Truthfulness and frugality ratio in the cheapest path auctions

Estonian Theory Days 2009

Nick Gravin
(based on the unpublished joint paper with E. Elkind, N. Chen, F. Petrov)

St.Petersburg Department of Steklov Mathematical Institute RAS,

Nanyang Technological University, Singapore

October 2009

General set system auctions. Model:

- There are given a set of agents $\mathcal{E} = \{1, 2, \dots, n\}$ and a family of feasible sets $\mathcal{F} \subseteq 2^{\mathcal{E}}$.
- Auctioneer is intent on hiring a team (some feasible set) of agents — winning set.
- Each agent i has a true cost c_i, but at the auction he can bid another price $b_i \geq c_i$.
- Utility u_i (profit) of an agent is $b_i - c_i$ if he wins and 0 if he loses.
- Every agent is selfish and wants to maximize u_i.

General set system auctions. Model:

- There are given a set of agents $\mathcal{E} = \{1, 2, \ldots, n\}$ and a family of feasible sets $\mathcal{F} \subseteq 2^{\mathcal{E}}$.
- Auctioneer is intent on hiring a team (some feasible set) of agents — winning set.
- Each agent i has a true cost c_i, but at the auction he can bid another price $b_i \geq c_i$.
- Utility u_i (profit) of an agent is $b_i - c_i$ if he wins and 0 if he loses.
- Every agent is selfish and wants to maximize u_i.

General set system auctions. Model:

- There are given a set of agents $\mathcal{E} = \{1, 2, \ldots, n\}$ and a family of feasible sets $\mathcal{F} \subseteq 2^{\mathcal{E}}$.
- Auctioneer is intent on hiring a team (some feasible set) of agents — winning set.
- Each agent i has a true cost c_i, but at the auction he can bid another price $b_i \geq c_i$.
- Utility u_i (profit) of an agent is $b_i - c_i$ if he wins and 0 if he loses.
- Every agent is selfish and wants to maximize u_i.

General set system auctions. Model:

- There are given a set of agents $\mathcal{E} = \{1, 2, \ldots, n\}$ and a family of feasible sets $\mathcal{F} \subseteq 2^{\mathcal{E}}$.
- Auctioneer is intent on hiring a team (some feasible set) of agents — winning set.
- Each agent i has a true cost c_i, but at the auction he can bid another price $b_i \geq c_i$.
- Utility u_i (profit) of an agent is $b_i - c_i$ if he wins and 0 if he loses.
- Every agent is selfish and wants to maximize u_i.

General set system auctions. Model:

- There are given a set of agents $\mathcal{E} = \{1, 2, \ldots, n\}$ and a family of feasible sets $\mathcal{F} \subseteq 2^{\mathcal{E}}$.
- Auctioneer is intent on hiring a team (some feasible set) of agents — winning set.
- Each agent i has a true cost c_i, but at the auction he can bid another price $b_i \geq c_i$.
- Utility u_i (profit) of an agent is $b_i - c_i$ if he wins and 0 if he loses.
- Every agent is selfish and wants to maximize u_i.

First price rule

Auction is called a first price if auctioneer always chooses the cheapest feasible set, i.e. a set F with minimal $\sum_{e \in F} b(e)$.

examples

Non-trivial issues concerning computational complexity appears when feasible sets are given implicitly.

1. **Path auction**: agents — edges in the graph; feasible sets — all paths between two given vertices s and t.
2. k-**Path**: agents — edges in the graph; feasible sets — k edge-disjoint paths from s to t.
3. **Vertex cover auction**: agents — vertices; feasible sets — all vertex covers of edges.
4. **Spanning tree auction**: agents — edges; feasible sets — spanning trees.
5. **Perfect matching auction**: agents — edges in bipartite graph; feasible sets — perfect matchings.

examples

Non-trivial issues concerning computational complexity appears when feasible sets are given implicitly.

1. **Path auction**: agents — edges in the graph; feasible sets — all paths between two given vertices s and t.
2. **k-Path**: agents — edges in the graph; feasible sets — k edge-disjoint paths from s to t.
3. **Vertex cover auction**: agents — vertices; feasible sets — all vertex covers of edges.
4. **Spanning tree auction**: agents — edges; feasible sets — spanning trees.
5. **Perfect matching auction**: agents — edges in bipartite graph; feasible sets — perfect matchings.

examples

Non-trivial issues concerning computational complexity appears when feasible sets are given implicitly.

