Games with pure equilibria

Spring 2024

Best response dynamics

Consider a strategic game $\Gamma = \langle N, (A_i)_{i\in N}, (u_i)_{i\in N}\rangle$ with $N = \{1, \dots, n\}$

- PNE are defined as the fix point among mutually best responses.

Best response dynamics

Consider a strategic game $\Gamma = \langle N, (A_i)_{i \in N}, (u_i)_{i \in N} \rangle$ with $N = \{1, \ldots, n\}$

- PNE are defined as the fix point among mutually best responses.
- It seems natural to consider variants of the process of local changes to try to get a PNE.

Best response dynamics

Consider a strategic game $\Gamma = \langle N, (A_i)_{i \in N}, (u_i)_{i \in N} \rangle$ with $N = \{1, \ldots, n\}$

- PNE are defined as the fix point among mutually best responses.
- It seems natural to consider variants of the process of local changes to try to get a PNE.
- Consider the algorithm:
  - choose $s \in A_1 \times \cdots \times A_n$
  - while s is not a NE do

    choose $i \in \{1, \ldots, n\}$ such that $s_i \notin BR(s_{-i})$

    Set s_i to be an action in $BR(s_{-i})$
- The process looks similar to local search algorithms. Is there any difference?

Best response graph

- The Nash dynamics or Best Response graph $G = (V, E)$ of Γ:
  - $V = A_1 \times \cdots \times A_n$
  - $(s, s') \in E$ iff $s' = (s_{-i}, s_i')$ for $i \in N$, $s_i \notin BR(s_{-i})$ and $s_i' \in BR(s_{-i})$.
- Performing local search on the best response graph
  - Does it produce a PNE?
  - If so, how much time?
  - Let's look to some examples.

Prisonner's dilemma

	Quiet	Fink
Quiet	2,2	0,3
Fink	3,0	1,1

Prisonner's dilemma

	Quiet	Fink
Quiet	2,2	0,3
Fink	3,0	1,1

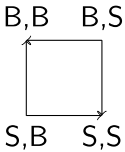
$$\begin{tikzcd}
\text{Q,Q} \arrow[r] \arrow[d] & \text{Q,F} \arrow[d] \\
\text{F,Q} \arrow[r] & \text{F,F}
\end{tikzcd}$$

Bach and Stravinsky

	Bach	Stravinsky
Bach	2,1	0,0
Stravinsky	0,0	1,2

Bach and Stravinsky

	Bach	Stravinsky
Bach	2,1	0,0
Stravinsky	0,0	1,2



Matching Pennies

	Head	Tail
Head	1,-1	-1,1
Tail	-1,1	1,-1

Matching Pennies

	Head	Tail
Head	1,-1	-1,1
Tail	-1,1	1,-1

$$\begin{tikzcd} \text{H,H} \arrow[r] & \text{H,T} \arrow[d] \\ \text{T,H} \arrow[u] & \text{T,T} \arrow[l] \end{tikzcd}$$

Other games

What about Congestion Games?

- In those games we cannot get the best response graph in polynomial time.
- However we can perform a local improvement step in polynomial time.
- Although, even assuring convergence, it might take exponential time to reach a NE.

Best response graph: Properties

- A NE is a sink, a node with out-degree 0, in the best response graph.
- The existence of a cycle in the best response graph does not rule out the existence of a PNE.
- If the best response graph is acyclic, the game has a PNE.

Best response graph: Properties

- A NE is a sink, a node with out-degree 0, in the best response graph.
- The existence of a cycle in the best response graph does not rule out the existence of a PNE.
- If the best response graph is acyclic, the game has a PNE.
- If best response dynamics converges to a PNE, maybe with a lot of time.

Potential games

(Monderer and Shapley 96)

- Consider a strategic game $\Gamma = \langle N, (A_i)_{i \in N}, (u_i)_{i \in N} \rangle$.
  Let $S = A_1 \times \cdots \times A_n$.

