Inapproximability of Congestion Games

Alexander Skopalik, Berthold Vöcking

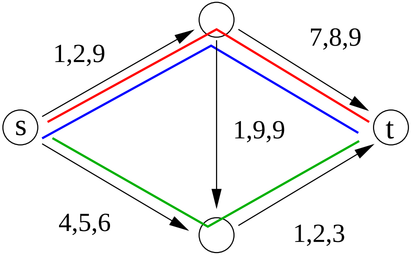
Department of Computer Science
RWTH Aachen

Warwick 2007

Network Congestion Games

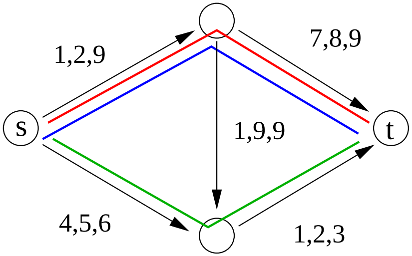
- Given a directed graph $G = (V, E)$ with delay functions $d_e : \{1, \ldots, n\} \rightarrow \mathbb{N}$, $e \in E$.
- Player i wants to allocate a path of minimal delay between a source s_i and a target t_i.

- Given a directed graph $G = (V, E)$ with delay functions $d_e : \{1, \ldots, n\} \to \mathbb{N}$, $e \in E$.
- Player i wants to allocate a path of minimal delay between a source s_i and a target t_i.



Network Congestion Games

- Given a directed graph $G = (V, E)$ with delay functions $d_e : \{1, \ldots, n\} \to \mathbb{N}$, $e \in E$.
- Player i wants to allocate a path of minimal delay between a source s_i and a target t_i.



- Game is called *symmetric* if all players have the same source/target pair.

Congestion Games - general Definition

Congestion game is a tuple $G = (\mathcal{N}, \mathcal{R}, (\Sigma_i)_{i \in \mathcal{N}}, (d_r)_{r \in \mathcal{R}})$ with

- $\mathcal{N} = \{1, \ldots, n\}$, set of players
- $\mathcal{R} = \{1, \ldots, m\}$, set of resources
- $\Sigma_i \subseteq 2^{[m]}$, strategy space of player i
- $d_r : \{1, \ldots, n\} \to \mathbb{R}$, delay function or resource r

For any state $S = (S_1, \ldots, S_n) \in \Sigma_1 \times \cdots \Sigma_n$

- n_r = number of players with $r \in S_i$
- $d_r(n_r)$ = delay of resource r
- $\sum_{r \in S_i} d_r(n_r)$ = delay of player i

S is Nash equilibrium if no player can unilaterally decrease its delay.

The transition graph

Definition

The *transition graph* of a congestion game Γ contains a node for every state S and a directed edge (S, S') if S' can be reached from S by the improvement step of a single player.

The transition graph

Definition

The *transition graph* of a congestion game Γ contains a node for every state S and a directed edge (S, S') if S' can be reached from S by the improvement step of a single player.

The sinks of the transition graph are the Nash equilibria of G.

The transition graph

Definition

The *transition graph* of a congestion game Γ contains a node for every state S and a directed edge (S, S') if S' can be reached from S by the improvement step of a single player.

The sinks of the transition graph are the Nash equilibria of G.

Nash equilibria are local optima wrt *Rosenthal's potential function*

$$\phi(S) = \sum_{r \in \mathcal{R}} \sum_{i=1}^{n_r(S)} d_r(i) \ .$$

Complexity of computing equilibria

Known Results

	matroid games	network games	general games
symmetric	$O(n^2 m^2)$	polynomial	
asymmetric	$O(n^2 m^2)$		

matroid game results by [Ackermann, Röglin, V. 2006]
all other results by [Fabrikant, Papadimitriou, Talwar 2004]

Complexity of computing equilibria

Known Results

	matroid games	network games	general games
symmetric	$O(n^2m^2)$	polynomial	PLS-complete
asymmetric	$O(n^2m^2)$	PLS-complete	PLS-complete

matroid game results by [Ackermann, Röglin, V. 2006]
all other results by [Fabrikant, Papadimitriou, Talwar 2004]

The complexity class PLS

PLS (Polynomial Local Search)

PLS contains optimization problems with a specified neighborhood relationship Γ. It is required that there is a poly-time algorithm that, given any solution S,

- either computes a solution in $\Gamma(S)$ with better objective value
- or certifies that S is a local optimum.