1. **Path auction**: agents — edges in the graph; feasible sets — all paths between two given vertices s and t.
2. k-**Path**: agents — edges in the graph; feasible sets — k edge-disjoint paths from s to t.
3. **Vertex cover auction**: agents — vertices; feasible sets — all vertex covers of edges.
4. **Spanning tree auction**: agents — edges; feasible sets — spanning trees.
5. **Perfect matching auction**: agents — edges in bipartite graph; feasible sets — perfect matchings.

examples

Non-trivial issues concerning computational complexity appears when feasible sets are given implicitly.

1. **Path auction**: agents — edges in the graph; feasible sets — all paths between two given vertices s and t.
2. k-**Path**: agents — edges in the graph; feasible sets — k edge-disjoint paths from s to t.
3. **Vertex cover auction**: agents — vertices; feasible sets — all vertex covers of edges.
4. **Spanning tree auction**: agents — edges; feasible sets — spanning trees.
5. **Perfect matching auction**: agents — edges in bipartite graph; feasible sets — perfect matchings.

examples

Non-trivial issues concerning computational complexity appears when feasible sets are given implicitly.

1. **Path auction**: agents — edges in the graph; feasible sets — all paths between two given vertices s and t.
2. k-**Path**: agents — edges in the graph; feasible sets — k edge-disjoint paths from s to t.
3. **Vertex cover auction**: agents — vertices; feasible sets — all vertex covers of edges.
4. **Spanning tree auction**: agents — edges; feasible sets — spanning trees.
5. **Perfect matching auction**: agents — edges in bipartite graph; feasible sets — perfect matchings.

examples

Non-trivial issues concerning computational complexity appears when feasible sets are given implicitly.

1. **Path auction**: agents — edges in the graph; feasible sets — all paths between two given vertices s and t.
2. k-**Path**: agents — edges in the graph; feasible sets — k edge-disjoint paths from s to t.
3. **Vertex cover auction**: agents — vertices; feasible sets — all vertex covers of edges.
4. **Spanning tree auction**: agents — edges; feasible sets — spanning trees.
5. **Perfect matching auction**: agents — edges in bipartite graph; feasible sets — perfect matchings.

Nash equilibrium in first price auction

Definition

Nash equilibrium(NE) in the first price auction is an assignment of agent bids b, such that no agent e can increase its utility u_e by varying its bid b_e.

NE exists if there is no agent belonging to all feasible sets(**monopolist**).

We can find a NE explicitly in polynomial time.

Nash equilibrium in first price auction

Definition

Nash equilibrium(NE) in the first price auction is an assignment of agent bids b, such that no agent e can increase its utility u_e by varying its bid b_e.

NE exists if there is no agent belonging to all feasible sets(**monopolist**).

We can find a NE explicitly in polynomial time.

Nash equilibrium in first price auction

Definition

Nash equilibrium(NE) in the first price auction is an assignment of agent bids b, such that no agent e can increase its utility u_e by varying its bid b_e.

NE exists if there is no agent belonging to all feasible sets(**monopolist**).
We can find a NE explicitly in polynomial time.

Truthful mechanisms

Mechanisms for winners and payments selection

- Suppose the true costs of all agents are private.
- Agents have only one chance to send their bids to the auctioneer.
- Auctioneer (buyer, center) on the base of these bids should choose the winning set and how much to pay to them.

Definition

The mechanism of winners selection and their payments is called **truthful** if each agent maximizes utility by bidding its true cost (for any fixed bids of other agents).

Mechanisms for winners and payments selection

- Suppose the true costs of all agents are private.
- Agents have only one chance to send their bids to the auctioneer.
- Auctioneer (buyer, center) on the base of these bids should choose the winning set and how much to pay to them.

Definition

The mechanism of winners selection and their payments is called **truthful** if each agent maximizes utility by bidding its true cost (for any fixed bids of other agents).

Mechanisms for winners and payments selection

- Suppose the true costs of all agents are private.
- Agents have only one chance to send their bids to the auctioneer.
- Auctioneer (buyer, center) on the base of these bids should choose the winning set and how much to pay to them.

Definition

The mechanism of winners selection and their payments is called **truthful** if each agent maximizes utility by bidding its true cost (for any fixed bids of other agents).

Mechanisms for winners and payments selection

- Suppose the true costs of all agents are private.
- Agents have only one chance to send their bids to the auctioneer.
- Auctioneer (buyer, center) on the base of these bids should choose the winning set and how much to pay to them.

