Strategic Behavior in Network Routing

Guido Schäfer
CWI Amsterdam / VU University Amsterdam
g.schaefer@cwi.nl

School:

Social, Information, and Routing Networks:
Models, Algorithms, and Strategic Behavior

February 29–March 2, 2012
Porto Alegre, Brazil

Motivation

Situations of strategic interaction

Viewpoint: many real-world problems are complex and distributed in nature:

1. involve several independent decision makers (lack of coordination)
2. each decision maker attempts to achieve his own goals (strategic behavior)
3. individual outcome depends on decisions made by others (interdependent)

Examples:

- routing in networks
- Internet applications
- auctions
- ...

Situations of strategic interaction

Viewpoint: many real-world problems are complex and distributed in nature:

1. involve several independent decision makers (lack of coordination)
2. each decision maker attempts to achieve his own goals (strategic behavior)
3. individual outcome depends on decisions made by others (interdependent)

Examples:

- routing in networks
- Internet applications
- auctions
- ...

Situations of strategic interaction

Viewpoint: many real-world problems are complex and distributed in nature:

1. involve several independent decision makers (lack of coordination)
2. each decision maker attempts to achieve his own goals (strategic behavior)
3. individual outcome depends on decisions made by others (interdependent)

Examples:

- routing in networks
- Internet applications
- auctions
- ...

Situations of strategic interaction

Viewpoint: many real-world problems are complex and distributed in nature:

1. involve several independent decision makers (lack of coordination)
2. each decision maker attempts to achieve his own goals (strategic behavior)
3. individual outcome depends on decisions made by others (interdependent)

Examples:

- routing in networks
- Internet applications
- auctions
- ...

Situations of strategic interaction

Viewpoint: many real-world problems are complex and distributed in nature:

1. involve several independent decision makers (lack of coordination)
2. each decision maker attempts to achieve his own goals (strategic behavior)
3. individual outcome depends on decisions made by others (interdependent)

Examples:

- routing in networks
- Internet applications
- auctions
- ...

Network routing

Network routing: (think of urban road traffic)

- large number of commuters want to travel from their origins to their destinations in a given network
- commuters are autonomous and choose their routes so as to minimize their individual travel times
- travel time along a network link is affected by congestion
- individual travel time depends on the choices made by others

Applications: road traffic, public transportation, Internet routing, etc.

Observation: selfish route choices lead to inefficient outcomes

Network routing

Network routing: (think of urban road traffic)

- large number of commuters want to travel from their origins to their destinations in a given network
- commuters are autonomous and choose their routes so as to minimize their individual travel times
- travel time along a network link is affected by congestion
- individual travel time depends on the choices made by others

Applications: road traffic, public transportation, Internet routing, etc.

Observation: selfish route choices lead to inefficient outcomes

Network routing

Network routing: (think of urban road traffic)

- large number of commuters want to travel from their origins to their destinations in a given network
- commuters are autonomous and choose their routes so as to minimize their individual travel times
- travel time along a network link is affected by congestion
- individual travel time depends on the choices made by others

Applications: road traffic, public transportation, Internet routing, etc.

Observation: selfish route choices lead to inefficient outcomes

Inefficient outcomes

Consequences

Negative consequences:

- environmental pollution
- waste of natural resources, time and money
- stress on the traffic participants
- ...

"In 2010, congestion caused urban Americans to travel 4.8 billion hours more and to purchase an extra 1.9 billion gallons of fuel for a congestion cost of $101 billion."

[Texas Transportation Institute, 2011 Urban Mobility Report]

Need: gain fundamental understanding of the effect of strategic interactions in such applications

Consequences

Negative consequences:

- environmental pollution
- waste of natural resources, time and money
- stress on the traffic participants
- ...

"In 2010, congestion caused urban Americans to travel 4.8 billion hours more and to purchase an extra 1.9 billion gallons of fuel for a congestion cost of $101 billion."

[Texas Transportation Institute, 2011 Urban Mobility Report]

Need: gain fundamental understanding of the effect of strategic interactions in such applications

Consequences

Negative consequences:

- environmental pollution
- waste of natural resources, time and money
- stress on the traffic participants
- ...

"In 2010, congestion caused urban Americans to travel 4.8 billion hours more and to purchase an extra 1.9 billion gallons of fuel for a congestion cost of $101 billion."

[Texas Transportation Institute, 2011 Urban Mobility Report]

Need: gain fundamental understanding of the effect of strategic interactions in such applications

Approach

Game theory: provides mathematical toolbox to study situations of strategic interaction

- different models of games
- several solution concepts for the prediction of “rational outcomes”

Algorithmic game theory:

- use game-theoretical models and solution concepts
- take computational/algorithmic issues into account

Goals:

- study the effect of strategic behavior
- analyze the efficiency loss due to lack of coordination
- provide algorithmic means to reduce the inefficiency

Approach

Game theory: provides mathematical toolbox to study situations of strategic interaction

- different models of games
- several solution concepts for the prediction of “rational outcomes”

Algorithmic game theory:

- use game-theoretical models and solution concepts
- take computational/algorithmic issues into account

Goals:

- study the effect of strategic behavior
- analyze the efficiency loss due to lack of coordination
- provide algorithmic means to reduce the inefficiency

Approach

Game theory: provides mathematical toolbox to study situations of strategic interaction

- different models of games
- several solution concepts for the prediction of “rational outcomes”

Algorithmic game theory:

- use game-theoretical models and solution concepts
- take computational/algorithmic issues into account

Goals:

- study the effect of strategic behavior
- analyze the efficiency loss due to lack of coordination
- provide algorithmic means to reduce the inefficiency

Today's goals

Goals for today:

- get a glimpse of the research in algorithmic game theory
- running example: network routing
- get an idea of the phenomena and questions that arise
- familiarize with used approaches and techniques
- approach the state-of-the-art of recent studies (perhaps)

Algorithmic game theory ...

... is a young, challenging, interdisciplinary research field
... addresses many aspects of practical relevance
... can have an impact on real-world applications
... unites ideas from game theory, algorithms, combinatorial optimization, complexity theory, etc.
... is fun!

Today's goals

Goals for today:

- get a glimpse of the research in algorithmic game theory
- running example: network routing
- get an idea of the phenomena and questions that arise
- familiarize with used approaches and techniques
- approach the state-of-the-art of recent studies (perhaps)

Algorithmic game theory ...

... is a young, challenging, interdisciplinary research field
... addresses many aspects of practical relevance
... can have an impact on real-world applications
... unites ideas from game theory, algorithms, combinatorial optimization, complexity theory, etc.
... is fun!

Today's goals

Goals for today:

- get a glimpse of the research in algorithmic game theory
- running example: network routing
- get an idea of the phenomena and questions that arise
- familiarize with used approaches and techniques
- approach the state-of-the-art of recent studies (perhaps)

Algorithmic game theory ...

... is a young, challenging, interdisciplinary research field
... addresses many aspects of practical relevance
... can have an impact on real-world applications
... unites ideas from game theory, algorithms, combinatorial optimization, complexity theory, etc.
... is fun!

Today's goals

Goals for today:

- get a glimpse of the research in algorithmic game theory
- running example: network routing
- get an idea of the phenomena and questions that arise
- familiarize with used approaches and techniques
- approach the state-of-the-art of recent studies (perhaps)

Algorithmic game theory ...

... is a young, challenging, interdisciplinary research field
... addresses many aspects of practical relevance
... can have an impact on real-world applications
... unites ideas from game theory, algorithms, combinatorial optimization, complexity theory, etc.
... is fun!

Today's goals

Goals for today:

- get a glimpse of the research in algorithmic game theory
- running example: network routing
- get an idea of the phenomena and questions that arise
- familiarize with used approaches and techniques
- approach the state-of-the-art of recent studies (perhaps)

Algorithmic game theory ...

... is a young, challenging, interdisciplinary research field
... addresses many aspects of practical relevance
... can have an impact on real-world applications
... unites ideas from game theory, algorithms, combinatorial optimization, complexity theory, etc.
... is fun!

Today's goals

Goals for today:

- get a glimpse of the research in algorithmic game theory
- running example: network routing
- get an idea of the phenomena and questions that arise
- familiarize with used approaches and techniques
- approach the state-of-the-art of recent studies (perhaps)

Algorithmic game theory ...

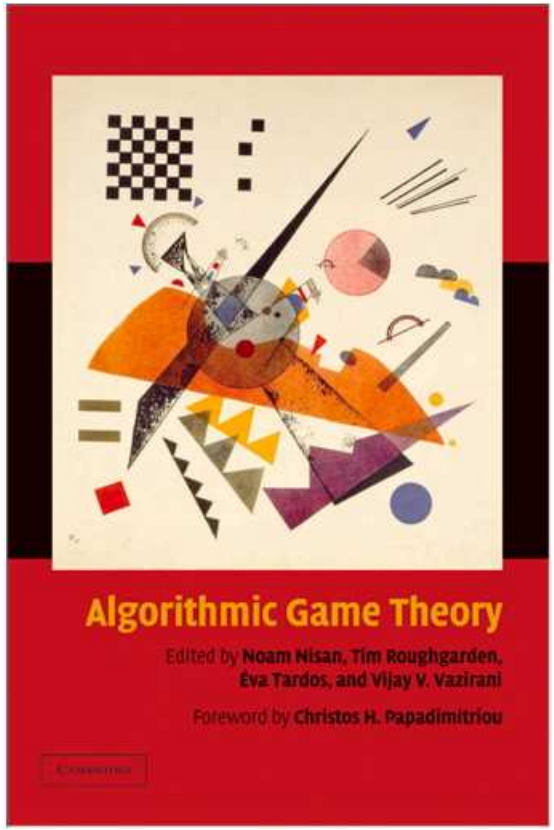
... is a young, challenging, interdisciplinary research field
... addresses many aspects of practical relevance
... can have an impact on real-world applications
... unites ideas from game theory, algorithms, combinatorial optimization, complexity theory, etc.
... is fun!

AGT book

N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani (Eds.)
Algorithmic Game Theory
Cambridge University Press, 2007.

Available online (free!) at:

http://www.cambridge.org/us/9780521872829



Roadmap

1. Examples of Some Phenomena
2. Wardrop Model and the Price of Anarchy
3. Coping with the Braess Paradox
4. Stackelberg Routing
5. Network Tolls

CWI

Examples of Some Phenomena

Cost minimization games

A **cost minimization game** G is given by

- set of players $N = \{1, \ldots, n\}$
- set of strategies S_i for every player $i \in N$
- cost function $C_i : S_1 \times \cdots \times S_n \to \mathbb{R}$

$S = S_1 \times \cdots \times S_n$ is called the set of strategy profiles.

Interpretation:

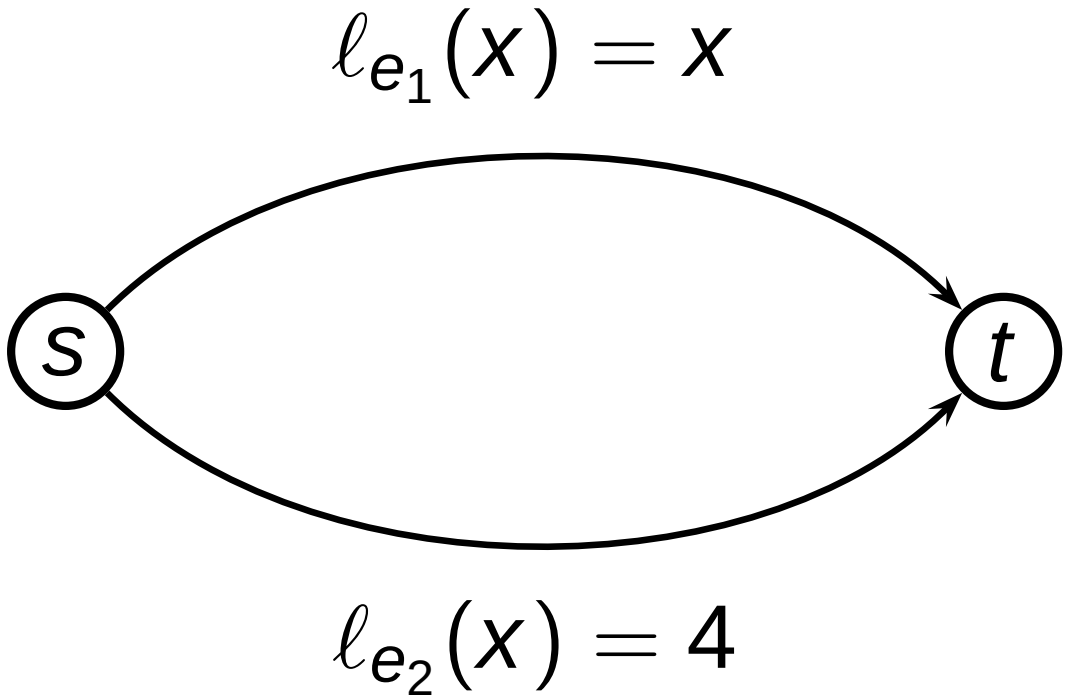
- every player $i \in N$ chooses a strategy $s_i \in S_i$ so as to minimize his individual cost $C_i(s_1, \ldots, s_n)$
- one-shot, simultaneous-move: players choose their strategies once and at the same time
- full-information: every player knows the strategies and cost function of every other player

[World Cup 2006 Example]

Example: Congestion game

$$n = 4$$

$$S_i = \{e_1, e_2\}$$

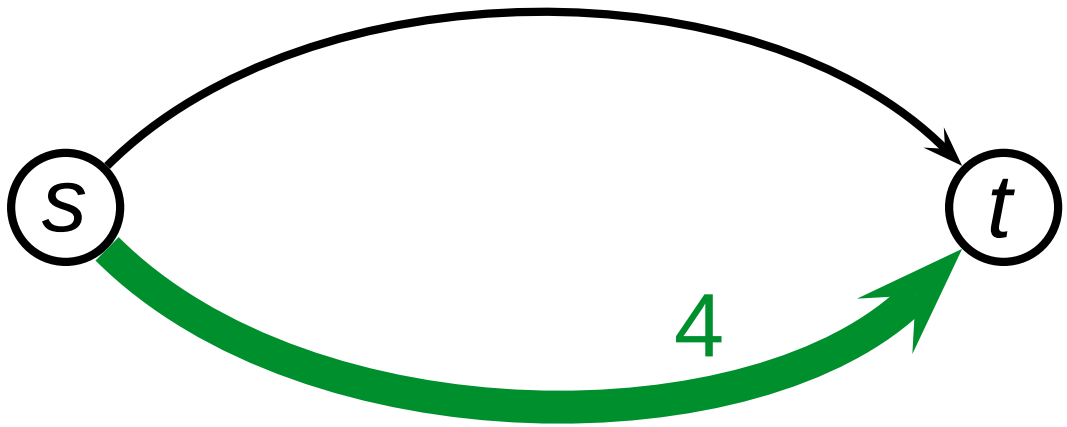


Example: Congestion game

$$n = 4$$

$$S_i = \{e_1, e_2\}$$

$$\ell_{e_1}(x) = x$$

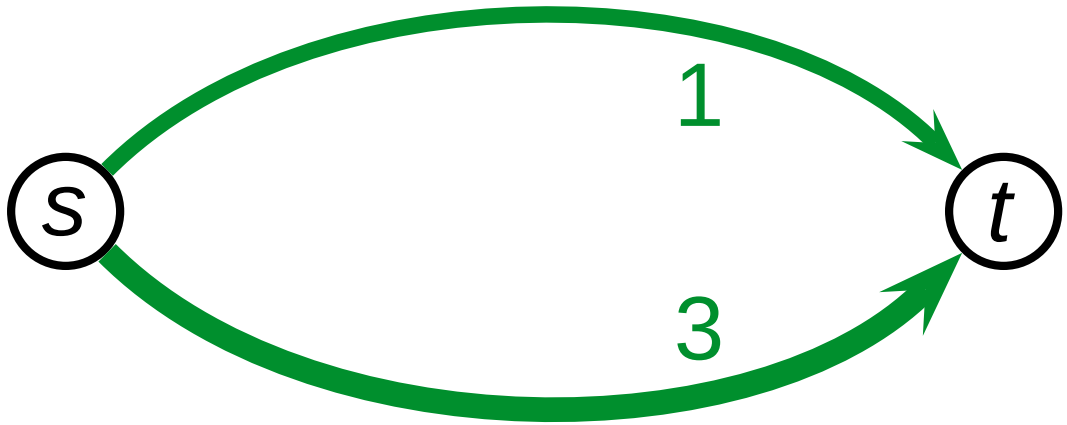


$$\ell_{e_2}(x) = 4$$

Example: Congestion game

$$n = 4$$
$$S_i = \{e_1, e_2\}$$

$$\ell_{e_1}(x) = x$$

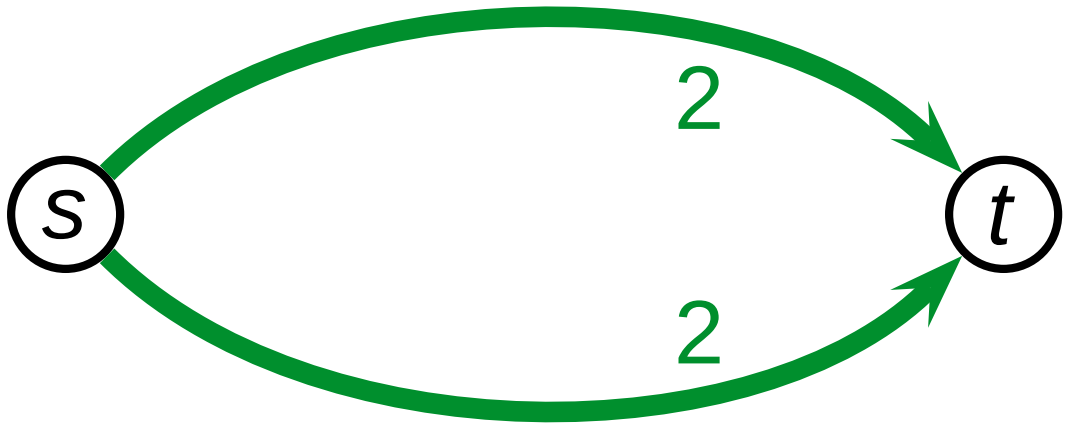


$$\ell_{e_2}(x) = 4$$

Example: Congestion game

$$n = 4$$
$$S_i = \{e_1, e_2\}$$

$$\ell_{e_1}(x) = x$$

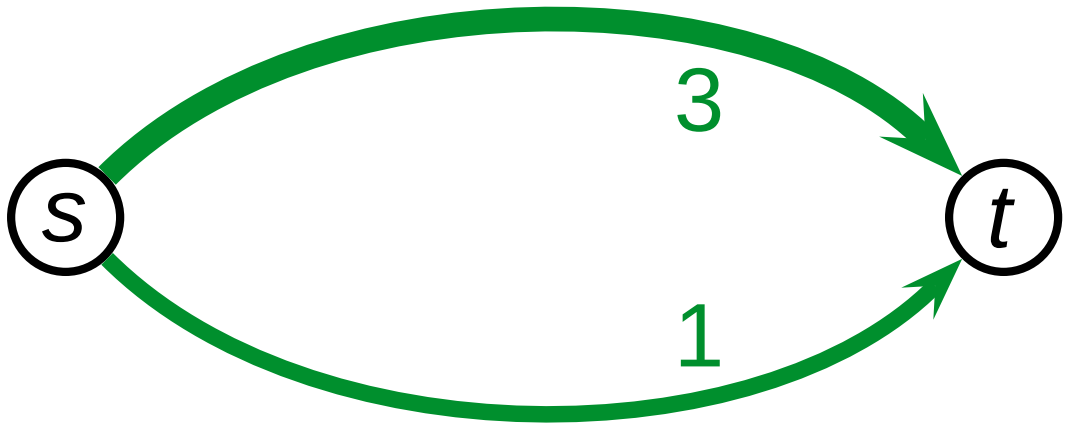


$$\ell_{e_2}(x) = 4$$

Example: Congestion game

$$n = 4$$
$$S_i = \{e_1, e_2\}$$

$$\ell_{e_1}(x) = x$$



$$\ell_{e_2}(x) = 4$$

Nash equilibrium and social cost

Nash equilibrium: strategy profile $s = (s_1, \ldots, s_n) \in S$ is a pure Nash equilibrium (PNE) if no player has an incentive to unilaterally deviate

$$\forall i \in N: \quad C_i(s_i, s_{-i}) \leq C_i(s_i', s_{-i}) \qquad \forall s_i' \in S_i$$

(s_{-i} refers to $(s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n)$)

Social cost of strategy profile $s = (s_1, \ldots, s_n) \in S$ is

$$C(s) = \sum_{i \in N} C_i(s)$$

A strategy profile s^* that minimizes the social cost function $C(\cdot)$ is called a social optimum.

Nash equilibrium and social cost

Nash equilibrium: strategy profile $s = (s_1, \ldots, s_n) \in S$ is a pure Nash equilibrium (PNE) if no player has an incentive to unilaterally deviate

$$\forall i \in N: \quad C_i(s_i, s_{-i}) \leq C_i(s'_i, s_{-i}) \qquad \forall s'_i \in S_i$$

(s_{-i} refers to $(s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n)$)

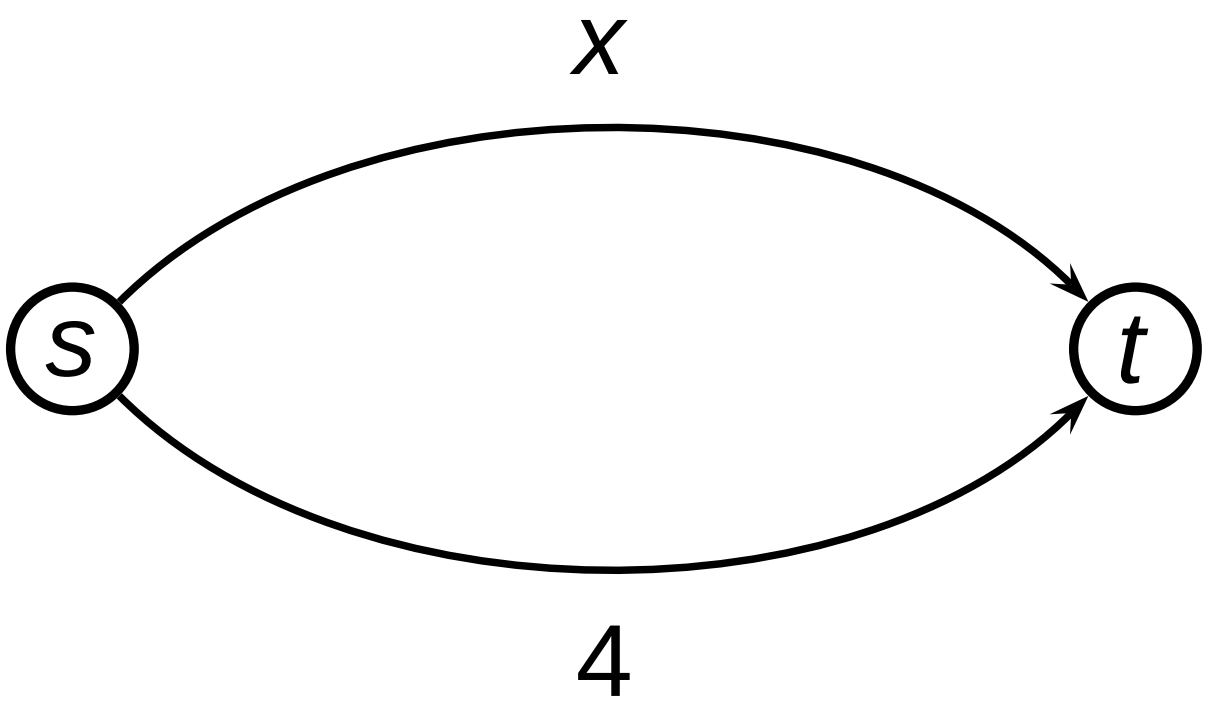
Social cost of strategy profile $s = (s_1, \ldots, s_n) \in S$ is

$$C(s) = \sum_{i \in N} C_i(s)$$

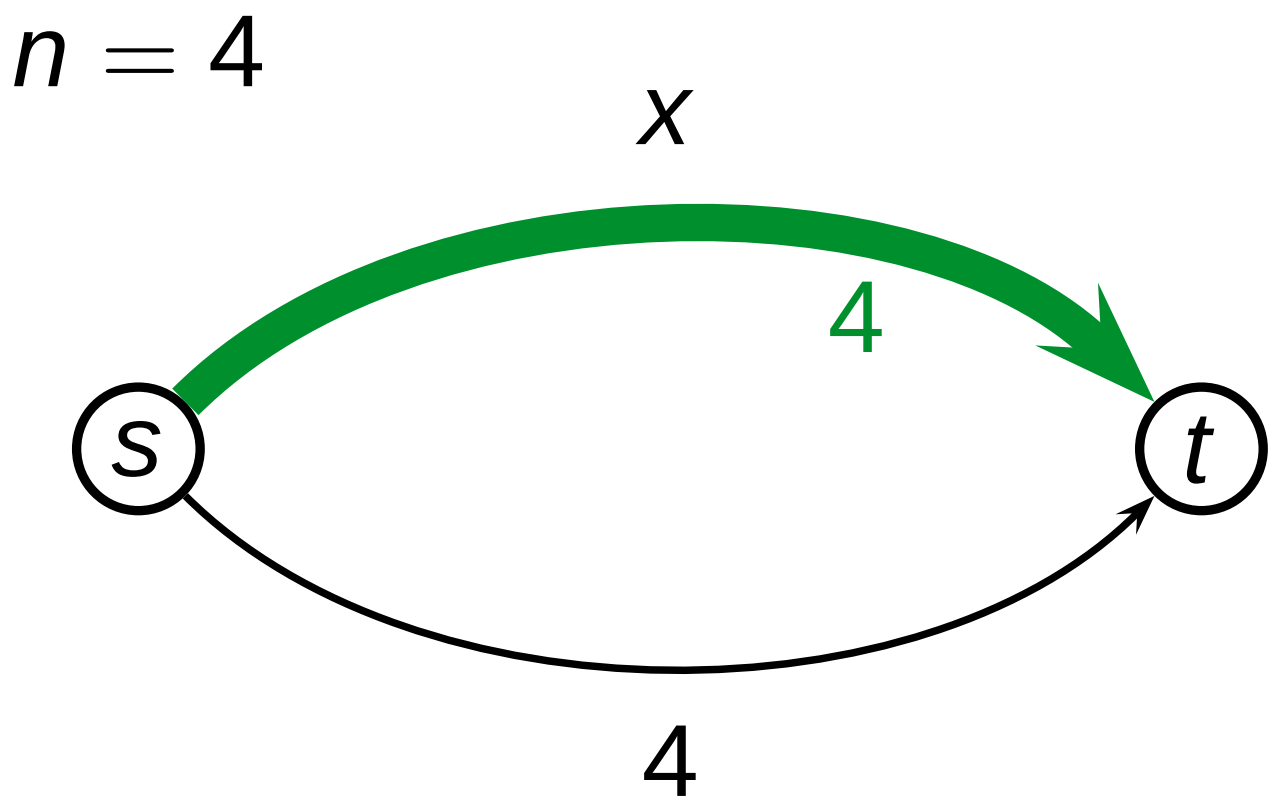
A strategy profile s^* that minimizes the social cost function $C(\cdot)$ is called a social optimum.