Potential games

(Monderer and Shapley 96)

- Consider a strategic game $\Gamma = \langle N, (A_i)_{i \in N}, (u_i)_{i \in N} \rangle$. Let $S = A_1 \times \cdots \times A_n$.
- A function $\Phi : S \to \mathbb{R}$ is an exact potential function for Γ if

$$\forall\, i \in N, \forall\, s \in S, \forall\, s'_i \in A_i : \; u_i(s) - u_i(s_{-i}, s'_i) = \Phi(s) - \Phi(s_{-i}, s'_i)$$

Potential games

(Monderer and Shapley 96)

- Consider a strategic game $\Gamma = \langle N, (A_i)_{i \in N}, (u_i)_{i \in N} \rangle$.
  Let $S = A_1 \times \cdots \times A_n$.
- A function $\Phi : S \to \mathbb{R}$ is an exact potential function for Γ if

$$\forall\, i \in N, \forall\, s \in S, \forall\, s_i' \in A_i : \; u_i(s) - u_i(s_{-i}, s_i') = \Phi(s) - \Phi(s_{-i}, s_i')$$

- A function $\Phi : S \to \mathbb{R}$ is a potential function for Γ if

$$\begin{aligned} &\forall\, i \in N, \forall\, s \in S, \forall\, s_i' \in A_i, \\ &\quad u_i(s) - u_i(s_{-i}, s_i') = \Phi(s) - \Phi(s_{-i}, s_i') = 0 \\ &\quad \text{or } (u_i(s) - u_i(s_{-i}, s_i'))(\Phi(s) - \Phi(s_{-i}, s_i')) > 0 \end{aligned}$$

Potential games

(Monderer and Shapley 96)

- Consider a strategic game $\Gamma = \langle N, (A_i)_{i \in N}, (u_i)_{i \in N} \rangle$. Let $S = A_1 \times \cdots \times A_n$.
- A function $\Phi : S \to \mathbb{R}$ is an exact potential function for Γ if

$$\forall\, i \in N, \forall\, s \in S, \forall\, s_i' \in A_i : \; u_i(s) - u_i(s_{-i}, s_i') = \Phi(s) - \Phi(s_{-i}, s_i')$$

- A function $\Phi : S \to \mathbb{R}$ is a potential function for Γ if

$$\begin{aligned} &\forall\, i \in N, \forall\, s \in S, \forall\, s_i' \in A_i, \\ &\quad u_i(s) - u_i(s_{-i}, s_i') = \Phi(s) - \Phi(s_{-i}, s_i') = 0 \\ &\quad \text{or } (u_i(s) - u_i(s_{-i}, s_i'))(\Phi(s) - \Phi(s_{-i}, s_i')) > 0 \end{aligned}$$

- Γ is a potential game if it admits a potential function.

Prisonner's dilemma

	Quiet	Fink
Quiet	2,2	0,3
Fink	3,0	1,1

$$\begin{tikzcd} Q,Q \arrow[r] \arrow[d] & Q,F \arrow[d] \\ F,Q \arrow[r] & F,F \end{tikzcd}$$

Prisonner's dilemma

	Quiet	Fink
Quiet	2,2	0,3
Fink	3,0	1,1

$$\begin{tikzcd}
Q,Q \arrow[r] \arrow[d] & Q,F \arrow[d] \\
F,Q \arrow[r] & F,F
\end{tikzcd}$$

Φ	Quiet	Fink
Quiet	1	2
Fink	2	3

Prisoner's dilemma

$$
\begin{tikzcd}
Q,Q \arrow[r] \arrow[d] & Q,F \arrow[d] \\
F,Q \arrow[r] & F,F
\end{tikzcd}
$$

