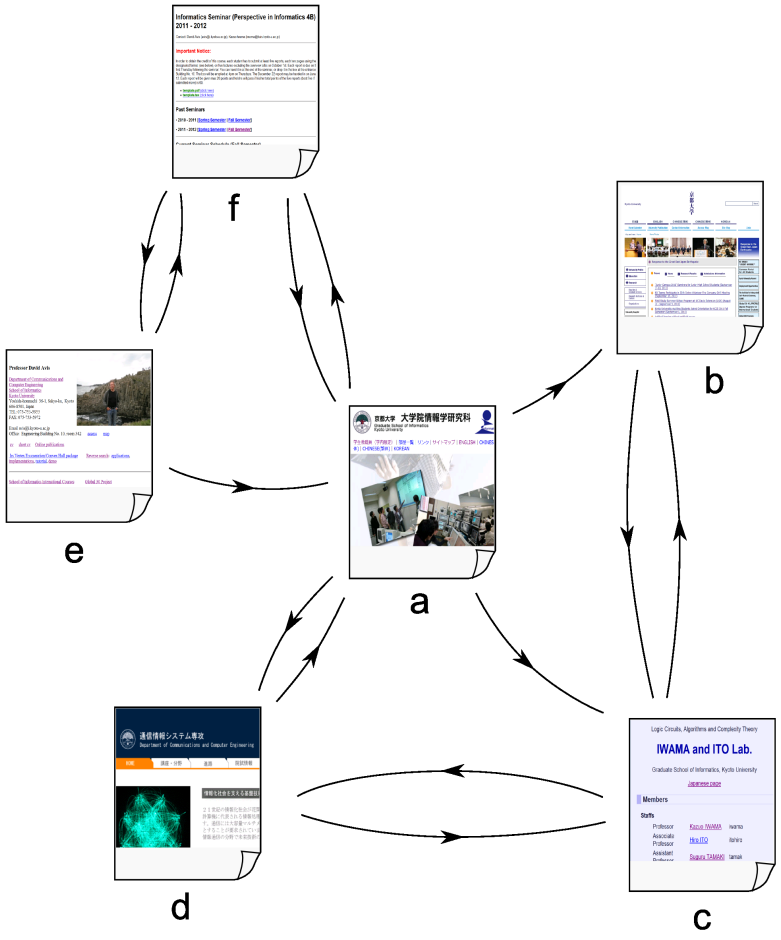
# Verifying Nash Equilibria in PageRank Games on Undirected Web Graphs

David Avis, Kazuo Iwama and Daichi Paku

ISAAC 2011, Yokohama

December 7, 2011

# Directed graphs: WWW



# $\alpha$-random walks

- Given: digraph $G = (V, E)$, probability distribution $q$ on $V$ and $0 < \alpha < 1$.

# $\alpha$-random walks

- Given: digraph $G = (V, E)$, probability distribution $q$ on $V$ and $0 < \alpha < 1$.
- Choose a starting vertex $v$ according to distribution $q$.

# $\alpha$-random walks

- Given: digraph $G = (V, E)$, probability distribution $q$ on $V$ and $0 < \alpha < 1$.
- Choose a starting vertex $v$ according to distribution $q$.
- With probability $1 - \alpha$ choose a random edge $vw$ and go to $w$.

# $\alpha$-random walks

- Given: digraph $G = (V, E)$, probability distribution $q$ on $V$ and $0 < \alpha < 1$.
- Choose a starting vertex $v$ according to distribution $q$.
- With probability $1 - \alpha$ choose a random edge $vw$ and go to $w$.
- With probability $\alpha$ jump to a vertex $u$ chosen according to $q$.

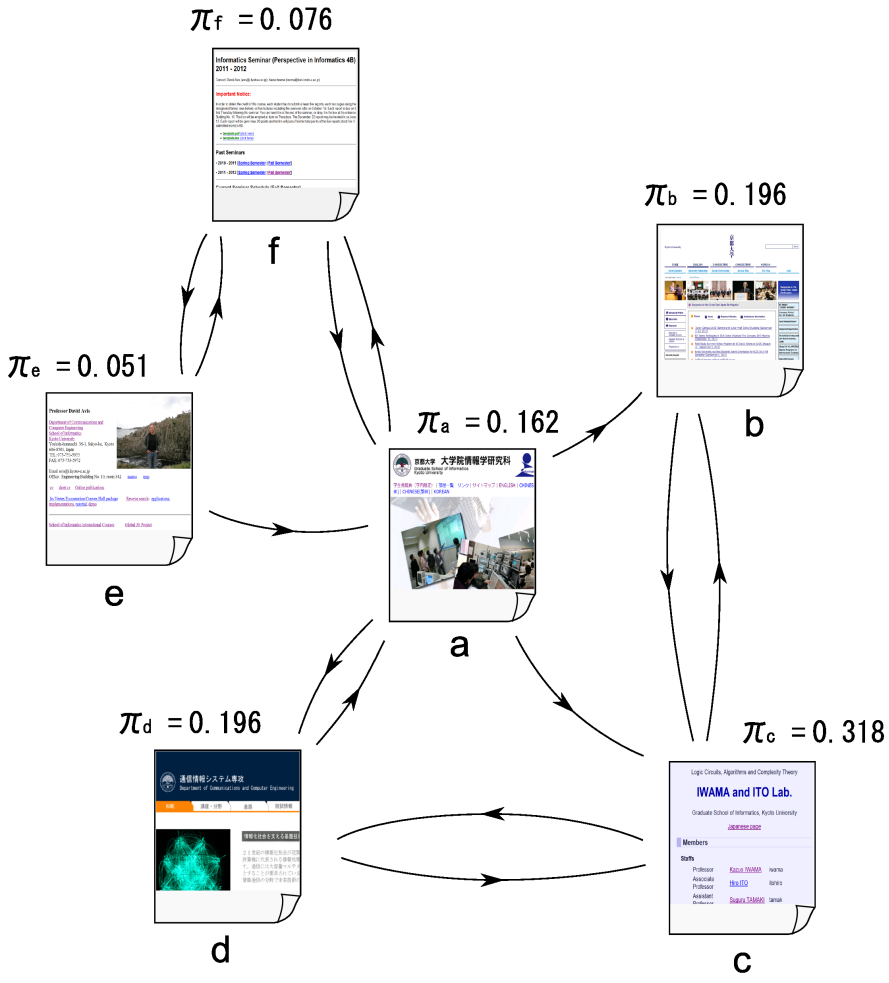
# $\alpha$-random walks

- Given: digraph $G = (V, E)$, probability distribution $q$ on $V$ and $0 < \alpha < 1$.
- Choose a starting vertex $v$ according to distribution $q$.
- With probability $1 - \alpha$ choose a random edge $vw$ and go to $w$.
- With probability $\alpha$ jump to a vertex $u$ chosen according to $q$.
- PageRank is the stationary distribution $\pi$ on $V$.