Example: Congestion game

$n = 4$



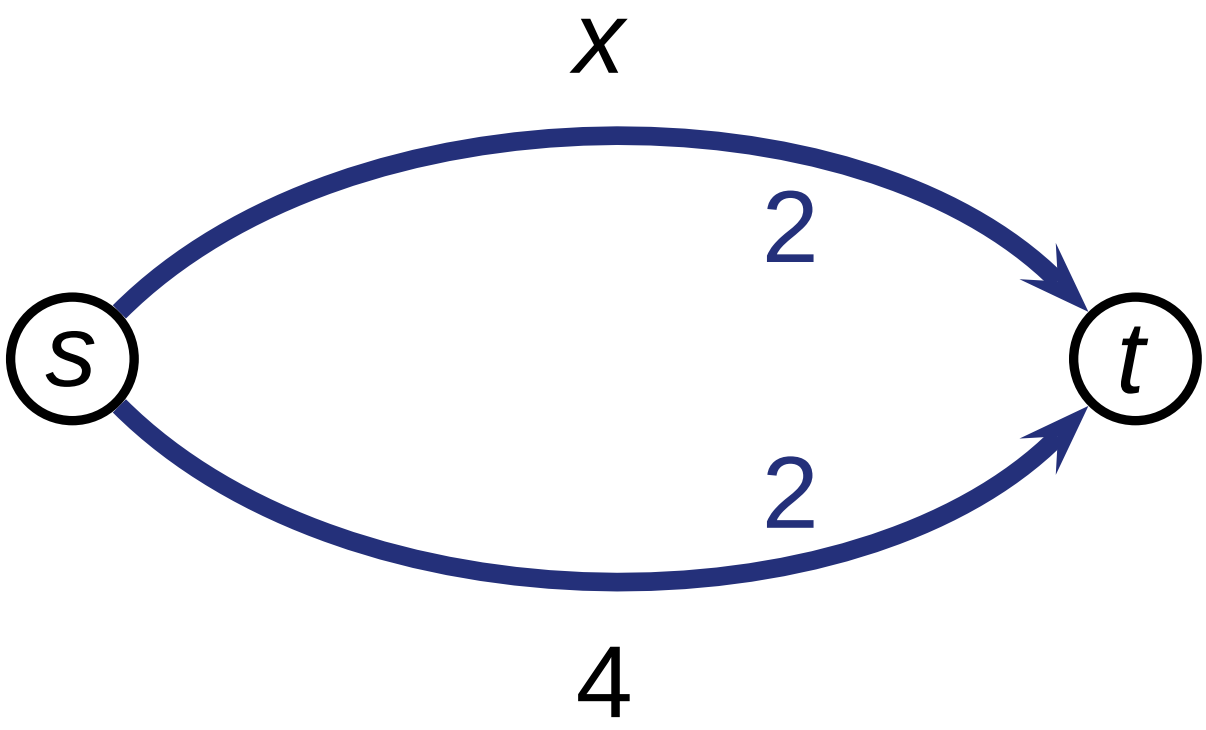
Example: Congestion game



Nash equilibrium: $C(s) = 4 \cdot 4 = 16$

Example: Congestion game

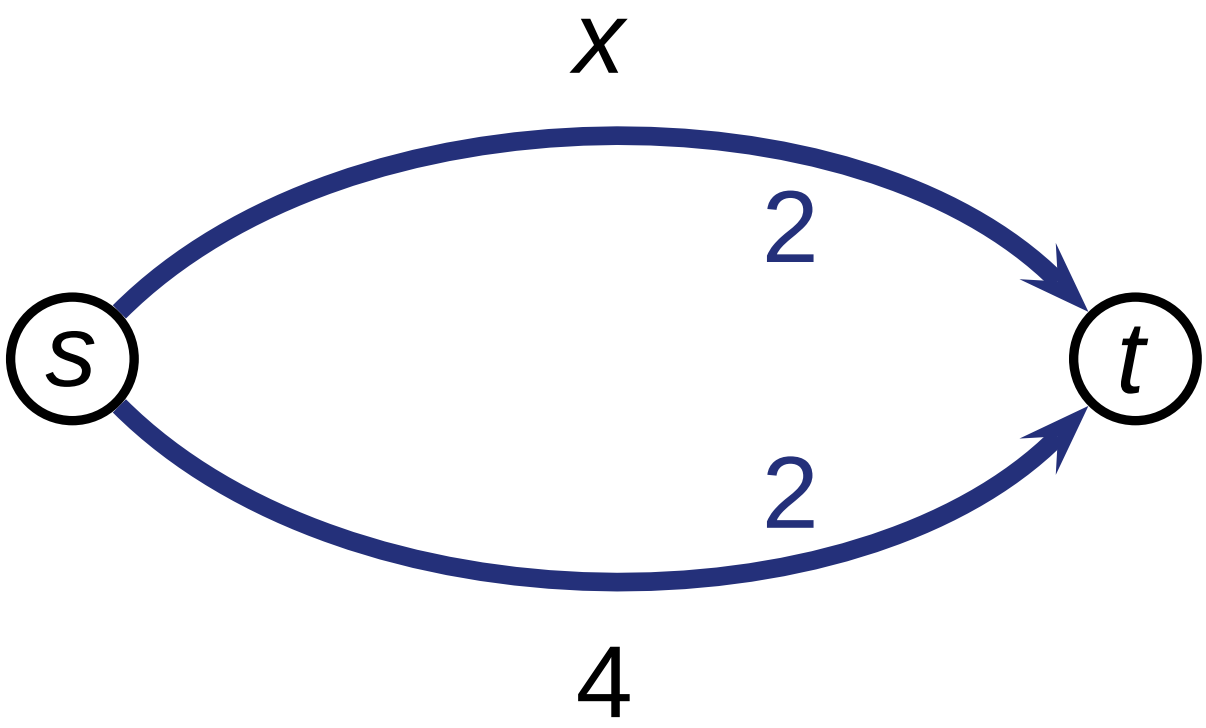
$n = 4$



social optimum: $C(s^*) = 4 + 8 = 12$

Example: Congestion game

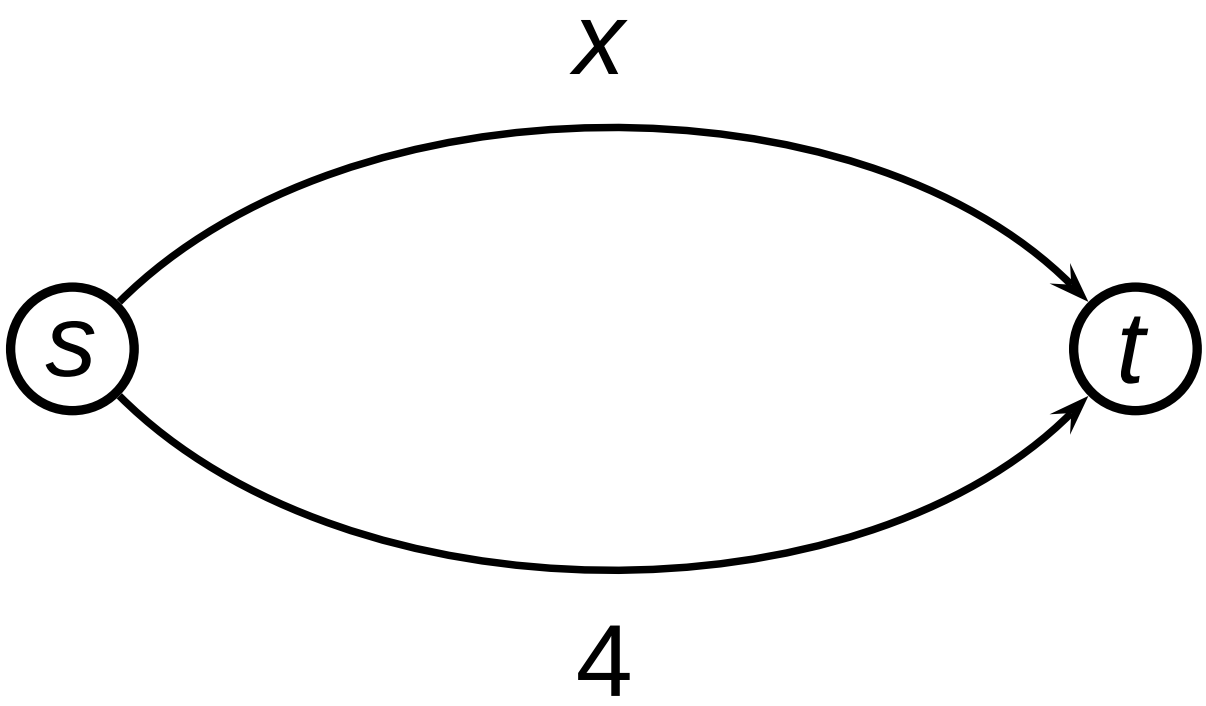
$n = 4$



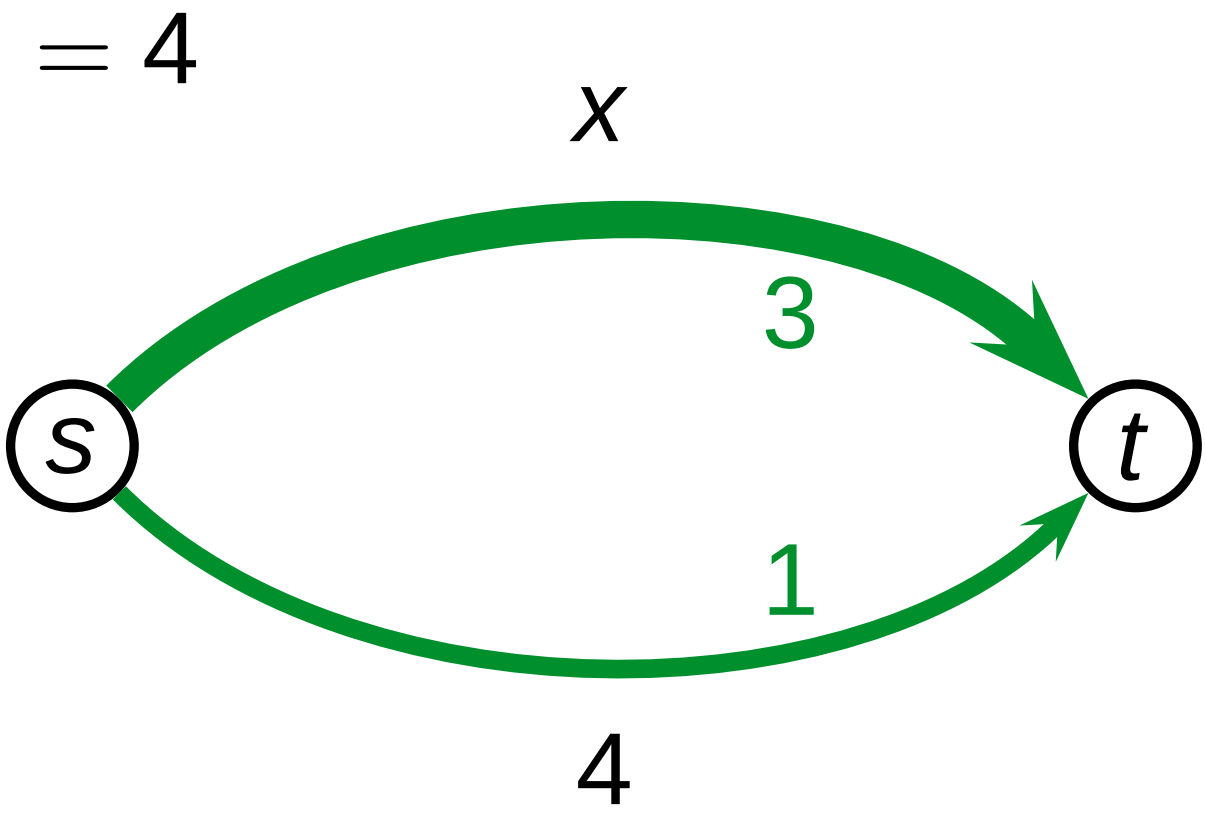
inefficiency: $\frac{C(s)}{C(s^*)} = \frac{16}{12} = \frac{4}{3}$

Example: Congestion game

$n = 4$



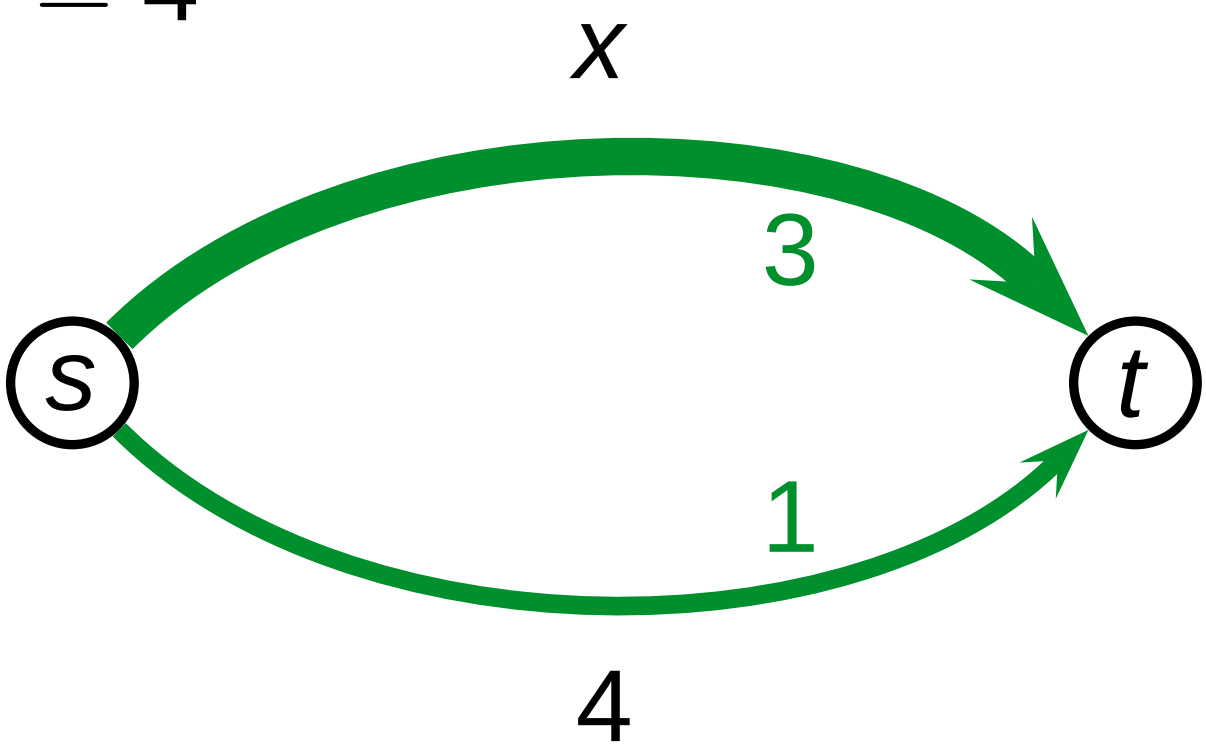
Example: Congestion game



Nash equilibrium: $C(s) = 9 + 4 = 13$

Example: Congestion game

$n = 4$



inefficiency: $\frac{C(s)}{C(s^*)} = \frac{13}{12} \approx 1.08$

Inefficiency of equilibria

Price of anarchy: worst-case inefficiency of equilibria

$$POA(G) = \max_{s \in PNE(G)} \frac{C(s)}{C(s^*)}$$

[Koutsoupias, Papadimitriou, STACS '99]

Price of stability: best-case inefficiency of equilibria

$$POS(G) = \min_{s \in PNE(G)} \frac{C(s)}{C(s^*)}$$

[Schulz, Moses, SODA '03]

Define the price of anarchy/stability of a class of games $\mathcal{G}$ as

$$POA(\mathcal{G}) = \max_{G \in \mathcal{G}} POA(G) \quad \text{and} \quad POS(\mathcal{G}) = \max_{G \in \mathcal{G}} POS(G)$$

Inefficiency of equilibria

Price of anarchy: worst-case inefficiency of equilibria

$$POA(G) = \max_{s \in PNE(G)} \frac{C(s)}{C(s^*)}$$

[Koutsoupias, Papadimitriou, STACS '99]

Price of stability: best-case inefficiency of equilibria

$$POS(G) = \min_{s \in PNE(G)} \frac{C(s)}{C(s^*)}$$

[Schulz, Moses, SODA '03]

Define the price of anarchy/stability of a class of games $\mathcal{G}$ as

$$POA(\mathcal{G}) = \max_{G \in \mathcal{G}} POA(G) \quad \text{and} \quad POS(\mathcal{G}) = \max_{G \in \mathcal{G}} POS(G)$$

Inefficiency of equilibria

Price of anarchy: worst-case inefficiency of equilibria

$$POA(G) = \max_{s \in PNE(G)} \frac{C(s)}{C(s^*)}$$

[Koutsoupias, Papadimitriou, STACS '99]

Price of stability: best-case inefficiency of equilibria

$$POS(G) = \min_{s \in PNE(G)} \frac{C(s)}{C(s^*)}$$

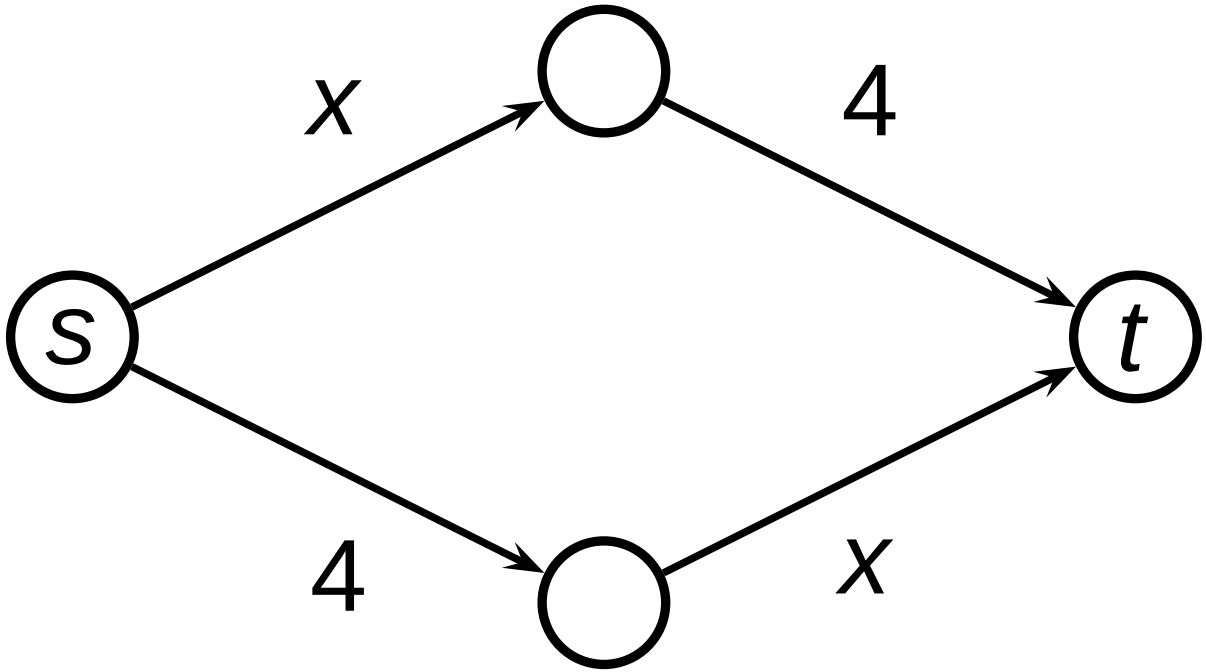
[Schulz, Moses, SODA '03]

Define the price of anarchy/stability of a class of games $\mathcal{G}$ as

$$POA(\mathcal{G}) = \max_{G \in \mathcal{G}} POA(G) \quad \text{and} \quad POS(\mathcal{G}) = \max_{G \in \mathcal{G}} POS(G)$$