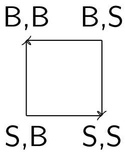
	Quiet	Fink
Quiet	2,2	0,3
Fink	3,0	1,1

Φ	Quiet	Fink
Quiet	1	2
Fink	2	3

Φ is an exact potential function

Bach and Stravinsky

	Bach	Stravinsky
Bach	2,1	0,0
Stravinsky	0,0	1,2



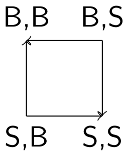
Bach and Stravinsky

$$
\begin{tikzcd}
B,B & B,S \arrow[l] \arrow[d] \\
S,B \arrow[u] \arrow[r] & S,S
\end{tikzcd}
$$

	Bach	Stravinsky
Bach	2,1	0,0
Stravinsky	0,0	1,2

Φ	Bach	Stravinsky
Bach	2	1
Stravinsky	0	2

Bach and Stravinsky



	Bach	Stravinsky
Bach	2,1	0,0
Stravinsky	0,0	1,2

Φ	Bach	Stravinsky
Bach	2	1
Stravinsky	0	2

Φ is an exact potential function

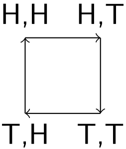
Matching Pennies

	Head	Tail
Head	1,-1	-1,1
Tail	-1,1	1,-1

$$\begin{tikzcd}
\text{H,H} \arrow[r] & \text{H,T} \arrow[d] \\
\text{T,H} \arrow[u] & \text{T,T} \arrow[l]
\end{tikzcd}$$

Matching Pennies

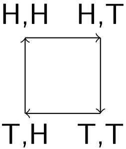
	Head	Tail
Head	1,-1	-1,1
Tail	-1,1	1,-1



This is not a potential game

Matching Pennies

	Head	Tail
Head	1,-1	-1,1
Tail	-1,1	1,-1



This is not a potential game

The property on Φ cannot hold along a cycle in the best response graph.

Potential games

Theorem

A strategic game is a potential game iff the best response graph is acyclic

Potential games

Theorem

A strategic game is a potential game iff the best response graph is acyclic

Proof.

Let G be the best response graph of Γ.

Potential games

Theorem

A strategic game is a potential game iff the best response graph is acyclic

Proof.

Let G be the best response graph of Γ.

- If G is acyclic, a topological sort of the graph provides a potential function for Γ.

Potential games

Theorem

A strategic game is a potential game iff the best response graph is acyclic

Proof.

Let G be the best response graph of Γ.

- If G is acyclic, a topological sort of the graph provides a potential function for Γ.
- The existence of a potential function Φ and the fact that, for each pair of connected strategy profiles in G, at least one player improves, implies the non existence of cycles in G.

$\square$

Potential games

Theorem

Any potential game has a PNE

Potential games

Theorem

Any potential game has a PNE

Proof.

As the best response graph is acyclic it must have a sink. $\square$

Potential games

Theorem

Any potential game has a PNE

Proof.

As the best response graph is acyclic it must have a sink. □

We have a way to show that a game has a PNE by showing that it is a potential game.

1. Best response dynamics
2. Potential games
3. Congestion games
4. References

Congestion games

A congestion game is defined by

- A finite set E of m resources
- A finite set N of n players
- A delay function d: $E \times \mathbb{N} \to \mathbb{Z}$
- For each player $i \in N$:
  - A set of actions $A_i \in 2^E$
  - A cost function c_i:

$$c_i(a_1, \dots, a_n) = \left(\sum_{e \in a_i} d(e, f(a_1, \dots, a_n, e)) \right)$$

being $f(a_1, \dots, a_n, e) = |\{i \mid e \in a_i\}|$.

Congestion games

A congestion game is defined by

- A finite set E of m resources
- A finite set N of n players
- A delay function d: $E \times \mathbb{N} \to \mathbb{Z}$
- For each player $i \in N$:
  - A set of actions $A_i \in 2^E$
  - A cost function c_i:

$$c_i(a_1, \ldots, a_n) = \left(\sum_{e \in a_i} d(e, f(a_1, \ldots, a_n, e)) \right)$$

being $f(a_1, \ldots, a_n, e) = |\{i \mid e \in a_i\}|$.

A singleton congestion game has $A_i = \{\{r\} \mid r \in E\}$.

