, P_k^*) \le \underset{\substack{\uparrow \\ \text{previous claim} \\ \text{applied to P*}}}{H(k) \cdot C(P_1^*, \ldots, P_k^*)}$$

Example: road traffic



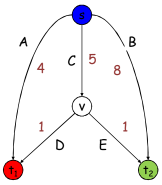
- resources
  - roads 1,2,3,4
- players
  - driver A, driver B
- strategies: which roads I use to reach my destination?
  - A wants to go in Salerno
  - e.g. S_A={{1,2},{3,4}}
  - B wants to go in Napoli
  - e.g. S_B={{1,4},{2,3}}
- what about the costs?

Congestion games [Rosenthal 73]

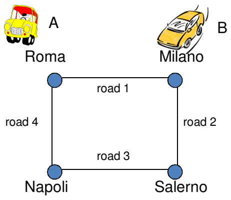
- There is a set of resources R
- Agent i's set of actions (pure strategies) A_i is a subset of 2^R, representing which subsets of resources would meet her needs
  - Note: different agents may need different resources
- There exist cost functions c_r: $\{1, 2, 3, \ldots\} \rightarrow \Re$ such that agent i's cost for a = (a_i, a_{-i}) is $\Sigma_{r \text{ in } a_i} c_r(n_r(a))$
  - $n_r(a)$ is the number of agents that chose r as one of their resources in the profile a

Another example: multicast routing

- Resources = edges
- Each resource r has a cost c_r
- Player 1's action set: {{A}, {C,D}}
- Player 2's action set: {{B}, {C,E}}
- For all resources r, $c_r(n_r(a)) = c_r / n_r(a)$



Road traffic



- A choose path 1,2
- B choose path 1,4
- $c_A = c_1(2) + c_2(1) = 4$
- $c_B = c_1(2) + c_4(1) = 5$

Costs for the roads

$c_1(1)=2$	$c_1(2)= 3$
$c_2(1)=1$	$c_2(2)= 4$
$c_3(1)=4$	$c_3(2)= 6$
$c_4(1)=2$	$c_4(2)= 5$

		B	
		{1,4}	{2,3}
A	{1,2}	(4,5)	(6,8)
	{3,4}	(9,7)	(8,7)

Every exact potential game is isomorphic to a congestion game

- Potential game:
  - n players, k pure strategies each, potential Φ
- Congestion game:
  - n players, k pure strategies each, 2^{kn} resources
  - every resource is associated with $\{0,1\}^{kn}$ vector

- Congestion game:
  - player $1 \leq i \leq n$ plays pure strategy $0 \leq q \leq k-1$: uses all resources r^b where bit $b_{ik+q} = 1$ (2^{kn-1} res.)
  - For every strategy vector s
    - define resource b(s) where $b(s)_{ik+q} = 1$ iff player i uses q in s
    - cost of resource b(s): $\Phi(s)$ for n agents, 0 for less agents
    - There are k^n such resources
  - In addition, for every strategy vector s
    - for agent *i*, let b'(s) be such that $b'(s)_{jk+q} =$ 0 if user $j \neq i$ uses strategy q in s, 1 otherwise.
    - cost of resource b'(s): $u^i(s)-\Phi(s)$ for 1 user, 0 otherwise

- Convince yourself at home that it actually works !

Every congestion game is an exact potential game

- Use potential $\phi(a) = \Sigma_r \Sigma_{i=1}^{nr(a)} c_r(i)$
  - One interpretation: the sum of the costs that the agents would have received if each agent were unaffected by all later agents
- Why is this a correct potential function?
  - Suppose an agent changes action: stop using some resources (R-), start using others (R+)
  - increase in the agent's cost equals $\Sigma_{r \text{ in } R+} c_r(n_r(a) + 1) - \Sigma_{r \text{ in } R-} c_r(n_r(a))$
  - This is exactly the change in the potential function above

– <u>Conclusion</u>: congestion games admit all properties of potential games

Computing NE in congestion games

- Sometimes easy (polytime) – symmetric network congestion games
- Sometimes hard (PLS complete) – general congestion games, symmetric congestion games, general network congestion games

Congestion games: special cases

- Symmetric CG
  - S_i are all the same and payoffs are identical symmetric function of n-1 variables
- Network CG
  - Each player has a starting and terminal node and the strategies are the paths in the network
    - E.g., the two examples above

Symmetric network congestion games (easy) (s –strategy, v – source, confusing)

- Convert to min-cost flow
- Replace edge e in G with n parallel links, of capacity 1, and with costs c(e,1), c(e,2), … c(e,n)
- The cost of a min cost flow of value n, (strategy s), from v to t is equal to the potential function

$$P(s) = \sum_{k=1}^{r} \sum_{y=1}^{n_k(s)} c_k(y)$$

Symmetric network congestion games (easy)

- Graph G=(V,E), source vertex v, sink t
- n players need to find a path from v to t.
- Congestion on edge e is given by c(e,k) where k is the number of players using the edge (c ≥ 0)
- Cost to player is sum of costs of edges player uses.