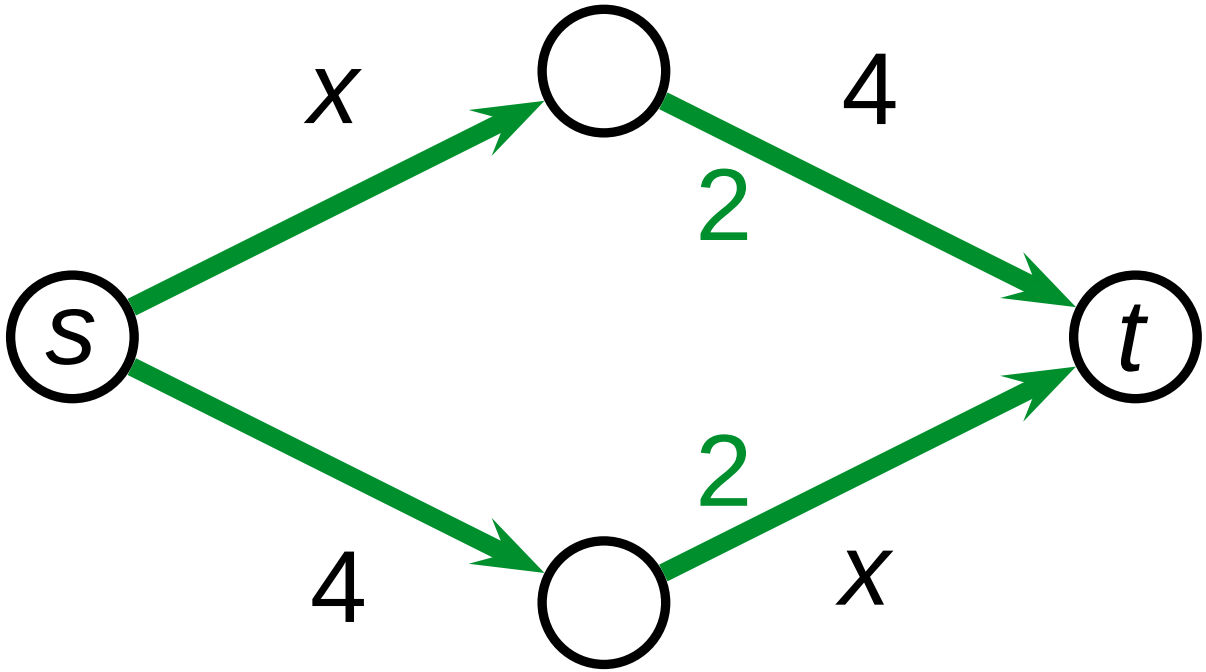
Example: Braess Paradox

$n = 4$



Example: Braess Paradox

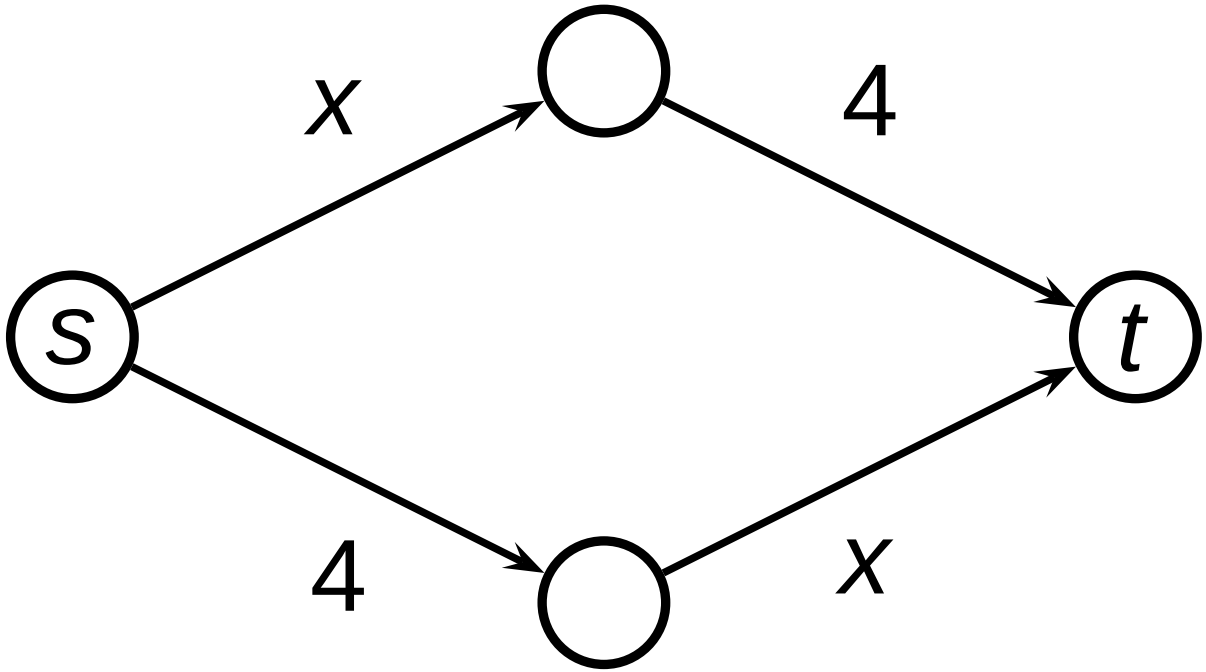
$n = 4$



Nash equilibrium: $C(s) = 24$

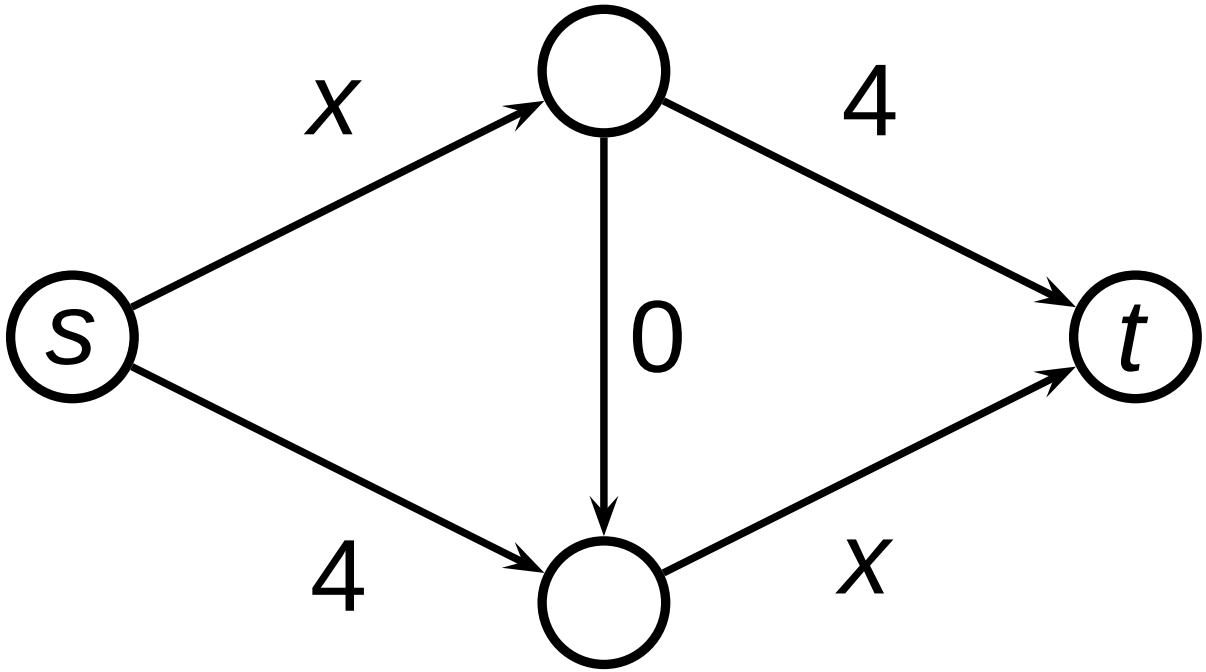
Example: Braess Paradox

$n = 4$



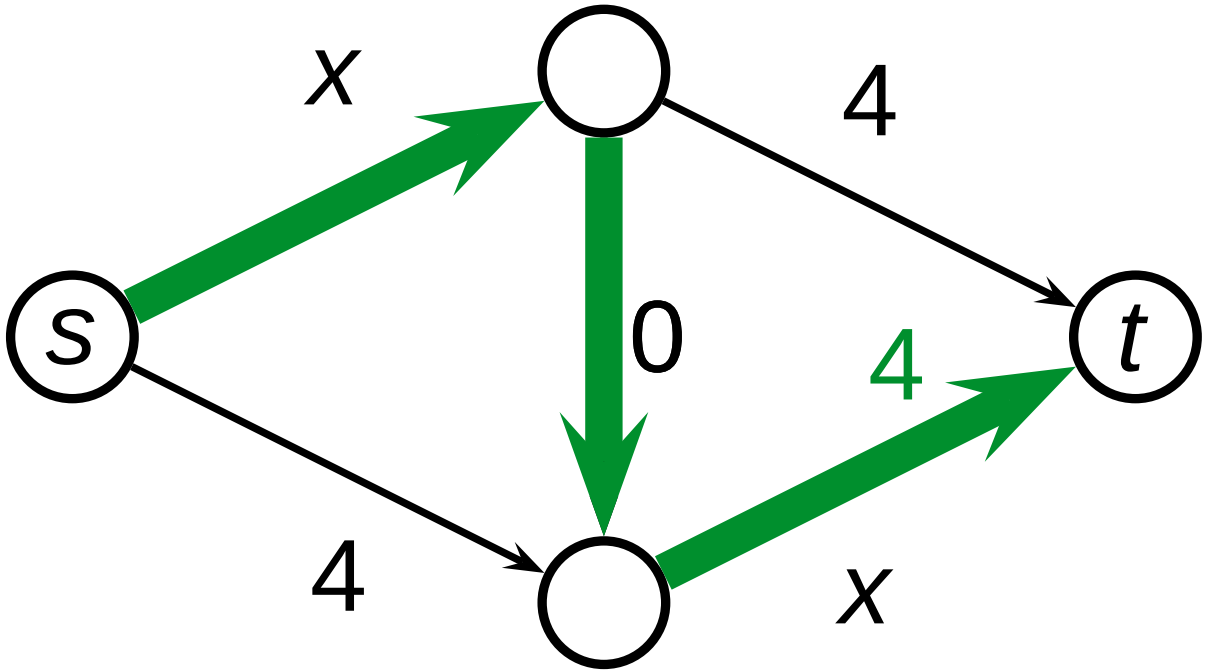
Example: Braess Paradox

$n = 4$



Example: Braess Paradox

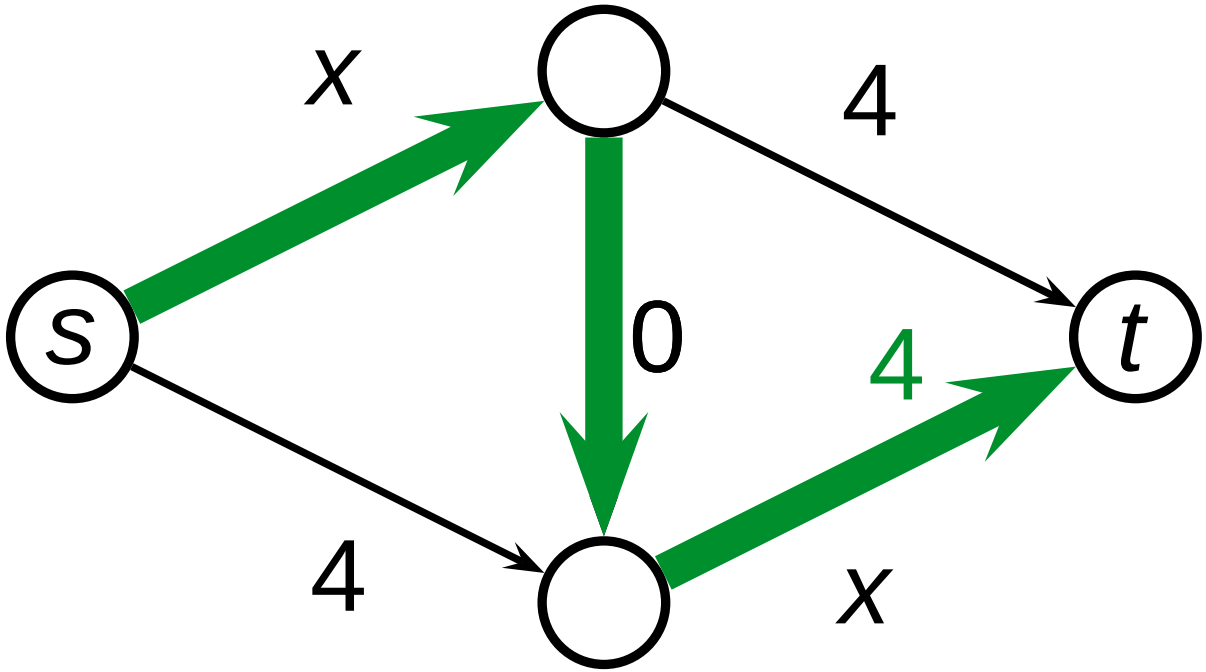
$n = 4$



Nash equilibrium: $C(s) = 32$

Example: Braess Paradox

$n = 4$

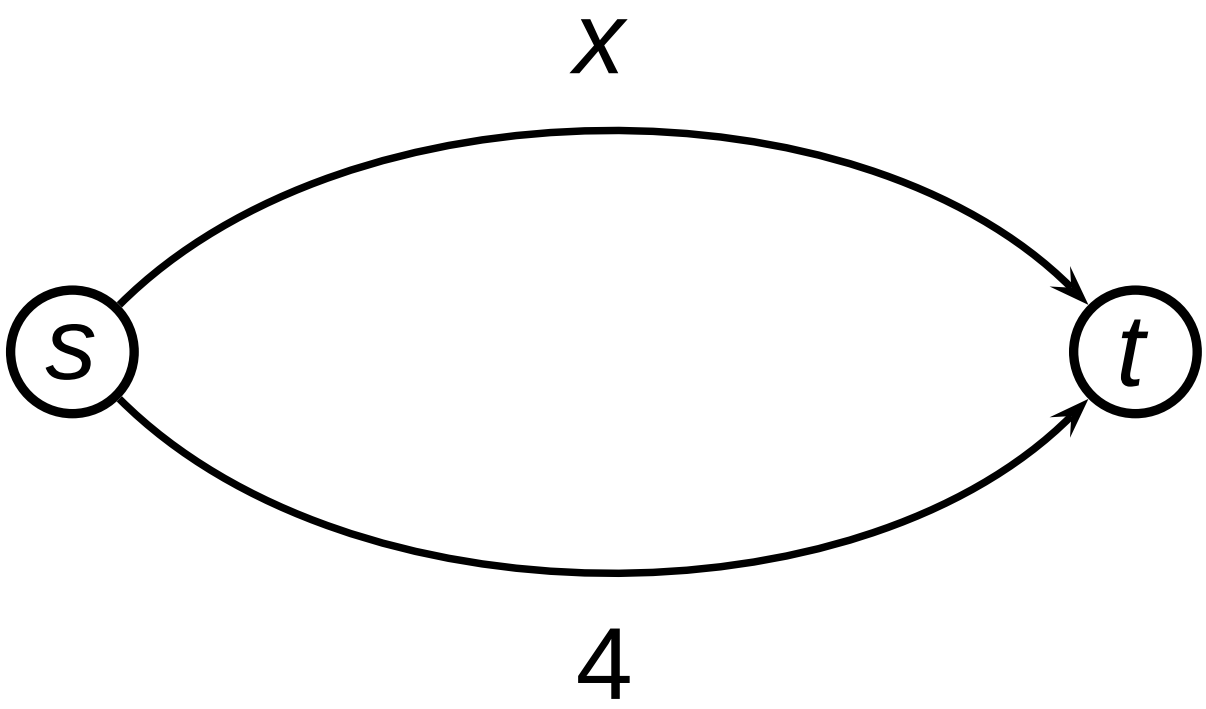


cost increase: $\frac{32}{24} = \frac{4}{3}$

[New York Times Article]

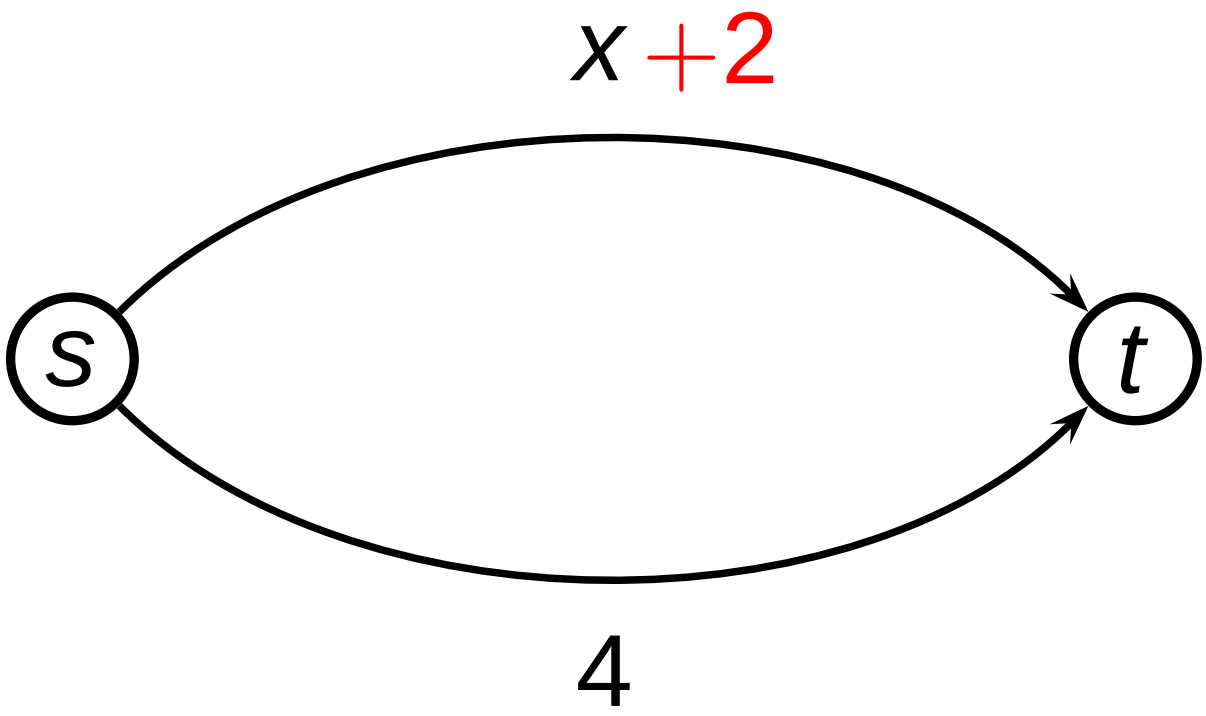
Example: Network tolls

$n = 4$

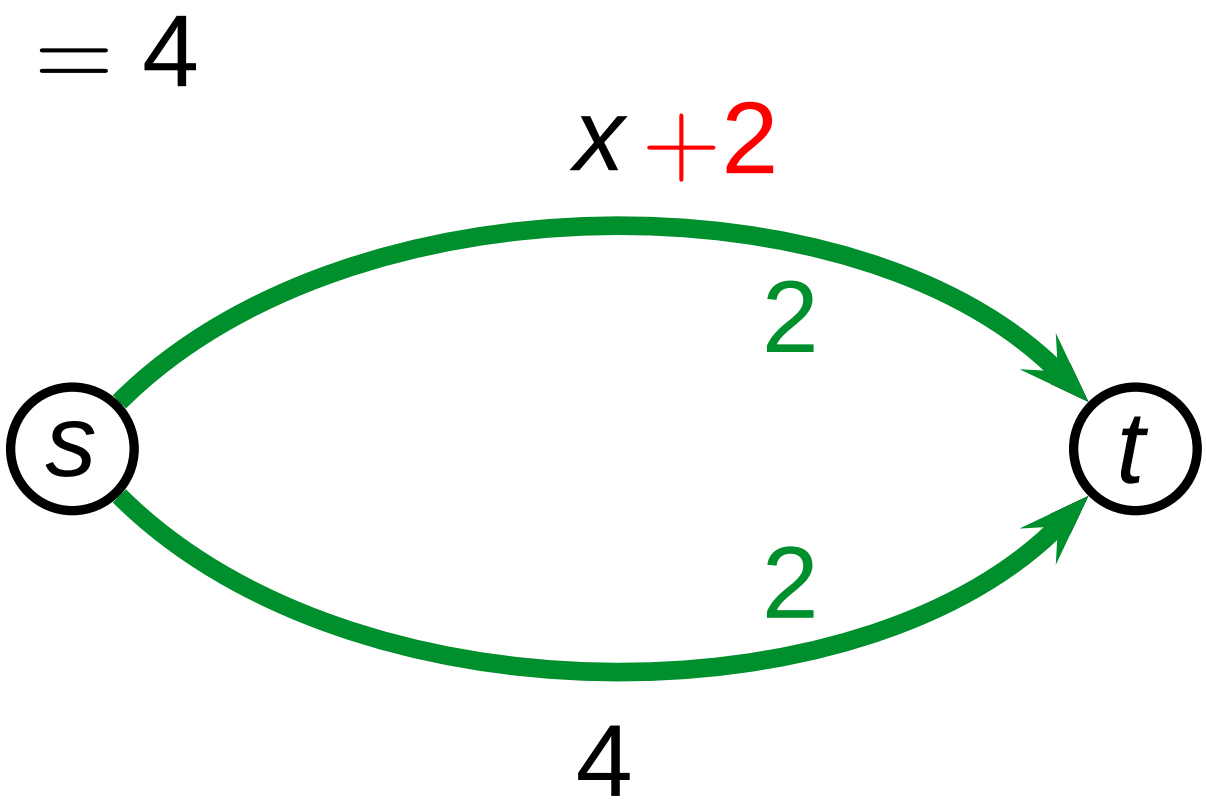


Example: Network tolls

$n = 4$



Example: Network tolls

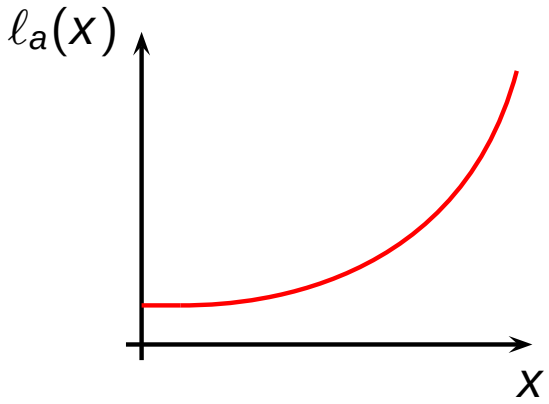


Nash equilibrium = social optimum

Wardrop Model and the Price of Anarchy

Wardrop model

Given:

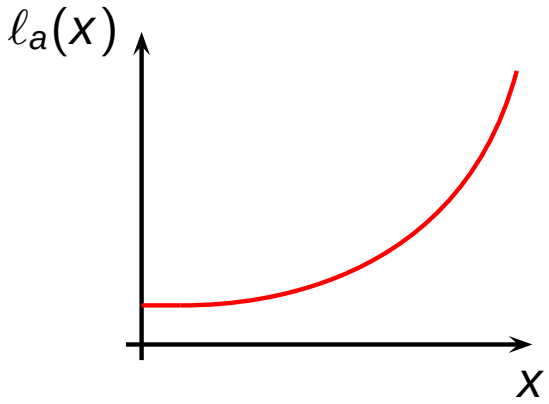


- directed network $G = (V, A)$
- k commodities $(s_1, t_1), \ldots, (s_k, t_k)$
- demand r_i for commodity $i \in [k]$
- flow-dependent latency function $\ell_a(\cdot)$ for every arc $a \in A$ (non-decreasing, differentiable, semi-convex)

Interpretation:

- send r_i units of flow from s_i to t_i, corresponding to an infinitely large population of non-atomic players
- latency experienced on arc $a \in A$ is $\ell_a(x)$, where x is the amount of flow on arc a
- selfishness: players choose minimum-latency paths
- one-shot, simultaneous-move, full information game

Wardrop model



Given:

- directed network $G = (V, A)$
- k commodities $(s_1, t_1), \ldots, (s_k, t_k)$
- demand r_i for commodity $i \in [k]$
- flow-dependent latency function $\ell_a(\cdot)$ for every arc $a \in A$ (non-decreasing, differentiable, semi-convex)

Interpretation:

- send r_i units of flow from s_i to t_i, corresponding to an infinitely large population of non-atomic players
- latency experienced on arc $a \in A$ is $\ell_a(x)$, where x is the amount of flow on arc a
- selfishness: players choose minimum-latency paths
- one-shot, simultaneous-move, full information game

Optimal flows

Notation:

- $\mathcal{P}_i =$ set of all (simple) directed s_i, t_i-paths
- $\mathcal{P} = \cup_{i \in [k]} \mathcal{P}_i =$ set of all relevant paths
- flow f is a function $f : \mathcal{P} \to \mathbb{R}^+$
- f is a feasible flow if $\sum_{P \in \mathcal{P}_i} f_P = r_i$ for all i
- flow on arc a: $f_a := \sum_{P \in \mathcal{P} : a \in P} f_P$
- latency of path P: $\ell_P(f) := \sum_{a \in P} \ell_a(f_a)$

Social cost of flow:

$$C(f) = \sum_{i \in [k]} \sum_{P \in \mathcal{P}_i} f_P \ell_P(f) = \sum_{a \in A} f_a \ell_a(f_a)$$

Optimal flow: feasible flow f^* that minimizes $C(\cdot)$

Optimal flows

Notation:

- $\mathcal{P}_i =$ set of all (simple) directed s_i, t_i-paths
- $\mathcal{P} = \cup_{i \in [k]} \mathcal{P}_i =$ set of all relevant paths
- flow f is a function $f : \mathcal{P} \to \mathbb{R}^+$
- f is a feasible flow if $\sum_{P \in \mathcal{P}_i} f_P = r_i$ for all i
- flow on arc a: $f_a := \sum_{P \in \mathcal{P} : a \in P} f_P$
- latency of path P: $\ell_P(f) := \sum_{a \in P} \ell_a(f_a)$

Social cost of flow:

$$C(f) = \sum_{i \in [k]} \sum_{P \in \mathcal{P}_i} f_P \ell_P(f) = \sum_{a \in A} f_a \ell_a(f_a)$$

Optimal flow: feasible flow f^* that minimizes $C(\cdot)$

Optimal flows

Notation:

- $\mathcal{P}_i =$ set of all (simple) directed s_i, t_i-paths
- $\mathcal{P} = \cup_{i \in [k]} \mathcal{P}_i =$ set of all relevant paths
- flow f is a function $f : \mathcal{P} \to \mathbb{R}^+$
- f is a feasible flow if $\sum_{P \in \mathcal{P}_i} f_P = r_i$ for all i
- flow on arc a: $f_a := \sum_{P \in \mathcal{P} : a \in P} f_P$
- latency of path P: $\ell_P(f) := \sum_{a \in P} \ell_a(f_a)$

Social cost of flow:

$$C(f) = \sum_{i \in [k]} \sum_{P \in \mathcal{P}_i} f_P \ell_P(f) = \sum_{a \in A} f_a \ell_a(f_a)$$

Optimal flow: feasible flow f^* that minimizes $C(\cdot)$

Nash flows

Nash flow: feasible flow f that satisfies for every $i \in [k]$

$$\forall P_1, P_2 \in \mathcal{P}_i \text{ with } f_{P_1} > 0,\ \forall \delta \in (0, f_{P_1}]: \quad \ell_{P_1}(f) \leq \ell_{P_2}(\tilde{f}),$$

where

$$\tilde{f}_P := \begin{cases} f_P - \delta & \text{if } P = P_1 \\ f_P + \delta & \text{if } P = P_2 \\ f_P & \text{otherwise.} \end{cases}$$

Exploiting continuity of $\ell_a(\cdot)$ and letting $\delta \to 0$, we obtain the following equivalent definition:

Wardrop flow: feasible flow f that satisfies for every $i \in [k]$

$$\forall P_1, P_2 \in \mathcal{P}_i \text{ with } f_{P_1} > 0: \quad \ell_{P_1}(f) \leq \ell_{P_2}(f)$$

Nash flows

Nash flow: feasible flow f that satisfies for every $i \in [k]$

$$\forall P_1, P_2 \in \mathcal{P}_i \text{ with } f_{P_1} > 0,\ \forall \delta \in (0, f_{P_1}]: \quad \ell_{P_1}(f) \leq \ell_{P_2}(\tilde{f}),$$

where

$$\tilde{f}_P := \begin{cases} f_P - \delta & \text{if } P = P_1 \\ f_P + \delta & \text{if } P = P_2 \\ f_P & \text{otherwise.} \end{cases}$$

Exploiting continuity of $\ell_a(\cdot)$ and letting $\delta \to 0$, we obtain the following equivalent definition:

Wardrop flow: feasible flow f that satisfies for every $i \in [k]$

$$\forall P_1, P_2 \in \mathcal{P}_i \text{ with } f_{P_1} > 0: \quad \ell_{P_1}(f) \leq \ell_{P_2}(f)$$

Nash flows

Nash flow: feasible flow f that satisfies for every $i \in [k]$

$$\forall P_1, P_2 \in \mathcal{P}_i \text{ with } f_{P_1} > 0,\ \forall \delta \in (0, f_{P_1}]: \quad \ell_{P_1}(f) \leq \ell_{P_2}(\tilde{f}),$$

where

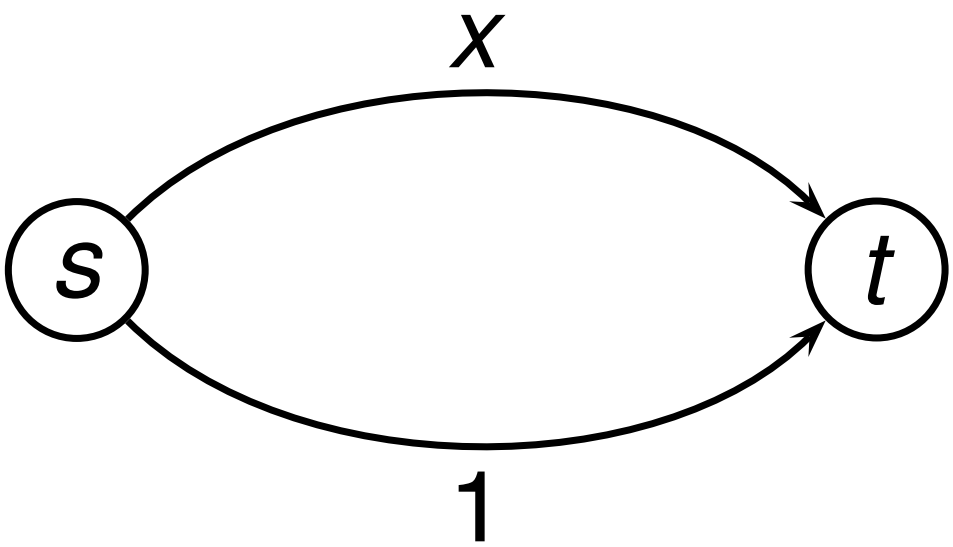
$$\tilde{f}_P := \begin{cases} f_P - \delta & \text{if } P = P_1 \\ f_P + \delta & \text{if } P = P_2 \\ f_P & \text{otherwise.} \end{cases}$$

Exploiting continuity of $\ell_a(\cdot)$ and letting $\delta \to 0$, we obtain the following equivalent definition:

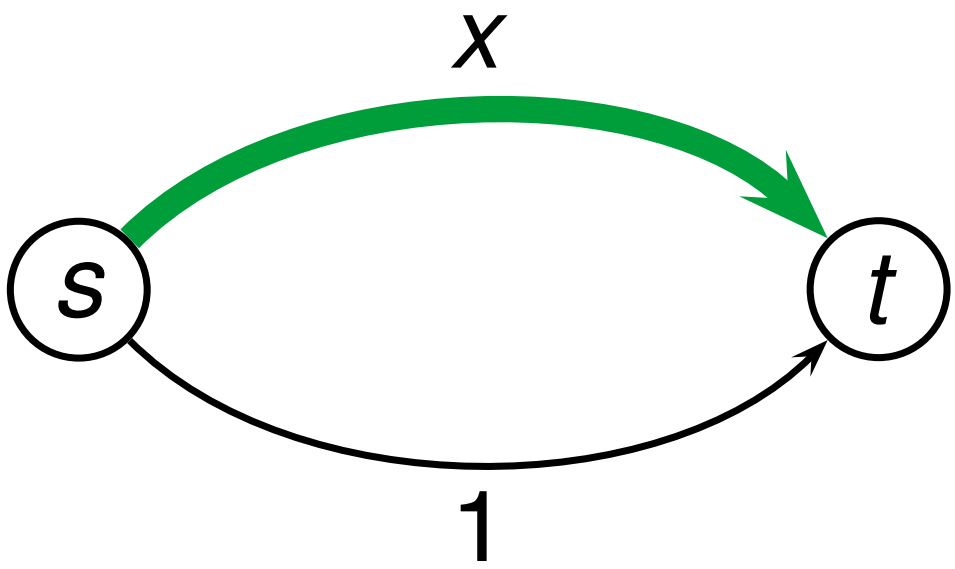
Wardrop flow: feasible flow f that satisfies for every $i \in [k]$

$$\forall P_1, P_2 \in \mathcal{P}_i \text{ with } f_{P_1} > 0: \quad \ell_{P_1}(f) \leq \ell_{P_2}(f)$$