An example of a congestion game

An example of a congestion game

- We have a factory with two end production lines, each having a cutting and a packing unit. Orders are cut down and then packed.

An example of a congestion game

- We have a factory with two end production lines, each having a cutting and a packing unit. Orders are cut down and then packed.
- We have 3 orders that have to be send to one of the end production lines.

An example of a congestion game

- We have a factory with two end production lines, each having a cutting and a packing unit. Orders are cut down and then packed.
- We have 3 orders that have to be send to one of the end production lines.
- The cutting machine on the first line takes 1 hour to process a single order, 2 hours to process 2 and 4 hours to process 3. The cutting machine on the second line takes 4, 5 and 9 hours respectively.

An example of a congestion game

- We have a factory with two end production lines, each having a cutting and a packing unit. Orders are cut down and then packed.
- We have 3 orders that have to be send to one of the end production lines.
- The cutting machine on the first line takes 1 hour to process a single order, 2 hours to process 2 and 4 hours to process 3. The cutting machine on the second line takes 4, 5 and 9 hours respectively.
- The packing machine on the first line takes 2 additional hours to pack a single order, 3 hours to pack 2 and 7 hours to pack 3. The packing machine on the second line takes instead 0, 2 and 9 hours respectively.

An example of a congestion game

- We have 4 resources C_1, C_2, P_1, P_2 and 3 players $N = \{1, 2, 3\}$
- $A_i = \{\{C_1, P_1\}, \{C_2, P_2\}\}$, $i = 1, 2, 3$
- Delay functions are defined by the processing times.

	1	2	3
C_1	1	2	4
C_2	4	5	9
P_1	2	3	7
P_2	0	2	9

An example of a congestion game

- We have 4 resources C_1, C_2, P_1, P_2 and 3 players $N = \{1, 2, 3\}$
- $A_i = \{\{C_1, P_1\}, \{C_2, P_2\}\}$, $i = 1, 2, 3$
- Delay functions are defined by the processing times.

	1	2	3
C_1	1	2	4
C_2	4	5	9
P_1	2	3	7
P_2	0	2	9

Does this game have a PNE?

Rosenthal's theorem

Theorem (Rosenthal 73)

Every congestion game is a potential game,

Rosenthal's theorem

Theorem (Rosenthal 73)

Every congestion game is a potential game,

- For a strategy profile $s = (a_1, \ldots, a_n)$, define

$$\Phi(s) = \sum_{e \in r(s)} \sum_{k=1}^{f(s,e)} d(e, k)$$

where $r(s) = \cup_{i \in N} a_i$.

Rosenthal's theorem

Theorem (Rosenthal 73)

Every congestion game is a potential game,

- For a strategy profile $s = (a_1, \ldots, a_n)$, define

$$\Phi(s) = \sum_{e \in r(s)} \sum_{k=1}^{f(s,e)} d(e,k)$$

where $r(s) = \cup_{i \in N} a_i$.

Let us show that Φ is a potential function.

- Let $s = (a_1, \ldots, a_n)$. Fix a player i and let $a_i' \subseteq E$ and $s' = (s_{-i}, a_i')$. We have

- Let $s = (a_1, \ldots, a_n)$. Fix a player i and let $a_i' \subseteq E$ and $s' = (s_{-i}, a_i')$. We have

$$c_i(s) - c_i(s_{-i}, a_i') = \left(\sum_{e \in a_i} d(e, f(s, e)) \right) - \left(\sum_{e' \in a_i'} d(e, f(s', e')) \right)$$

- Let $s = (a_1, \ldots, a_n)$. Fix a player i and let $a_i' \subseteq E$ and $s' = (s_{-i}, a_i')$. We have

$$c_i(s) - c_i(s_{-i}, a_i') = \left(\sum_{e \in a_i} d(e, f(s, e)) \right) - \left(\sum_{e' \in a_i'} d(e, f(s', e')) \right)$$

$$\Phi(s) - \Phi(s') = \sum_{e \in r(s)} \sum_{k=1}^{f(s,e)} d(e, k) - \sum_{e' \in r(s')} \sum_{k=1}^{f(s',e')} d(e', k)$$

