Algorithms for Decision Support

Introduction to Algorithmic Game Theory

Outline

- Fundamental concepts
  - Game, players, strategies, payoffs/costs
- Nash Equilibrium
- Price of Anarchy
  - Selfish load balancing
- Mechanism design
  - Auction
  - Vickrey-Clarke-Groves mechanism

Prisoner’s Dilemma

- Two prisoners A and B are on trial for a crime

Prisoner’s Dilemma

- Two prisoners A and B are on trial for a crime
  - Each of them faces a choice of confessing to the crime or remaining silent

A B	confess	silent
confess		
silent		

Prisoner’s Dilemma

- Two prisoners A and B are on trial for a crime
  - Each of them faces a choice of confessing to the crime or remaining silent
    - If both remain silent, they both serve a short prison term (2 years)

A B	confess	silent
confess		
silent		2 2

Prisoner’s Dilemma

- Two prisoners A and B are on trial for a crime
    - Each of them faces a choice of confessing to the crime or remaining silent
        - If both remain silent, they both serve a short prison term (2 years)
        - If only one of them confesses, his term will be reduced to 1 year and the other get a sentence of 5 years

B \ A	confess	silent
confess		5 1
silent	1 5	

Prisoner’s Dilemma

- Two prisoners A and B are on trial for a crime
  - Each of them faces a choice of confessing to the crime or remaining silent
    - If both remain silent, they both serve a short prison term (2 years)
    - If only one of them confesses, his term will be reduced to 1 year and the other get a sentence of 5 years
    - If both confess, they both serve prison sentence of 4 years