Inefficiency of equilibria

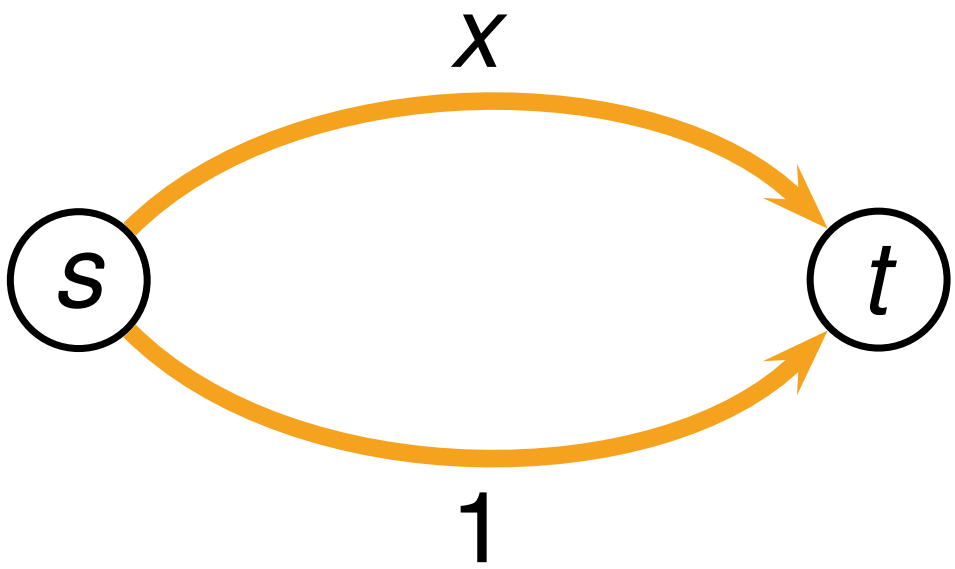


Inefficiency of equilibria



Nash flow f: $C(f) = 1$

Inefficiency of equilibria



Optimal flow f^*: $C(f^*) = \frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot 1 = \frac{3}{4}$

Existence and computation of optimal flows

$$
\begin{aligned}
\min \quad & \sum_{a \in A} \ell_a(f_a) f_a & & & \\
\text{s.t.} \quad & \sum_{P \in \mathcal{P}_i} f_P & = & r_i & \forall i \in [k] \\
& f_a & = & \sum_{P \in \mathcal{P}: a \in P} f_P & \forall a \in A \\
& f_P & \geq & 0 & \forall P \in \mathcal{P}
\end{aligned}
$$

Observations:

- minimize a continuous function (social cost) over a compact set (feasible flows)
- optimal flow must exist (extreme value theorem)
- optimal flow can be computed efficiently (convex program)

Existence and computation of optimal flows

$$
\begin{aligned}
\min \quad & \sum_{a \in A} \ell_a(f_a) f_a & & & \\
\text{s.t.} \quad & \sum_{P \in \mathcal{P}_i} f_P & = & r_i & \forall i \in [k] \\
& f_a & = & \sum_{P \in \mathcal{P}: a \in P} f_P & \forall a \in A \\
& f_P & \geq & 0 & \forall P \in \mathcal{P}
\end{aligned}
$$

Observations:

- minimize a continuous function (social cost) over a compact set (feasible flows)
- optimal flow must exist (extreme value theorem)
- optimal flow can be computed efficiently (convex program)

KKT optimality conditions

$$\begin{aligned} \min \quad & \sum_{a \in A} h_a(f_a) \\ \text{s.t.} \quad & \sum_{P \in \mathcal{P}_i} f_P = r_i && \forall i \in [k] \\ & f_a = \sum_{P \in \mathcal{P}: a \in P} f_P && \forall a \in A \\ & f_P \geq 0 && \forall P \in \mathcal{P} \end{aligned} \tag{CP}$$

Theorem (KKT optimality conditions)

Suppose $(h_a)_{a \in A}$ are continuously differentiable and convex. f is an optimal solution for (CP) if and only if for every $i \in [k]$

$$\forall P_1, P_2 \in \mathcal{P}_i,\ f_{P_1} > 0 : \quad \sum_{a \in P_1} h'_a(f_a) \leq \sum_{a \in P_2} h'_a(f_a)$$

KKT optimality conditions

$$
\begin{aligned}
\min \quad & \sum_{a \in A} h_a(f_a) & & & \\
\text{s.t.} \quad & \sum_{P \in \mathcal{P}_i} f_P & = & r_i & \forall i \in [k] \\
& f_a & = & \sum_{P \in \mathcal{P}: a \in P} f_P & \forall a \in A \\
& f_P & \geq & 0 & \forall P \in \mathcal{P}
\end{aligned}
\tag{CP}
$$

Theorem (KKT optimality conditions)

Suppose $(h_a)_{a \in A}$ are continuously differentiable and convex. f is an optimal solution for (CP) if and only if for every $i \in [k]$

$$
\forall P_1, P_2 \in \mathcal{P}_i,\ f_{P_1} > 0 : \quad \sum_{a \in P_1} h'_a(f_a) \leq \sum_{a \in P_2} h'_a(f_a)
$$

Existence and computation of Nash flows

Idea: Can we determine functions $(h_a)_{a \in A}$ such that KKT optimality conditions reduce to Nash flow conditions?

$$\begin{aligned} \min \quad & \sum_{a \in A} \int_0^{f_a} \ell_a(x)\,dx & & \\ \text{s.t.} \quad & \sum_{P \in \mathcal{P}_i} f_P = r_i & & \forall i \in [k] \\ & f_a = \sum_{P \in \mathcal{P}: a \in P} f_P & & \forall a \in A \\ & f_P \geq 0 & & \forall P \in \mathcal{P} \end{aligned}$$

Observations:

- optimal solutions of (CP) above coincides with Nash flows
- Nash flow must exist (extreme value theorem)
- similar argument: cost of Nash flow is unique
- Nash flow can be computed efficiently (convex program)

Existence and computation of Nash flows

Idea: Can we determine functions $(h_a)_{a \in A}$ such that KKT optimality conditions reduce to Nash flow conditions?

$$\begin{aligned} \min \quad & \sum_{a \in A} \int_0^{f_a} \ell_a(x)\,dx & & & \\ \text{s.t.} \quad & \sum_{P \in \mathcal{P}_i} f_P = r_i & & & \forall i \in [k] \\ & f_a = \sum_{P \in \mathcal{P}: a \in P} f_P & & & \forall a \in A \\ & f_P \geq 0 & & & \forall P \in \mathcal{P} \end{aligned}$$

Observations:

- optimal solutions of (CP) above coincides with Nash flows
- Nash flow must exist (extreme value theorem)
- similar argument: cost of Nash flow is unique
- Nash flow can be computed efficiently (convex program)

Existence and computation of Nash flows

Idea: Can we determine functions $(h_a)_{a \in A}$ such that KKT optimality conditions reduce to Nash flow conditions?

$$\begin{aligned} \min \quad & \sum_{a \in A} \int_0^{f_a} \ell_a(x)\,dx & & & \\ \text{s.t.} \quad & \sum_{P \in \mathcal{P}_i} f_P & = & r_i & \forall i \in [k] \\ & f_a & = & \sum_{P \in \mathcal{P}: a \in P} f_P & \forall a \in A \\ & f_P & \geq & 0 & \forall P \in \mathcal{P} \end{aligned}$$

Observations:

- optimal solutions of (CP) above coincides with Nash flows
- Nash flow must exist (extreme value theorem)
- similar argument: cost of Nash flow is unique
- Nash flow can be computed efficiently (convex program)

Characterization of optimal flows

$$
\begin{aligned}
\min \quad & \sum_{a \in A} \ell_a(f_a) f_a & & & \\
\text{s.t.} \quad & \sum_{P \in \mathcal{P}_i} f_P & = & r_i & \forall i \in [k] \\
& f_a & = & \sum_{P \in \mathcal{P}: a \in P} f_P & \forall a \in A \\
& f_P & \geq & 0 & \forall P \in \mathcal{P}
\end{aligned}
$$

Applying KKT optimality conditions yields:

Theorem

f is an optimal flow with respect to $(\ell_a)_{a \in A}$ if and only if f is a Nash flow with respect to $(\ell_a^*)_{a \in A}$, where $\ell_a^*(x) = (x \cdot \ell_a(x))'$.

Characterization of optimal flows

$$
\begin{aligned}
\min \quad & \sum_{a \in A} \ell_a(f_a) f_a & & & \\
\text{s.t.} \quad & \sum_{P \in \mathcal{P}_i} f_P & = & r_i & \forall i \in [k] \\
& f_a & = & \sum_{P \in \mathcal{P}: a \in P} f_P & \forall a \in A \\
& f_P & \geq & 0 & \forall P \in \mathcal{P}
\end{aligned}
$$

Applying KKT optimality conditions yields:

Theorem

f is an optimal flow with respect to $(\ell_a)_{a \in A}$ if and only if f is a Nash flow with respect to $(\ell_a^*)_{a \in A}$, where $\ell_a^*(x) = (x \cdot \ell_a(x))'$.

Price of anarchy in selfish routing

Price of anarchy: given an instance $I = (G, (s_i, t_i), (r_i), (\ell_a))$

$$POA(I) = \max_{f \text{ Nash flow}} \frac{C(f)}{C(f^*)}$$

POA for selfish routing is independent of the network topology, but depends on the class of latency functions:

- $POA = \frac{4}{3}$ for linear latency functions
- $POA = \Theta(\frac{p}{\ln p})$ for polynomial latency functions of degree p
- POA is unbounded in general

[Roughgarden, Tardos, JACM '02]

Price of anarchy in selfish routing

Price of anarchy: given an instance $I = (G, (s_i, t_i), (r_i), (\ell_a))$

$$POA(I) = \max_{f \text{ Nash flow}} \frac{C(f)}{C(f^*)}$$

POA for selfish routing is independent of the network topology, but depends on the class of latency functions:

- $POA = \frac{4}{3}$ for linear latency functions
- $POA = \Theta(\frac{p}{\ln p})$ for polynomial latency functions of degree p
- POA is unbounded in general

[Roughgarden, Tardos, JACM '02]

[Upper bound on POA for linear latency functions]

POA for polynomial latency functions

p	1	2	3	...
POA	≈ 1.333	≈ 1.626	≈ 1.896	

Table: POA for polynomial latency functions of degree p

Theorem

Let $(\ell_a)_{a \in A}$ be polynomial latency functions of degree at most p.

$$POA \leq \left(1 - \frac{p}{(p+1)^{(p+1)/p}}\right)^{-1} = \Theta\left(\frac{p}{\ln p}\right)$$

[Roughgarden, Tardos, JACM '02]

POA for polynomial latency functions

p	1	2	3	...
POA	≈ 1.333	≈ 1.626	≈ 1.896	

Table: POA for polynomial latency functions of degree p

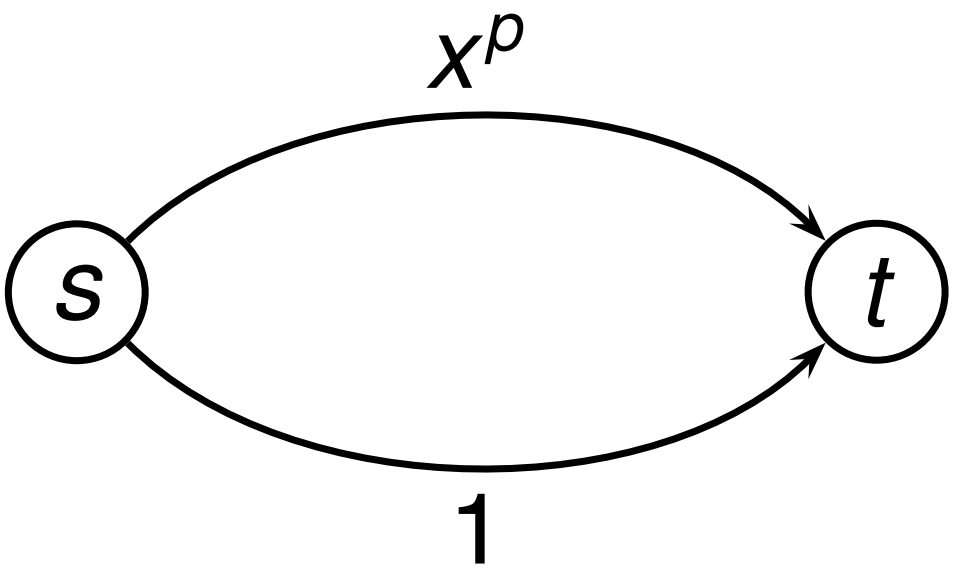
Theorem

Let $(\ell_a)_{a \in A}$ be polynomial latency functions of degree at most p.

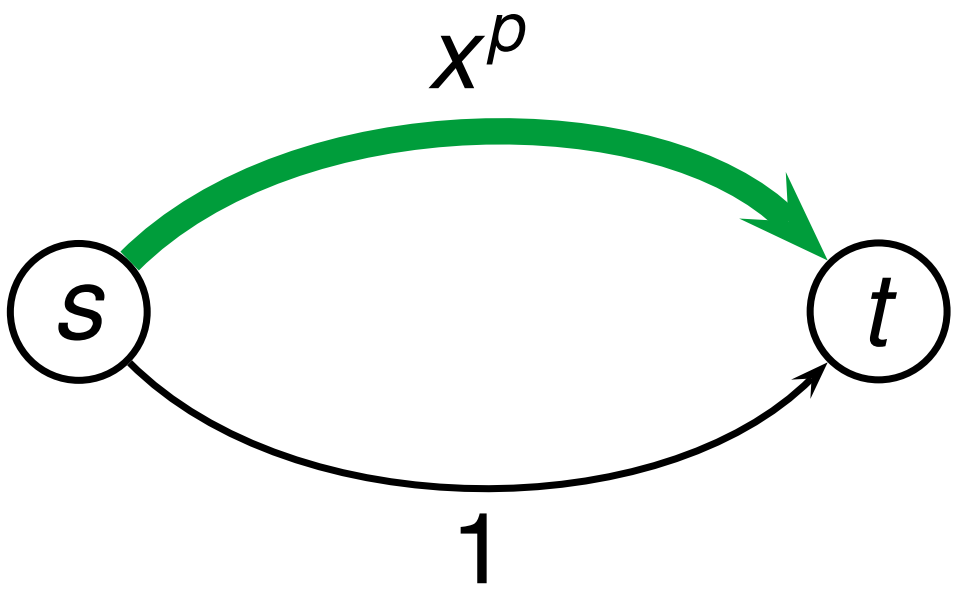
$$POA \leq \left(1 - \frac{p}{(p+1)^{(p+1)/p}}\right)^{-1} = \Theta\left(\frac{p}{\ln p}\right)$$

[Roughgarden, Tardos, JACM '02]

Example: Unbounded POA

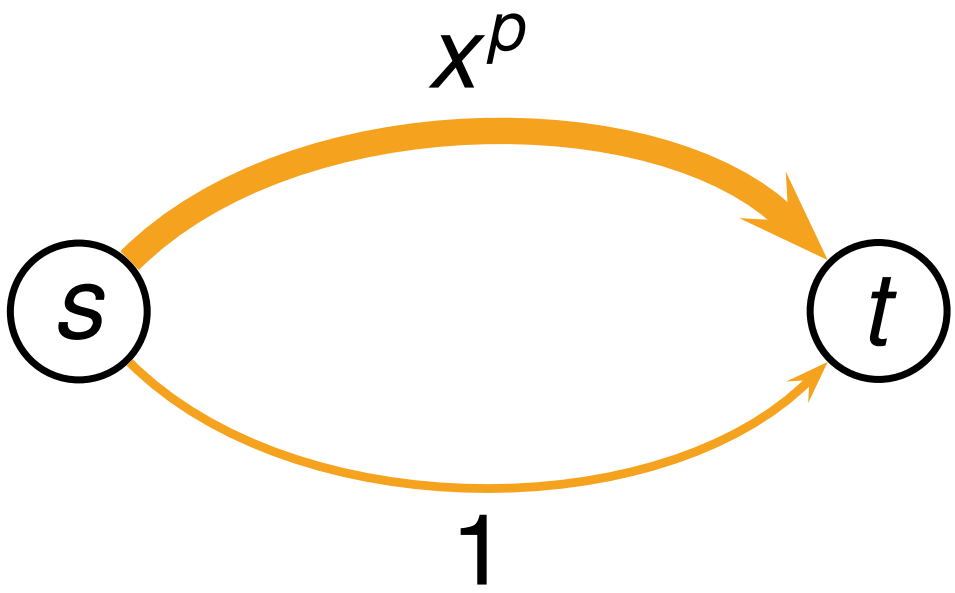


Example: Unbounded POA



Nash flow f: $C(f) = 1$

Example: Unbounded POA



Optimal flow f^*: $C(f^*) = (1-\epsilon)^{1+p} + \epsilon \cdot 1 \to 0$

Example: Unbounded POA

$$POA \to \infty \quad \text{as} \quad p \to \infty$$

Reducing inefficiency

Means to reduce inefficiency:

- infrastructure improvement (expensive, Braess paradox)
- centralized routing (difficult to implement)
- Stackelberg routing (control part of the traffic)
- traffic regulation (taxes on fuel, pay-as-you-go tax)
- network tolls

Coping with the Braess Paradox

Braess subgraph problem

Consider a single-commodity instance $I = (G, (s,t), r, (\ell_a))$ with linear latency functions. Define

$$d(H) = \text{common latency of flow carrying paths in a Nash flow for } H \subseteq G$$

Braess subgraph problem: Given $(G, (s,t), r, (\ell_a))$, find a subgraph $H \subseteq G$ that minimizes $d(H)$.

Algorithm TRIVIAL: simply return the original graph G

Theorem

TRIVIAL is a $\frac{4}{3}$-approximation algorithm for the Braess subgraph problem.

Braess subgraph problem

Consider a single-commodity instance $I = (G, (s,t), r, (\ell_a))$ with linear latency functions. Define

$$d(H) = \text{common latency of flow carrying paths in a Nash flow for } H \subseteq G$$

Braess subgraph problem: Given $(G, (s,t), r, (\ell_a))$, find a subgraph $H \subseteq G$ that minimizes $d(H)$.

Algorithm TRIVIAL: simply return the original graph G

Theorem

TRIVIAL is a $\frac{4}{3}$-approximation algorithm for the Braess subgraph problem.

Braess subgraph problem

Consider a single-commodity instance $I = (G, (s,t), r, (\ell_a))$ with linear latency functions. Define

$$d(H) = \text{common latency of flow carrying paths in a Nash flow for } H \subseteq G$$

Braess subgraph problem: Given $(G, (s,t), r, (\ell_a))$, find a subgraph $H \subseteq G$ that minimizes $d(H)$.

Algorithm TRIVIAL: simply return the original graph G

Theorem

TRIVIAL *is a* $\frac{4}{3}$*-approximation algorithm for the Braess subgraph problem.*

Braess subgraph problem

Consider a single-commodity instance $I = (G, (s,t), r, (\ell_a))$ with linear latency functions. Define

$$d(H) = \text{common latency of flow carrying paths in a Nash flow for } H \subseteq G$$

Braess subgraph problem: Given $(G, (s,t), r, (\ell_a))$, find a subgraph $H \subseteq G$ that minimizes $d(H)$.

Algorithm TRIVIAL: simply return the original graph G

Theorem

TRIVIAL is a $\frac{4}{3}$-approximation algorithm for the Braess subgraph problem.

Braess subgraph problem

Consider a single-commodity instance $I = (G, (s,t), r, (\ell_a))$ with linear latency functions. Define

$$d(H) = \text{common latency of flow carrying paths in a Nash flow for } H \subseteq G$$

Braess subgraph problem: Given $(G, (s,t), r, (\ell_a))$, find a subgraph $H \subseteq G$ that minimizes $d(H)$.

Algorithm TRIVIAL: simply return the original graph G

Theorem

TRIVIAL is a $\frac{4}{3}$-approximation algorithm for the Braess subgraph problem.

Braess subgraph problem

Theorem

TRIVIAL *is a* $\frac{4}{3}$*-approximation algorithm for the Braess subgraph problem.*

Proof:

1. Algorithm computes a feasible solution in polynomial time.
2. Consider an arbitrary subgraph H of G. Let

$$f = \text{Nash flow for } G \quad \text{and} \quad h = \text{Nash flow for } H$$

The bound of $\frac{4}{3}$ on the price of anarchy yields

$$C(f) \leq \frac{4}{3} \cdot C(f^*) \leq \frac{4}{3} \cdot C(h)$$

$$r \cdot d(G) = C(f) \leq \frac{4}{3} \cdot C(f^*) \leq \frac{4}{3} \cdot C(h) = \frac{4}{3} \cdot r \cdot d(H)$$

Braess subgraph problem

Theorem

TRIVIAL *is a* $\frac{4}{3}$*-approximation algorithm for the Braess subgraph problem.*

Proof:

1. Algorithm computes a feasible solution in polynomial time.
2. Consider an arbitrary subgraph H of G. Let

$$f = \text{Nash flow for } G \quad \text{and} \quad h = \text{Nash flow for } H$$

The bound of $\frac{4}{3}$ on the price of anarchy yields

$$C(f) \leq \frac{4}{3} \cdot C(f^*) \leq \frac{4}{3} \cdot C(h)$$

$$r \cdot d(G) = C(f) \leq \frac{4}{3} \cdot C(f^*) \leq \frac{4}{3} \cdot C(h) = \frac{4}{3} \cdot r \cdot d(H)$$

Braess subgraph problem

Theorem

TRIVIAL *is a* $\frac{4}{3}$*-approximation algorithm for the Braess subgraph problem.*

Proof:

1. Algorithm computes a feasible solution in polynomial time.
2. Consider an arbitrary subgraph H of G. Let

$$f = \text{Nash flow for } G \quad \text{and} \quad h = \text{Nash flow for } H$$

The bound of $\frac{4}{3}$ on the price of anarchy yields

$$C(f) \leq \frac{4}{3} \cdot C(f^*) \leq \frac{4}{3} \cdot C(h)$$

$$r \cdot d(G) = C(f) \leq \frac{4}{3} \cdot C(f^*) \leq \frac{4}{3} \cdot C(h) = \frac{4}{3} \cdot r \cdot d(H)$$

Braess subgraph problem

Theorem

TRIVIAL *is a* $\frac{4}{3}$*-approximation algorithm for the Braess subgraph problem.*

Proof:

1. Algorithm computes a feasible solution in polynomial time.
2. Consider an arbitrary subgraph H of G. Let

$$f = \text{Nash flow for } G \quad \text{and} \quad h = \text{Nash flow for } H$$

The bound of $\frac{4}{3}$ on the price of anarchy yields

$$C(f) \leq \frac{4}{3} \cdot C(f^*) \leq \frac{4}{3} \cdot C(h)$$

$$r \cdot d(G) = C(f) \leq \frac{4}{3} \cdot C(f^*) \leq \frac{4}{3} \cdot C(h) = \frac{4}{3} \cdot r \cdot d(H)$$

Braess subgraph problem

Theorem

TRIVIAL *is a* $\frac{4}{3}$*-approximation algorithm for the Braess subgraph problem.*

Proof:

1. Algorithm computes a feasible solution in polynomial time.
2. Consider an arbitrary subgraph H of G. Let

$$f = \text{Nash flow for } G \quad \text{and} \quad h = \text{Nash flow for } H$$

The bound of $\frac{4}{3}$ on the price of anarchy yields

$$C(f) \leq \frac{4}{3} \cdot C(f^*) \leq \frac{4}{3} \cdot C(h)$$

$$r \cdot d(G) = C(f) \leq \frac{4}{3} \cdot C(f^*) \leq \frac{4}{3} \cdot C(h) = \frac{4}{3} \cdot r \cdot d(H)$$

Braess subgraph problem

Theorem

TRIVIAL *is a* $\frac{4}{3}$*-approximation algorithm for the Braess subgraph problem.*

Proof:

1. Algorithm computes a feasible solution in polynomial time.
2. Consider an arbitrary subgraph H of G. Let

$$f = \text{Nash flow for } G \quad \text{and} \quad h = \text{Nash flow for } H$$

The bound of $\frac{4}{3}$ on the price of anarchy yields

$$C(f) \leq \frac{4}{3} \cdot C(f^*) \leq \frac{4}{3} \cdot C(h)$$

$$r \cdot d(G) = C(f) \leq \frac{4}{3} \cdot C(f^*) \leq \frac{4}{3} \cdot C(h) = \frac{4}{3} \cdot r \cdot d(H)$$

Braess subgraph problem

Theorem

TRIVIAL *is a* $\frac{4}{3}$*-approximation algorithm for the Braess subgraph problem.*

Proof:

1. Algorithm computes a feasible solution in polynomial time.
2. Consider an arbitrary subgraph H of G. Let

$$f = \text{Nash flow for } G \quad \text{and} \quad h = \text{Nash flow for } H$$

The bound of $\frac{4}{3}$ on the price of anarchy yields

$$C(f) \leq \frac{4}{3} \cdot C(f^*) \leq \frac{4}{3} \cdot C(h)$$

$$r \cdot d(G) = C(f) \leq \frac{4}{3} \cdot C(f^*) \leq \frac{4}{3} \cdot C(h) = \frac{4}{3} \cdot r \cdot d(H)$$

Braess subgraph problem

Theorem

TRIVIAL *is a* $\frac{4}{3}$*-approximation algorithm for the Braess subgraph problem.*

Proof:

1. Algorithm computes a feasible solution in polynomial time.
2. Consider an arbitrary subgraph H of G. Let

$$f = \text{Nash flow for } G \quad \text{and} \quad h = \text{Nash flow for } H$$

The bound of $\frac{4}{3}$ on the price of anarchy yields

$$C(f) \leq \frac{4}{3} \cdot C(f^*) \leq \frac{4}{3} \cdot C(h)$$

$$r \cdot d(G) = C(f) \leq \frac{4}{3} \cdot C(f^*) \leq \frac{4}{3} \cdot C(h) = \frac{4}{3} \cdot r \cdot d(H)$$

Braess subgraph problem

Theorem

TRIVIAL *is a* $\frac{4}{3}$*-approximation algorithm for the Braess subgraph problem.*

Proof:

1. Algorithm computes a feasible solution in polynomial time.
2. Consider an arbitrary subgraph H of G. Let

$$f = \text{Nash flow for } G \quad \text{and} \quad h = \text{Nash flow for } H$$

The bound of $\frac{4}{3}$ on the price of anarchy yields

$$C(f) \leq \frac{4}{3} \cdot C(f^*) \leq \frac{4}{3} \cdot C(h)$$

$$r \cdot d(G) = C(f) \leq \frac{4}{3} \cdot C(f^*) \leq \frac{4}{3} \cdot C(h) = \frac{4}{3} \cdot r \cdot d(H)$$

Braess subgraph problem

Theorem

TRIVIAL *is a* $\frac{4}{3}$*-approximation algorithm for the Braess subgraph problem.*

Proof:

1. Algorithm computes a feasible solution in polynomial time.
2. Consider an arbitrary subgraph H of G. Let

$$f = \text{Nash flow for } G \quad \text{and} \quad h = \text{Nash flow for } H$$

The bound of $\frac{4}{3}$ on the price of anarchy yields

$$C(f) \leq \frac{4}{3} \cdot C(f^*) \leq \frac{4}{3} \cdot C(h)$$

$$r \cdot d(G) = C(f) \leq \frac{4}{3} \cdot C(f^*) \leq \frac{4}{3} \cdot C(h) = \frac{4}{3} \cdot r \cdot d(H)$$

□

As good as it gets...