Definition

The mechanism of winners selection and their payments is called **truthful** if each agent maximizes utility by bidding its true cost (for any fixed bids of other agents).

Example

(Vickrey) Second price auction (for particular set system).
There are n sellers and auctioneer needs to bay exactly one thing.
Mechanism selects the cheapest agent and pays to him the bid of the second by the price agent.

Is it clear why this mechanism is truthful?

Example

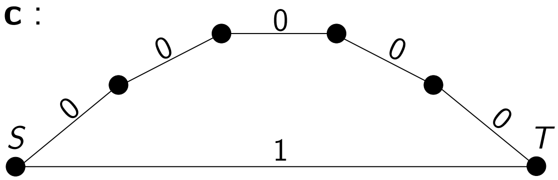
(Vickrey) Second price auction (for particular set system).
There are n sellers and auctioneer needs to bay exactly one thing.
Mechanism selects the cheapest agent and pays to him the bid of the second by the price agent.

Is it clear why this mechanism is truthful?

Example

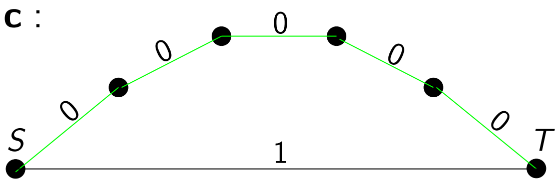
(Vickrey-Clarke-Groves) VCG for general set system.
Choose the cheapest feasible set F and pay **independently** to each agent in F its threshold bid, i.e. such bid that F is still the cheapest feasible set.

VCG for cheapest path



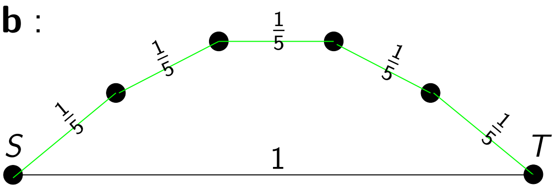
Initial true costs.

VCG for cheapest path



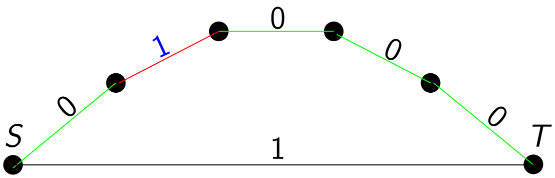
Cheapest path.

VCG for cheapest path



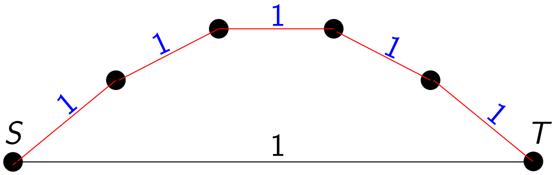
Cheapest Nash equilibrium. Total payment is 1.

VCG for cheapest path



Payment to each agent on the winning path should be 1.

VCG for cheapest path



The total payment is 5, but for NE it was only 1.

How much mechanism overpays

- $(\mathcal{E}, \mathcal{F})$ — monopoly free set system;
- $\mathcal{M}$ — truthful mechanism for $(\mathcal{E}, \mathcal{F})$.
- c is a true cost vector; $\nu(c)$ — the total payment in the **cheapest** Nash for $(\mathcal{E}, \mathcal{F}, c)$;
- $P_{\mathcal{M}}(c)$ — total payment of $\mathcal{M}$ for c;

Definition

Frugality ratio(Karlin, Kempe, Tamir FOCS 05)measures a mechanism efficiency.

$$\Phi_{\mathcal{M}} = \sup_c \frac{P_{\mathcal{M}}(c)}{\nu(c)}.$$

How much mechanism overpays

- $(\mathcal{E}, \mathcal{F})$ — monopoly free set system;
- $\mathcal{M}$ — truthful mechanism for $(\mathcal{E}, \mathcal{F})$.
- c is a true cost vector; $\nu(c)$ — the total payment in the **cheapest** Nash for $(\mathcal{E}, \mathcal{F}, c)$;
- $P_{\mathcal{M}}(c)$ — total payment of $\mathcal{M}$ for c;

Definition

Frugality ratio(Karlin, Kempe, Tamir FOCS 05)measures a mechanism efficiency.