# $\alpha$-random walks

- Given: digraph $G = (V, E)$, probability distribution $q$ on $V$ and $0 < \alpha < 1$.
- Choose a starting vertex $v$ according to distribution $q$.
- With probability $1 - \alpha$ choose a random edge $vw$ and go to $w$.
- With probability $\alpha$ jump to a vertex $u$ chosen according to $q$.
- PageRank is the stationary distribution $\pi$ on $V$.
- In this talk we assume $q$ is uniform.

# PageRank



# Computing PageRank

- Potential:

$$\begin{aligned}\phi_{uv} &= P\{\alpha \textit{ random walk from } u \textit{ visits } v \textit{ before jumping}\} \\ &= \frac{1-\alpha}{|\Gamma(u)|}\sum_{i\in\Gamma(u)}\phi_{iv}.\end{aligned}$$

where $\Gamma(u)$ are the out neighbours of $u$.

# Computing PageRank

- Potential:

$$\begin{aligned}\phi_{uv} &= P\{\alpha \textit{ random walk from } u \textit{ visits } v \textit{ before jumping}\} \\ &= \frac{1-\alpha}{|\Gamma(u)|}\sum_{i\in\Gamma(u)}\phi_{iv}.\end{aligned}$$

  where $\Gamma(u)$ are the out neighbours of $u$.

- PageRank = stationary distribution $\pi$

$$\pi_v = \alpha\frac{\sum_{u\in V} q_u\phi_{uv}}{1-\frac{(1-\alpha)}{|\Gamma(v)|}\sum_{i\in\Gamma(v)}\phi_{iv}}.$$

# Directed PageRank games: Hopcroft-Sheldon '08

- Any player=page can add or delete any outlinks.

# Directed PageRank games: Hopcroft-Sheldon '08

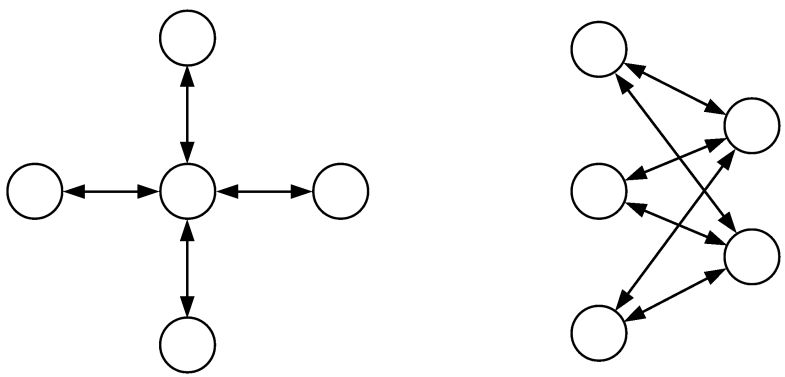
- Any player=page can add or delete any outlinks.
- Page $u$ is in best response if no add or delete improves its PageRank.

# Directed PageRank games: Hopcroft-Sheldon '08

- Any player=page can add or delete any outlinks.
- Page $u$ is in best response if no add or delete improves its PageRank.
- Graph $G$ is in Nash Equilibrium(NE) if each page is in best response.

# Directed PageRank games: Hopcroft-Sheldon '08

- Any player=page can add or delete any outlinks.
- Page $u$ is in best response if no add or delete improves its PageRank.
- Graph $G$ is in Nash Equilibrium(NE) if each page is in best response.
- $\alpha$-insensitive: NE for all $0 < \alpha < 1$.



(a) (b)

# Directed PageRank games: Hopcroft-Sheldon

$$\phi_{uv} = P\{\alpha \textit{ random walk from } u \textit{ visits } v \textit{ before jumping}\}$$

# Directed PageRank games: Hopcroft-Sheldon

$$\phi_{uv} = P\{\alpha \textit{ random walk from } u \textit{ visits } v \textit{ before jumping}\}$$

- A page $u$ is in best response if it links only to nodes $v$ that maximize $\phi_{uv}$

# Directed PageRank games: Hopcroft-Sheldon

$$\phi_{uv} = P\{\alpha \textit{ random walk from } u \textit{ visits } v \textit{ before jumping}\}$$

- A page $u$ is in best response if it links only to nodes $v$ that maximize $\phi_{uv}$
- If a strongly connected graph $G$ is a NE all edges are bidirected.

# Directed PageRank games: Hopcroft-Sheldon

$$\phi_{uv} = P\{\alpha \textit{ random walk from } u \textit{ visits } v \textit{ before jumping}\}$$