Theorem

Assuming $P \neq NP$, for every $\varepsilon > 0$ there is no approximation algorithm for the Braess subgraph problem achieving an approximation guarantee of $(\frac{4}{3} - \varepsilon)$.

Proof idea: A $(\frac{4}{3} - \varepsilon)$-approximation algorithm can be used to determine whether there exist two disjoint paths between (s_1, t_1) and (s_2, t_2) in a directed graph. The latter problem is NP-complete.

As good as it gets...

Theorem

Assuming $P \neq NP$, for every $\varepsilon > 0$ there is no approximation algorithm for the Braess subgraph problem achieving an approximation guarantee of $(\frac{4}{3} - \varepsilon)$.

Proof idea: A $(\frac{4}{3} - \varepsilon)$-approximation algorithm can be used to determine whether there exist two disjoint paths between (s_1, t_1) and (s_2, t_2) in a directed graph. The latter problem is NP-complete.

As good as it gets...

Theorem

Assuming $P \neq NP$, for every $\varepsilon > 0$ there is no approximation algorithm for the Braess subgraph problem achieving an approximation guarantee of $(\frac{4}{3} - \varepsilon)$.

Proof idea: A $(\frac{4}{3} - \varepsilon)$-approximation algorithm can be used to determine whether there exist two disjoint paths between (s_1, t_1) and (s_2, t_2) in a directed graph. The latter problem is NP-complete.

Stackelberg Routing

Stackelberg routing

Setting:

- central authority (Stackelberg leader):
  - controls an $\alpha \in [0, 1]$ fraction of the entire demand
  - routes demand according to a predefined policy, called Stackelberg strategy
- selfish followers: once the Stackelberg leader has fixed his flow, remaining $(1 - \alpha)$ fraction of the demand is routed selfishly

Viewpoint:

- network provider controls flow through his sub-network
- agents are guided through a congested network by a central authority

Stackelberg routing

Setting:

- central authority (Stackelberg leader):
  - controls an $\alpha \in [0, 1]$ fraction of the entire demand
  - routes demand according to a predefined policy, called Stackelberg strategy
- selfish followers: once the Stackelberg leader has fixed his flow, remaining $(1 - \alpha)$ fraction of the demand is routed selfishly

Viewpoint:

- network provider controls flow through his sub-network
- agents are guided through a congested network by a central authority

Stackelberg routing

Setting:

- central authority (Stackelberg leader):
  - controls an $\alpha \in [0, 1]$ fraction of the entire demand
  - routes demand according to a predefined policy, called Stackelberg strategy
- selfish followers: once the Stackelberg leader has fixed his flow, remaining $(1 - \alpha)$ fraction of the demand is routed selfishly

Viewpoint:

- network provider controls flow through his sub-network
- agents are guided through a congested network by a central authority

Yet another viewpoint

Network routing games:

- **self-interest hypothesis:** all players are entirely selfish
- experiments in economics show that this viewpoint is too simplistic in many scenarios: player's behavior may vary

$$\text{altruistic} \quad \leftrightarrow \quad \text{selfish} \quad \leftrightarrow \quad \text{spiteful}$$

- recent studies in network routing attempt to capture this phenomenon [Chen, Kempe, EC '08]

Stackelberg routing:

- abandon the above assumption (at least partially)
- fraction of the players act selfishly, others behave arbitrarily
- aggregated behavior of non-selfish players can be seen as a Stackelberg strategy

Model

Stackelberg leader: fixes some flow g (Stackelberg strategy)

- weak: $\sum_{P \in \mathcal{P}_i} g_P = \alpha r_i$ for all i
- strong: $\sum_{P \in \mathcal{P}_i} g_P = \alpha_i r_i$ for all i and $\sum_i \alpha_i r_i = \alpha \sum_i r_i$

Selfish followers:

- route flow selfishly with respect to modified latency functions $\tilde{\ell}_a(x) := \ell_a(g_a + x)$ for every arc $a \in A$
- let h be the resulting Nash flow for $\tilde{\ell}$

Goal: compute Stackelberg strategy g that achieves good price of anarchy

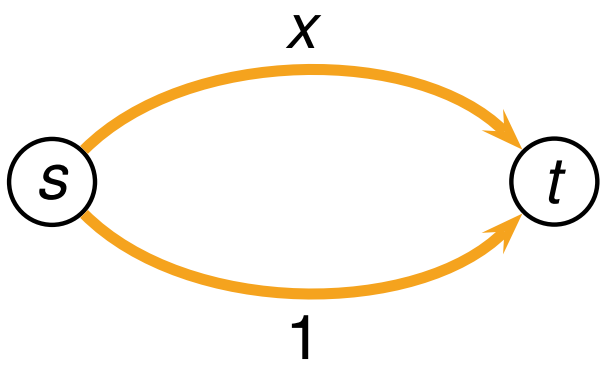
$$\frac{C(g+h)}{C(f^*)}$$

[Roughgarden, SICOMP '04]

Example: Stackelberg routing helps

Stackelberg strategy: put $\min\{\alpha, \frac{1}{2}\}$ units of flow on lower arc; rest on upper arc

- $\alpha \geq \frac{1}{2}$: enforce optimal flow $\Rightarrow POA = 1$
- $\alpha < \frac{1}{2}$: combined flow has cost $C(g+h) = (1-\alpha)^2 + \alpha$ $\Rightarrow POA = \frac{4}{3}(1-\alpha+\alpha^2)$



optimal flow f^* sends

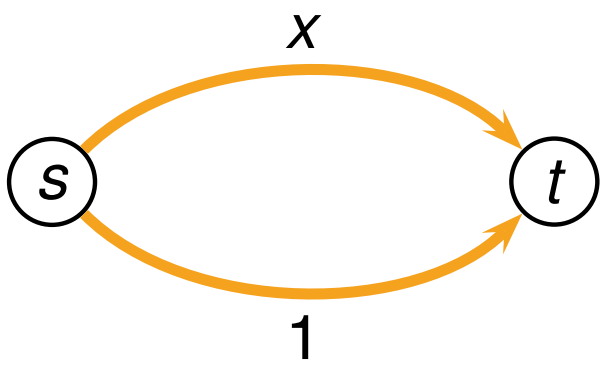
- $\frac{1}{2}$ unit of flow on upper arc
- $\frac{1}{2}$ unit of flow on lower arc

$C(f^*) = \frac{3}{4}$

Example: Stackelberg routing helps

Stackelberg strategy: put $\min\{\alpha, \frac{1}{2}\}$ units of flow on lower arc; rest on upper arc

- $\alpha \geq \frac{1}{2}$: enforce optimal flow
  $\Rightarrow POA = 1$
- $\alpha < \frac{1}{2}$: combined flow has cost
  $C(g+h) = (1-\alpha)^2 + \alpha$
  $\Rightarrow POA = \frac{4}{3}(1-\alpha+\alpha^2)$



optimal flow f^* sends

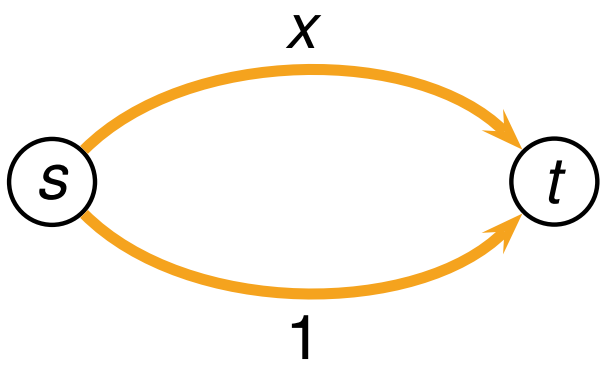
- $\frac{1}{2}$ unit of flow on upper arc
- $\frac{1}{2}$ unit of flow on lower arc

$C(f^*) = \frac{3}{4}$

Example: Stackelberg routing helps

Stackelberg strategy: put $\min\{\alpha, \frac{1}{2}\}$ units of flow on lower arc; rest on upper arc

- $\alpha \geq \frac{1}{2}$: enforce optimal flow
  $\Rightarrow POA = 1$
- $\alpha < \frac{1}{2}$: combined flow has cost
  $C(g+h) = (1-\alpha)^2 + \alpha$
  $\Rightarrow POA = \frac{4}{3}(1-\alpha+\alpha^2)$



optimal flow f^* sends

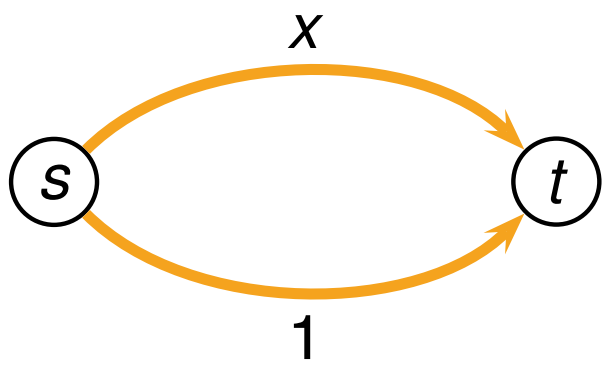
- $\frac{1}{2}$ unit of flow on upper arc
- $\frac{1}{2}$ unit of flow on lower arc

$C(f^*) = \frac{3}{4}$

Example: Stackelberg routing helps

Stackelberg strategy: put $\min\{\alpha, \frac{1}{2}\}$ units of flow on lower arc; rest on upper arc

- $\alpha \geq \frac{1}{2}$: enforce optimal flow
  $\Rightarrow POA = 1$
- $\alpha < \frac{1}{2}$: combined flow has cost
  $C(g+h) = (1-\alpha)^2 + \alpha$
  $\Rightarrow POA = \frac{4}{3}(1 - \alpha + \alpha^2)$

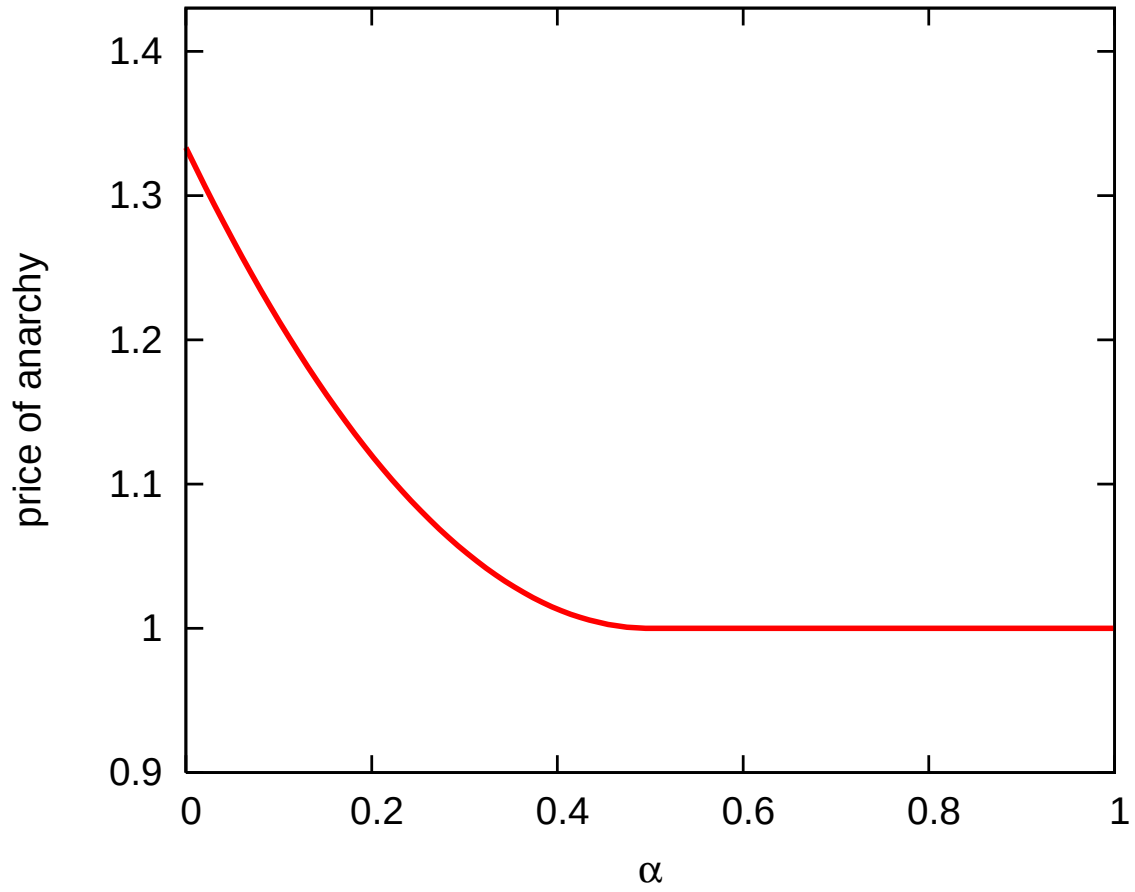


optimal flow f^* sends

- $\frac{1}{2}$ unit of flow on upper arc
- $\frac{1}{2}$ unit of flow on lower arc

$C(f^*) = \frac{3}{4}$

Example: Stackelberg routing helps



Stackelberg strategies

Computing the best Stackelberg strategy is NP-hard

[Roughgarden, SICOMP '04]

Largest-Latency-First (LLF):

1. compute an optimal flow f^* for the entire demand r
2. saturate paths used by f^* in decreasing order of latencies until a total flow of αr units has been routed

SCALE:

1. compute an optimal flow f^* for the entire demand r
2. define $g := \alpha f^*$

Impact of Stackelberg routing

Theorem

LLF reduces POA in parallel-arc networks to $\frac{1}{\alpha}$, independently of latency functions.

[Roughgarden, SICOMP '04]

Theorem

LLF reduces POA in series-parallel networks to $1 + \frac{1}{\alpha}$, independently of latency functions.

[Swamy, SODA '07]

Consequence: POA becomes constant if Stackelberg leader controls a constant fraction of the entire demand.

Main open question: Can we always find a Stackelberg strategy such that the POA is bounded as a function of α?

Impact of Stackelberg routing

Theorem

LLF reduces POA in parallel-arc networks to $\frac{1}{\alpha}$*, independently of latency functions.*

[Roughgarden, SICOMP '04]

Theorem

LLF reduces POA in series-parallel networks to $1 + \frac{1}{\alpha}$*, independently of latency functions.*

[Swamy, SODA '07]

Consequence: POA becomes constant if Stackelberg leader controls a constant fraction of the entire demand.

Main open question: Can we always find a Stackelberg strategy such that the POA is bounded as a function of α?

Impact of Stackelberg routing

Theorem

LLF reduces POA in parallel-arc networks to $\frac{1}{\alpha}$*, independently of latency functions.*

[Roughgarden, SICOMP '04]

Theorem

LLF reduces POA in series-parallel networks to $1 + \frac{1}{\alpha}$*, independently of latency functions.*

[Swamy, SODA '07]

Consequence: POA becomes constant if Stackelberg leader controls a constant fraction of the entire demand.

Main open question: Can we always find a Stackelberg strategy such that the POA is bounded as a function of α?

Impact of Stackelberg routing

Theorem

LLF reduces POA in parallel-arc networks to $\frac{1}{\alpha}$*, independently of latency functions.*

[Roughgarden, SICOMP '04]

Theorem

LLF reduces POA in series-parallel networks to $1 + \frac{1}{\alpha}$*, independently of latency functions.*

[Swamy, SODA '07]

Consequence: POA becomes constant if Stackelberg leader controls a constant fraction of the entire demand.

Main open question: Can we always find a Stackelberg strategy such that the POA is bounded as a function of α?

Limits of Stackelberg routing

Theorem

POA is unbounded for weak Stackelberg strategies in multi-commodity networks.

[Roughgarden, SICOMP '04]

→ does not rule out the existence of a strong Stackelberg strategy with a bounded POA

Theorem

The POA achievable by an arbitrary Stackelberg strategy is unbounded even for single-commodity networks.

[Bonifaci, Harks, Schäfer, MOR '10]

Limits of Stackelberg routing

Theorem

POA is unbounded for weak Stackelberg strategies in multi-commodity networks.

[Roughgarden, SICOMP '04]

$\rightarrow$ does not rule out the existence of a strong Stackelberg strategy with a bounded POA

Theorem

The POA achievable by an arbitrary Stackelberg strategy is unbounded even for single-commodity networks.

[Bonifaci, Harks, Schäfer, MOR '10]

Limits of Stackelberg routing

Theorem

POA is unbounded for weak Stackelberg strategies in multi-commodity networks.

[Roughgarden, SICOMP '04]

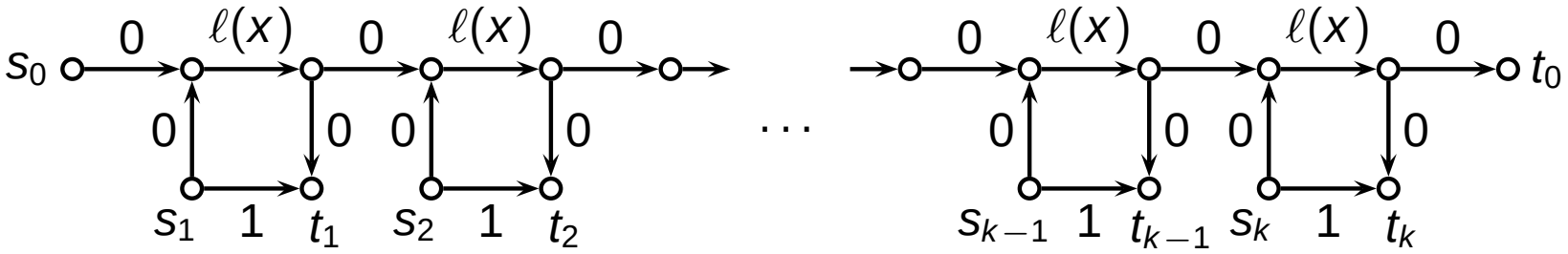
$\rightarrow$ does not rule out the existence of a strong Stackelberg strategy with a bounded POA

Theorem

The POA achievable by an arbitrary Stackelberg strategy is unbounded even for single-commodity networks.

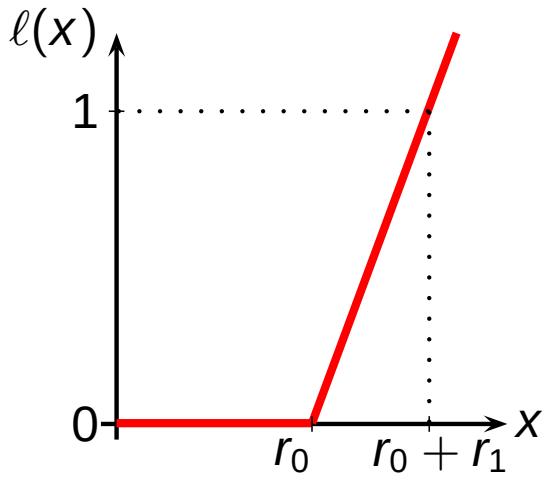
[Bonifaci, Harks, Schäfer, MOR '10]

Proof for multi-commodity networks

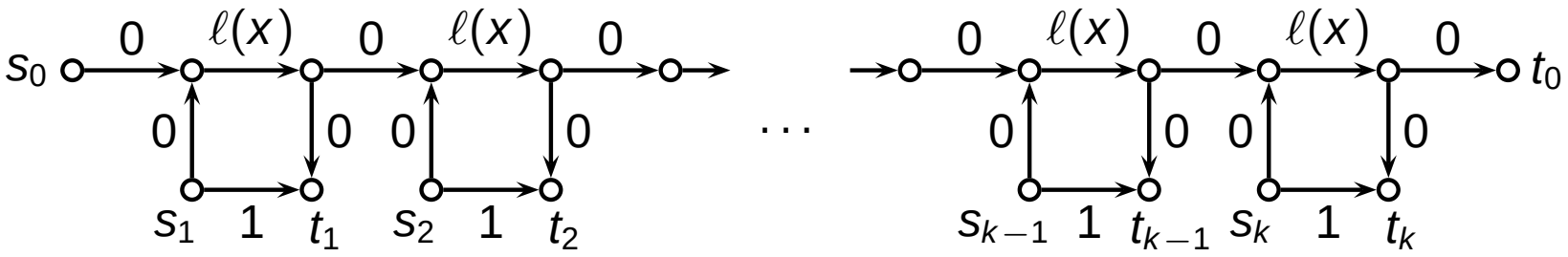


Instance:

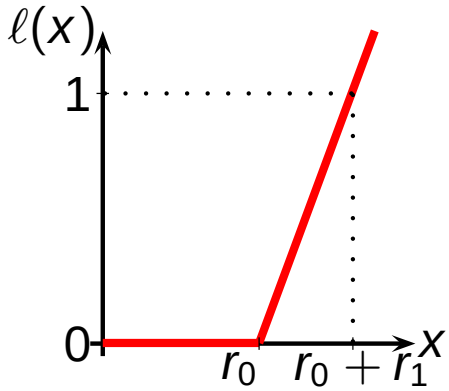
- $k+1$ commodities
- demand $r_0 := \frac{1-\alpha}{2}$ for (s_0, t_0)
- demand $r_1 := \frac{1+\alpha}{2k}$ for (s_i, t_i), $i \in [k]$
- latency function $\ell(x)$
  - $\ell(r_0) := 0$
  - $\ell(x) \geq \frac{x-r_0}{r_1}$ for all x



Proof for multi-commodity networks



Cost of optimal flow: $C(f^*) \leq 1$



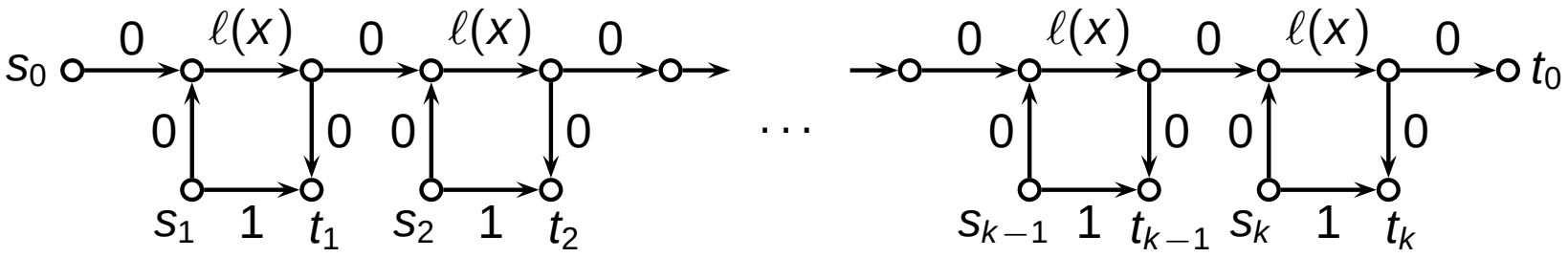
Stackelberg strategy: fix an arbitrary Stackelberg strategy and let g_i be the flow on arc (s_i, t_i), $i \in [k]$

Selfish followers: remaining flow of commodity $i \in [k]$ is sent along the upper path

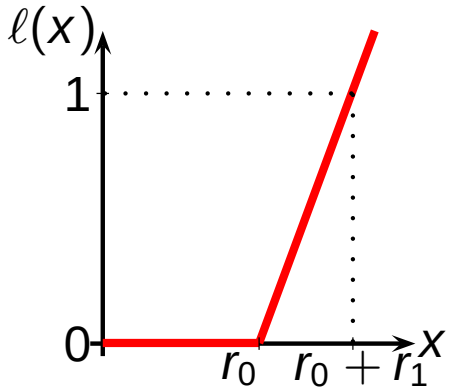
$$\ell_{P_0}(g+h) \geq \sum_{i \in [k]} \frac{r_1 - g_i}{r_1} \geq k - \frac{\alpha}{r_1} = \frac{1-\alpha}{1+\alpha} k$$

$$C(g+h) \geq r_0 \ell_{P_0}(g+h) \geq \frac{(1-\alpha)^2}{2(1+\alpha)} k = \Omega(k) \qquad \square$$

Proof for multi-commodity networks



Cost of optimal flow: $C(f^*) \leq 1$



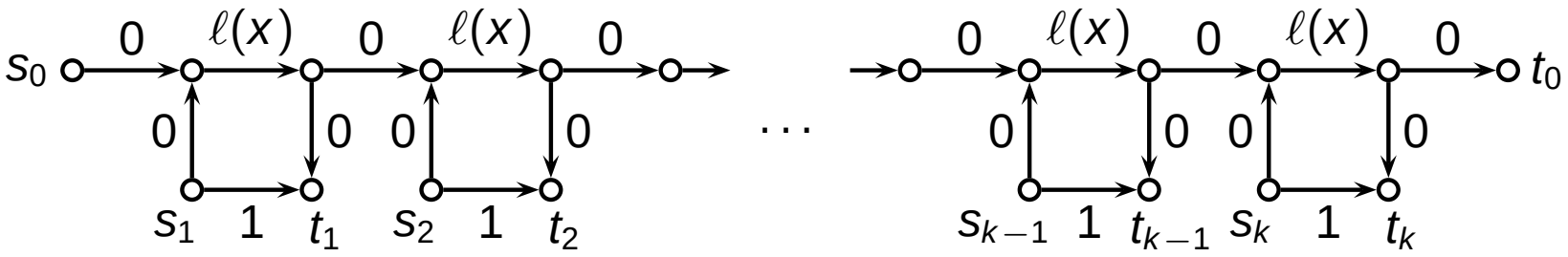
Stackelberg strategy: fix an arbitrary Stackelberg strategy and let g_i be the flow on arc (s_i, t_i), $i \in [k]$

Selfish followers: remaining flow of commodity $i \in [k]$ is sent along the upper path

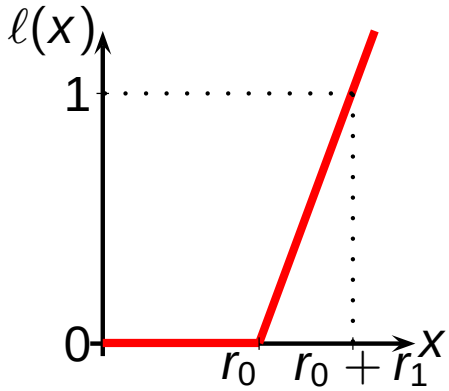
$$\ell_{P_0}(g+h) \geq \sum_{i \in [k]} \frac{r_1 - g_i}{r_1} \geq k - \frac{\alpha}{r_1} = \frac{1-\alpha}{1+\alpha} k$$

$$C(g+h) \geq r_0 \ell_{P_0}(g+h) \geq \frac{(1-\alpha)^2}{2(1+\alpha)} k = \Omega(k) \qquad \square$$