Cost difference

- Note that

Cost difference

- Note that
  - $e \in a_i \cap a_i'$: $f(s,e) = f(s',e)$
  - $e \notin a_i$ and $e \notin a_i'$: $f(s,e) = f(s',e)$

Cost difference

- Note that
  - $e \in a_i \cap a_i'$: $f(s,e) = f(s',e)$
  - $e \notin a_i$ and $e \notin a_i'$: $f(s,e) = f(s',e)$

$$c_i(s) - c_i(s_{-i}, s_i') = \left(\sum_{e \in a_i} d(e, f(s,e))\right) - \left(\sum_{e' \in a_i'} d(e, f(s',e'))\right)$$

$$= \sum_{e \in a_i, e \notin a_i'} d(e, f(s,e)) - \sum_{e \notin a_i, e \in a_i'} d(e, f(s',e'))$$

Potential difference

- Furthermore,

Potential difference

- Furthermore,
  - $e \in a_i$ and $e \notin a'_i$: $f(s,e) = f(s',e) + 1$
  - $e \notin a_i$ and $e \in a'_i$: $f(s,e) + 1 = f(s',e)$

Potential difference

- Furthermore,
  - $e \in a_i$ and $e \notin a'_i$: $f(s,e) = f(s',e) + 1$
  - $e \notin a_i$ and $e \in a'_i$: $f(s,e) + 1 = f(s',e)$

$$\begin{aligned}
\Phi(s) - \Phi(s') &= \sum_{e \in r(s)} \sum_{k=1}^{f(s,e)} d(e,k) - \sum_{e \in r(s')} \sum_{k=1}^{f(s',e)} d(e,k) \\
&= \sum_{e \in a_i, e \notin a'_i} [\sum_{k=1}^{f(s',e)+1} d(e,k) - \sum_{k=1}^{f(s',e)} d(e,k)] \\
&\quad + \sum_{e \notin a_i, e \in a'_i} [\sum_{k=1}^{f(s,e)} d(e,k) - \sum_{k=1}^{f(s,e)+1} d(e,k)]
\end{aligned}$$

Potential difference

$$= \sum_{e \in a_i, e \notin a_i'} [\sum_{k=1}^{f(s',e)+1} d(e,k) - \sum_{k=1}^{f(s',e)} d(e,k)]$$

$$+ \sum_{e \notin a_i, e \in a_i'} [\sum_{k=1}^{f(s,e)} d(e,k) - \sum_{k=1}^{f(s,e)+1} d(e,k)]$$

$$= \sum_{e \in a_i, e \notin a_i'} d(e, f(s',e)+1) - \sum_{e \notin a_i, e \in a_i'} d(e, f(s,e)+1)$$

$$= \sum_{e \in a_i, e \notin a_i'} d(e, f(s,e)) - \sum_{e \notin a_i, e \in a_i'} d(e, f(s',e))$$

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

Network congestion games

- A network congestion game is a congestion game defined by a directed graph G and a collection of pairs of vertices (s_i, t_i).
  - The set of resources are the arcs in G.
  - The actions, for player i, are the $s_i \to t_i$ paths on G.
- A network congestion game is symmetric when $s_i = s$ and $t_i = t$, for $i \in N$.

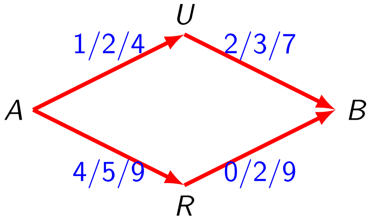
An example of a network congestion game

An example of a network congestion game

- There are three players.
- and a network (with a delay function on arcs)

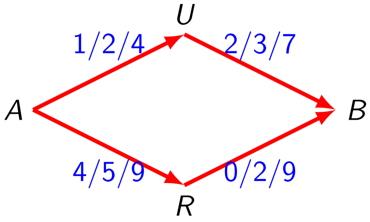
An example of a network congestion game

- There are three players.
- and a network (with a delay function on arcs)