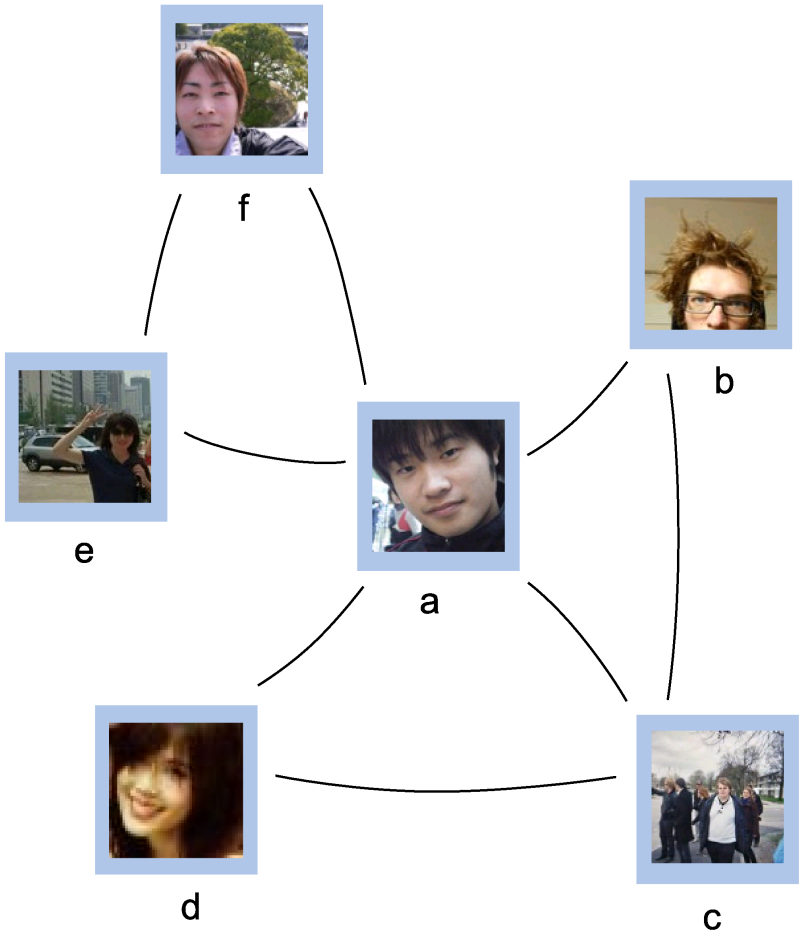
- A page $u$ is in best response if it links only to nodes $v$ that maximize $\phi_{uv}$
- If a strongly connected graph $G$ is a NE all edges are bidirected.
- $\alpha$-sensitive NE exist (Chen et al.'09)

# Undirected graphs

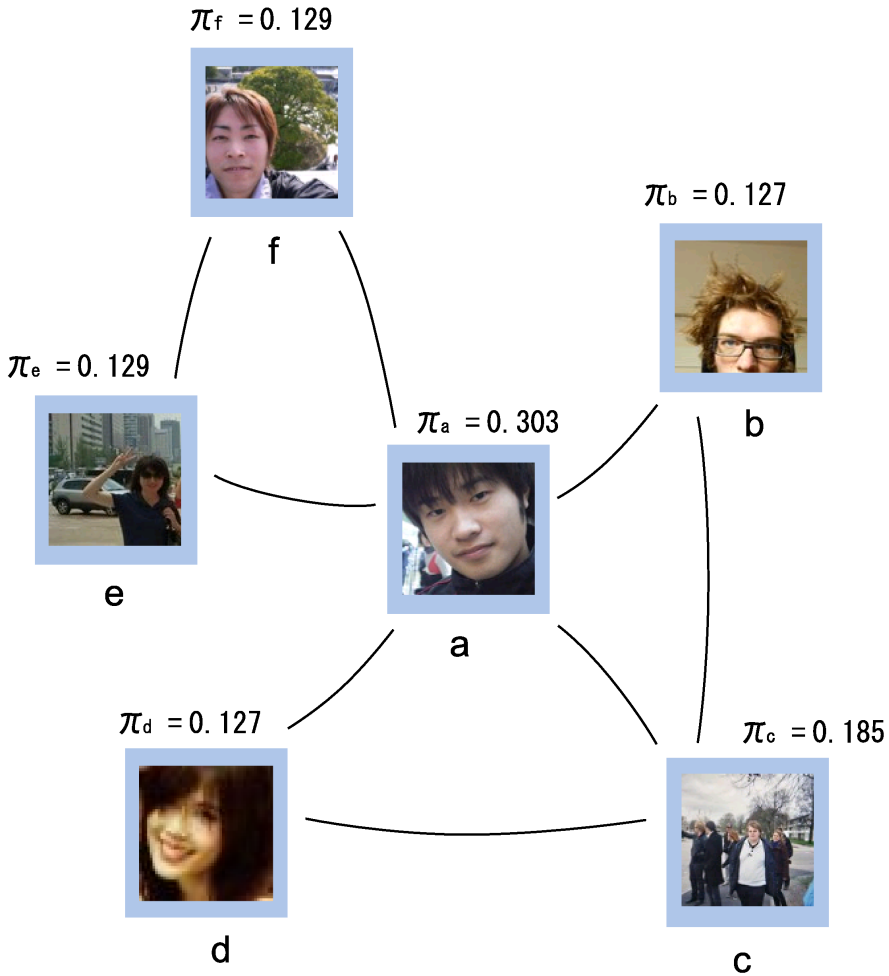
We study PageRank for undirected graphs, such as:

- Friends on Facebook
- Co-authorship (Erdös graph)
- Bilateral agreements

# Undirected graphs: Facebook



# Undirected graphs: PageRank

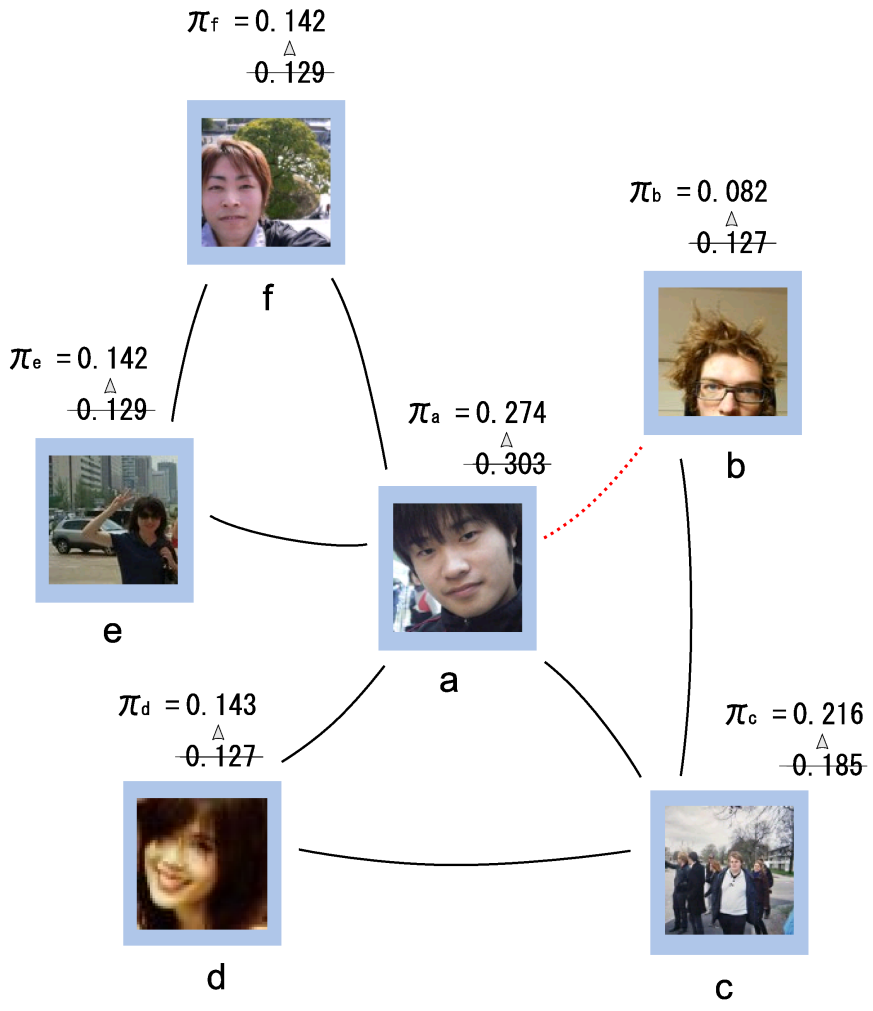


# Undirected PageRank games - 1

- A player=page can delete any of its edges.

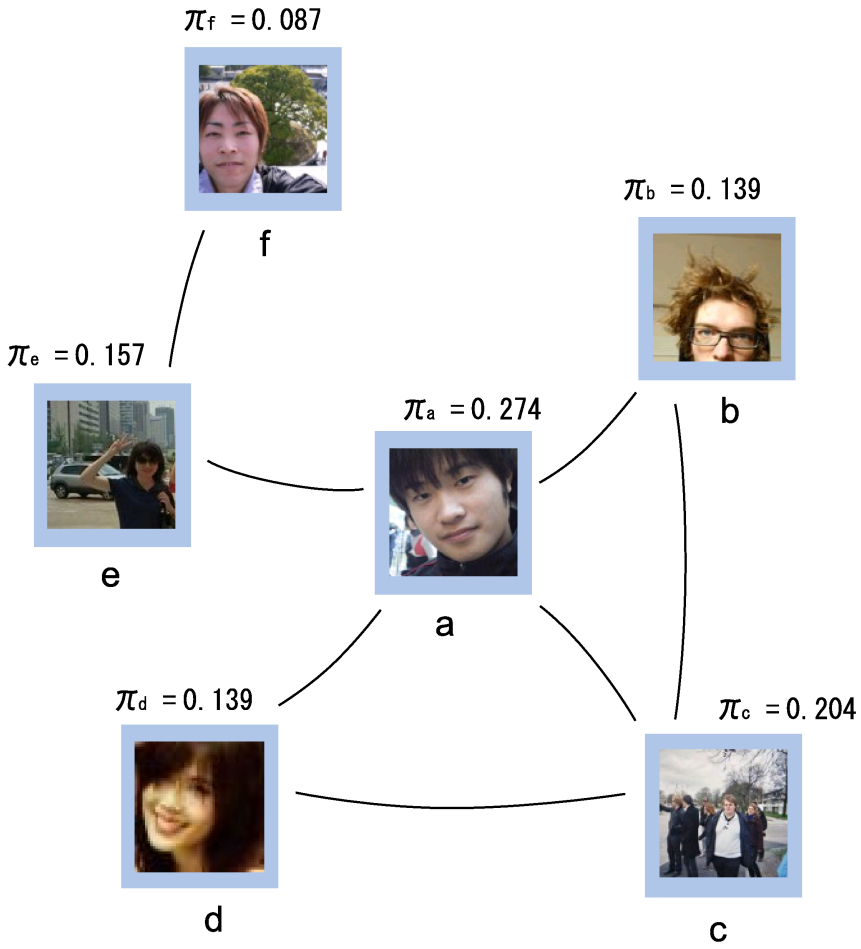
# Undirected PageRank games - 1

- A player=page can delete any of its edges.