B \ A	confess	silent
confess	4 4	
silent		

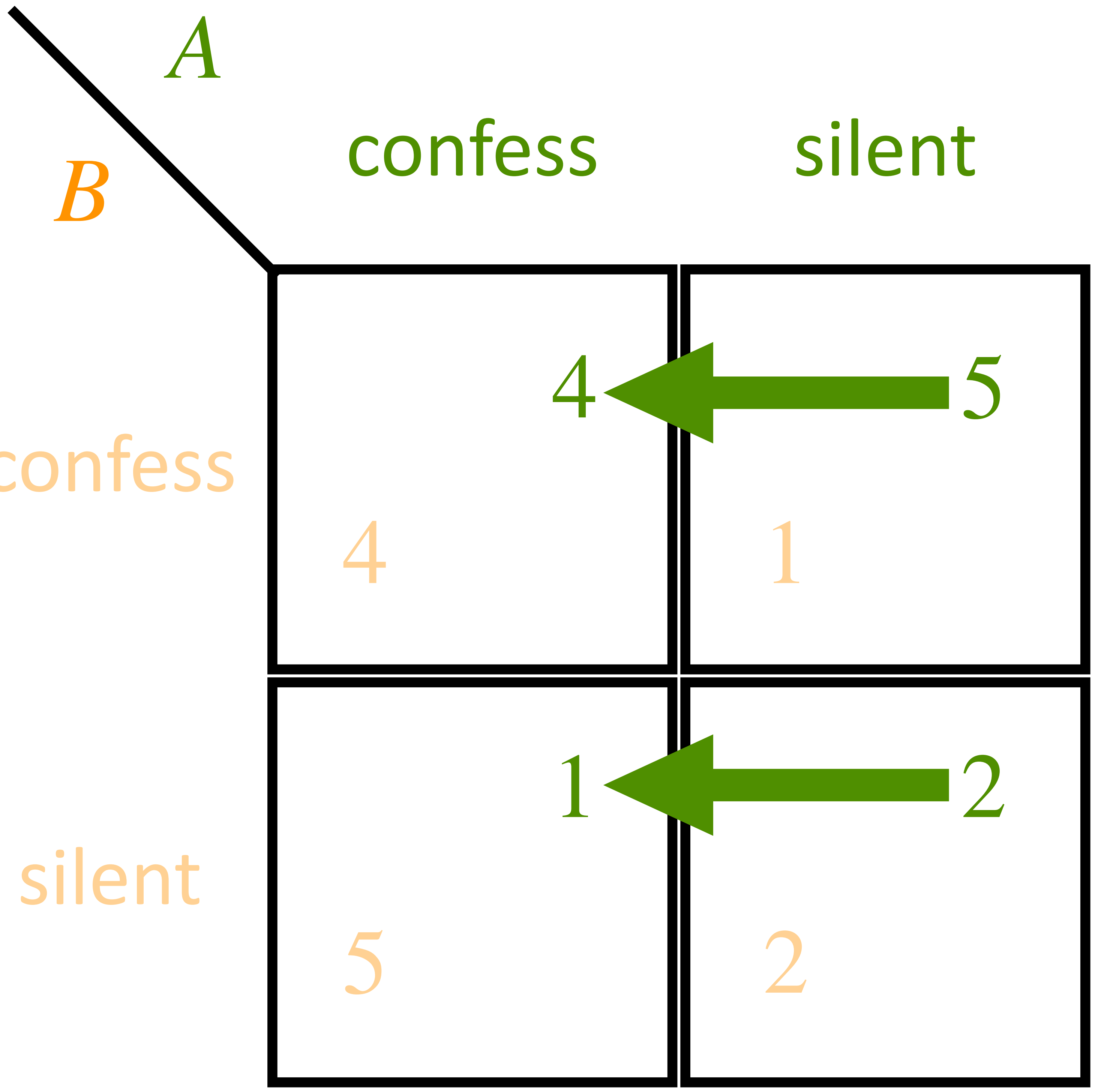
Prisoner’s Dilemma

- Two prisoners A and B are on trial for a crime
    - Each of them faces a choice of confessing to the crime or remaining silent
        - If both remain silent, they both serve a short prison term (2 years)
        - If only one of them confesses, his term will be reduced to 1 year and the other get a sentence of 5 years
        - If both confess, they both serve prison sentence of 4 years