$$\Phi_{\mathcal{M}} = \sup_{c} \frac{P_{\mathcal{M}}(c)}{\nu(c)}.$$

How much mechanism overpays

- $(\mathcal{E}, \mathcal{F})$ — monopoly free set system;
- $\mathcal{M}$ — truthful mechanism for $(\mathcal{E}, \mathcal{F})$.
- c is a true cost vector; $\nu(c)$ — the total payment in the **cheapest** Nash for $(\mathcal{E}, \mathcal{F}, c)$;
- $P_{\mathcal{M}}(c)$ — total payment of $\mathcal{M}$ for c;

Definition

Frugality ratio(Karlin, Kempe, Tamir FOCS 05)measures a mechanism efficiency.

$$\Phi_{\mathcal{M}} = \sup_{c} \frac{P_{\mathcal{M}}(c)}{\nu(c)}.$$

How much mechanism overpays

- $(\mathcal{E}, \mathcal{F})$ — monopoly free set system;
- $\mathcal{M}$ — truthful mechanism for $(\mathcal{E}, \mathcal{F})$.
- c is a true cost vector; $\nu(c)$ — the total payment in the **cheapest** Nash for $(\mathcal{E}, \mathcal{F}, c)$;
- $P_{\mathcal{M}}(c)$ — total payment of $\mathcal{M}$ for c;

Definition

Frugality ratio(Karlin, Kempe, Tamir FOCS 05)measures a mechanism efficiency.

$$\Phi_{\mathcal{M}} = \sup_c \frac{P_{\mathcal{M}}(c)}{\nu(c)}.$$

How much mechanism overpays

- $(\mathcal{E}, \mathcal{F})$ — monopoly free set system;
- $\mathcal{M}$ — truthful mechanism for $(\mathcal{E}, \mathcal{F})$.
- c is a true cost vector; $\nu(c)$ — the total payment in the **cheapest** Nash for $(\mathcal{E}, \mathcal{F}, c)$;
- $P_{\mathcal{M}}(c)$ — total payment of $\mathcal{M}$ for c;

Definition

Frugality ratio(Karlin, Kempe, Tamir FOCS 05)measures a mechanism efficiency.

$$\Phi_{\mathcal{M}} = \sup_{c} \frac{P_{\mathcal{M}}(c)}{\nu(c)}.$$

Motivation for frugality ratio

- $\Phi_{\mathcal{M}} \geq 1$, since one can take c to be by itself a Nash equilibrium and then $P_{\mathcal{M}}(c) \geq \nu(c)$.
- $\Phi_{\mathcal{M}} = 1 \Leftrightarrow (\mathcal{E}, \mathcal{F})$ is a matroid.(KKT 05)
- $(\mathcal{E}, \mathcal{F})$ is a Matroid iff for every two sets $S, T \in \mathcal{F}$, there is a bijection f between $S \setminus T$ and $T \setminus S$ such that $S \setminus \{e\} \cup \{f(e)\}$ is in $\mathcal{F}$ for every $e \in S \setminus T$.

Motivation for frugality ratio

- $\Phi_{\mathcal{M}} \geq 1$, since one can take c to be by itself a Nash equilibrium and then $P_{\mathcal{M}}(c) \geq \nu(c)$.
- $\Phi_{\mathcal{M}} = 1 \Leftrightarrow (\mathcal{E}, \mathcal{F})$ is a matroid.(KKT 05)
- $(\mathcal{E}, \mathcal{F})$ is a Matroid iff for every two sets $S, T \in \mathcal{F}$, there is a bijection f between $S \setminus T$ and $T \setminus S$ such that $S \setminus \{e\} \cup \{f(e)\}$ is in $\mathcal{F}$ for every $e \in S \setminus T$.

Motivation for frugality ratio

- $\Phi_{\mathcal{M}} \geq 1$, since one can take c to be by itself a Nash equilibrium and then $P_{\mathcal{M}}(c) \geq \nu(c)$.
- $\Phi_{\mathcal{M}} = 1 \Leftrightarrow (\mathcal{E}, \mathcal{F})$ is a matroid.(KKT 05)
- $(\mathcal{E}, \mathcal{F})$ is a Matroid iff for every two sets $S, T \in \mathcal{F}$, there is a bijection f between $S \setminus T$ and $T \setminus S$ such that $S \setminus \{e\} \cup \{f(e)\}$ is in $\mathcal{F}$ for every $e \in S \setminus T$.

