On the Impact of Combinatorial Structure on Congestion Games

Heiko Röglin

Department of Computer Science
RHEINISCH-WESTFÄLISCHE TECHNISCHE HOCHSCHULE AACHEN

joint work with Heiner Ackermann and Berthold Vöcking

MPI
September 25, 2006

Congestion Games – Definition

Congestion game $\Gamma = (\mathcal{N}, \mathcal{R}, (\Sigma_i)_{i \in \mathcal{N}}, (d_r)_{r \in \mathcal{R}})$

Congestion Games – Definition

Congestion game $\Gamma = (\mathcal{N}, \mathcal{R}, (\Sigma_i)_{i \in \mathcal{N}}, (d_r)_{r \in \mathcal{R}})$

Main Ingredients

- set of **players**
  $\mathcal{N} = \{1, \ldots, n\}$

Congestion Games – Definition

Congestion game $\Gamma = (\mathcal{N}, \mathcal{R}, (\Sigma_i)_{i \in \mathcal{N}}, (d_r)_{r \in \mathcal{R}})$

Main Ingredients

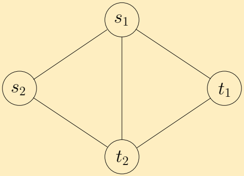
- set of **players**
  $\mathcal{N} = \{1, \ldots, n\}$
- set of **resources**
  $\mathcal{R} = \{1, \ldots, m\}$

Congestion Games – Definition

Congestion game $\Gamma = (\mathcal{N}, \mathcal{R}, (\Sigma_i)_{i \in \mathcal{N}}, (d_r)_{r \in \mathcal{R}})$

Main Ingredients

- set of **players**
  $\mathcal{N} = \{1, \ldots, n\}$
- set of **resources**
  $\mathcal{R} = \{1, \ldots, m\}$
  e. g., $\mathcal{R}$ = set of edges

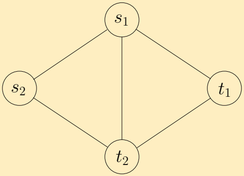


Congestion Games – Definition

Congestion game $\Gamma = (\mathcal{N}, \mathcal{R}, (\Sigma_i)_{i \in \mathcal{N}}, (d_r)_{r \in \mathcal{R}})$

Main Ingredients

- set of **players**
  $\mathcal{N} = \{1, \ldots, n\}$
- set of **resources**
  $\mathcal{R} = \{1, \ldots, m\}$
  e. g., $\mathcal{R}$ = set of edges



Further Ingredients

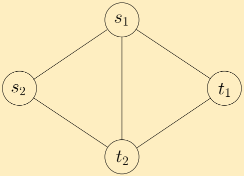
- set of **strategies**
  $\forall i \in \mathcal{N} : \Sigma_i \subseteq 2^{\mathcal{R}}$

Congestion Games – Definition

Congestion game $\Gamma = (\mathcal{N}, \mathcal{R}, (\Sigma_i)_{i \in \mathcal{N}}, (d_r)_{r \in \mathcal{R}})$

Main Ingredients

- set of **players**
  $\mathcal{N} = \{1, \dots, n\}$
- set of **resources**
  $\mathcal{R} = \{1, \dots, m\}$
  e. g., $\mathcal{R}$ = set of edges



Further Ingredients

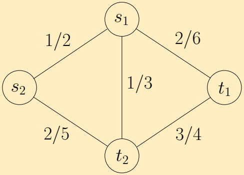
- set of **strategies**
  $\forall i \in \mathcal{N} : \Sigma_i \subseteq 2^{\mathcal{R}}$
  e. g.,
  $\Sigma_i = \{P \subseteq \mathcal{R} \mid P \text{ path } s_i \to t_i\}$

Congestion Games – Definition

Congestion game $\Gamma = (\mathcal{N}, \mathcal{R}, (\Sigma_i)_{i\in\mathcal{N}}, (d_r)_{r\in\mathcal{R}})$

Main Ingredients

- set of **players**
  $\mathcal{N} = \{1, \ldots, n\}$
- set of **resources**
  $\mathcal{R} = \{1, \ldots, m\}$
  e. g., $\mathcal{R}$ = set of edges



Further Ingredients

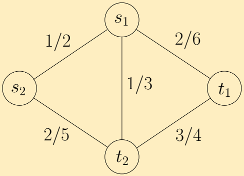
- set of **strategies**
  $\forall i \in \mathcal{N} : \Sigma_i \subseteq 2^{\mathcal{R}}$
  e. g.,
  $\Sigma_i = \{P \subseteq \mathcal{R} \mid P \text{ path } s_i \to t_i\}$
- **delay functions**
  $\forall r \in \mathcal{R} : \; d_r : \mathbb{N} \to \mathbb{N}$
- Every player wants to **minimize his delay**.