Examples:

- FLIP (circuit evaluation with Flip-neighborhood)
- Max-Sat with Flip-neighborhood
- Max-Cut with Flip-neighborhood
- TSP with 2-Opt-neighborhood
- Congestion games wrt improvement steps

The complexity class PLS

PLS reductions

Given two PLS problems Π_1 and Π_2 find a mapping from the instances of Π_1 to the instances of Π_2 such that

- the mapping can be computed in polynomial time,
- the local optima of Π_1 are mapped to local optima of Π_2, and
- given any local optimum of Π_2, one can construct a local optimum of Π_1 in polynomial time.

Examples for PLS-complete problem:

- FLIP (via a master reduction)
- Max-Sat and POS-NAE-SAT
- Max-Cut

Approximation of local search problems

Definition

Consider any local search problem Π. Let $\alpha > 1$. An α-approximation for an instance of Π is a state S with the property that every state in $\Gamma(S)$ has a value of at most α times better than the value of S.

Approximation of local search problems

Definition

Consider any local search problem Π. Let $\alpha > 1$. An α-approximation for an instance of Π is a state S with the property that every state in $\Gamma(S)$ has a value of at most α times better than the value of S.

Orlin, Punnen, Abraham, Schulz 2004

There is a fully polynomial time approximation scheme for every problem in PLS.

Approximation of congestion games

Definition

An α-approximate equilibrium, for $\alpha > 1$, is a state S with the property that none of the players can improve its delay by a factor of more than α.

Approximation of congestion games

Definition

An α-approximate equilibrium, for $\alpha > 1$, is a state S with the property that none of the players can improve its delay by a factor of more than α.

Chien & Sinclair 2007

In any symmetric network congestion game in which all edges satisfy the β-bounded jump condition, i.e., $d_e(i+1) \leq \beta d_e(i)$ for all $i \in \mathbb{N}$, there is a sequence of improvement steps converging in

$$O(n\beta\epsilon^{-1}\log(nD))$$

steps, where D is an upper bound on the maximum delay.

New results

- For any poly-time computable $\alpha > 1$, finding an α-approximate Nash equilibrium in general congestion games with positive and increasing delay functions is PLS-hard.

New results

- For any poly-time computable $\alpha > 1$, finding an α-approximate Nash equilibrium in general congestion games with positive and increasing delay functions is PLS-hard.
- For every $n \in \mathbb{N}$, there is a congestion game with n players having a state with the property that every sequence of improvement steps leading from this state to an approximate equilibrium has exponential length in n.

New results

- For any poly-time computable $\alpha > 1$, finding an α-approximate Nash equilibrium in general congestion games with positive and increasing delay functions is PLS-hard.
- For every $n \in \mathbb{N}$, there is a congestion game with n players having a state with the property that every sequence of improvement steps leading from this state to an approximate equilibrium has exponential length in n.
- It is PSPACE-hard to compute an α-equilibrium reachable from a given state in a given congestion games.

Sketch of the analysis

We do a PLS-Reduction from FLIP.

Definition (FLIP)

An instance of the problem FLIP consists of a Boolean circuit C with input bits $x_1, \dots, x_n$ and output bits $y_1, \dots, y_m$. The neighborhood $N(x)$ of solution x is set of bit vectors x' that differ from x in one bit and $c(x') < c(x)$.

- We transform C into a congestion game $G(C)$ such that Nash equilibria of $G(C)$ correspond to a local optimum of f_C.