Mechanism design for cheapest path auctions

Known results:

- **VCG** — Φ_{VCG} for example with long path and an edge is quadratic in terms of the optimal frugality ratio.
- **KKT $\sqrt{}$-mechanism** gives a linear approximation to the optimal frugality ratio ($\Phi_{\sqrt{}} \leq 2X$ and any truthful mechanism has frugality ratio at least $\frac{1}{\sqrt{2}}X$).
- **Pruning-lifting mechanism** (our paper) for k-path auctions provides optimal frugality ratio.

Mechanism design for cheapest path auctions

Known results:

- **VCG** — Φ_{VCG} for example with long path and an edge is quadratic in terms of the optimal frugality ratio.
- **KKT $\sqrt{}$-mechanism** gives a linear approximation to the optimal frugality ratio ($\Phi_{\sqrt{}} \leq 2X$ and any truthful mechanism has frugality ratio at least $\frac{1}{\sqrt{2}}X$).
- **Pruning-lifting mechanism** (our paper) for k-path auctions provides optimal frugality ratio.

Mechanism design for cheapest path auctions

Known results:

- **VCG** — Φ_{VCG} for example with long path and an edge is quadratic in terms of the optimal frugality ratio.
- **KKT $\sqrt{\ }$-mechanism** gives a linear approximation to the optimal frugality ratio ($\Phi_{\sqrt{\ }} \leq 2X$ and any truthful mechanism has frugality ratio at least $\frac{1}{\sqrt{2}}X$).
- **Pruning-lifting mechanism** (our paper) for k-path auctions provides optimal frugality ratio.

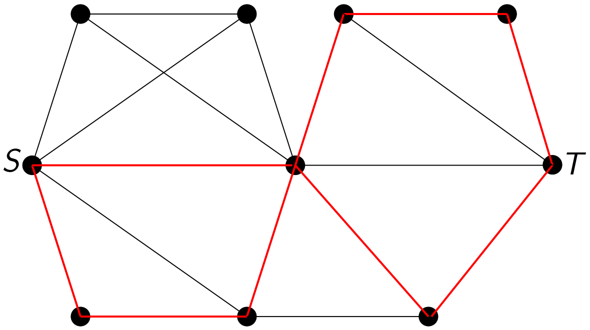
KKT mechanism for cheapest path

$\sqrt{}$-**mechanism** for cheapest path auctions.

We will see that for some graphs it will have much better frugality ratio than VCG.

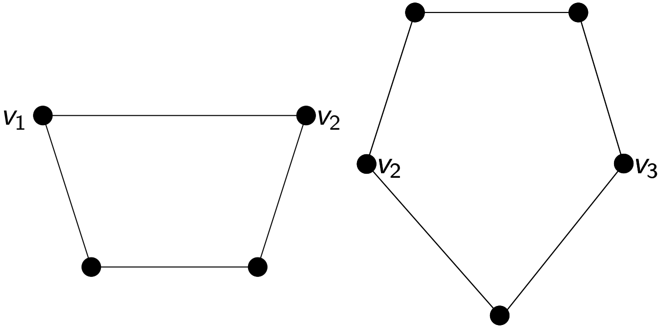
First step.

Find two edge-disjoint paths P, P' minimizing $b(P) + b(P')$.
(Ignore the rest of the graph)



Second step.

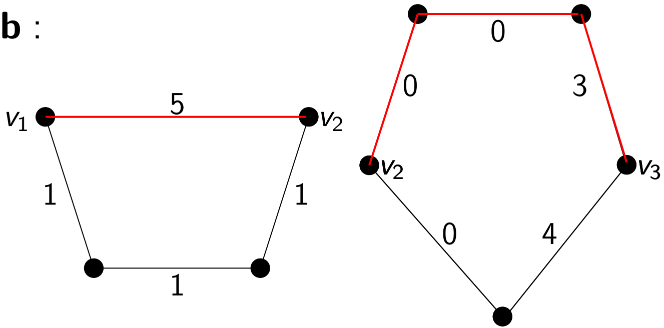
Let $s = v_1, v_2, \ldots, v_{k+1} = t$ be the vertices that P, P' have in common, in the order in which they appear in P and P'. Let P_i (resp. P'_i) be the subpath of P (resp. P') from v_i to v_{i+1}.



Third step.