Congestion Games – Definition

Congestion game $\Gamma = (\mathcal{N}, \mathcal{R}, (\Sigma_i)_{i \in \mathcal{N}}, (d_r)_{r \in \mathcal{R}})$

Main Ingredients

- set of **players**
  $\mathcal{N} = \{1, \ldots, n\}$
- set of **resources**
  $\mathcal{R} = \{1, \ldots, m\}$
  e. g., $\mathcal{R}$ = set of edges



Further Ingredients

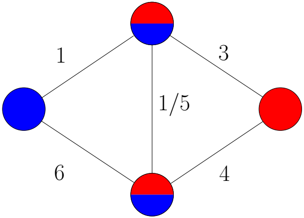
- set of **strategies**
  $\forall i \in \mathcal{N} : \Sigma_i \subseteq 2^{\mathcal{R}}$
  e. g.,
  $\Sigma_i = \{P \subseteq \mathcal{R} \mid P \text{ path } s_i \rightarrow t_i\}$
- **delay functions**
  $\forall r \in \mathcal{R} : \; d_r : \mathbb{N} \rightarrow \mathbb{N}$
- Every player wants to **minimize his delay**.
- Every player is faced with **optimization problem with varying delays**.

Congestion Games – Example

$\mathcal{N} = \{1, 2\}$, $\mathcal{R} = E$, $n = 2$, $m = 5$

Σ_1 = set of spanning trees on blue vertices

Σ_2 = set of spanning trees on red vertices

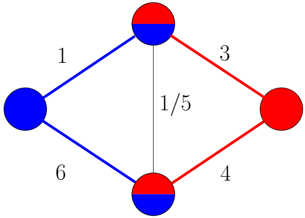


Congestion Games – Example

$\mathcal{N} = \{1, 2\}$, $\mathcal{R} = E$, $n = 2$, $m = 5$

Σ_1 = set of spanning trees on blue vertices

Σ_2 = set of spanning trees on red vertices



Congestion Games – Example

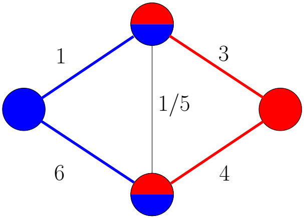
$\mathcal{N} = \{1, 2\}$, $\mathcal{R} = E$, $n = 2$, $m = 5$

Σ_1 = set of spanning trees on blue vertices

Σ_2 = set of spanning trees on red vertices

Delay: 7

Opt: 2



Delay: 7

Opt: 4

Congestion Games – Example

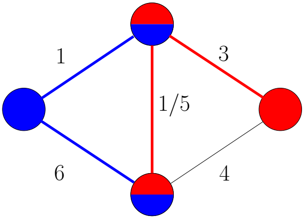
$\mathcal{N} = \{1, 2\}$, $\mathcal{R} = E$, $n = 2$, $m = 5$

Σ_1 = set of spanning trees on blue vertices

Σ_2 = set of spanning trees on red vertices

Delay: 7

Opt: 6



Delay: 4

Opt: 4

Congestion Games – Example

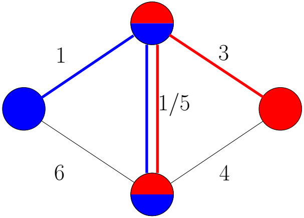
$\mathcal{N} = \{1, 2\}$, $\mathcal{R} = E$, $n = 2$, $m = 5$

Σ_1 = set of spanning trees on blue vertices

Σ_2 = set of spanning trees on red vertices

Delay: 6

Opt: 6



Delay: 8

Opt: 7

Congestion Games – Example

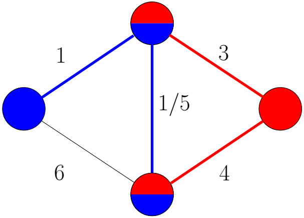
$\mathcal{N} = \{1,2\}$, $\mathcal{R} = E$, $n = 2$, $m = 5$