A B	confess	silent
confess	4 4	5 1
silent	1 5	2 2

Prisoner’s Dilemma

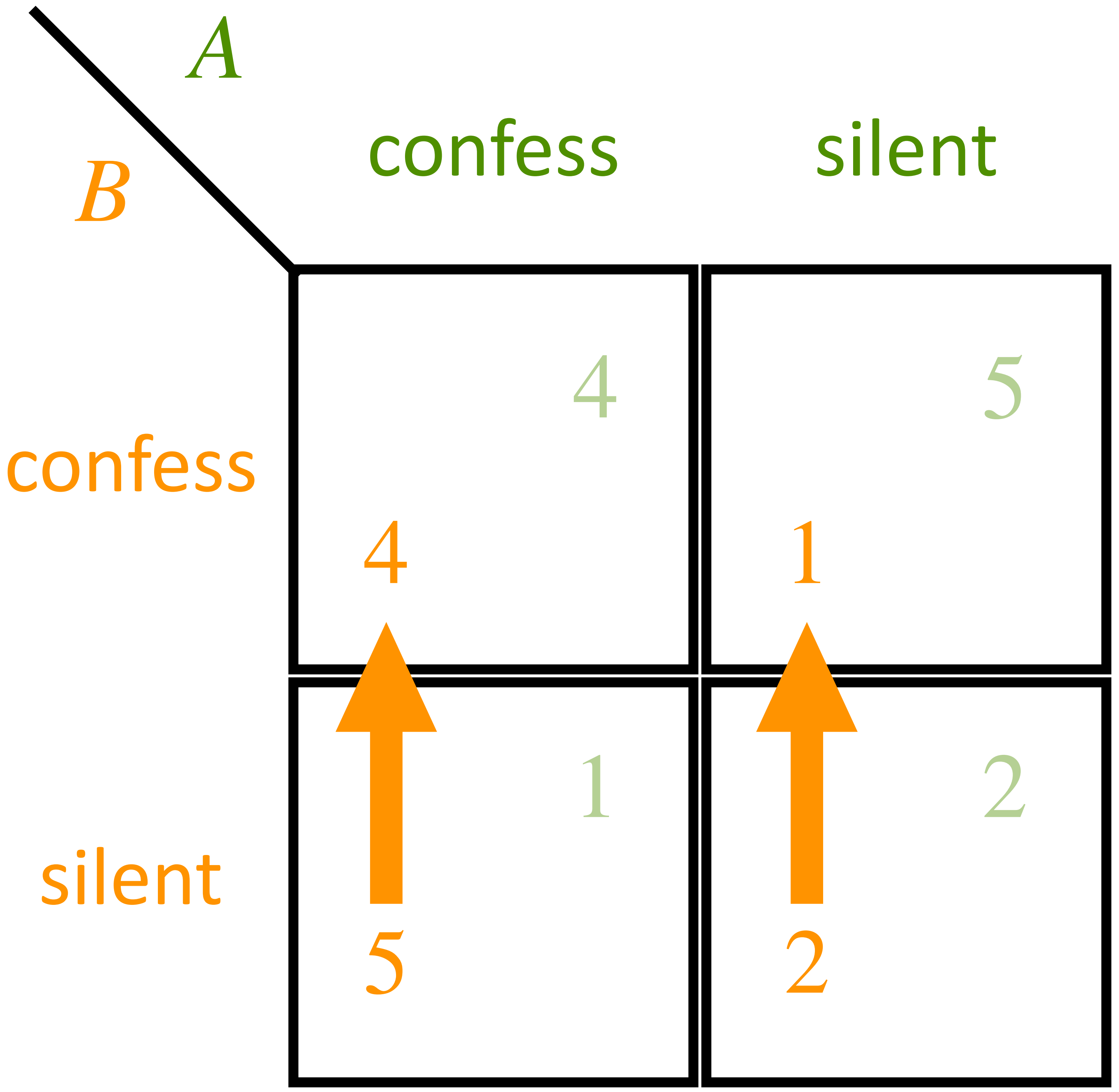
- Two prisoners A and B are on trial for a crime
  - Each of them faces a choice of confessing to the crime or remaining silent
    - If both remain silent, they both serve a short prison term (2 years)
    - If only one of them confesses, his term will be reduced to 1 year and the other get a sentence of 5 years
    - If both confess, they both serve prison sentence of 4 years