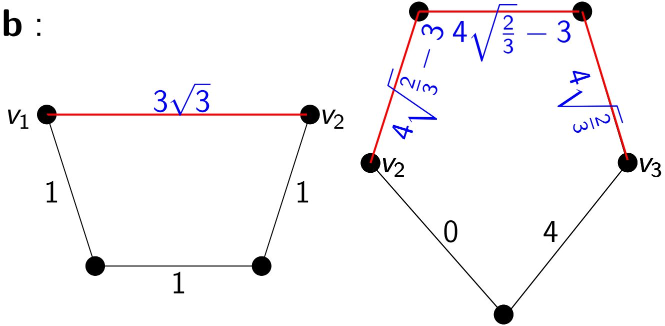
For each i, include P_i in the solution iff $\sqrt{|P_i|}b(P_i) \leq \sqrt{|P_i'|}b(P_i')$; otherwise, include P_i'.

b :



Forth step.

Pay to each winner its threshold bid (i.e. the largest value that he can bid and still win, if the others bid the same).



Bounds on $\Phi_{\sqrt{}}$

- $\sqrt{}$-mechanism is truthful, thus $\mathbf{b} = \mathbf{c}$.
- $\Phi_{\sqrt{}} \leq 2 \max_i \sqrt{|P_i||P'_i|}$.
- Thus for the edge and a path graph we have $\Phi_{\sqrt{}} = O(2\sqrt{l})$, while $\Phi_{VCG} = O(l)$.
- There is a lower bound $\frac{1}{\sqrt{2}} \max_i \sqrt{|P_i||P'_i|}$ on any truthful mechanism, thus

$$\frac{1}{\sqrt{2}} \max_i \sqrt{|P_i||P'_i|} \leq \Phi_{\sqrt{}} \leq 2 \max_i \sqrt{|P_i||P'_i|}.$$

Bounds on $\Phi_{\sqrt{}}$

- $\sqrt{}$-mechanism is truthful, thus $\mathbf{b} = \mathbf{c}$.
- $\Phi_{\sqrt{}} \leq 2\max_i \sqrt{|P_i||P_i'|}$.
- Thus for the edge and a path graph we have $\Phi_{\sqrt{}} = O(2\sqrt{l})$, while $\Phi_{VCG} = O(l)$.
- There is a lower bound $\frac{1}{\sqrt{2}}\max_i \sqrt{|P_i||P_i'|}$ on any truthful mechanism, thus

$$\frac{1}{\sqrt{2}}\max_i \sqrt{|P_i||P_i'|} \leq \Phi_{\sqrt{}} \leq 2\max_i \sqrt{|P_i||P_i'|}.$$

Bounds on $\Phi_{\sqrt{}}$

- $\sqrt{}$-mechanism is truthful, thus $\mathbf{b} = \mathbf{c}$.
- $\Phi_{\sqrt{}} \leq 2 \max_i \sqrt{|P_i||P_i'|}$.
- Thus for the edge and a path graph we have $\Phi_{\sqrt{}} = O(2\sqrt{l})$, while $\Phi_{VCG} = O(l)$.
- There is a lower bound $\frac{1}{\sqrt{2}} \max_i \sqrt{|P_i||P_i'|}$ on any truthful mechanism, thus

$$\frac{1}{\sqrt{2}} \max_i \sqrt{|P_i||P_i'|} \leq \Phi_{\sqrt{}} \leq 2 \max_i \sqrt{|P_i||P_i'|}.$$

Bounds on $\Phi_{\sqrt{\ }}$

- $\sqrt{\ }$-mechanism is truthful, thus $\mathbf{b} = \mathbf{c}$.
- $\Phi_{\sqrt{\ }} \leq 2 \max_i \sqrt{|P_i||P_i'|}$.
- Thus for the edge and a path graph we have $\Phi_{\sqrt{\ }} = O(2\sqrt{l})$, while $\Phi_{VCG} = O(l)$.
- There is a lower bound $\frac{1}{\sqrt{2}} \max_i \sqrt{|P_i||P_i'|}$ on any truthful mechanism, thus

$$\frac{1}{\sqrt{2}} \max_i \sqrt{|P_i||P_i'|} \leq \Phi_{\sqrt{\ }} \leq 2 \max_i \sqrt{|P_i||P_i'|}.$$

k-paths and cheapest path auctions

k-paths and cheapest path auctions

Nash for k-paths

Theorem

Any Nash equilibrium for the cheapest path auction with respect to the bid vector b should have two non-intersecting by the edges cheapest paths.

Theorem

Any Nash equilibrium for the k-paths auction with respect to the bid vector b should have $k+1$ non-intersecting by the edges cheapest paths.