Sketch of the analysis

We do a PLS-Reduction from FLIP.

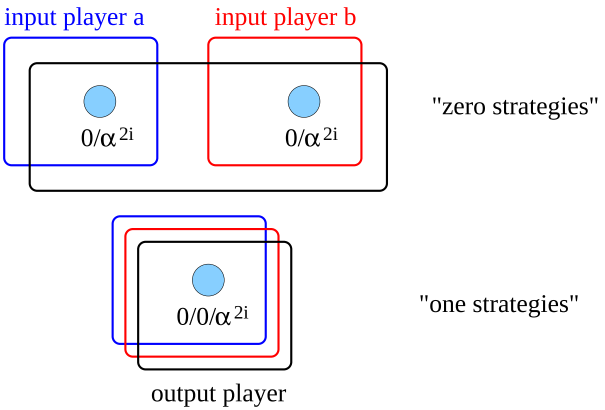
Definition (FLIP)

An instance of the problem FLIP consists of a Boolean circuit C with input bits $x_1, \ldots, x_n$ and output bits $y_1, \ldots, y_m$. The neighborhood $N(x)$ of solution x is set of bit vectors x' that differ from x in one bit and $c(x') < c(x)$.

- We transform C into a congestion game $G(C)$ such that Nash equilibria of $G(C)$ correspond to a local optimum of f_C.
- Delays of different strategies of any player in $G(C)$ deviate at least by a factor of α.
- Thus all equilibria are α-approximate equilibria.

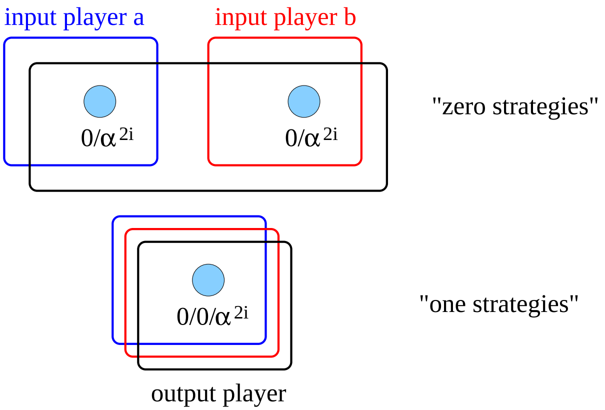
Representing circuits by congestion games

W.l.o.g. the circuit consists only of NAND-Gates. Sort the gates in reverse topological order. Design of i-th gate:



Representing circuits by congestion games

W.l.o.g. the circuit consists only of NAND-Gates. Sort the gates in reverse topological order. Design of i-th gate:



Input players "trigger" the output player because of the reverse topological order of the gates.

The feedback problem

A simple Idea that does not work out ...

1. Construct a circuit S with n input and n output bits that, for $x \in \{0,1\}^n$ not being a local optimum, computes $x' \in \{0,1\}^n$ with $f_C(x') \leq f_C(x)$.
2. Represent S by a congestion game $G(S)$.

The feedback problem

A simple Idea that does not work out ...

1. Construct a circuit S with n input and n output bits that, for $x \in \{0,1\}^n$ not being a local optimum, computes $x' \in \{0,1\}^n$ with $f_C(x') \leq f_C(x)$.
2. Represent S by a congestion game $G(S)$.
3. Additionally, ensure that the output bits in $G(S)$ trigger the input bits such that the circuit is in an equilibrium only when input and output bits are identical.

The feedback problem

A simple Idea that does not work out ...

1. Construct a circuit S with n input and n output bits that, for $x \in \{0,1\}^n$ not being a local optimum, computes $x' \in \{0,1\}^n$ with $f_C(x') \leq f_C(x)$.
2. Represent S by a congestion game $G(S)$.
3. Additionally, ensure that the output bits in $G(S)$ trigger the input bits such that the circuit is in an equilibrium only when input and output bits are identical.