Σ_1 = set of spanning trees on blue vertices

Σ_2 = set of spanning trees on red vertices

Delay: 2

Opt: 2



Delay: 7

Opt: 7

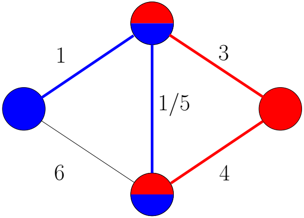
Congestion Games – Example

$\mathcal{N} = \{1, 2\}$, $\mathcal{R} = E$, $n = 2$, $m = 5$

Σ_1 = set of spanning trees on blue vertices

Σ_2 = set of spanning trees on red vertices

Delay: 2
Opt: 2



Delay: 7
Opt: 7

A state $S \in \Sigma_1 \times \cdots \times \Sigma_n$ is called pure Nash equilibrium if no player can improve his delay unilaterally.

Questions

Rosenthal (1973)

- Every congestion game possess a pure Nash equilibrium.
- Every better response sequence terminates after a pseudopolynomial number of steps.

Questions

Rosenthal (1973)

- Every congestion game possess a pure Nash equilibrium.
- Every better response sequence terminates after a pseudopolynomial number of steps.

Questions

- How many best responses are needed to find an equilibrium?
- What is the complexity of computing equilibria?

How many best responses are needed?

1. How many best responses are needed?
2. What is the complexity of finding equilibria?

Known Results

Fabrikant, Papadimitriou, Talwar (STOC 2004)

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

Known Results

Fabrikant, Papadimitriou, Talwar (STOC 2004)

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

Ieong, McGrew, Nudelman, Shoham, Sun (AAAI 2005)

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

Known Results

Fabrikant, Papadimitriou, Talwar (STOC 2004)

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

Ieong, McGrew, Nudelman, Shoham, Sun (AAAI 2005)

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

Question

What about Spanning Tree congestion games?
Is there a characterization which congestion games converge in polynomial time?

Rosenthal's Potential Function

Properties

- $\Phi\colon \Sigma_1 \times \cdots \times \Sigma_n \to \mathbb{Z}$
- $\forall S : 0 \leq \phi(S) \leq m \cdot n \cdot d_{\max}$.
- If one player decreases his delay by x, then also Φ decreases by x.

Rosenthal's Potential Function

Properties

- $\Phi\colon \Sigma_1 \times \cdots \times \Sigma_n \to \mathbb{Z}$
- $\forall S : 0 \leq \phi(S) \leq m \cdot n \cdot d_{\max}$.
- If one player decreases his delay by x, then also Φ decreases by x.

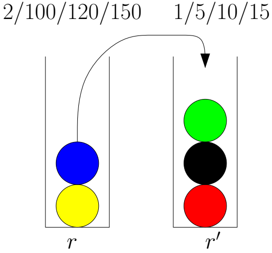
Corollary

- pure Nash equilibria = states in which no player can decrease the potential Φ
- After at most $m \cdot n \cdot d_{\max}$ better responses a pure Nash equilibrium is reached.

Singleton Games

Singleton Games

- Idea: Reduce delays without changing the game!



$$d_r(n_r) > d_{r'}(n_{r'} + 1)$$

Singleton Games

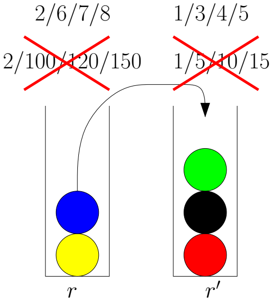
Singleton Games

- Idea: Reduce delays without changing the game!
- equivalent delays $\overline{d}_r(x) \leq n \cdot m$

$$\forall r, r' \in \mathcal{R}, n_r, n_{r'} :$$

$$d_r(n_r) > d_{r'}(n_{r'}+1)$$

$$\iff \quad \overline{d}_r(n_r) > \overline{d}_{r'}(n_{r'}+1)$$



$d_r(n_r) > d_{r'}(n_{r'}+1)$

Singleton Games

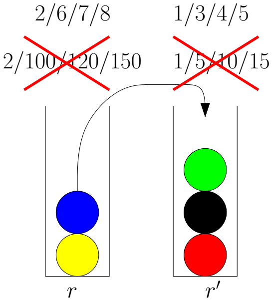
Singleton Games

- Idea: Reduce delays without changing the game!
- equivalent delays $\overline{d}_r(x) \leq n \cdot m$