An example of a network congestion game

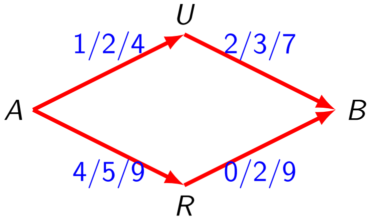
- There are three players.
- and a network (with a delay function on arcs)



- Player's objective: going from $s = A$ to $t = B$ as fast as possible.

An example of a network congestion game

- There are three players.
- and a network (with a delay function on arcs)



- Player's objective: going from $s = A$ to $t = B$ as fast as possible.
- Strategy profiles: paths from A to B.
- A NE?

An example of a weighted network congestion game

An example of a weighted network congestion game

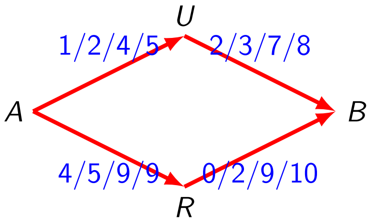
- There are three players with weights 1,1,2

An example of a weighted network congestion game

- There are three players with weights 1,1,2
- and a network (with a delay function on arcs)

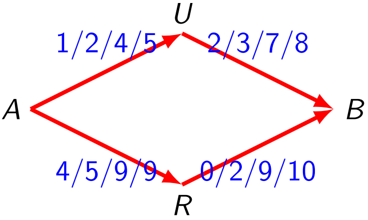
An example of a weighted network congestion game

- There are three players with weights 1,1,2
- and a network (with a delay function on arcs)



An example of a weighted network congestion game

- There are three players with weights 1,1,2
- and a network (with a delay function on arcs)



- Player's objective: send w_i units from $s = A$ to $t = B$ as fast as possible.
- Strategy profiles: paths from A to B.
- A NE?

Results on convergence time

Theorem (Fabrikant, Papadimitriou, Talwar (STOC 04))

There exist network congestion games with an initial strategy profile from which all better response sequences have exponential length.

Theorem (Fabrikant, Papadimitriou, Talwar (STOC 04))

There is a polynomial time algorithm for finding a PNE in symmetric network congestion games.

Results on convergence time

Theorem (Fabrikant, Papadimitriou, Talwar (STOC 04))

There exist network congestion games with an initial strategy profile from which all better response sequences have exponential length.

Theorem (Fabrikant, Papadimitriou, Talwar (STOC 04))

There is a polynomial time algorithm for finding a PNE in symmetric network congestion games.

Theorem (Ieong, McGrew, Nudelman, Shoham, Sun (AAAI 05))

In singleton congestion games all best response sequences have length at most $n^2 m$.

Complexity classification?

Optimization problem

An optimization problem is defined by a structure $\Pi = (\mathsf{I}, \mathsf{sol}, \mathsf{m}, \mathsf{goal})$, where

- I is the input set to Π;
- $\mathsf{sol}(x)$ is the set of feasible solutions for an input x.
- m is an integer measure defined over pairs (x, y), $x \in I$ and $y \in \mathsf{sol}(x)$.
- goal is the optimization criterium MAX or MIN.

An optimization problem is a functional problem whose goal, given an instance x, is to find an optimum solution

$$y = \mathsf{goal}\{(m(x, y') \mid y' \in \mathsf{sol}(x)\}.$$

Example: Given a graph and two vertices, obtain a path joining them with minimum length.

PLS

- A local search problem is an optimization problem with a neighborhood structure defined on the solution set $\mathcal{N}(\text{sol}(x))$.

PLS

- A local search problem is an optimization problem with a neighborhood structure defined on the solution set $\mathcal{N}(\text{sol}(x))$.
- A local optimum is a solution such that all its neighbors have equal or worse cost.