If this construction would be possible then Nash equilibria correspond to local optimal.

The feedback problem

A simple Idea that does not work out ...

1. Construct a circuit S with n input and n output bits that, for $x \in \{0,1\}^n$ not being a local optimum, computes $x' \in \{0,1\}^n$ with $f_C(x') \leq f_C(x)$.
2. Represent S by a congestion game $G(S)$.
3. Additionally, ensure that the output bits in $G(S)$ trigger the input bits such that the circuit is in an equilibrium only when input and output bits are identical.

If this construction would be possible then Nash equilibria correspond to local optimal.

Problem: output players have much smaller delay differences than input players

PLS Reduction – High Level Description

Idea: Construct a congestion game simulating a processor

The players

- *n input players* $X_1, \ldots, X_n$
- *m clock players* $Z_1, \ldots, Z_m$ (with large delay differences)
- *gate players* for several circuits
- the *controller*

PLS Reduction – High Level Description

Idea: Construct a congestion game simulating a processor

The players

- *n input players* $X_1, \ldots, X_n$
- *m clock players* $Z_1, \ldots, Z_m$ (with large delay differences)
- *gate players* for several circuits
- the *controller*

Two kinds of states

Let M be a very large integer.

- In the *expensive states* at least one player has a delay of at least M.
- In the *inexpensive states* all players have a delay significantly smaller than M.

PLS-Reduction – High Level Description

The expensive states do not contain Nash equilibria. So we only have to consider the inexpensive states.

PLS-Reduction – High Level Description

The expensive states do not contain Nash equilibria. So we only have to consider the inexpensive states.

In the inexpensive states ...

- the clock players $Z_1, \ldots, Z_m$ count downward
- while counting downward they trigger improvement steps (flips) of the input players $X_1, \ldots, X_n$

PLS-Reduction – High Level Description

The expensive states do not contain Nash equilibria. So we only have to consider the inexpensive states.

In the inexpensive states ...

- the clock players $Z_1, \ldots, Z_m$ count downward
- while counting downward they trigger improvement steps (flips) of the input players $X_1, \ldots, X_n$
- the controller together with the circuits guarantee the property $z \geq f_C(x)$ (*upper bound condition*)
- this ensures that the clock can only stop because it cannot trigger an improvement step

PLS-Reduction – High Level Description

The expensive states do not contain Nash equilibria. So we only have to consider the inexpensive states.

In the inexpensive states ...

- the clock players $Z_1, \ldots, Z_m$ count downward
- while counting downward they trigger improvement steps (flips) of the input players $X_1, \ldots, X_n$
- the controller together with the circuits guarantee the property $z \geq f_C(x)$ *(upper bound condition)*
- this ensures that the clock can only stop because it cannot trigger an improvement step
- consequently, every Nash equilibrium corresponds to a local optimum of C

Network Congestion Games

Conjecture

For any poly-time computable $\alpha > 1$, finding an α-approximate Nash equilibrium in network congestion games with positive and increasing delay functions is PLS-hard.

Conjecture

For any poly-time computable $\alpha > 1$, finding an α-approximate Nash equilibrium in network congestion games with positive and increasing delay functions is PLS-hard.

Theorem

Let $\alpha, \beta > 1$ be appropriate constants. For every $n \in \mathbb{N}$, there is an n-player game with $O(n)$ resources and β-jump bounded delay function such that there is a state that has distance exponential in n to all α-Nash equilibria.

Open problems

- Is there a set of reasonable assumptions on the delay functions such that
  - computing an approximate Nash equilibrium has polynomial complexity?
  - improvement sequences reach an approximate Nash equilibrium after polynomially many steps?

Open problems

- Is there a set of reasonable assumptions on the delay functions such that
  - computing an approximate Nash equilibrium has polynomial complexity?
  - improvement sequences reach an approximate Nash equilibrium after polynomially many steps?
- What about other games? – e.g. the party affiliation game (Max-Cut)?