$$\forall r, r' \in \mathcal{R}, n_r, n_{r'} :$$

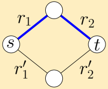
$$d_r(n_r) > d_{r'}(n_{r'}+1)$$

$$\iff \quad \overline{d}_r(n_r) > \overline{d}_{r'}(n_{r'}+1)$$

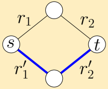


$$d_r(n_r) > d_{r'}(n_{r'}+1)$$

Network Congestion Games



$$d_{r_1}(n_{r_1}) + d_{r_2}(n_{r_2}) > d_{r'_1}(n_{r'_1}+1) + d_{r'_2}(n_{r'_2}+1)$$



What about Spanning Tree Congestion Games?

Theorem

- In spanning tree congestion games all best response sequences have length at most $n^3 \cdot m$.

What about Spanning Tree Congestion Games?

Theorem

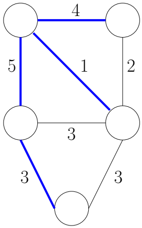
- In spanning tree congestion games all best response sequences have length at most $n^3 \cdot m$.
- In matroid congestion games all best response sequences have length at most $n^2 \cdot m \cdot rank$.

Proof

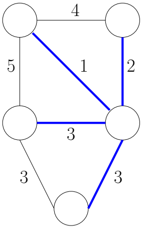
Lemma

- weighted graph: $G = (V, E, w)$
- Let T_0 be a ST, let T^{OPT} be a MST: $w(T_0) \geq w(T^{\mathrm{OPT}})$.

There exists sequence $T_0, \ldots, T_l = T^{\mathrm{OPT}}$ of STs with $w(T_0) \geq w(T_1) \geq \ldots \geq w(T_l)$ with $|T_i \setminus T_{i-1}| = 1$.



$w(T_0) = 13$



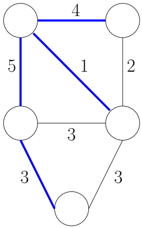
$w(T_3) = 9$

Proof

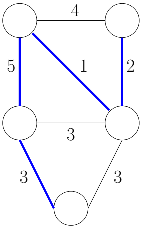
Lemma

- weighted graph: $G = (V, E, w)$
- Let T_0 be a ST, let T^{OPT} be a MST: $w(T_0) \geq w(T^{\mathrm{OPT}})$.

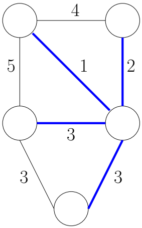
There exists sequence $T_0, \ldots, T_l = T^{\mathrm{OPT}}$ of STs with $w(T_0) \geq w(T_1) \geq \ldots \geq w(T_l)$ with $|T_i \setminus T_{i-1}| = 1$.



$w(T_0) = 13$



$w(T_1) = 11$



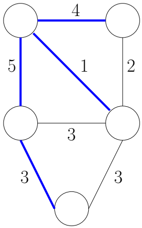
$w(T_3) = 9$

Proof

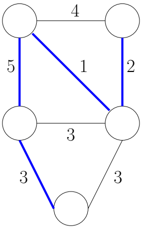
Lemma

- weighted graph: $G = (V, E, w)$
- Let T_0 be a ST, let T^{OPT} be a MST: $w(T_0) \geq w(T^{\mathrm{OPT}})$.

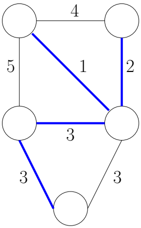
There exists sequence $T_0, \ldots, T_l = T^{\mathrm{OPT}}$ of STs with $w(T_0) \geq w(T_1) \geq \ldots \geq w(T_l)$ with $|T_i \setminus T_{i-1}| = 1$.



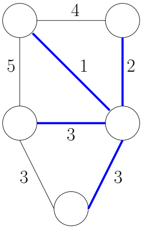
$w(T_0) = 13$



$w(T_1) = 11$



$w(T_2) = 9$



$w(T_3) = 9$

Fast Convergence beyond Matroids

Theorem:

Let $\mathcal{I}$ be any inclusion-free non-matroid set system. Then, for every n, there exists an n-player congestion game with the following properties.