Proof for multi-commodity networks



Cost of optimal flow: $C(f^*) \leq 1$



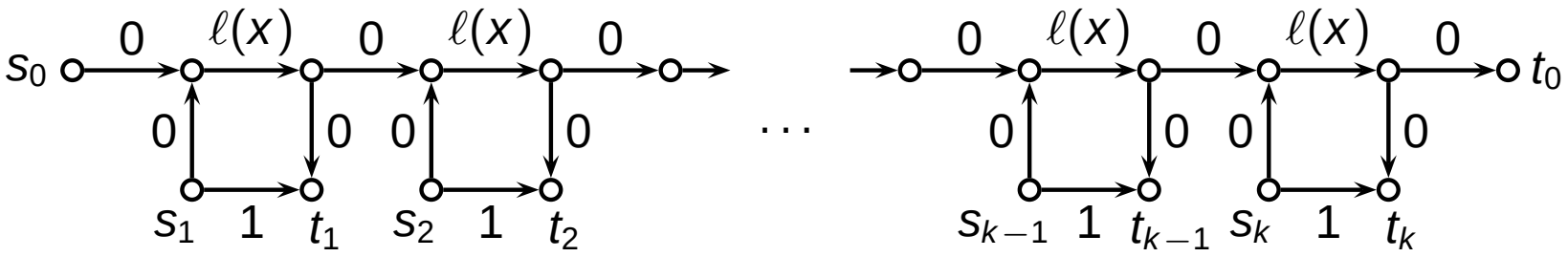
Stackelberg strategy: fix an arbitrary Stackelberg strategy and let g_i be the flow on arc (s_i, t_i), $i \in [k]$

Selfish followers: remaining flow of commodity $i \in [k]$ is sent along the upper path

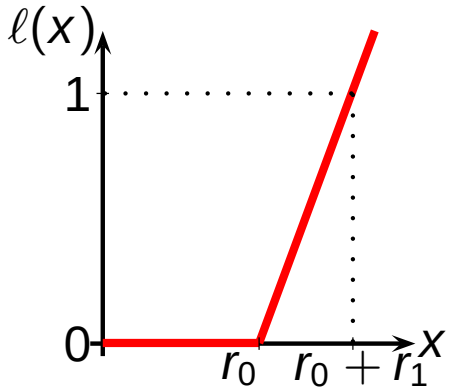
$$\ell_{P_0}(g+h) \geq \sum_{i \in [k]} \frac{r_1 - g_i}{r_1} \geq k - \frac{\alpha}{r_1} = \frac{1-\alpha}{1+\alpha} k$$

$$C(g+h) \geq r_0 \ell_{P_0}(g+h) \geq \frac{(1-\alpha)^2}{2(1+\alpha)} k = \Omega(k) \qquad \square$$

Proof for multi-commodity networks



Cost of optimal flow: $C(f^*) \leq 1$



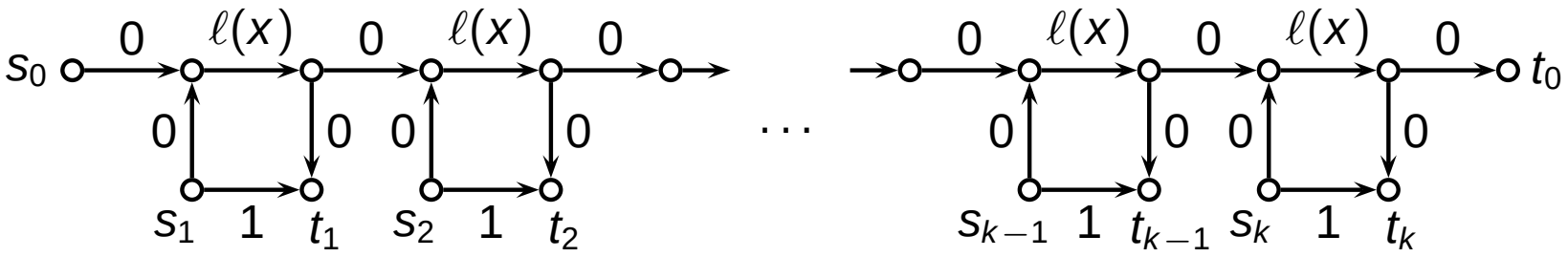
Stackelberg strategy: fix an arbitrary Stackelberg strategy and let g_i be the flow on arc (s_i, t_i), $i \in [k]$

Selfish followers: remaining flow of commodity $i \in [k]$ is sent along the upper path

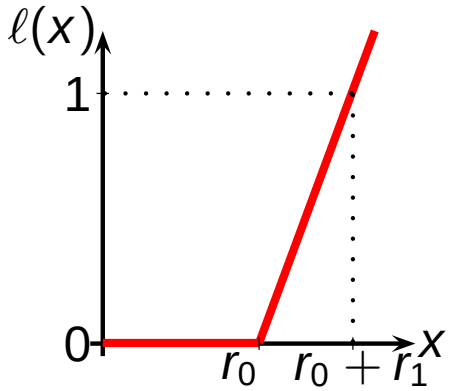
$$\ell_{P_0}(g+h) \geq \sum_{i \in [k]} \frac{r_1 - g_i}{r_1} \geq k - \frac{\alpha}{r_1} = \frac{1-\alpha}{1+\alpha} k$$

$$C(g+h) \geq r_0 \ell_{P_0}(g+h) \geq \frac{(1-\alpha)^2}{2(1+\alpha)} k = \Omega(k)$$

Proof for multi-commodity networks



Cost of optimal flow: $C(f^*) \leq 1$



Stackelberg strategy: fix an arbitrary Stackelberg strategy and let g_i be the flow on arc (s_i, t_i), $i \in [k]$

Selfish followers: remaining flow of commodity $i \in [k]$ is sent along the upper path

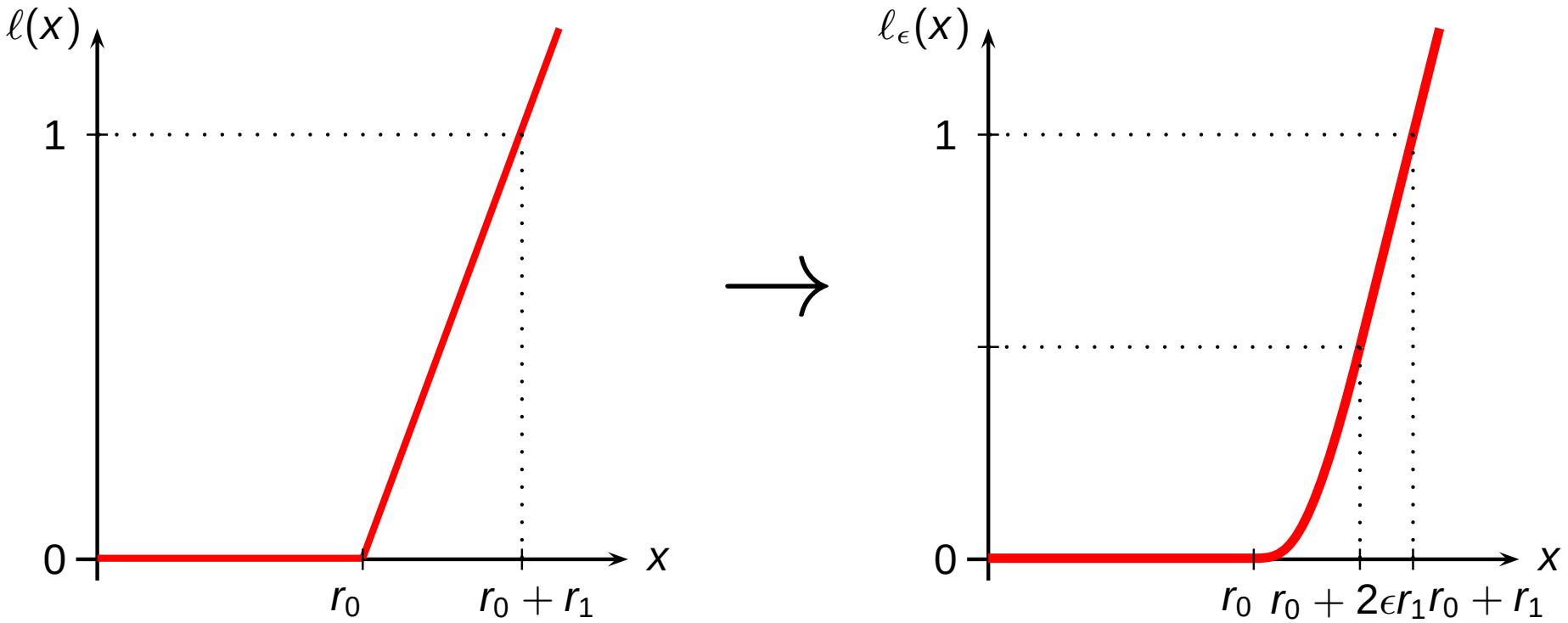
$$\ell_{P_0}(g+h) \geq \sum_{i \in [k]} \frac{r_1 - g_i}{r_1} \geq k - \frac{\alpha}{r_1} = \frac{1-\alpha}{1+\alpha} k$$

$$C(g+h) \geq r_0 \ell_{P_0}(g+h) \geq \frac{(1-\alpha)^2}{2(1+\alpha)} k = \Omega(k) \qquad \square$$

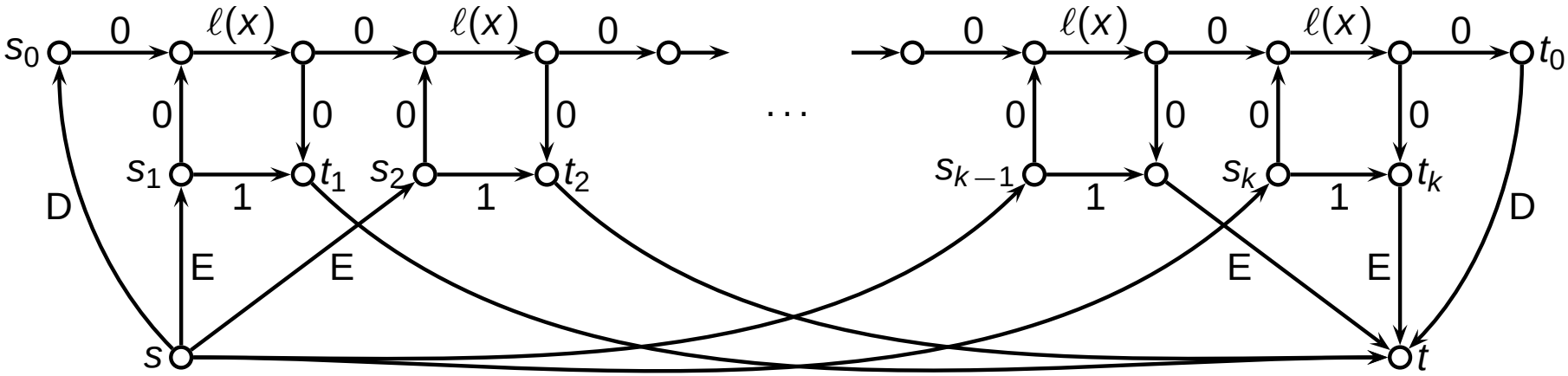
Standard latency functions

Problem: latency function $\ell(\cdot)$ is not continuous

Solution: can interpolate



Extension to single-commodity networks



Basic idea:

- add super-source and super-sink to multi-commodity instance and connect to sources and sinks, respectively
- use latency functions of type *D* and *E* to mimic arc capacities
- can essentially enforce a similar effect as in the multi-commodity instance

Recap

Proof shows:

- for a given $\alpha \in (0, 1)$: can construct an instance such that the POA of an arbitrary (strong) Stackelberg strategy grows with the size of the network
- implies that the POA is unbounded if we let the size of the network grow arbitrarily
- can also show that this is unavoidable: no constant size network can prove an unbounded POA

Bicriteria Bounds

Theorem

There is an efficiently computable Stackelberg strategy g such that $C(g+h) \leq C(f^*)$, where f^* is an optimal flow with respect to demands scaled by $1+\sqrt{1-\alpha}$.

[Bonifaci, Harks, Schäfer, MOR '10]

Consider instance with scaled latency functions

$$\hat{\ell}_a(x) := \frac{1}{\beta} \cdot \ell_a\left(\frac{x}{\beta}\right), \quad \text{where} \quad \beta := 1+\sqrt{1-\alpha}$$

Theorem

If SCALE is used as the Stackelberg strategy, then the cost of the resulting flow satisfies $\hat{C}(g+h) \leq C(f^*)$.

[Bonifaci, Harks, Schäfer, MOR '10]

Bicriteria Bounds

Theorem

There is an efficiently computable Stackelberg strategy g such that $C(g+h) \leq C(f^*)$, where f^* is an optimal flow with respect to demands scaled by $1+\sqrt{1-\alpha}$.

[Bonifaci, Harks, Schäfer, MOR '10]

Consider instance with scaled latency functions

$$\hat{\ell}_a(x) := \frac{1}{\beta} \cdot \ell_a\left(\frac{x}{\beta}\right), \quad \text{where} \quad \beta := 1+\sqrt{1-\alpha}$$

Theorem

If SCALE is used as the Stackelberg strategy, then the cost of the resulting flow satisfies $\hat{C}(g+h) \leq C(f^*)$.

[Bonifaci, Harks, Schäfer, MOR '10]

Bicriteria Bounds

Theorem

There is an efficiently computable Stackelberg strategy g such that $C(g+h) \leq C(f^*)$, where f^* is an optimal flow with respect to demands scaled by $1+\sqrt{1-\alpha}$.

[Bonifaci, Harks, Schäfer, MOR '10]

Consider instance with scaled latency functions

$$\hat{\ell}_a(x) := \frac{1}{\beta} \cdot \ell_a\left(\frac{x}{\beta}\right), \quad \text{where} \quad \beta := 1+\sqrt{1-\alpha}$$

Theorem

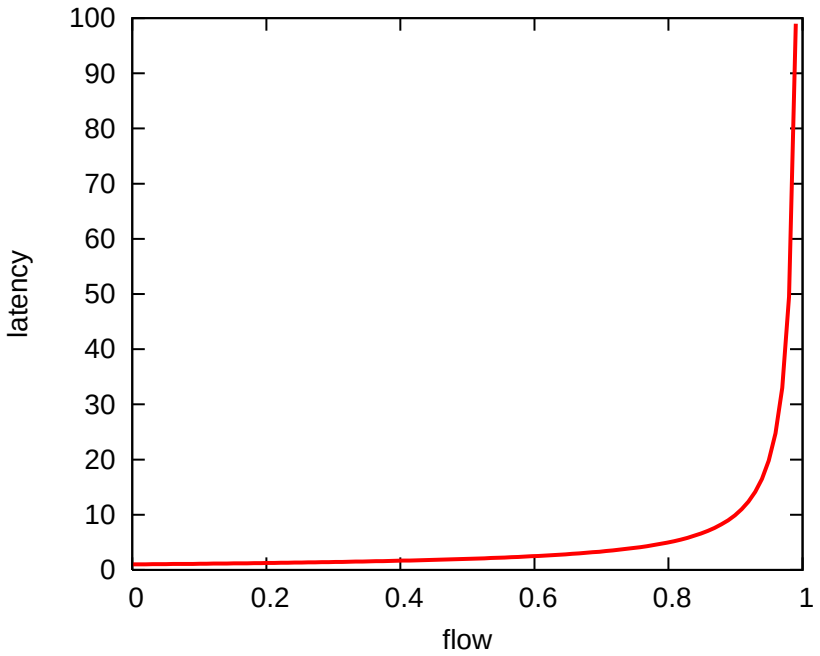
If SCALE is used as the Stackelberg strategy, then the cost of the resulting flow satisfies $\hat{C}(g+h) \leq C(f^*)$.

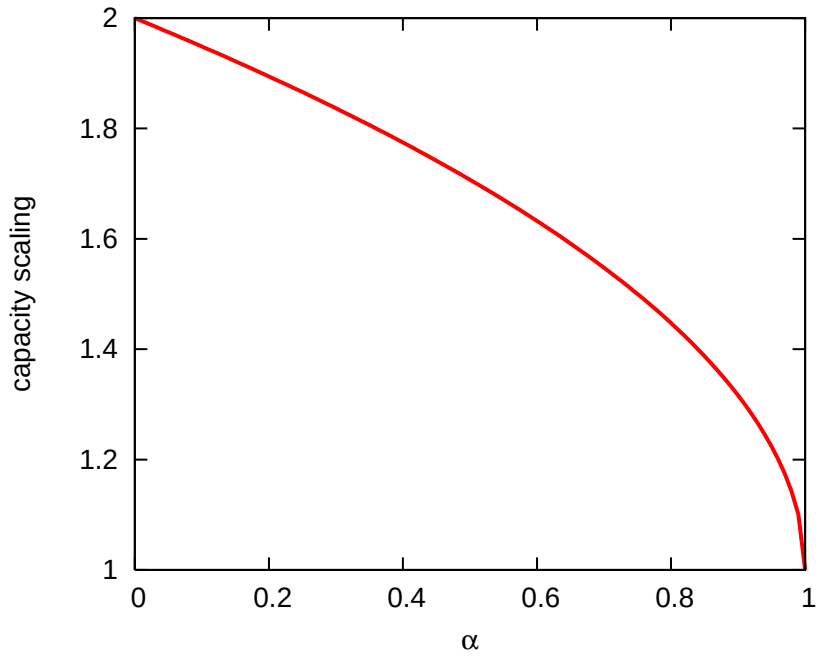
[Bonifaci, Harks, Schäfer, MOR '10]

Implications

Consequence for M/M/1 latency functions: simulating arc of capacity u_a

$$\ell_a(x) = \frac{1}{u_a - x}$$





Bounds for specific classes of latency functions

Let $\mathcal{L}_d$ be the class of latency functions satisfying

$$\ell(cz) \geq c^d \ell(z) \quad \forall c \in [0,1]$$

Example: $\mathcal{L}_d$ contains all polynomials with non-negative coefficients of degree d

Theorem

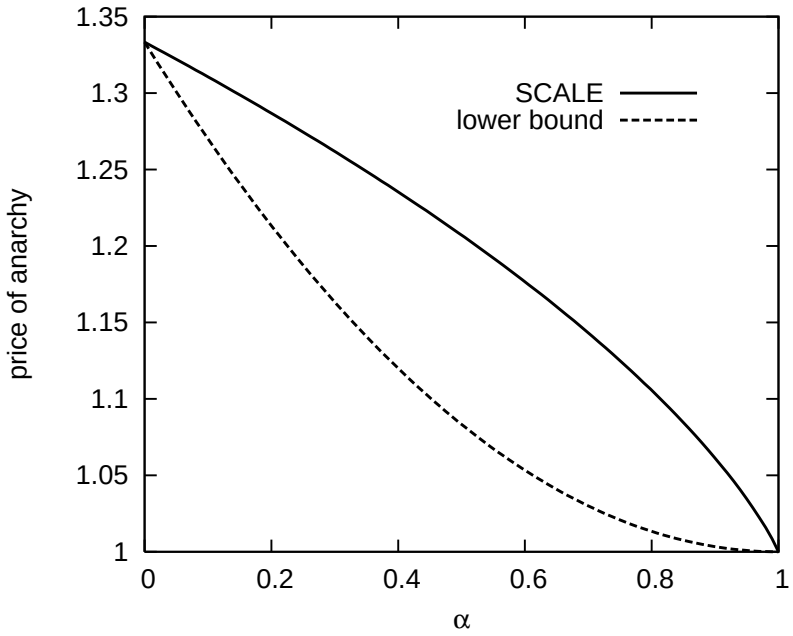
The POA of the SCALE strategy for latency functions in $\mathcal{L}_1$ is at most

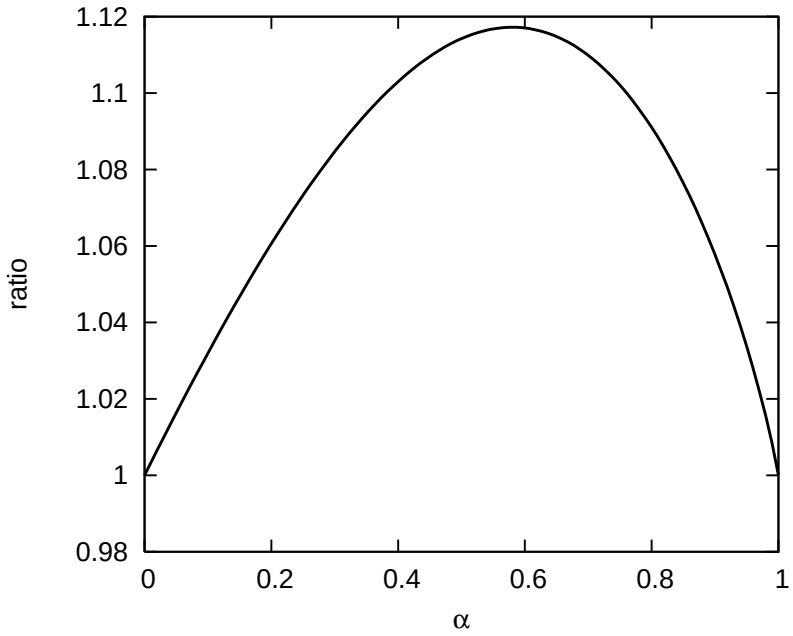
$$\frac{(1+\sqrt{1-\alpha})^2}{2(1+\sqrt{1-\alpha})-1}$$

and this bound is tight

[Bonifaci, Harks, Schäfer, MOR '10]

Implications





- can also prove a lower bound on the POA of arbitrary Stackelberg strategies
- SCALE is a 1.12-approximation for latency functions in $\mathcal{L}_1$ with respect to the best Stackelberg strategy (which is NP-hard to compute)

Bounds for specific classes of latency functions

Theorem

The POA of the SCALE strategy for latency functions in $\mathcal{L}_d$ is at most

$$\frac{(d+1)z_d - \alpha d}{(d+1)z_d - d},$$

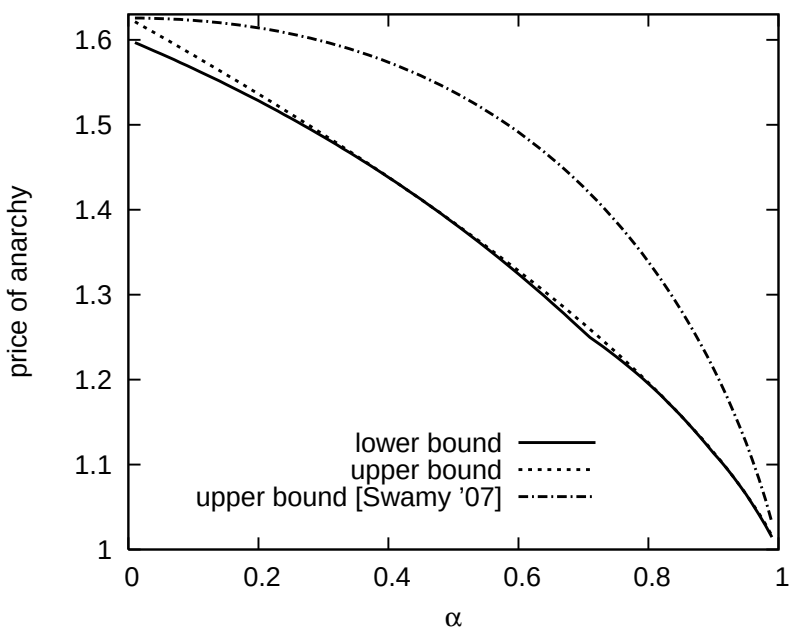
where $z_d \geq 1$ is the unique solution of the equation

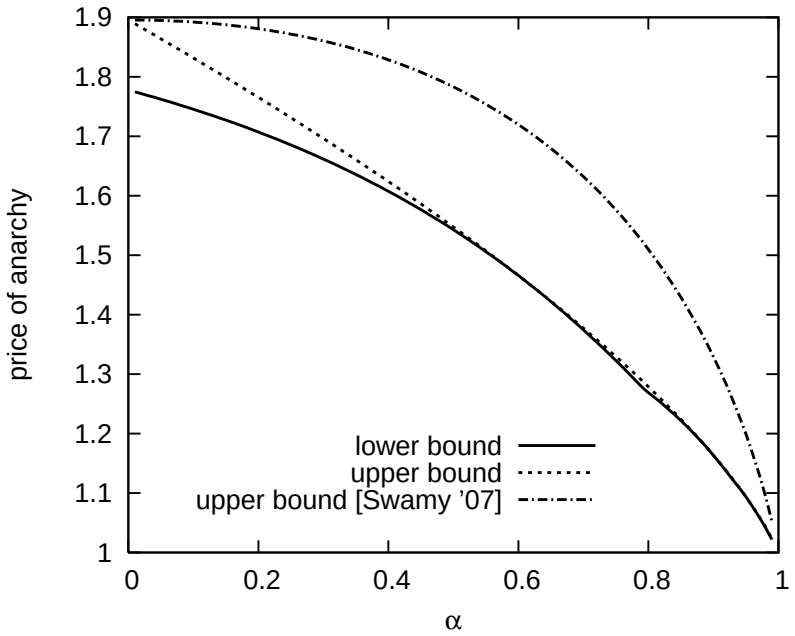
$$z^{d+1} - (d+1)z + \alpha d = 0$$

and this bound is almost tight.

[Bonifaci, Harks, Schäfer, MOR '10]

Implications





Network Tolls

Reducing inefficiency

Means to reduce inefficiency:

- infrastructure improvement (expensive, Braess paradox)
- centralized routing (difficult to implement)
- Stackelberg routing (control part of the traffic)
- traffic regulation (taxes on fuel, pay-as-you-go tax)
- network tolls

Advantages of network tolls:

- fine-grained means: impose "local" taxes
- new technologies facilitate electronic collection of tolls
- dynamic pricing schemes have potential to change participants' behavior
- . . .

Reducing inefficiency

Means to reduce inefficiency:

- infrastructure improvement (expensive, Braess paradox)
- centralized routing (difficult to implement)
- Stackelberg routing (control part of the traffic)
- traffic regulation (taxes on fuel, pay-as-you-go tax)
- network tolls

Advantages of network tolls:

- fine-grained means: impose “local” taxes
- new technologies facilitate electronic collection of tolls
- dynamic pricing schemes have potential to change participants’ behavior
- . . .