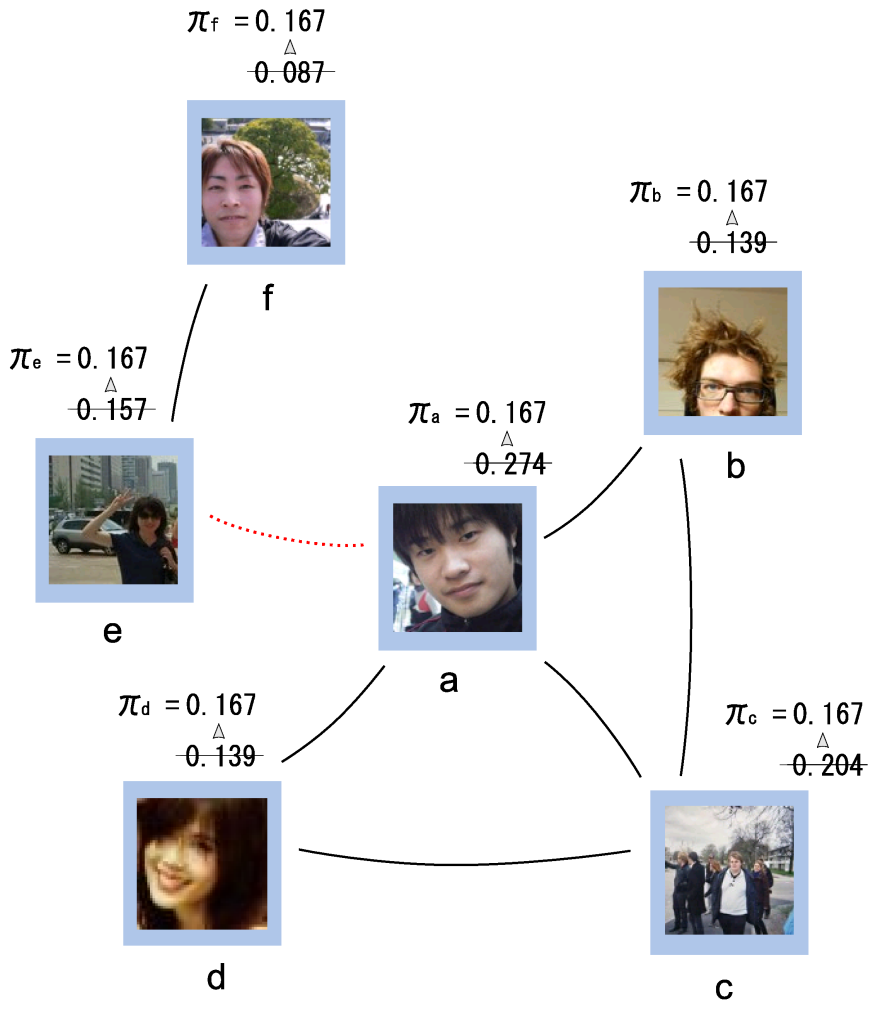
# Undirected PageRank games - 2

- A player=page can delete any of its edges.



# Undirected PageRank games - 3

- A player=page can delete any of its edges.

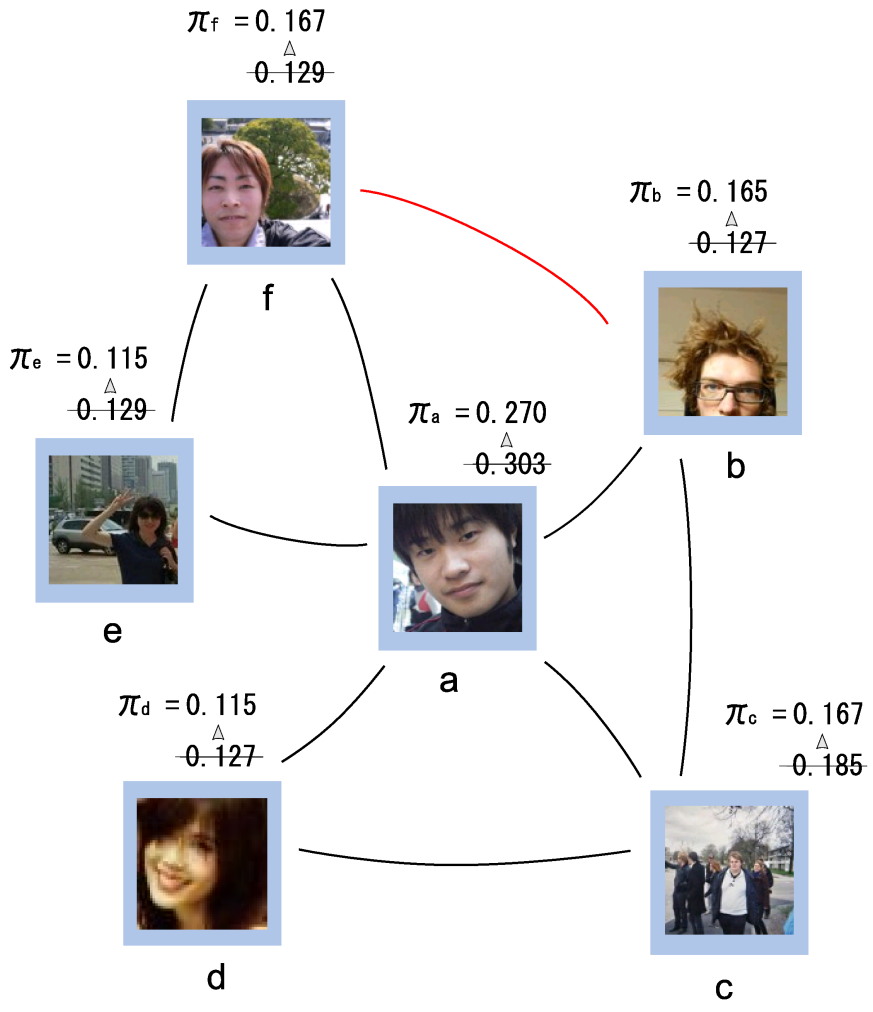


# Undirected PageRank games - 4

- A player=page can add an edge if the other page agrees.

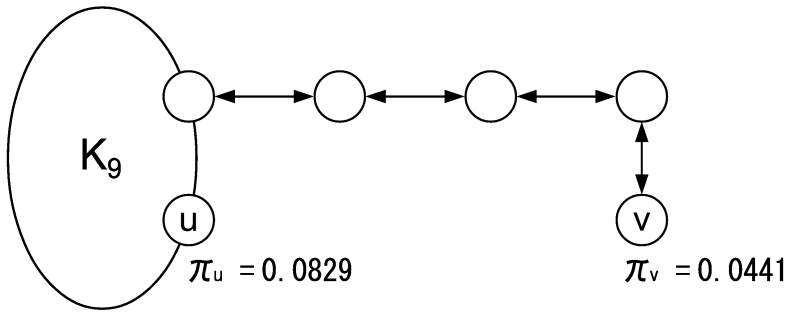
# Undirected PageRank games - 4

- A player=page can add an edge if the other page agrees.



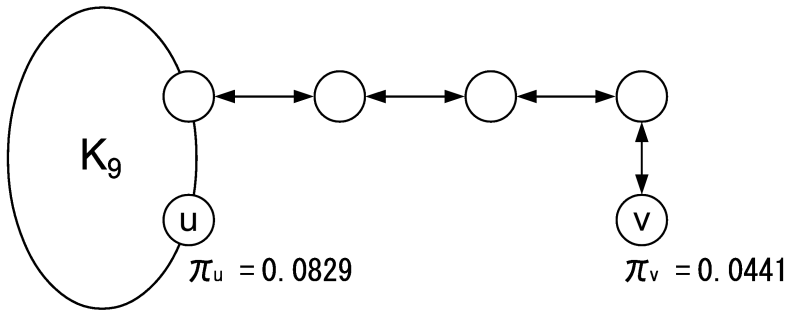
# Undirected PageRank games - 5

- $u$ tries to add edge $uv$ ...