These theorems are non trivial results of graph theory.

Nash for k-paths

Theorem

Any Nash equilibrium for the cheapest path auction with respect to the bid vector b should have two non-intersecting by the edges cheapest paths.

Theorem

Any Nash equilibrium for the k-paths auction with respect to the bid vector b should have $k+1$ non-intersecting by the edges cheapest paths.

These theorems are non trivial results of graph theory.

Nash for k-paths

Theorem

Any Nash equilibrium for the cheapest path auction with respect to the bid vector b should have two non-intersecting by the edges cheapest paths.

Theorem

Any Nash equilibrium for the k-paths auction with respect to the bid vector b should have $k+1$ non-intersecting by the edges cheapest paths.

These theorems are non trivial results of graph theory.

Our mechanism for k-path auctions

Pruning-lifting mechanism

Pruning step

- Pick $k+1$ edge-disjoint paths $P_1, \ldots, P_{k+1}$ in G such that

$$\delta(P_1, \ldots, P_{k+1}) = \max_{P \in \cup P_i} b(P)$$

  is minimized. (Ignore the rest of the graph.)
- It is NP hard to find $P_1, \ldots, P_{k+1}$ even with $< k+1$ factor approximation of $\delta(P_1, \ldots, P_{k+1})$ to the optimal one.
- The cheapest $k+1$ flow gives $k+1$ approximation.

Pruning step

- Pick $k+1$ edge-disjoint paths $P_1, \ldots, P_{k+1}$ in G such that

$$\delta(P_1, \ldots, P_{k+1}) = \max_{P \in \cup P_i} b(P)$$

is minimized. (Ignore the rest of the graph.)

- It is NP hard to find $P_1, \ldots, P_{k+1}$ even with $< k+1$ factor approximation of $\delta(P_1, \ldots, P_{k+1})$ to the optimal one.
- The cheapest $k+1$ flow gives $k+1$ approximation.

Pruning step

- Pick $k+1$ edge-disjoint paths $P_1, \ldots, P_{k+1}$ in G such that

$$\delta(P_1, \ldots, P_{k+1}) = \max_{P \in \cup P_i} b(P)$$

  is minimized. (Ignore the rest of the graph.)
- It is NP hard to find $P_1, \ldots, P_{k+1}$ even with $< k+1$ factor approximation of $\delta(P_1, \ldots, P_{k+1})$ to the optimal one.
- The cheapest $k+1$ flow gives $k+1$ approximation.

Lifting step

We construct a graph $\mathcal{H}$ as follows:

- we take arcs of $\cup_i P_i$ as vertices for $\mathcal{H}$;
- we draw an edge between e and e' iff there is no path from s to t containing both e, e'.

Lifting step

We construct a graph $\mathcal{H}$ as follows:

- we take arcs of $\cup_i P_i$ as vertices for $\mathcal{H}$;
- we draw an edge between e and e' iff there is no path from s to t containing both e, e'.

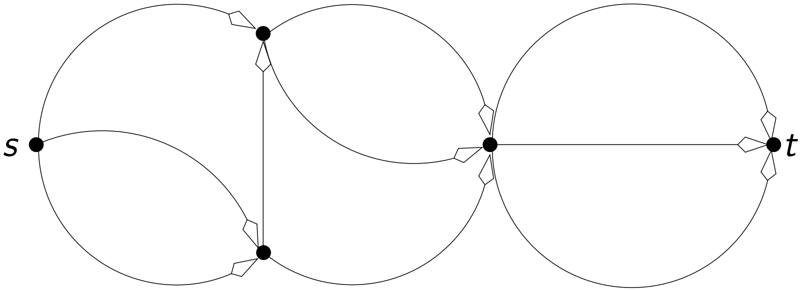
Lifting step

We construct a graph $\mathcal{H}$ as follows:

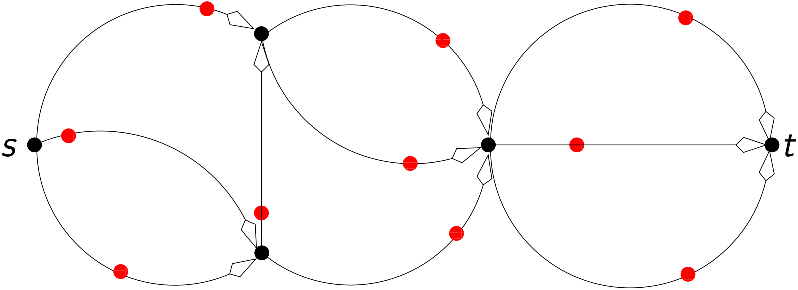
- we take arcs of $\cup_i P_i$ as vertices for $\mathcal{H}$;
- we draw an edge between e and e' iff there is no path from s to t containing both e, e'.