- each Σ_i is isomorphic to $\mathcal{I}$,
- the delay functions are non-negative and non-decreasing, and
- there is a best response sequence of length $2^{\Omega(n)}$.

Fast Convergence beyond Matroids

Theorem:

Let $\mathcal{I}$ be any inclusion-free non-matroid set system. Then, for every n, there exists an n-player congestion game with the following properties.

- each Σ_i is isomorphic to $\mathcal{I}$,
- the delay functions are non-negative and non-decreasing, and
- there is a best response sequence of length $2^{\Omega(n)}$.

Conclusion: The matroid property is the maximal property on the individual players' strategy spaces that guarantees polynomial convergence.

Proof Idea for Exponential Convergence

Because of the non-matroid property, one can show:

1-2-exchange property

There exist three resources a, b, c with the property that, if the delays of the other resources are chosen appropriately, an optimal solution of $\mathcal{I}$ contains

- $d(a) < d(b) + d(c) \Rightarrow a \in \mathrm{OPT}$ and $b, c \notin \mathrm{OPT}$,
- $d(a) > d(b) + d(c) \Rightarrow a \notin \mathrm{OPT}$ and $b, c \in \mathrm{OPT}$.

Proof Idea for Exponential Convergence

Because of the non-matroid property, one can show:

1-2-exchange property

There exist three resources a, b, c with the property that, if the delays of the other resources are chosen appropriately, an optimal solution of $\mathcal{I}$ contains

- $d(a) < d(b) + d(c) \Rightarrow a \in \mathrm{OPT}$ and $b, c \notin \mathrm{OPT}$,
- $d(a) > d(b) + d(c) \Rightarrow a \notin \mathrm{OPT}$ and $b, c \in \mathrm{OPT}$.

Using this property one can interweave the strategy spaces in the form of a counter that yields a best response sequence of length $2^{\Omega(n)}$.

What is the complexity of finding equilibria?

1. How many best responses are needed?
2. What is the complexity of finding equilibria?

PLS

Local Search Problem Π

- set of instances $\mathcal{I}_\Pi$
- for $I \in \mathcal{I}_\Pi$: set of feasible solutions $\mathcal{F}(I)$
- for $I \in \mathcal{I}_\Pi$: objective function $c : \mathcal{F}(I) \to \mathbb{Z}$
- for $I \in \mathcal{I}_\Pi$ and $S \in \mathcal{F}(I)$: neighborhood $\mathcal{N}(S, I) \subseteq \mathcal{F}(I)$

PLS

Local Search Problem Π

- set of instances $\mathcal{I}_\Pi$
- for $I \in \mathcal{I}_\Pi$: set of feasible solutions $\mathcal{F}(I)$
- for $I \in \mathcal{I}_\Pi$: objective function $c : \mathcal{F}(I) \to \mathbb{Z}$
- for $I \in \mathcal{I}_\Pi$ and $S \in \mathcal{F}(I)$: neighborhood $\mathcal{N}(S, I) \subseteq \mathcal{F}(I)$

PLS

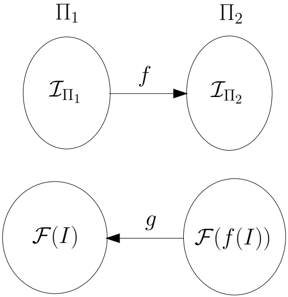
Π is in PLS if polynomial time algorithms exists for

- finding initial feasible solution $S \in \mathcal{F}(I)$,
- computing the objective value $c(S)$,
- finding a better solution in the neighborhood $\mathcal{N}(S, I)$ if S is not locally optimal.