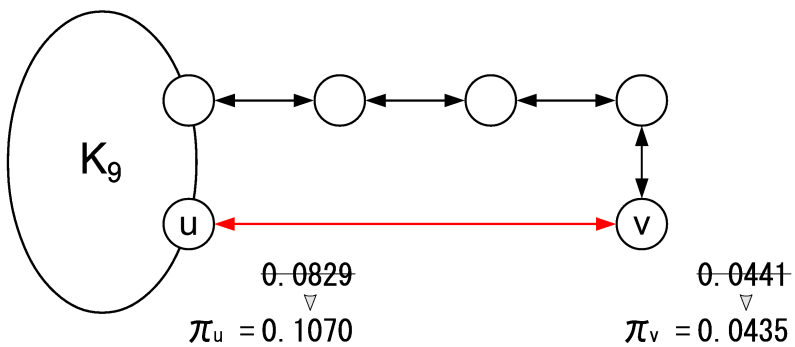


# Undirected PageRank games - 5

- $u$ tries to add edge $uv$ ...




- ... but $v$ refuses!



# Undirected PageRank games - 6

- The addition of an edge requires the other page's approval.

# Undirected PageRank games - 6

- The addition of an edge requires the other page's approval.
- We consider only edge deletions in this talk.

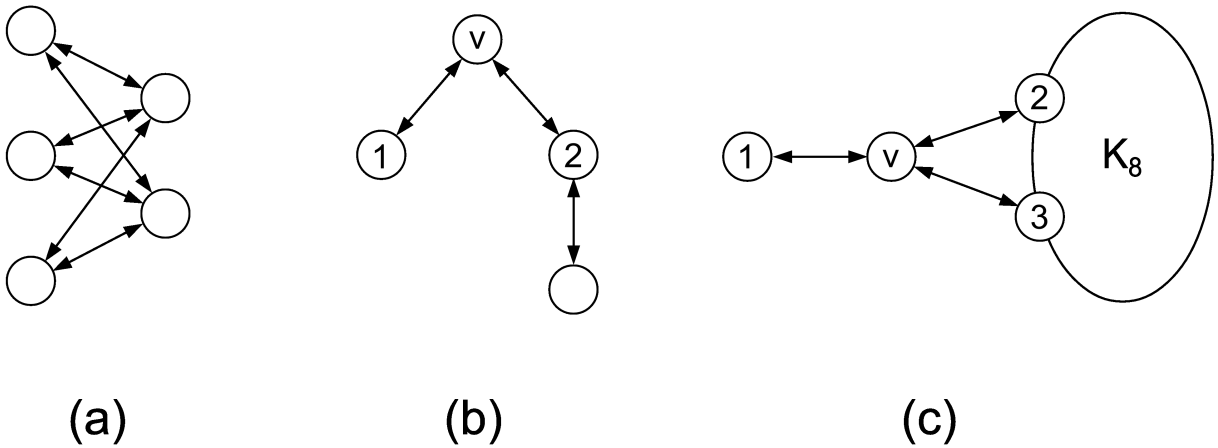
# Undirected PageRank games - 6

- The addition of an edge requires the other page's approval.
- We consider only edge deletions in this talk.
- A page is in best response if no deletion improves its PageRank.

# Undirected PageRank games - 6

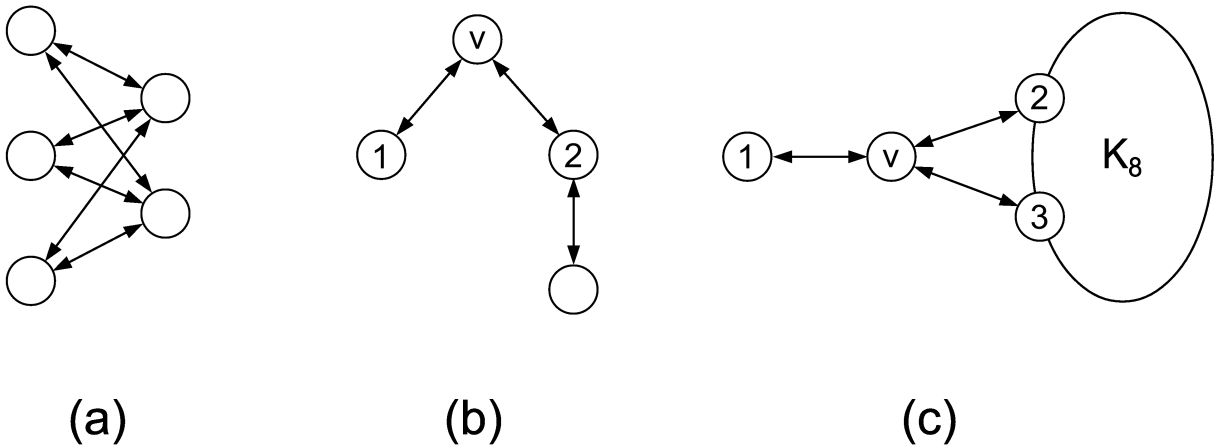
- The addition of an edge requires the other page's approval.
- We consider only edge deletions in this talk.
- A page is in best response if no deletion improves its PageRank.
- Graph $G$ is in Nash Equilibrium(NE) if each page is in best response.