B \ A	confess	silent
confess	4 4	5 1
silent	1 5	2 2

Prisoner’s Dilemma

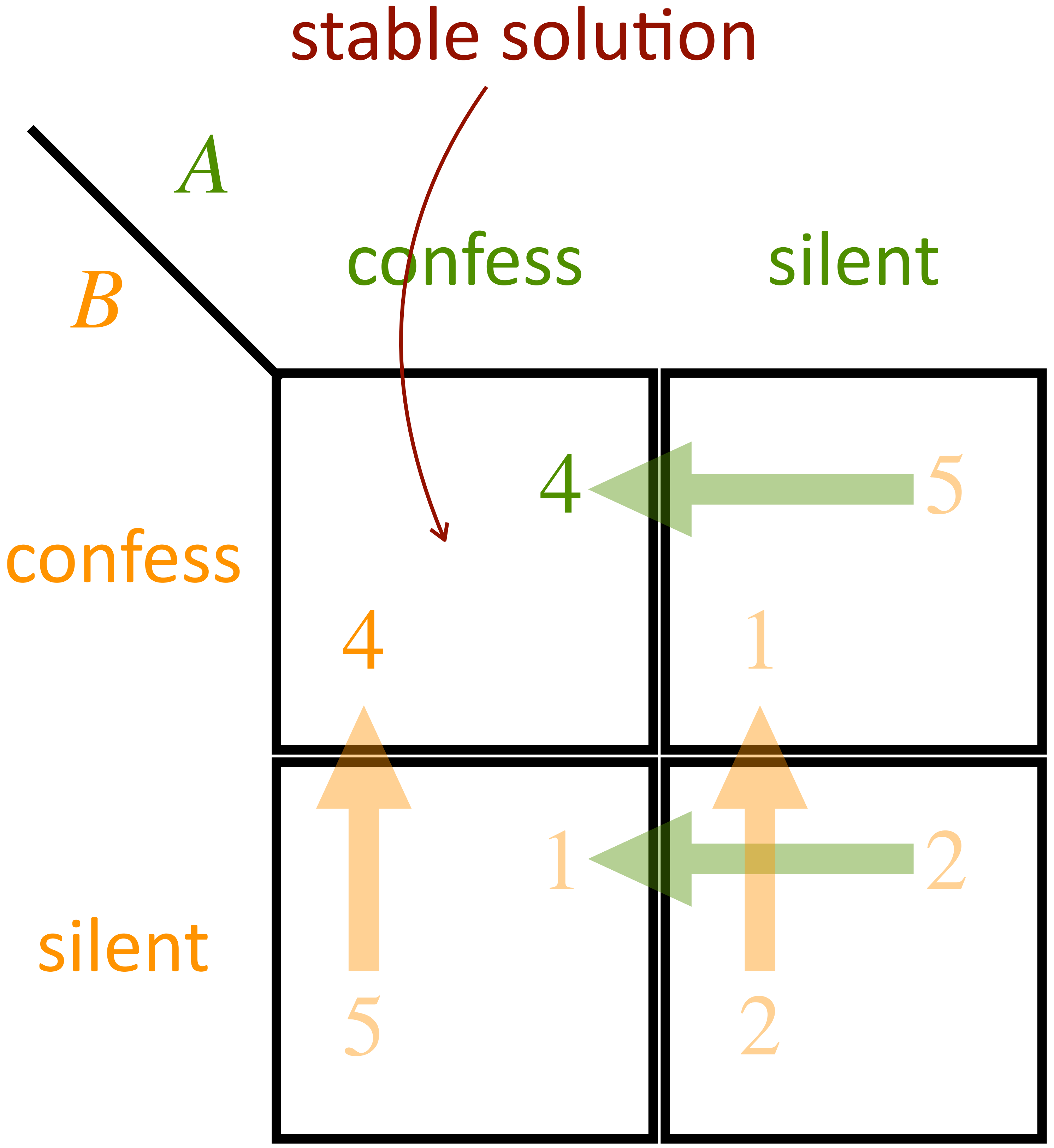
- Two prisoners A and B are on trial for a crime
  - Each of them faces a choice of confessing to the crime or remaining silent
    - If both remain silent, they both serve a short prison term (2 years)
    - If only one of them confesses, his term will be reduced to 1 year and the other get a sentence of 5 years
    - If both confess, they both serve prison sentence of 4 years



B \ A	confess	silent
confess	4 4	5 1
silent	1 5	2 2

Prisoner’s Dilemma

- Two prisoners A and B are on trial for a crime
  - Each of them faces a choice of confessing to the crime or remaining silent
    - If both remain silent, they both serve a short prison term (2 years)
    - If only one of them confesses, his term will be reduced to 1 year and the other get a sentence of 5 years
    - If both confess, they both serve prison sentence of 4 years



stable solution

A

B

	confess	silent
confess	4 4	5 1
silent	1 5	2 2

Evening Together

Evening Together

- Two players B and G are deciding on how to spend their evening

Evening Together

- Two players B and G are deciding on how to spend their evening
    - Possibilities: going to a baseball game or going to a softball game