Dynamic pricing: ERP in Singapore

Road pricing

Network tolls: impose non-negative tolls $\tau := (\tau_a)_{a \in A}$ on arcs

- dynamic: flow-dependent function $\tau_a(x)$ (non-decreasing, continuous)
- static: flow-independent constant τ_a

Interpretation: by traversing an arc $a \in A$ with flow value x, a player experiences a delay of $\ell_a(x)$ and has to pay a toll of $\tau_a(x)$

Assumption: players are homogeneous

$$\phi_a(x) := \ell_a(x) + \alpha \cdot \tau_a(x)$$

α specifies how players value time over money (w.l.o.g. $\alpha = 1$)

Road pricing

Network tolls: impose non-negative tolls $\tau := (\tau_a)_{a \in A}$ on arcs

- dynamic: flow-dependent function $\tau_a(x)$ (non-decreasing, continuous)
- static: flow-independent constant τ_a

Interpretation: by traversing an arc $a \in A$ with flow value x, a player experiences a delay of $\ell_a(x)$ and has to pay a toll of $\tau_a(x)$

Assumption: players are homogeneous

$$\phi_a(x) := \ell_a(x) + \alpha \cdot \tau_a(x)$$

α specifies how players value time over money (w.l.o.g. $\alpha = 1$)

Road pricing

Network tolls: impose non-negative tolls $\tau := (\tau_a)_{a \in A}$ on arcs

- dynamic: flow-dependent function $\tau_a(x)$ (non-decreasing, continuous)
- static: flow-independent constant τ_a

Interpretation: by traversing an arc $a \in A$ with flow value x, a player experiences a delay of $\ell_a(x)$ and has to pay a toll of $\tau_a(x)$

Assumption: players are homogeneous

$$\phi_a(x) := \ell_a(x) + \alpha \cdot \tau_a(x)$$

α specifies how players value time over money (w.l.o.g. $\alpha = 1$)

Marginal cost tolls

Marginal cost tolls: $\tau_a(x) := x \cdot \ell'_a(x) \quad \forall a \in A$

Theorem

f is an optimal flow with respect to ℓ iff f is a Nash flow with respect to $\ell + \tau$.

[Beckman, McGuire and Winsten 1956]

Drawbacks:

1. potentially every arc of the network is tolled
2. imposed tolls can be arbitrarily large

Most previous studies: no restrictions on tolls that can be imposed on the arcs of the network

Question: What can be done if such restrictions are given exogenously?

Marginal cost tolls

Marginal cost tolls: $\tau_a(x) := x \cdot \ell_a'(x) \quad \forall a \in A$

Theorem

f is an optimal flow with respect to ℓ iff f is a Nash flow with respect to $\ell + \tau$.

[Beckman, McGuire and Winsten 1956]

Drawbacks:

1. potentially every arc of the network is tolled
2. imposed tolls can be arbitrarily large

Most previous studies: no restrictions on tolls that can be imposed on the arcs of the network

Question: What can be done if such restrictions are given exogenously?

Marginal cost tolls

Marginal cost tolls: $\tau_a(x) := x \cdot \ell'_a(x) \quad \forall a \in A$

Theorem

f is an optimal flow with respect to ℓ iff f is a Nash flow with respect to $\ell + \tau$.

[Beckman, McGuire and Winsten 1956]

Drawbacks:

1. potentially every arc of the network is tolled
2. imposed tolls can be arbitrarily large

Most previous studies: no restrictions on tolls that can be imposed on the arcs of the network

Question: What can be done if such restrictions are given exogenously?

Marginal cost tolls

Marginal cost tolls: $\tau_a(x) := x \cdot \ell_a'(x) \quad \forall a \in A$

Theorem

f is an optimal flow with respect to ℓ iff f is a Nash flow with respect to $\ell + \tau$.

[Beckman, McGuire and Winsten 1956]

Drawbacks:

1. potentially every arc of the network is tolled
2. imposed tolls can be arbitrarily large

Most previous studies: no restrictions on tolls that can be imposed on the arcs of the network

Question: What can be done if such restrictions are given exogenously?

Marginal cost tolls

Marginal cost tolls: $\tau_a(x) := x \cdot \ell'_a(x) \quad \forall a \in A$

Theorem

f is an optimal flow with respect to ℓ iff f is a Nash flow with respect to $\ell + \tau$.

[Beckman, McGuire and Winsten 1956]

Drawbacks:

1. potentially every arc of the network is tolled
2. imposed tolls can be arbitrarily large

Most previous studies: no restrictions on tolls that can be imposed on the arcs of the network

Question: What can be done if such restrictions are given exogenously?

Effective road pricing in Singapore

Restricted Network Toll Problem

Restricted Network Toll Problem

Our setting: suppose we are given threshold functions $\theta := (\theta_a)_{a \in A}$ (dynamic or static)

Tolls $\tau = (\tau_a)_{a \in A}$ are θ-restricted if

$$\forall a \in A: \qquad 0 \leq \tau_a(x) \leq \theta_a(x) \quad \forall x \geq 0$$

Special cases:

1. taxing subnetworks: allow tolls only on arcs $T \subseteq A$
2. restrict the toll on each arc $a \in A$ by a (flow-independent) threshold value θ_a
3. toll on each arc $a \in A$ does not exceed a certain fraction of the latency of that arc, e.g., $\theta_a(x) = \varepsilon \ell_a(x)$ for some $\varepsilon > 0$

Restricted Network Toll Problem

Our setting: suppose we are given threshold functions $\theta := (\theta_a)_{a \in A}$ (dynamic or static)

Tolls $\tau = (\tau_a)_{a \in A}$ are θ-restricted if

$$\forall a \in A: \qquad 0 \leq \tau_a(x) \leq \theta_a(x) \quad \forall x \geq 0$$

Special cases:

1. taxing subnetworks: allow tolls only on arcs $T \subseteq A$
2. restrict the toll on each arc $a \in A$ by a (flow-independent) threshold value θ_a
3. toll on each arc $a \in A$ does not exceed a certain fraction of the latency of that arc, e.g., $\theta_a(x) = \varepsilon \ell_a(x)$ for some $\varepsilon > 0$

Restricted Network Toll Problem

Our setting: suppose we are given threshold functions $\theta := (\theta_a)_{a \in A}$ (dynamic or static)

Tolls $\tau = (\tau_a)_{a \in A}$ are θ-restricted if

$$\forall a \in A: \qquad 0 \leq \tau_a(x) \leq \theta_a(x) \quad \forall x \geq 0$$

Special cases:

1. taxing subnetworks: allow tolls only on arcs $T \subseteq A$
2. restrict the toll on each arc $a \in A$ by a (flow-independent) threshold value θ_a
3. toll on each arc $a \in A$ does not exceed a certain fraction of the latency of that arc, e.g., $\theta_a(x) = \varepsilon \ell_a(x)$ for some $\varepsilon > 0$

Efficiency measures

Given θ-restricted tolls τ, let f^τ denote a Nash flow that is induced by τ, i.e., f^τ is a Nash flow with respect to $\ell + \tau$.

θ-restricted tolls τ are ρ-approximate if for all Nash flows $f^{\bar{\tau}}$ induced by θ-restricted tolls $\bar{\tau}$

$$C(f^\tau) \leq \rho \cdot C(f^{\bar{\tau}})$$

The efficiency of θ-restricted tolls is

$$\min_{\theta\text{-restricted tolls } \tau} \frac{C(f^\tau)}{C(f^*)}$$

Note: interested in the cost of the induced Nash flow (system performance) rather than the total disutility of the players

Efficiency measures

Given θ-restricted tolls τ, let f^τ denote a Nash flow that is induced by τ, i.e., f^τ is a Nash flow with respect to $\ell + \tau$.

θ-restricted tolls τ are ρ-approximate if for all Nash flows $f^{\bar{\tau}}$ induced by θ-restricted tolls $\bar{\tau}$

$$C(f^\tau) \leq \rho \cdot C(f^{\bar{\tau}})$$

The efficiency of θ-restricted tolls is

$$\min_{\theta\text{-restricted tolls } \tau} \frac{C(f^\tau)}{C(f^*)}$$

Note: interested in the cost of the induced Nash flow (system performance) rather than the total disutility of the players

Efficiency measures

Given θ-restricted tolls τ, let f^τ denote a Nash flow that is induced by τ, i.e., f^τ is a Nash flow with respect to $\ell + \tau$.

θ-restricted tolls τ are ρ-approximate if for all Nash flows $f^{\bar{\tau}}$ induced by θ-restricted tolls $\bar{\tau}$

$$C(f^\tau) \leq \rho \cdot C(f^{\bar{\tau}})$$

The efficiency of θ-restricted tolls is

$$\min_{\theta\text{-restricted tolls } \tau} \frac{C(f^\tau)}{C(f^*)}$$

Note: interested in the cost of the induced Nash flow (system performance) rather than the total disutility of the players

Efficiency measures

Given θ-restricted tolls τ, let f^τ denote a Nash flow that is induced by τ, i.e., f^τ is a Nash flow with respect to $\ell + \tau$.

θ-restricted tolls τ are ρ-approximate if for all Nash flows $f^{\bar{\tau}}$ induced by θ-restricted tolls $\bar{\tau}$

$$C(f^\tau) \leq \rho \cdot C(f^{\bar{\tau}})$$

The efficiency of θ-restricted tolls is

$$\min_{\theta\text{-restricted tolls } \tau} \frac{C(f^\tau)}{C(f^*)}$$

Note: interested in the cost of the induced Nash flow (system performance) rather than the total disutility of the players

Results in a nutshell

1. Derive an algorithm to compute optimal θ-restricted tolls for parallel-arc networks and affine latency/threshold functions
   - algorithm works for general latency/threshold functions
   - can guarantee polynomial running time only for affine case
   - **key:** characterize inducible flows for single-commodity networks

2. Provide bounds on the efficiency of θ-restricted tolls for multi-commodity networks and polynomial latency functions
   - derive a pricing scheme that realizes respective efficiency
   - bounds are tight, even for parallel-arc networks
   - approach based on (λ, μ)-smoothness technique
   - **insight:** impose tolls on arcs that are sensitive to flow changes

Results in a nutshell

1. Derive an algorithm to compute optimal θ-restricted tolls for parallel-arc networks and affine latency/threshold functions
   - algorithm works for general latency/threshold functions
   - can guarantee polynomial running time only for affine case
   - **key:** characterize inducible flows for single-commodity networks

2. Provide bounds on the efficiency of θ-restricted tolls for multi-commodity networks and polynomial latency functions
   - derive a pricing scheme that realizes respective efficiency
   - bounds are tight, even for parallel-arc networks
   - approach based on (λ, μ)-smoothness technique
   - **insight:** impose tolls on arcs that are sensitive to flow changes

Related work

1. special case: taxing subnetworks
   - NP-hard to compute optimal tolls for two-commodity networks and affine latency functions
   - algorithm to compute optimal tolls for parallel-arc networks and affine latency functions

   [Hoefer, Olbrich, Skopalik, WINE 08]

2. existence/computation of optimal-inducing tolls for heterogenous players
   - single-commodity, non-atomic — [Cole, Dodis, Roughgarden, STOC 03]
   - multi-commodity, non-atomic — [Fleischer, Jain, Mahdian, FOCS 04]; [Karakostas, Kolliopoulos, FOCS 04]
   - multi-commodity, atomic — [Swamy, SODA 07]

3. price of anarchy and price of stability of ε-Nash flows

   [Christodoulou, Koutsoupias, Spirakis, ESA 09]

Computing Optimal Tolls for Parallel-Arc Networks

Characterization of inducible flows

Question: Given an arbitrary flow f, can we determine whether f is inducible by θ-restricted tolls?

Note: If f is given, then $\ell_a(f_a)$ and $\theta_a(f_a)$ are constants.

Nash conditions: f is a Nash flow with respect to $\ell + \tau$ if

$$\forall P_1, P_2 \in \mathcal{P} \text{ with } f_{P_1} > 0 : \quad (\ell + \tau)_{P_1}(f) \leq (\ell + \tau)_{P_2}(f)$$

Can describe the set of θ-restricted tolls that induce f as

$$\left\{ (\tau_a)_{a \in A} \;\middle|\; \begin{array}{rcll} \delta_v & \leq & \delta_u + \ell_a + \tau_a & \forall a = (u,v) \in A \setminus A^+ \\ \delta_v & = & \delta_u + \ell_a + \tau_a & \forall a = (u,v) \in A^+ \\ \tau_a & \geq & 0 & \forall a \in A \\ \tau_a & \leq & \theta_a & \forall a \in A \end{array} \right\},$$

where $A^+ := \{a \in A : f_a > 0\}$.

Characterization of inducible flows

Question: Given an arbitrary flow f, can we determine whether f is inducible by θ-restricted tolls?

Note: If f is given, then $\ell_a(f_a)$ and $\theta_a(f_a)$ are constants.

Nash conditions: f is a Nash flow with respect to $\ell + \tau$ if

$$\forall P_1, P_2 \in \mathcal{P} \text{ with } f_{P_1} > 0 : \quad (\ell + \tau)_{P_1}(f) \leq (\ell + \tau)_{P_2}(f)$$

Can describe the set of θ-restricted tolls that induce f as

$$\left\{ (\tau_a)_{a \in A} \;\middle|\; \begin{array}{rcll} \delta_v & \leq & \delta_u + \ell_a + \tau_a & \forall a = (u,v) \in A \setminus A^+ \\ \delta_v & = & \delta_u + \ell_a + \tau_a & \forall a = (u,v) \in A^+ \\ \tau_a & \geq & 0 & \forall a \in A \\ \tau_a & \leq & \theta_a & \forall a \in A \end{array} \right\},$$

where $A^+ := \{a \in A : f_a > 0\}$.

Characterization of inducible flows

Question: Given an arbitrary flow f, can we determine whether f is inducible by θ-restricted tolls?

Note: If f is given, then $\ell_a(f_a)$ and $\theta_a(f_a)$ are constants.

Nash conditions: f is a Nash flow with respect to $\ell + \tau$ if

$$\forall P_1, P_2 \in \mathcal{P} \text{ with } f_{P_1} > 0: \quad (\ell+\tau)_{P_1}(f) \leq (\ell+\tau)_{P_2}(f)$$

Can describe the set of θ-restricted tolls that induce f as

$$\left\{ (\tau_a)_{a\in A} \;\middle|\; \begin{array}{lcll} \delta_v & \leq & \delta_u + \ell_a + \tau_a & \forall a=(u,v) \in A \setminus A^+ \\ \delta_v & = & \delta_u + \ell_a + \tau_a & \forall a=(u,v) \in A^+ \\ \tau_a & \geq & 0 & \forall a \in A \\ \tau_a & \leq & \theta_a & \forall a \in A \end{array} \right\},$$

where $A^+ := \{a \in A : f_a > 0\}$.

Characterization of inducible flows

Question: Given an arbitrary flow f, can we determine whether f is inducible by θ-restricted tolls?

Note: If f is given, then $\ell_a(f_a)$ and $\theta_a(f_a)$ are constants.

Nash conditions: f is a Nash flow with respect to $\ell + \tau$ if

$$\forall P_1, P_2 \in \mathcal{P} \text{ with } f_{P_1} > 0: \quad (\ell+\tau)_{P_1}(f) \leq (\ell+\tau)_{P_2}(f)$$

Can describe the set of θ-restricted tolls that induce f as

$$\left\{ (\tau_a)_{a\in A} \;\middle|\; \begin{array}{rcll} \delta_v & \leq & \delta_u + \ell_a + \tau_a & \forall a = (u,v) \in A \setminus A^+ \\ \delta_v & = & \delta_u + \ell_a + \tau_a & \forall a = (u,v) \in A^+ \\ \tau_a & \geq & 0 & \forall a \in A \\ \tau_a & \leq & \theta_a & \forall a \in A \end{array} \right\},$$

where $A^+ := \{a \in A : f_a > 0\}$.

Characterization of inducible flows

Auxiliary graph: $\hat{G}(f) = (V, \hat{A})$ with arc-costs $c : \hat{A} \to \mathbb{R}$:

- for every arc $(u,v) \in A$: introduce a forward arc $(u,v) \in \hat{A}$ with $c_{(u,v)} := \ell_{(u,v)} + \theta_{(u,v)}$
- for every arc $(u,v) \in A^+$: introduce a backward arc $(v,u) \in \hat{A}$ with $c_{(v,u)} := -\ell_{(u,v)}$

Theorem

f is inducible by θ-restricted tolls if and only if $\hat{G}(f)$ does not contain a cycle of negative cost.

[Bonifaci, Mahyar, Schäfer, SAGT '11]

Remark: can also extract respective θ-restricted tolls from $\hat{G}(f)$

Computing optimal tolls: compute a minimum cost flow f such that $\hat{G}(f)$ does not contain a negative cycle

Characterization of inducible flows

Auxiliary graph: $\hat{G}(f) = (V, \hat{A})$ with arc-costs $c : \hat{A} \to \mathbb{R}$:

- for every arc $(u, v) \in A$: introduce a forward arc $(u, v) \in \hat{A}$ with $c_{(u,v)} := \ell_{(u,v)} + \theta_{(u,v)}$
- for every arc $(u, v) \in A^+$: introduce a backward arc $(v, u) \in \hat{A}$ with $c_{(v,u)} := -\ell_{(u,v)}$

Theorem

f is inducible by θ-restricted tolls if and only if $\hat{G}(f)$ does not contain a cycle of negative cost.

[Bonifaci, Mahyar, Schäfer, SAGT '11]

Remark: can also extract respective θ-restricted tolls from $\hat{G}(f)$

Computing optimal tolls: compute a minimum cost flow f such that $\hat{G}(f)$ does not contain a negative cycle

Characterization of inducible flows

Auxiliary graph: $\hat{G}(f) = (V, \hat{A})$ with arc-costs $c : \hat{A} \to \mathbb{R}$:

- for every arc $(u, v) \in A$: introduce a forward arc $(u, v) \in \hat{A}$ with $c_{(u,v)} := \ell_{(u,v)} + \theta_{(u,v)}$
- for every arc $(u, v) \in A^+$: introduce a backward arc $(v, u) \in \hat{A}$ with $c_{(v,u)} := -\ell_{(u,v)}$

Theorem

f is inducible by θ-restricted tolls if and only if $\hat{G}(f)$ does not contain a cycle of negative cost.

[Bonifaci, Mahyar, Schäfer, SAGT '11]

Remark: can also extract respective θ-restricted tolls from $\hat{G}(f)$

Computing optimal tolls: compute a minimum cost flow f such that $\hat{G}(f)$ does not contain a negative cycle

Characterization of inducible flows

Auxiliary graph: $\hat{G}(f) = (V, \hat{A})$ with arc-costs $c : \hat{A} \to \mathbb{R}$:

- for every arc $(u,v) \in A$: introduce a forward arc $(u,v) \in \hat{A}$ with $c_{(u,v)} := \ell_{(u,v)} + \theta_{(u,v)}$
- for every arc $(u,v) \in A^+$: introduce a backward arc $(v,u) \in \hat{A}$ with $c_{(v,u)} := -\ell_{(u,v)}$

Theorem

f is inducible by θ-restricted tolls if and only if $\hat{G}(f)$ does not contain a cycle of negative cost.

[Bonifaci, Mahyar, Schäfer, SAGT '11]

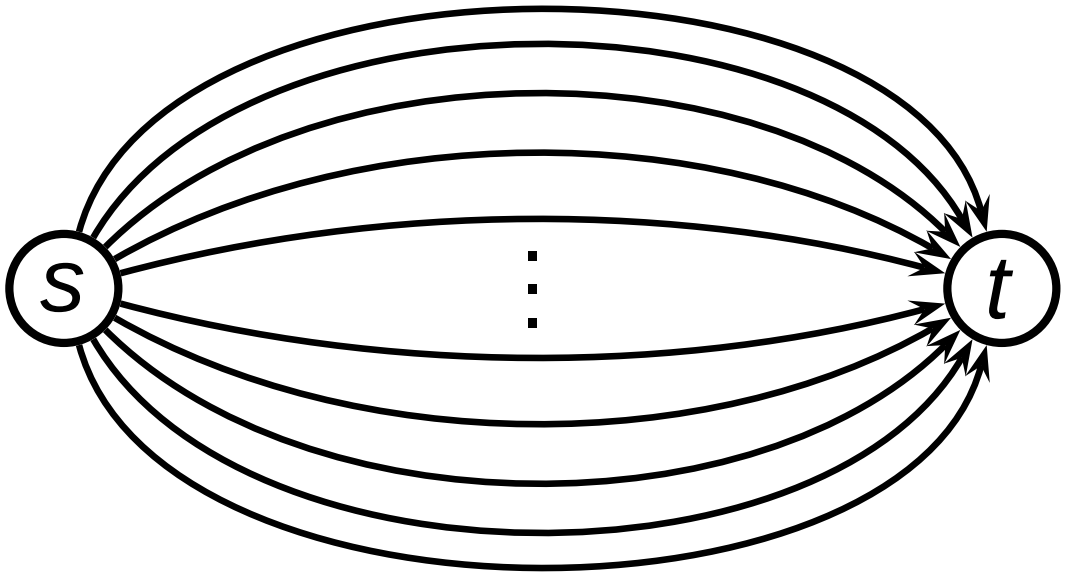
Remark: can also extract respective θ-restricted tolls from $\hat{G}(f)$

Computing optimal tolls: compute a minimum cost flow f such that $\hat{G}(f)$ does not contain a negative cycle

Algorithm for parallel-arc networks

Goal: compute a minimum cost flow f such that

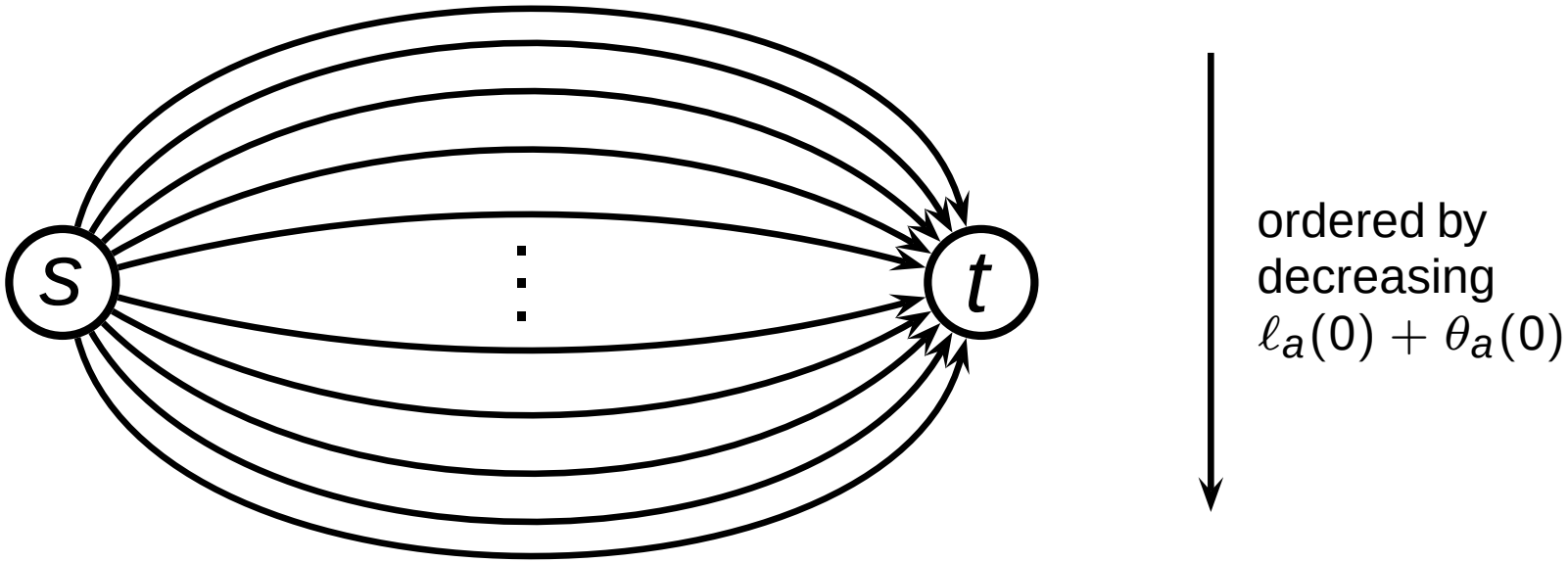
$$\forall a \in A,\ f_a > 0 : \qquad \ell_a(f_a) \leq \ell_{a'}(f_{a'}) + \theta_{a'}(f_{a'}) \quad \forall a' \in A.$$



Algorithm for parallel-arc networks

Goal: compute a minimum cost flow f such that

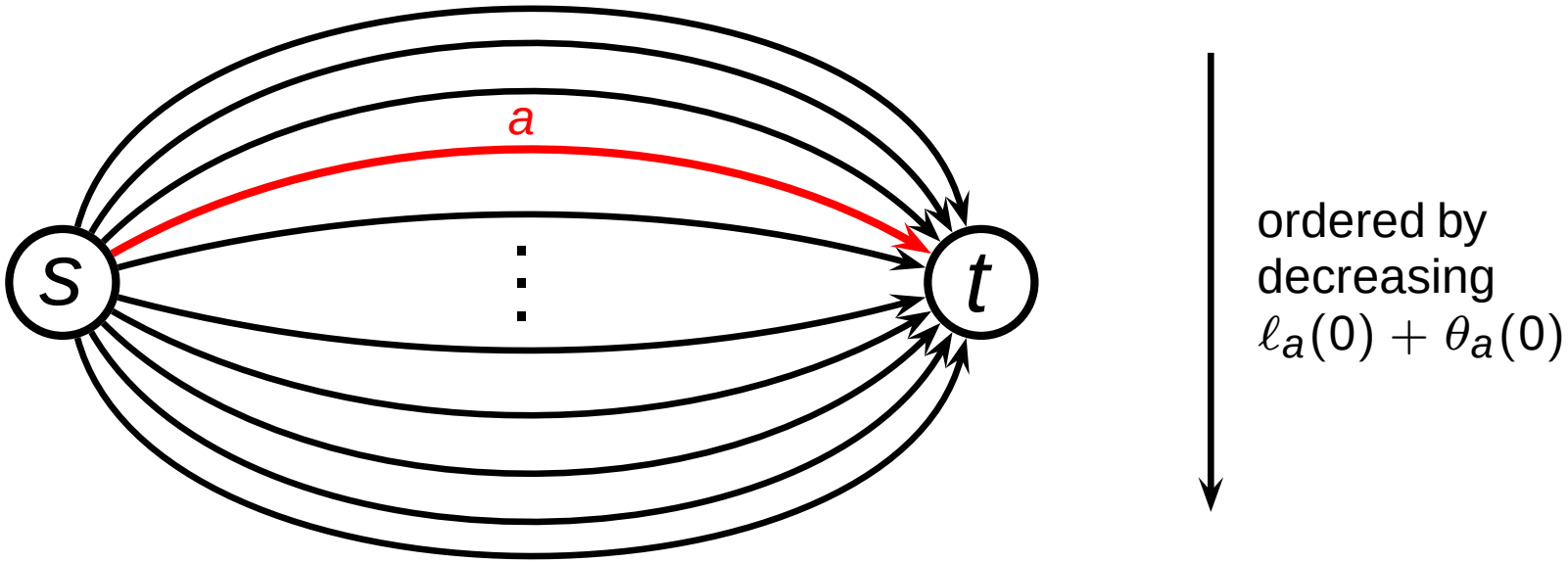
$$\forall a \in A,\ f_a > 0 : \qquad \ell_a(f_a) \leq \ell_{a'}(f_{a'}) + \theta_{a'}(f_{a'}) \quad \forall a' \in A.$$