# Directed vs Undirected NE



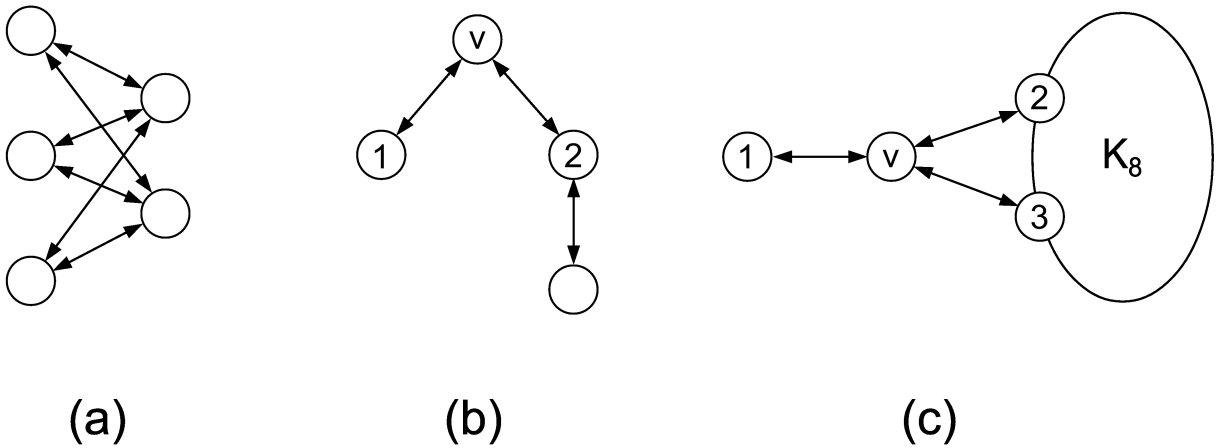

- (a) $\alpha$-insensitive NE in both models

# Directed vs Undirected NE




- (a) $\alpha$-insensitive NE in both models
- (b) Undirected: $\alpha$-insensitive NE
  Directed: delete v2

# Directed vs Undirected NE




- (a) $\alpha$-insensitive NE in both models
- (b) Undirected: $\alpha$-insensitive NE
  Directed: delete v2
- (c) Undirected: NE only for $4/7 \leq \alpha < 1$
  else delete both v2 and v3
  Directed: Not NE

# Our results

Input: An undirected graph $G$, $\alpha$, $q$.

Output: Is the input a Nash equilibrium in the deletions only model? (yes/no)

# Our results

Input: An undirected graph $G$, $\alpha$, $q$.

Output: Is the input a Nash equilibrium in the deletions only model? (yes/no)

- $G$ is a tree: $O(n^2)$ algorithm.

# Our results

Input: An undirected graph $G$, $\alpha$, $q$.

Output: Is the input a Nash equilibrium in the deletions only model? (yes/no)

- $G$ is a tree: $O(n^2)$ algorithm.
- Let $k = k(G)$ the max vertex degree in any biconnected component

# Our results

Input: An undirected graph $G$, $\alpha$, $q$.

Output: Is the input a Nash equilibrium in the deletions only model? (yes/no)

- $G$ is a tree: $O(n^2)$ algorithm.
- Let $k = k(G)$ the max vertex degree in any biconnected component
- $G$ is a graph: $O(2^k n^4)$ algorithm.

# Algorithm for trees

- A strategy for page $v$ is a $|\Gamma(v)|$-vector $x$ where $x_i = 1$ if $i$-th edge retained, else $x_i = 0$.

# Algorithm for trees

- A strategy for page $v$ is a $|\Gamma(v)|$-vector $x$ where $x_i = 1$ if $i$-th edge retained, else $x_i = 0$.
- The PageRank for $v$ under strategy $x$ is

$$\pi_v(x) = \alpha \frac{\sum_{u \in V} q_u \phi_{uv}(x)}{1 - (1-\alpha)\frac{\sum_{i \in \Gamma(v)} \phi_{iv}(x) x_i}{\mathbf{1}^T x}}$$

# Algorithm for trees

- A strategy for page $v$ is a $|\Gamma(v)|$-vector $x$ where $x_i = 1$ if $i$-th edge retained, else $x_i = 0$.
- The PageRank for $v$ under strategy $x$ is

$$\pi_v(x) = \alpha \frac{\sum_{u \in V} q_u \phi_{uv}(x)}{1 - (1-\alpha)\frac{\sum_{i \in \Gamma(v)} \phi_{iv}(x) x_i}{\mathbf{1}^T x}}$$

- For known vectors $a \geq 0$ and $b \geq 0$ we can rewrite as:

$$\pi_v(x) = \mathbf{1}^T x \ \frac{a^T x}{b^T x},$$

# Algorithm for trees

- A strategy for page $v$ is a $|\Gamma(v)|$-vector $x$ where $x_i = 1$ if $i$-th edge retained, else $x_i = 0$.
- The PageRank for $v$ under strategy $x$ is

$$\pi_v(x) = \alpha \frac{\sum_{u \in V} q_u \phi_{uv}(x)}{1 - (1-\alpha)\frac{\sum_{i \in \Gamma(v)} \phi_{iv}(x) x_i}{\mathbf{1}^T x}}$$

- For known vectors $a \geq 0$ and $b \geq 0$ we can rewrite as:

$$\pi_v(x) = \mathbf{1}^T x \ \frac{a^T x}{b^T x},$$

- $G$ is a NE iff for any strategy $x$

$$\pi_v(\mathbf{1}) \geq \pi_v(x)$$

# Fractional programming (Megiddo's method)

- To solve:

$$P : \max \quad \pi_v(x) = \mathbf{1}^T x \, \frac{a^T x}{b^T x}, \quad x \in \{0,1\}^n,$$

# Fractional programming (Megiddo's method)

- To solve:

$$P : \max \quad \pi_v(x) = \mathbf{1}^T x \, \frac{a^T x}{b^T x}, \quad x \in \{0,1\}^n,$$

- We solve for $l = 1, ..., |\Gamma(v)|$:

$$Q : \max \quad f(x) = \frac{a^T x}{b^T x} \qquad \text{s.t.} \quad \mathbf{1}^T x = l.$$

# Fractional programming (Megiddo's method)

- To solve:

$$P : \max \quad \pi_v(x) = \mathbf{1}^T x \, \frac{a^T x}{b^T x}, \quad x \in \{0,1\}^n,$$

- We solve for $l = 1, ..., |\Gamma(v)|$:

$$Q : \max \quad f(x) = \frac{a^T x}{b^T x} \qquad \text{s.t.} \quad \mathbf{1}^T x = l.$$

- For fixed $\delta > 0$ let

$$h(\delta) = \max \quad g(x) = (a - b\delta)^T x \qquad \text{s.t.} \quad \mathbf{1}^T x = l.$$