G \ B	baseball	softball
baseball		
softball		

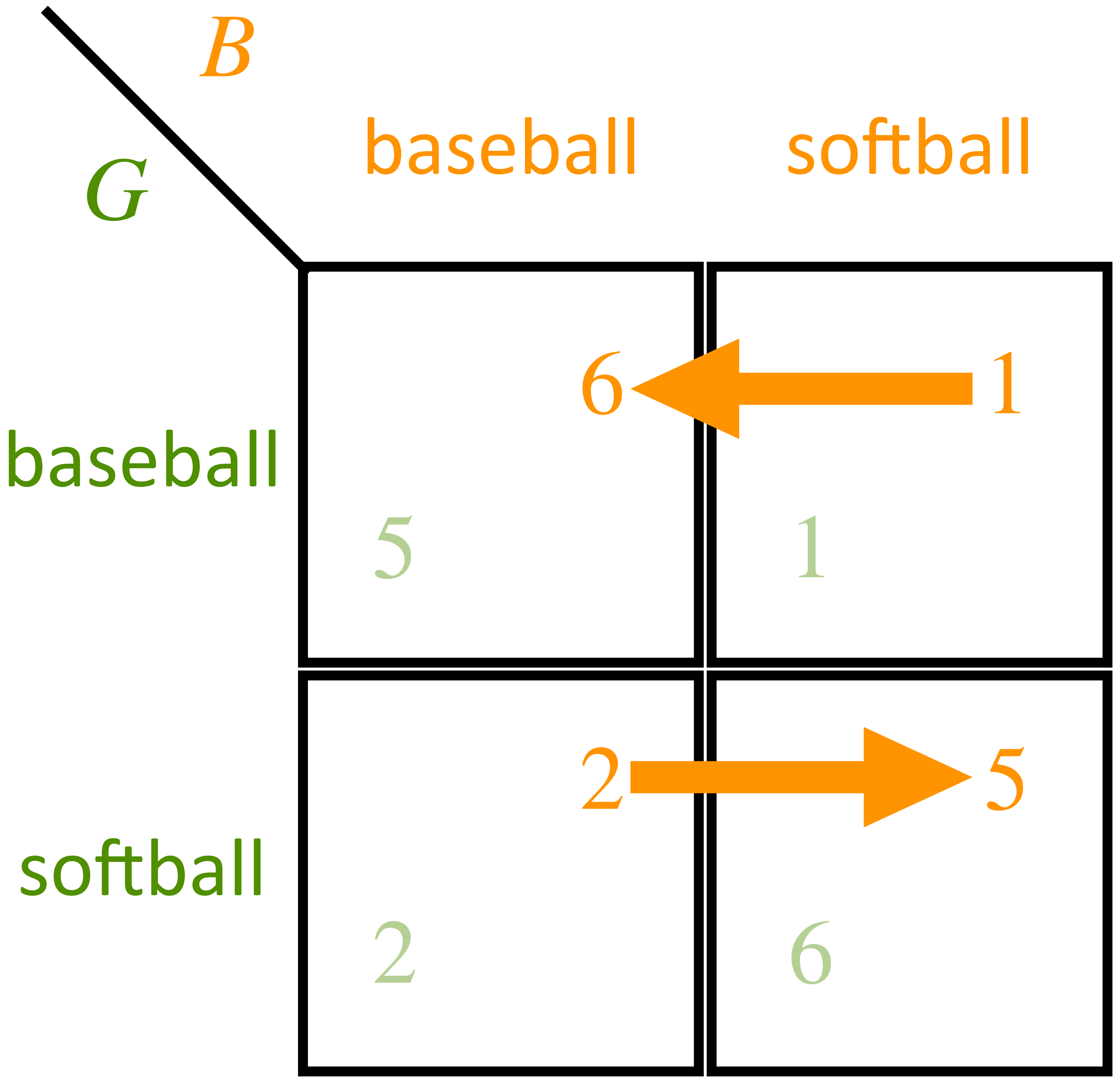
Evening Together

- Two players B and G are deciding on how to spend their evening
  - Possibilities: going to a baseball game or going to a softball game
    - B prefers baseball game and G prefers softball
    - But they both would like to spend the evening together rather than separately