Algorithm for parallel-arc networks

Goal: compute a minimum cost flow f such that

$$\forall a \in A,\ f_a > 0: \qquad \ell_a(f_a) \leq \ell_{a'}(f_{a'}) + \theta_{a'}(f_{a'}) \quad \forall a' \in A.$$

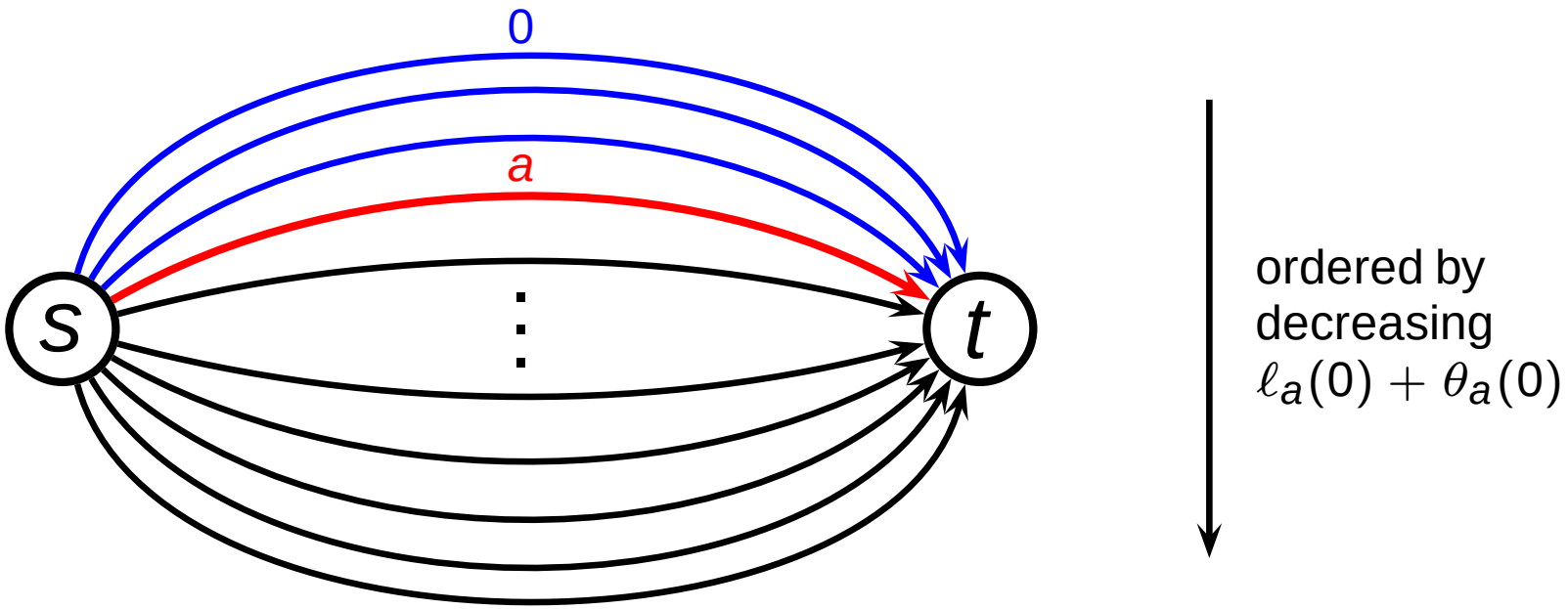


suppose we knew the arc a with $f_a = 0$ and $\ell_a(0) + \theta_a(0)$ minimum

Algorithm for parallel-arc networks

Goal: compute a minimum cost flow f such that

$$\forall a \in A,\ f_a > 0: \qquad \ell_a(f_a) \leq \ell_{a'}(f_{a'}) + \theta_{a'}(f_{a'}) \quad \forall a' \in A.$$

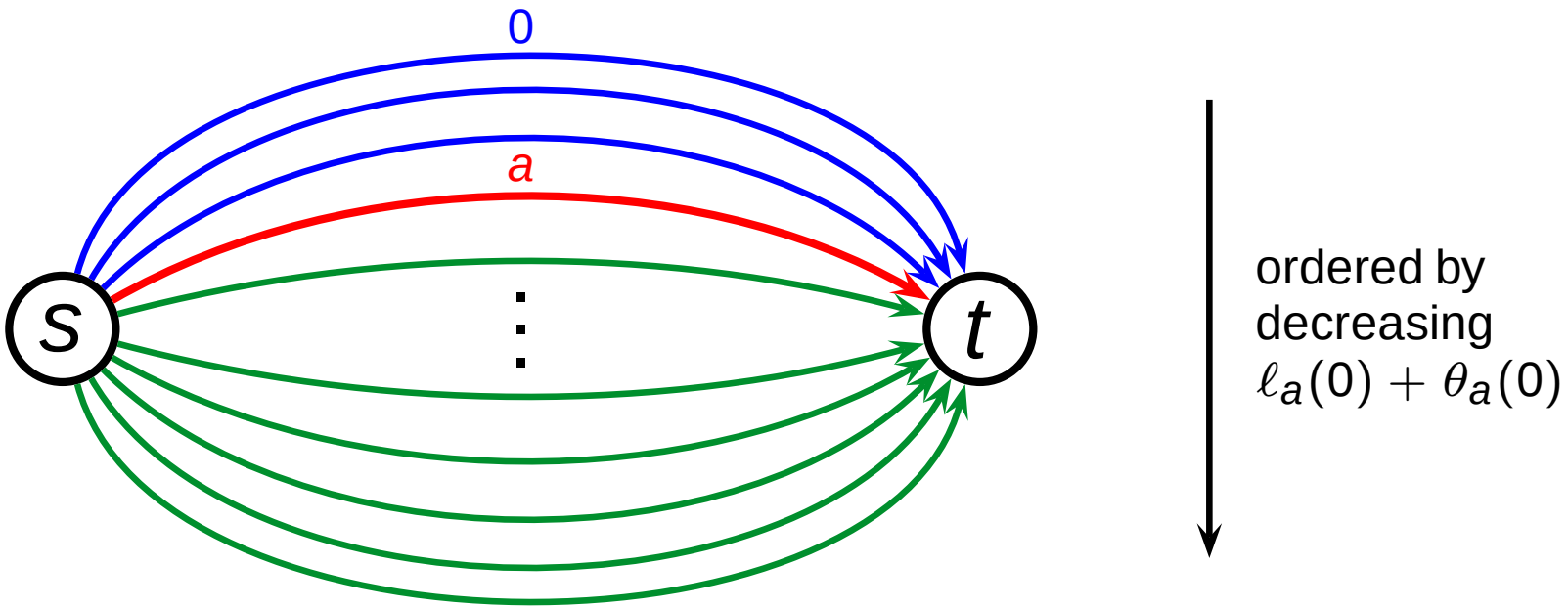


suppose we knew the arc a with $f_a = 0$ and $\ell_a(0) + \theta_a(0)$ minimum

Algorithm for parallel-arc networks

Goal: compute a minimum cost flow f such that

$$\forall a \in A,\ f_a > 0: \qquad \ell_a(f_a) \leq \ell_{a'}(f_{a'}) + \theta_{a'}(f_{a'}) \quad \forall a' \in A.$$



suppose we knew the arc a with $f_a = 0$ and $\ell_a(0) + \theta_a(0)$ minimum

Algorithm for parallel-arc networks

Goal: compute a minimum cost flow f such that

$$\forall a \in A,\ f_a > 0: \qquad \ell_a(f_a) \leq \ell_{a'}(f_{a'}) + \theta_{a'}(f_{a'}) \quad \forall a' \in A. \qquad (1)$$

Algorithm:

1. Guess the minimum value $z = \ell_a(0) + \theta_a(0)$ of a zero-flow arc a in a minimum cost flow satisfying (1).
2. Set $f_a = 0$ for every arc $a \in A$ with $\ell_a(0) > z$.
3. Let $A' = \{a \in A \mid \ell_a(0) \leq z\}$ be the remaining arcs and solve

$$\begin{array}{lrcll} \min & \sum_{a \in A'} f_a \ell_a(f_a) & & & \\ \text{s.t.} & \sum_{a \in A'} f_a & = & r & \\ & f_a & \geq & 0 & \forall a \in A' \\ & \ell_a(f_a) & \leq & z & \forall a \in A' \\ & \ell_a(f_a) & \leq & \ell_{a'}(f_{a'}) + \theta_{a'}(f_{a'}) & \forall a, a' \in A' \end{array}$$

Algorithm for parallel-arc networks

Goal: compute a minimum cost flow f such that

$$\forall a \in A,\ f_a > 0: \qquad \ell_a(f_a) \leq \ell_{a'}(f_{a'}) + \theta_{a'}(f_{a'}) \quad \forall a' \in A. \qquad (1)$$

Algorithm:

1. Guess the minimum value $z = \ell_a(0) + \theta_a(0)$ of a zero-flow arc a in a minimum cost flow satisfying (1).
2. Set $f_a = 0$ for every arc $a \in A$ with $\ell_a(0) > z$.
3. Let $A' = \{a \in A \mid \ell_a(0) \leq z\}$ be the remaining arcs and solve

$$\begin{array}{lrcll} \min & \sum_{a \in A'} f_a \ell_a(f_a) & & & \\ \text{s.t.} & \sum_{a \in A'} f_a & = & r & \\ & f_a & \geq & 0 & \forall a \in A' \\ & \ell_a(f_a) & \leq & z & \forall a \in A' \\ & \ell_a(f_a) & \leq & \ell_{a'}(f_{a'}) + \theta_{a'}(f_{a'}) & \forall a, a' \in A' \end{array}$$

Algorithm for parallel-arc networks

Goal: compute a minimum cost flow f such that

$$\forall a \in A,\ f_a > 0: \qquad \ell_a(f_a) \leq \ell_{a'}(f_{a'}) + \theta_{a'}(f_{a'}) \quad \forall a' \in A. \qquad (1)$$

Algorithm:

1. Guess the minimum value $z = \ell_a(0) + \theta_a(0)$ of a zero-flow arc a in a minimum cost flow satisfying (1).
2. Set $f_a = 0$ for every arc $a \in A$ with $\ell_a(0) > z$.
3. Let $A' = \{a \in A \mid \ell_a(0) \leq z\}$ be the remaining arcs and solve

$$\begin{array}{lrcll} \min & \sum_{a \in A'} f_a \ell_a(f_a) & & & \\ \text{s.t.} & \sum_{a \in A'} f_a & = & r & \\ & f_a & \geq & 0 & \forall a \in A' \\ & \ell_a(f_a) & \leq & z & \forall a \in A' \\ & \ell_a(f_a) & \leq & \ell_{a'}(f_{a'}) + \theta_{a'}(f_{a'}) & \forall a, a' \in A' \end{array}$$

Algorithm for parallel-arc networks

Goal: compute a minimum cost flow f such that

$$\forall a \in A,\ f_a > 0: \qquad \ell_a(f_a) \leq \ell_{a'}(f_{a'}) + \theta_{a'}(f_{a'}) \quad \forall a' \in A. \qquad (1)$$

Algorithm:

1. Guess the minimum value $z = \ell_a(0) + \theta_a(0)$ of a zero-flow arc a in a minimum cost flow satisfying (1).
2. Set $f_a = 0$ for every arc $a \in A$ with $\ell_a(0) > z$.
3. Let $A' = \{a \in A \mid \ell_a(0) \leq z\}$ be the remaining arcs and solve

$$\begin{array}{lrcll} \min & \sum_{a \in A'} f_a \ell_a(f_a) & & & \\ \text{s.t.} & \sum_{a \in A'} f_a & = & r & \\ & f_a & \geq & 0 & \forall a \in A' \\ & \ell_a(f_a) & \leq & z & \forall a \in A' \\ & \ell_a(f_a) & \leq & \ell_{a'}(f_{a'}) + \theta_{a'}(f_{a'}) & \forall a, a' \in A' \end{array}$$

Algorithm for parallel-arc networks

Goal: compute a minimum cost flow f such that

$$\forall a \in A,\ f_a > 0: \qquad \ell_a(f_a) \leq \ell_{a'}(f_{a'}) + \theta_{a'}(f_{a'}) \quad \forall a' \in A. \tag{1}$$

Algorithm:

1. Guess the minimum value $z = \ell_a(0) + \theta_a(0)$ of a zero-flow arc a in a minimum cost flow satisfying (1).
2. Set $f_a = 0$ for every arc $a \in A$ with $\ell_a(0) > z$.
3. Let $A' = \{a \in A \mid \ell_a(0) \leq z\}$ be the remaining arcs and solve

$$\begin{array}{lrcll} \min & \sum_{a \in A'} f_a \ell_a(f_a) & & & \\ \text{s.t.} & \sum_{a \in A'} f_a & = & r & \\ & f_a & \geq & 0 & \forall a \in A' \\ & \ell_a(f_a) & \leq & z & \forall a \in A' \\ & \ell_a(f_a) & \leq & \ell_{a'}(f_{a'}) + \theta_{a'}(f_{a'}) & \forall a, a' \in A' \end{array}$$

Algorithm for parallel-arc networks

Goal: compute a minimum cost flow f such that

$$\forall a \in A,\ f_a > 0: \qquad \ell_a(f_a) \leq \ell_{a'}(f_{a'}) + \theta_{a'}(f_{a'}) \quad \forall a' \in A. \qquad (1)$$

Algorithm:

1. Guess the minimum value $z = \ell_a(0) + \theta_a(0)$ of a zero-flow arc a in a minimum cost flow satisfying (1).
2. Set $f_a = 0$ for every arc $a \in A$ with $\ell_a(0) > z$.
3. Let $A' = \{a \in A \mid \ell_a(0) \leq z\}$ be the remaining arcs and solve

$$\begin{array}{lrcll} \min & \sum_{a \in A'} f_a \ell_a(f_a) & & & \\ \text{s.t.} & \sum_{a \in A'} f_a & = & r & \\ & f_a & \geq & 0 & \forall a \in A' \\ & \ell_a(f_a) & \leq & z & \forall a \in A' \\ & \ell_a(f_a) & \leq & \ell_{a'}(f_{a'}) + \theta_{a'}(f_{a'}) & \forall a, a' \in A' \end{array}$$

Main result

Theorem

The algorithm computes optimal θ-restricted tolls for parallel-arc networks in polynomial time if all latency and threshold functions are affine.

[Bonifaci, Mahyar, Schäfer, SAGT '11]

Remark: [Hoefer et al., WINE 08] derived a similar result for the special case that restrictions are of the form $\theta_a \in \{0, \infty\}$ for every arc $a \in A$.

Recall: problem is NP-hard for two-commodity networks and affine latency functions

Open problem: Is the restricted network toll problem NP-hard for single-commodity networks and affine latency functions?

Main result

Theorem

The algorithm computes optimal θ-restricted tolls for parallel-arc networks in polynomial time if all latency and threshold functions are affine.

[Bonifaci, Mahyar, Schäfer, SAGT '11]

Remark: [Hoefer et al., WINE 08] derived a similar result for the special case that restrictions are of the form $\theta_a \in \{0, \infty\}$ for every arc $a \in A$.

Recall: problem is NP-hard for two-commodity networks and affine latency functions

Open problem: Is the restricted network toll problem NP-hard for single-commodity networks and affine latency functions?

Main result

Theorem

The algorithm computes optimal θ-restricted tolls for parallel-arc networks in polynomial time if all latency and threshold functions are affine.

[Bonifaci, Mahyar, Schäfer, SAGT '11]

Remark: [Hoefer et al., WINE 08] derived a similar result for the special case that restrictions are of the form $\theta_a \in \{0, \infty\}$ for every arc $a \in A$.

Recall: problem is NP-hard for two-commodity networks and affine latency functions

Open problem: Is the restricted network toll problem NP-hard for single-commodity networks and affine latency functions?

Main result

Theorem

The algorithm computes optimal θ-restricted tolls for parallel-arc networks in polynomial time if all latency and threshold functions are affine.

[Bonifaci, Mahyar, Schäfer, SAGT '11]

Remark: [Hoefer et al., WINE 08] derived a similar result for the special case that restrictions are of the form $\theta_a \in \{0, \infty\}$ for every arc $a \in A$.

Recall: problem is NP-hard for two-commodity networks and affine latency functions

Open problem: Is the restricted network toll problem NP-hard for single-commodity networks and affine latency functions?

Efficiency of Restricted Tolls in Multi-Commodity Networks

Pricing scheme

Idea: charge marginal cost tolls if possible; otherwise, charge maximum amount $\theta_a(x)$

Bounded marginal cost tolls:

$$\tau_a(x) := \min\{x \cdot \ell_a'(x),\ \theta_a(x)\}$$

Let f^τ be a Nash flow with respect to $\ell + \tau$. Basically, the efficiency of f^τ depends on

$$\bar{\varepsilon}_d = \min\left\{ \frac{\theta_a(f_a^\tau)}{\ell_a(f_a^\tau)} \;\middle|\; \ell_a \text{ is polynomial function of degree } d \right\}$$

Insight: more important to impose tolls on arcs with high degree latency functions than on arcs with low degree latency functions.

Pricing scheme

Idea: charge marginal cost tolls if possible; otherwise, charge maximum amount $\theta_a(x)$

Bounded marginal cost tolls:

$$\tau_a(x) := \min\{x \cdot \ell_a'(x),\ \theta_a(x)\}$$

Let f^τ be a Nash flow with respect to $\ell + \tau$. Basically, the efficiency of f^τ depends on

$$\bar{\varepsilon}_d = \min\left\{ \frac{\theta_a(f_a^\tau)}{\ell_a(f_a^\tau)} \;\middle|\; \ell_a \text{ is polynomial function of degree } d \right\}$$

Insight: more important to impose tolls on arcs with high degree latency functions than on arcs with low degree latency functions.

Pricing scheme

Idea: charge marginal cost tolls if possible; otherwise, charge maximum amount $\theta_a(x)$

Bounded marginal cost tolls:

$$\tau_a(x) := \min\{x \cdot \ell_a'(x),\ \theta_a(x)\}$$

Let f^τ be a Nash flow with respect to $\ell + \tau$. Basically, the efficiency of f^τ depends on

$$\bar{\varepsilon}_d = \min\left\{ \frac{\theta_a(f_a^\tau)}{\ell_a(f_a^\tau)} \;\middle|\; \ell_a \text{ is polynomial function of degree } d \right\}$$

Insight: more important to impose tolls on arcs with high degree latency functions than on arcs with low degree latency functions.

Pricing scheme

Idea: charge marginal cost tolls if possible; otherwise, charge maximum amount $\theta_a(x)$

Bounded marginal cost tolls:

$$\tau_a(x) := \min\{x \cdot \ell_a'(x),\ \theta_a(x)\}$$

Let f^τ be a Nash flow with respect to $\ell + \tau$. Basically, the efficiency of f^τ depends on

$$\bar{\varepsilon}_d = \min\left\{ \frac{\theta_a(f_a^\tau)}{\ell_a(f_a^\tau)} \;\middle|\; \ell_a \text{ is polynomial function of degree } d \right\}$$

Insight: more important to impose tolls on arcs with high degree latency functions than on arcs with low degree latency functions.

Pricing scheme

Idea: charge marginal cost tolls if possible; otherwise, charge maximum amount $\theta_a(x)$

Bounded marginal cost tolls:

$$\tau_a(x) := \min\{x \cdot \ell_a'(x),\ \theta_a(x)\}$$

Let f^τ be a Nash flow with respect to $\ell + \tau$. Basically, the efficiency of f^τ depends on

$$\bar{\varepsilon}_d = \min\left\{ \frac{\theta_a(f_a^\tau)}{\ell_a(f_a^\tau)} \;\middle|\; \ell_a \text{ is polynomial function of degree } d \right\}$$

Insight: more important to impose tolls on arcs with high degree latency functions than on arcs with low degree latency functions.

Consequence

Theorem

Suppose all threshold functions are of the form $\theta_a(x) = \varepsilon \ell_a(x)$ and latency functions are polynomials of degree at most p. The efficiency of bounded marginal cost tolls is 1 if $\varepsilon \geq p$ and at most

$$\left((1+\varepsilon)\left(1 - \frac{p}{p+1}\left(\frac{1+\varepsilon}{p+1}\right)^{1/p}\right)\right)^{-1}$$

otherwise.

[Bonifaci, Mahyar, Schäfer, SAGT '11]

Remark: bound coincides with the price of stability of ε-Nash flows shown by [Christodoulou, Koutsoupias, Spirakis, ESA 09]

Consequence

Theorem

Suppose all threshold functions are of the form $\theta_a(x) = \varepsilon \ell_a(x)$ and latency functions are polynomials of degree at most p. The efficiency of bounded marginal cost tolls is 1 if $\varepsilon \geq p$ and at most

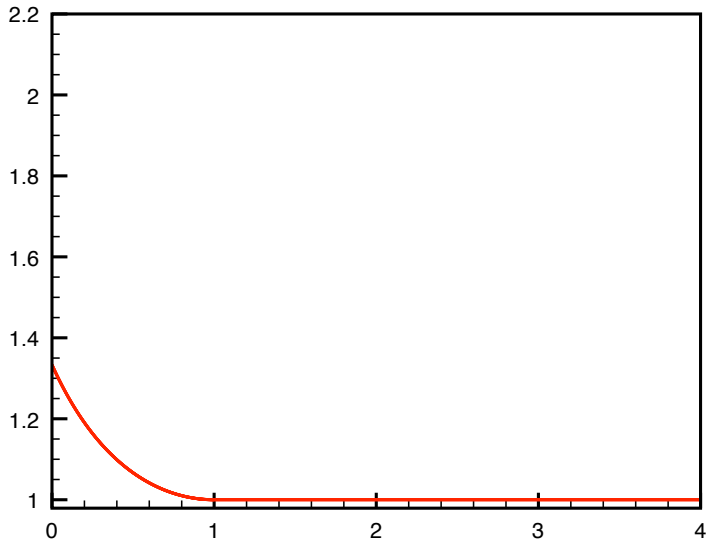
$$\left((1+\varepsilon)\left(1 - \frac{p}{p+1}\left(\frac{1+\varepsilon}{p+1}\right)^{1/p}\right)\right)^{-1}$$

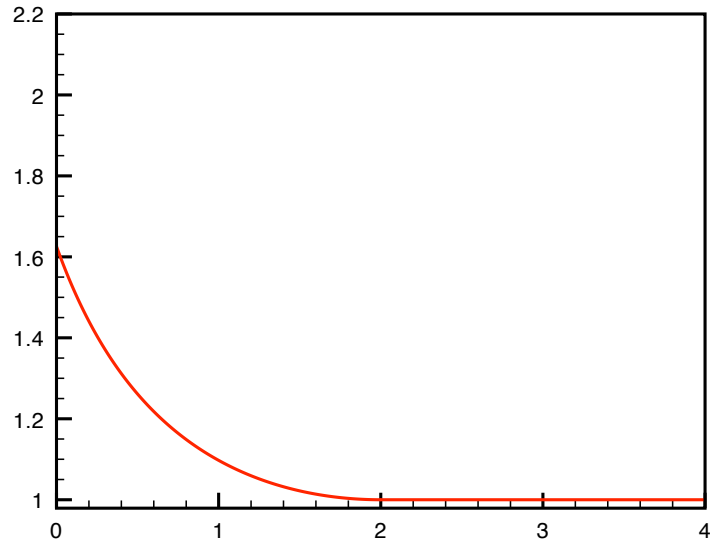
otherwise.

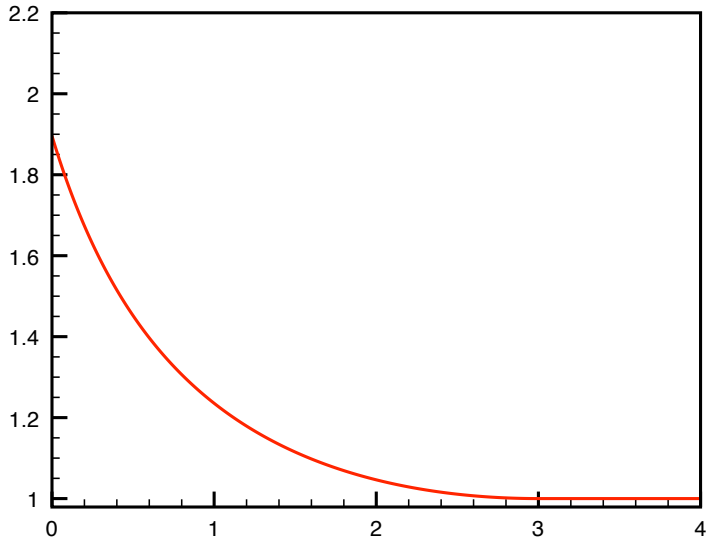
[Bonifaci, Mahyar, Schäfer, SAGT '11]

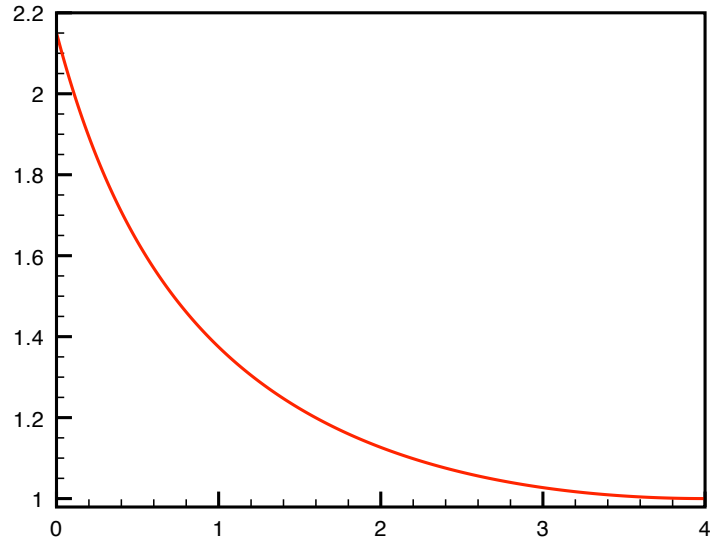
Remark: bound coincides with the price of stability of ε-Nash flows shown by [Christodoulou, Koutsoupias, Spirakis, ESA 09]

Bounds for $p = 1, 2, 3, 4$









Concluding Remarks

Today's goals

Goals for today:

- get a glimpse of the research in algorithmic game theory
- running example: network routing
- get an idea of the phenomena and questions that arise
- familiarize with used approaches and techniques
- approach the state-of-the-art of recent studies (perhaps)

Algorithmic game theory ...

... is a young, challenging, interdisciplinary research field
... addresses many aspects of practical relevance
... can have an impact on real-world applications
... unites ideas from game theory, algorithms, combinatorial optimization, complexity theory, etc.
... is fun!

Exercises

1. Show how to compute constant tolls that induce an optimal flow as Nash flow.

2. Prove that the price of anarchy is 1 for the network routing game if all latency functions are of the form $\ell_a(x) = q_a \cdot x$.

3. Prove that there always exists a trichromatic triangle in the Sperner triangle.