PLS-reductions

PLS-reduction

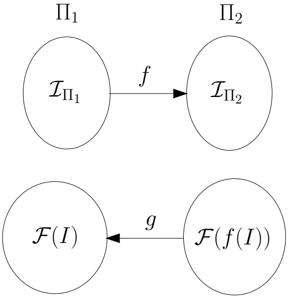
- Polynomial-time computable function $f\colon \mathcal{I}_{\Pi_1} \to \mathcal{I}_{\Pi_2}$.
- Polynomial-time computable function $(S_2 \in \mathcal{F}(f(I)))$ $g\colon S_2 \mapsto S_1 \in \mathcal{F}(I)$



PLS-reductions

PLS-reduction

- Polynomial-time computable function $f\colon \mathcal{I}_{\Pi_1} \rightarrow \mathcal{I}_{\Pi_2}$.
- Polynomial-time computable function $(S_2 \in \mathcal{F}(f(I)))$ $g\colon S_2 \mapsto S_1 \in \mathcal{F}(I)$
- S_2 locally optimal $\Rightarrow g(S_2)$ locally optimal.



MaxCut

Schäffer, Yannakakis (1991)

Finding a locally optimal cut is PLS-complete.

MaxCut

Schäffer, Yannakakis (1991)

Finding a locally optimal cut is PLS-complete.

Congestion Games are in PLS

- objective function = Rosenthal's potential function
- $S' \in \mathcal{N}(S)$ if S' is obtained from S by better response of some player

MaxCut

Schäffer, Yannakakis (1991)

Finding a locally optimal cut is PLS-complete.

Congestion Games are in PLS

- objective function = Rosenthal's potential function
- $S' \in \mathcal{N}(S)$ if S' is obtained from S by better response of some player

Fabrikant, Papadimitriou, Talwar (STOC 2004)

Finding a pure Nash equilibrium in network congestion games is PLS-complete.

Threshold Congestion Games

Threshold Congestion Games

$\mathcal{R} = \mathcal{R}_{\mathrm{in}} \dot{\cup} \mathcal{R}_{\mathrm{out}}$. Every player i has two strategies:

- in: an arbitrary subset $\mathcal{S}_i \subseteq \mathcal{R}_{\mathrm{in}}$
- out: $\{r_i\}$ for a unique resource $r_i \in \mathcal{R}_{\mathrm{out}}$ with fixed delay, the so-called threshold t_i

Threshold Congestion Games

Threshold Congestion Games

$\mathcal{R} = \mathcal{R}_{\text{in}} \dot\cup \mathcal{R}_{\text{out}}$. Every player i has two strategies:

in: an arbitrary subset $\mathcal{S}_i \subseteq \mathcal{R}_{\text{in}}$

out: $\{r_i\}$ for a unique resource $r_i \in \mathcal{R}_{\text{out}}$ with fixed delay, the so-called threshold t_i

2-threshold games

Each resource $r \in \mathcal{R}_{\text{in}}$ is contained in the strategies of exactly two players.

Threshold Congestion Games

Threshold Congestion Games

$\mathcal{R} = \mathcal{R}_{\text{in}} \dot\cup \mathcal{R}_{\text{out}}$. Every player i has two strategies:

in: an arbitrary subset $\mathcal{S}_i \subseteq \mathcal{R}_{\text{in}}$

out: $\{r_i\}$ for a unique resource $r_i \in \mathcal{R}_{\text{out}}$ with fixed delay, the so-called threshold t_i

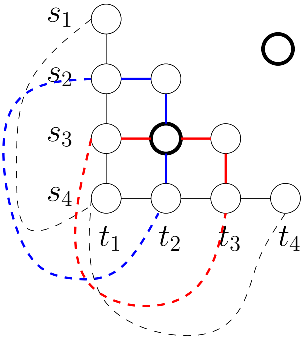
2-threshold games

Each resource $r \in \mathcal{R}_{\text{in}}$ is contained in the strategies of exactly two players.

Theorem

2-threshold congestion games are PLS-complete.

Reduction



$r_{2,3} \in \mathcal{S}_2 \cup \mathcal{S}_3$

Theorem

Network congestion games are PLS-complete for (un)directed networks with linear delay functions.

Conclusions and Open Questions

- 1-2-exchanges ⇒ exponentially long best response sequences
- 1-k-exchanges ⇒ PLS-completeness
- Threshold Congestion Games are a good starting point for PLS-reductions.

Conclusions and Open Questions

- 1-2-exchanges ⇒ exponentially long best response sequences
- 1-k-exchanges ⇒ PLS-completeness
- Threshold Congestion Games are a good starting point for PLS-reductions.

Open Questions

- Are 1-2-exchanges sufficient to construct a state from which every best response sequence is exponentially long?
- How large has k to be in order to prove PLS-completeness?

Thank you!

Questions?