PLS

- A local search problem is an optimization problem with a neighborhood structure defined on the solution set $\mathcal{N}(\mathrm{sol}(x))$.
- A local optimum is a solution such that all its neighbors have equal or worse cost.

(Johnson, Papadimitriou, Yannakakis, FOCS 85)
A local search problem belongs to PLS (Polynomial Local Search)

PLS

- A local search problem is an optimization problem with a neighborhood structure defined on the solution set $\mathcal{N}(\mathrm{sol}(x))$.
- A local optimum is a solution such that all its neighbors have equal or worse cost.

(Johnson, Papadimitriou, Yannakakis, FOCS 85)
A local search problem belongs to PLS (Polynomial Local Search) if polynomial time algorithms exist for

PLS

- A local search problem is an optimization problem with a neighborhood structure defined on the solution set $\mathcal{N}(\mathrm{sol}(x))$.
- A local optimum is a solution such that all its neighbors have equal or worse cost.

(Johnson, Papadimitriou, Yannakakis, FOCS 85)
A local search problem belongs to PLS (Polynomial Local Search) if polynomial time algorithms exist for

- finding initial feasible solution $s \in \mathrm{sol}(x)$,

PLS

- A local search problem is an optimization problem with a neighborhood structure defined on the solution set $\mathcal{N}(\mathrm{sol}(x))$.
- A local optimum is a solution such that all its neighbors have equal or worse cost.

(Johnson, Papadimitriou, Yannakakis, FOCS 85)
A local search problem belongs to PLS (Polynomial Local Search) if polynomial time algorithms exist for

- finding initial feasible solution $s \in \mathrm{sol}(x)$,
- computing the objective measure $m(x, y)$,

PLS

- A local search problem is an optimization problem with a neighborhood structure defined on the solution set $\mathcal{N}(\mathrm{sol}(x))$.
- A local optimum is a solution such that all its neighbors have equal or worse cost.

(Johnson, Papadimitriou, Yannakakis, FOCS 85)
A local search problem belongs to PLS (Polynomial Local Search) if polynomial time algorithms exist for

- finding initial feasible solution $s \in \mathrm{sol}(x)$,
- computing the objective measure $m(x, y)$,
- checking whether a solution is a local optimum and if not finding a better solution in the neighborhood.

PLS reductions

A PLS reduction from $(\Pi_1, \mathcal{N}_1)$ to $(\Pi_2, \mathcal{N}_2)$ is

- a polynomial time computable function $f : \mathrm{I}_{\Pi_1} \to \mathrm{I}_{\Pi_2}$ and
- a polynomial time computable function $g : \mathrm{sol}_{\Pi_2}(f(x)) \to \mathrm{sol}_{\Pi_1}(x)$, for $x \in \mathrm{I}_{\Pi_1}$ such that
- if $s_2 \in \mathrm{sol}_{\Pi_2}(f(x))$ locally optimal then $g(s_2)$ is locally optimal.

PLS reductions

A PLS reduction from $(\Pi_1, \mathcal{N}_1)$ to $(\Pi_2, \mathcal{N}_2)$ is

- a polynomial time computable function $f : \mathrm{I}_{\Pi_1} \to \mathrm{I}_{\Pi_2}$ and
- a polynomial time computable function $g : \mathrm{sol}_{\Pi_2}(f(x)) \to \mathrm{sol}_{\Pi_1}(x)$, for $x \in \mathrm{I}_{\Pi_1}$ such that
- if $s_2 \in \mathrm{sol}_{\Pi_2}(f(x))$ locally optimal then $g(s_2)$ is locally optimal.

So that,

- If a local opt of Π_2 is "easy" to find then a local opt of Π_1 is easy to find.
- If a local opt of Π_1 is "hard" to find then a local opt of Π_2 is hard to find.