# Fractional programming (Megiddo's method)

- To solve:

$$P : \max \quad \pi_v(x) = \mathbf{1}^T x \, \frac{a^T x}{b^T x}, \quad x \in \{0,1\}^n,$$

- We solve for $l = 1, ..., |\Gamma(v)|$:

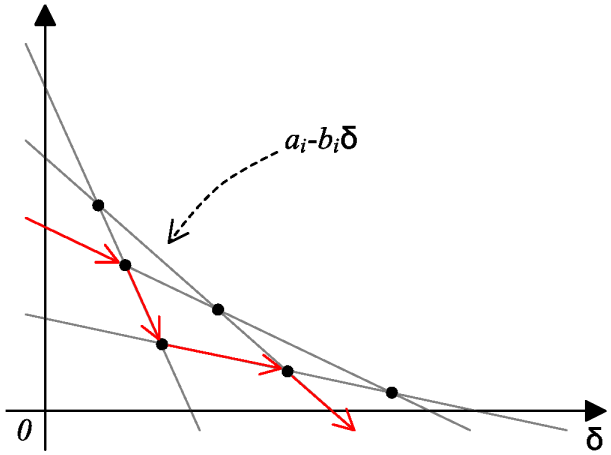
$$Q : \max \quad f(x) = \frac{a^T x}{b^T x} \qquad \text{s.t.} \quad \mathbf{1}^T x = l.$$

- For fixed $\delta > 0$ let

$$h(\delta) = \max \quad g(x) = (a - b\delta)^T x \qquad \text{s.t.} \quad \mathbf{1}^T x = l.$$

- $z^*$ is the optimum solution to $Q$ iff $h(z^*) = 0$.

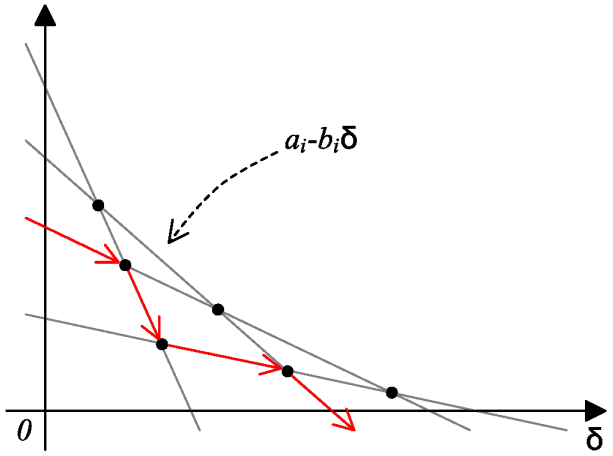
# Root finding




- Find the root $h(z^*) = 0$ of

$$h(\delta) = \max \quad g(x) = (a - b\delta)^T x \qquad \text{s.t.} \quad \mathbf{1}^T x = I.$$

# Root finding

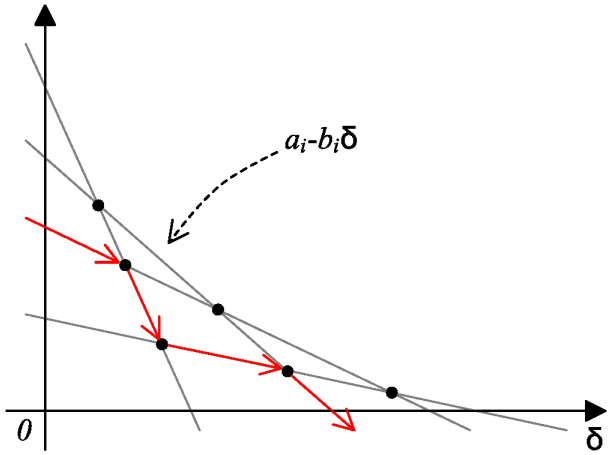



- Find the root $h(z^*) = 0$ of

$$h(\delta) = \max \quad g(x) = (a - b\delta)^T x \qquad \text{s.t.} \quad \mathbf{1}^T x = l.$$

- The root lies on the $l$-th layer of the line arrangement.

# Root finding

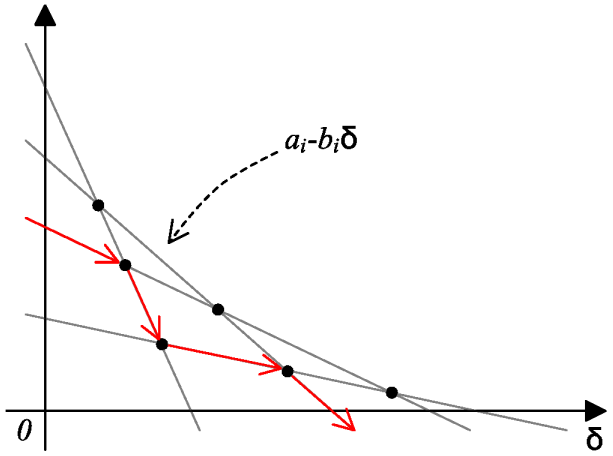



- Find the root $h(z^*) = 0$ of

$$h(\delta) = \max \quad g(x) = (a - b\delta)^T x \qquad \text{s.t.} \quad \mathbf{1}^T x = l.$$

- The root lies on the $l$-th layer of the line arrangement.
- We find all roots in $O(|\Gamma(v)|^2)$-time for given vertex $v$

# Root finding




- Find the root $h(z^*) = 0$ of

$$h(\delta) = \max \quad g(x) = (a - b\delta)^T x \qquad \text{s.t.} \quad \mathbf{1}^T x = l.$$

- The root lies on the $l$-th layer of the line arrangement.
- We find all roots in $O(|\Gamma(v)|^2)$-time for given vertex $v$
- Since $G$ is a tree we have

$$\sum_{v \in V} (|\Gamma(v)|^2 \leq 4n^2$$

# Future work

- Is there a polynomial time algorithm for general graphs?

# Future work

- Is there a polynomial time algorithm for general graphs?
- Is there a characterization of graphs which are NE?

# Future work

- Is there a polynomial time algorithm for general graphs?
- Is there a characterization of graphs which are NE?
- How about edge additions?