Lifting step

$\bigcup P_i$

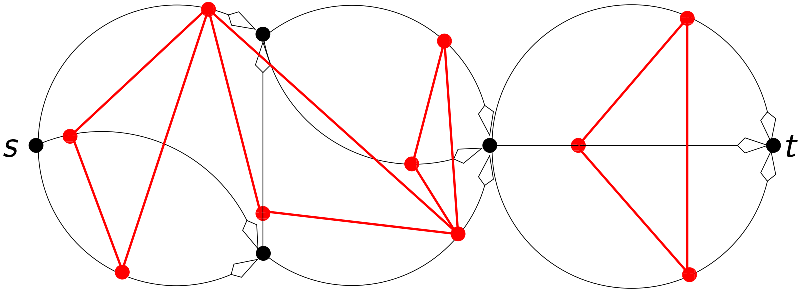


Lifting step



Lifting step

$\mathcal{H}$



Weighting step

- We split $\mathcal{H}$ into its components of connectivity $\mathcal{H}_1, \ldots, \mathcal{H}_l$. Let A_i be an adjacency matrix of $\mathcal{H}_i$.
- For each i we find the positive eigenvector (w_i) and eigenvalue α_i of the matrix A_i. Let $\alpha = \max_i \alpha_i$.

$$(A_i)(w_i) = \alpha_i (w_i)$$

- Define $b'(e) = \frac{b(e)}{w(e)}$. Let P the maximum weight path with respec to $b'()$. Take $\cup P_i \setminus P$ as the set of winners.
- Pay to each agent its threshold bid.

Weighting step

- We split $\mathcal{H}$ into its components of connectivity $\mathcal{H}_1, \ldots, \mathcal{H}_l$. Let A_i be an adjacency matrix of $\mathcal{H}_i$.
- For each i we find the positive eigenvector (w_i) and eigenvalue α_i of the matrix A_i. Let $\alpha = \max_i \alpha_i$.

$$(A_i)\,(w_i) = \alpha_i\,(w_i)$$

- Define $b'(e) = \frac{b(e)}{w(e)}$. Let P the maximum weight path with respec to $b'()$. Take $\cup P_i \setminus P$ as the set of winners.
- Pay to each agent its threshold bid.

Weighting step

- We split $\mathcal{H}$ into its components of connectivity $\mathcal{H}_1, \ldots, \mathcal{H}_l$. Let A_i be an adjacency matrix of $\mathcal{H}_i$.
- For each i we find the positive eigenvector (w_i) and eigenvalue α_i of the matrix A_i. Let $\alpha = \max_i \alpha_i$.

$$(A_i)\,(w_i) = \alpha_i\,(w_i)$$

- Define $b'(e) = \frac{b(e)}{w(e)}$. Let P the maximum weight path with respec to $b'()$. Take $\cup P_i \setminus P$ as the set of winners.
- Pay to each agent its threshold bid.

Weighting step

- We split $\mathcal{H}$ into its components of connectivity $\mathcal{H}_1, \ldots, \mathcal{H}_l$. Let A_i be an adjacency matrix of $\mathcal{H}_i$.
- For each i we find the positive eigenvector (w_i) and eigenvalue α_i of the matrix A_i. Let $\alpha = \max_i \alpha_i$.

$$(A_i)\,(w_i) = \alpha_i\,(w_i)$$

- Define $b'(e) = \frac{b(e)}{w(e)}$. Let P the maximum weight path with respec to $b'()$. Take $\cup P_i \setminus P$ as the set of winners.
- Pay to each agent its threshold bid.

Φ_{PL}

$$\Phi_{PL} = \frac{\alpha}{k},$$

where $\frac{\alpha}{k}$ is a lower bound on the frugality ratio of any truthful mechanism.

So PL is the optimal mechanism with respect to frugality ratio.

Φ_{PL}

$$\Phi_{PL} = \frac{\alpha}{k},$$

where $\frac{\alpha}{k}$ is a lower bound on the frugality ratio of any truthful mechanism.

So PL is the optimal mechanism with respect to frugality ratio.