PLS reductions

A PLS reduction from $(\Pi_1, \mathcal{N}_1)$ to $(\Pi_2, \mathcal{N}_2)$ is

- a polynomial time computable function $f : \mathrm{I}_{\Pi_1} \to \mathrm{I}_{\Pi_2}$ and
- a polynomial time computable function $g : \mathrm{sol}_{\Pi_2}(f(x)) \to \mathrm{sol}_{\Pi_1}(x)$, for $x \in \mathrm{I}_{\Pi_1}$ such that
- if $s_2 \in \mathrm{sol}_{\Pi_2}(f(x))$ locally optimal then $g(s_2)$ is locally optimal.

So that,

- If a local opt of Π_2 is "easy" to find then a local opt of Π_1 is easy to find.
- If a local opt of Π_1 is "hard" to find then a local opt of Π_2 is hard to find.

A PLS problem $(\Pi, \mathcal{N})$ is PLS-complete if every problem in PLS is PLS-reducible to $(\Pi, \mathcal{N})$.

PLS complete problems

- MAX-SAT (maximum satisfiability) problem
  - Given a Boolean formula in conjunctive normal form with a positive integer weight for each clause.
  - A solution is an assignment of the value 0 or 1 to all variables.
  - Its weight, to be maximized, is the sum of the weights of all satisfied clauses.
  - As neighborhood consider the Flip-neighborhood, where two assignments are neighbors if one can be obtained from the other by fliipping the value of a single variable.

PLS complete problems

- MaxCut problem.
  - Given a graph $G = (V, E)$ with non-negative edge weights.
  - A feasible solution is a partition of V into two sets A and B.
  - The objective is to maximize the weight of the edges between the two sets A and B.
  - In the Flip-neighborhood two solutions are neighbors if one can be obtained from the other by moving a single vertex from one set to the other.

PLS completeness

Theorem

Computing a PNE in congestion games is PLS-complete.

PLS completeness

Theorem

Computing a PNE in congestion games is PLS-complete.

- The problem belongs to PLS taking as neighborhood the Nash dynamics because the Rosenthal's potential function can be evaluated in polynomial time.

PLS completeness

Theorem

Computing a PNE in congestion games is PLS-complete.

- The problem belongs to PLS taking as neighborhood the Nash dynamics because the Rosenthal's potential function can be evaluated in polynomial time.
- We provide a reduction from MaxCut under the Flip-neigborhood.

PLS completeness

Reduction from MaxCut under the Flip-neigborhood:

PLS completeness

Reduction from MaxCut under the Flip-neigborhood:

Let $(G, E, (w_e)_{e \in E})$ be an instance of MaxCut, define a congestion game as follows:

PLS completeness

Reduction from MaxCut under the Flip-neigborhood:

Let $(G, E, (w_e)_{e \in E})$ be an instance of MaxCut, define a congestion game as follows:

- For each edge e, we add resources e^a and e^b, with delay 0 if used by only one player and delay w_e if used by more players.

PLS completeness

Reduction from MaxCut under the Flip-neigborhood:

Let $(G, E, (w_e)_{e \in E})$ be an instance of MaxCut, define a congestion game as follows:

- For each edge e, we add resources e^a and e^b, with delay 0 if used by only one player and delay w_e if used by more players.
- Players correspond to nodes V, and the action set of $v \in V$, $A_v = \{e^a, e^b \mid e \in E \text{ incident to } v\}$

PLS completeness

- Each solution (A, B) of MaxCut corresponds to strategy profile $s = (s_1, \ldots, s_n)$ where for each $v \in V$, $s_v = \{e^a | e = (u, v) \wedge v \in A\} \cup \{e^b | e = (u, v) \wedge v \in B\}$

PLS completeness

- Each solution (A, B) of MaxCut corresponds to strategy profile $s = (s_1, \ldots, s_n)$ where for each $v \in V$, $s_v = \{e^a | e = (u, v) \wedge v \in A\} \cup \{e^b | e = (u, v) \wedge v \in B\}$
- Local optima of the MaxCut instance coincide with the Nash equilibria of the congestion game.

1. Best response dynamics
2. Potential games
3. Congestion games
4. References

Reference

B. Vöcking, Congestion Games: Optimization in Competition