G \ B	baseball	softball
baseball	6 5	1 1
softball	2 2	5 6

Evening Together

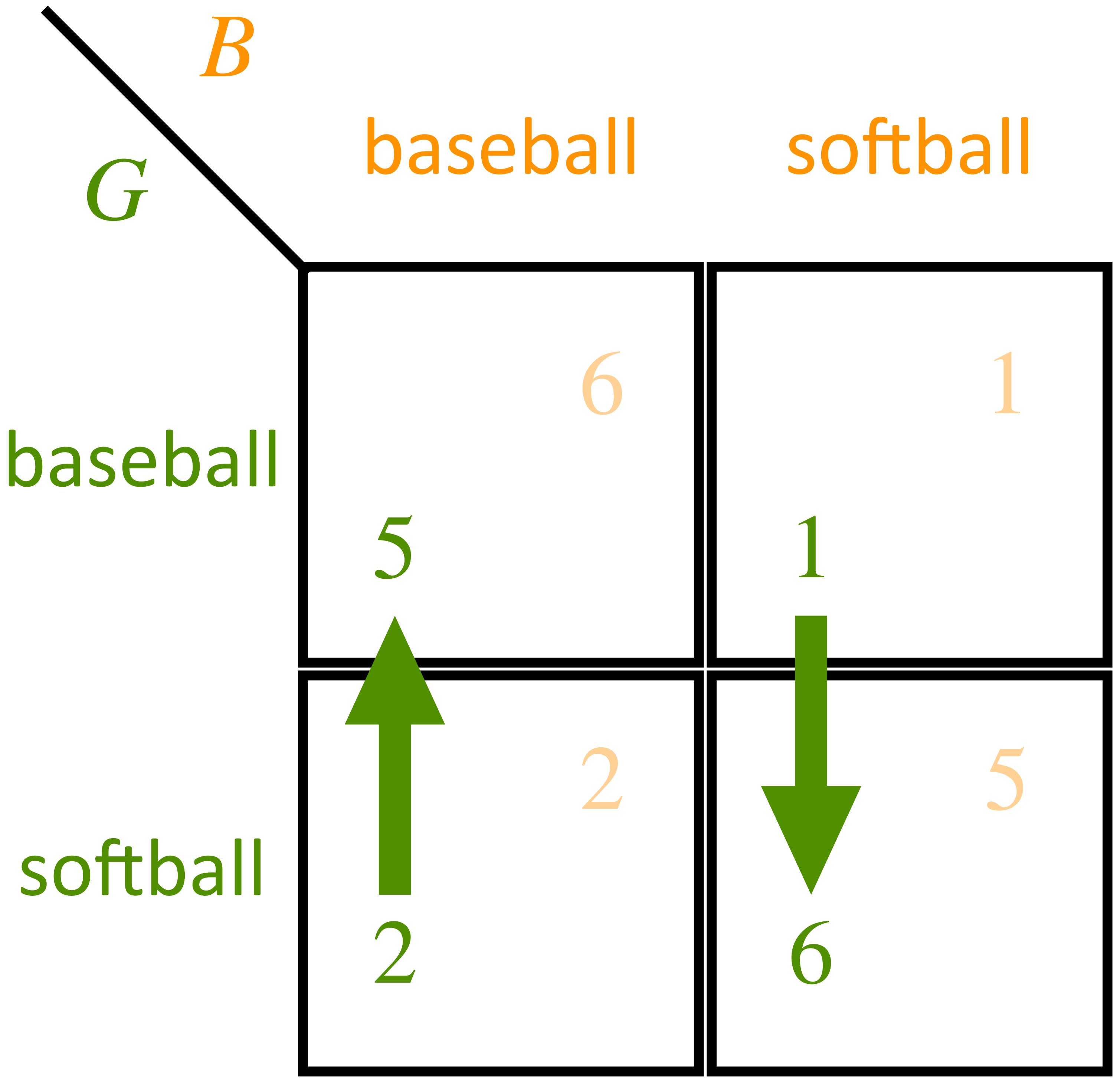
- Two players B and G are deciding on how to spend their evening
  - Possibilities: going to a baseball game or going to a softball game
    - B prefers baseball game and G prefers softball
    - But they both would like to spend the evening together rather than separately



G \ B	baseball	softball
baseball	6 5	1 1
softball	2 2	5 6

Evening Together

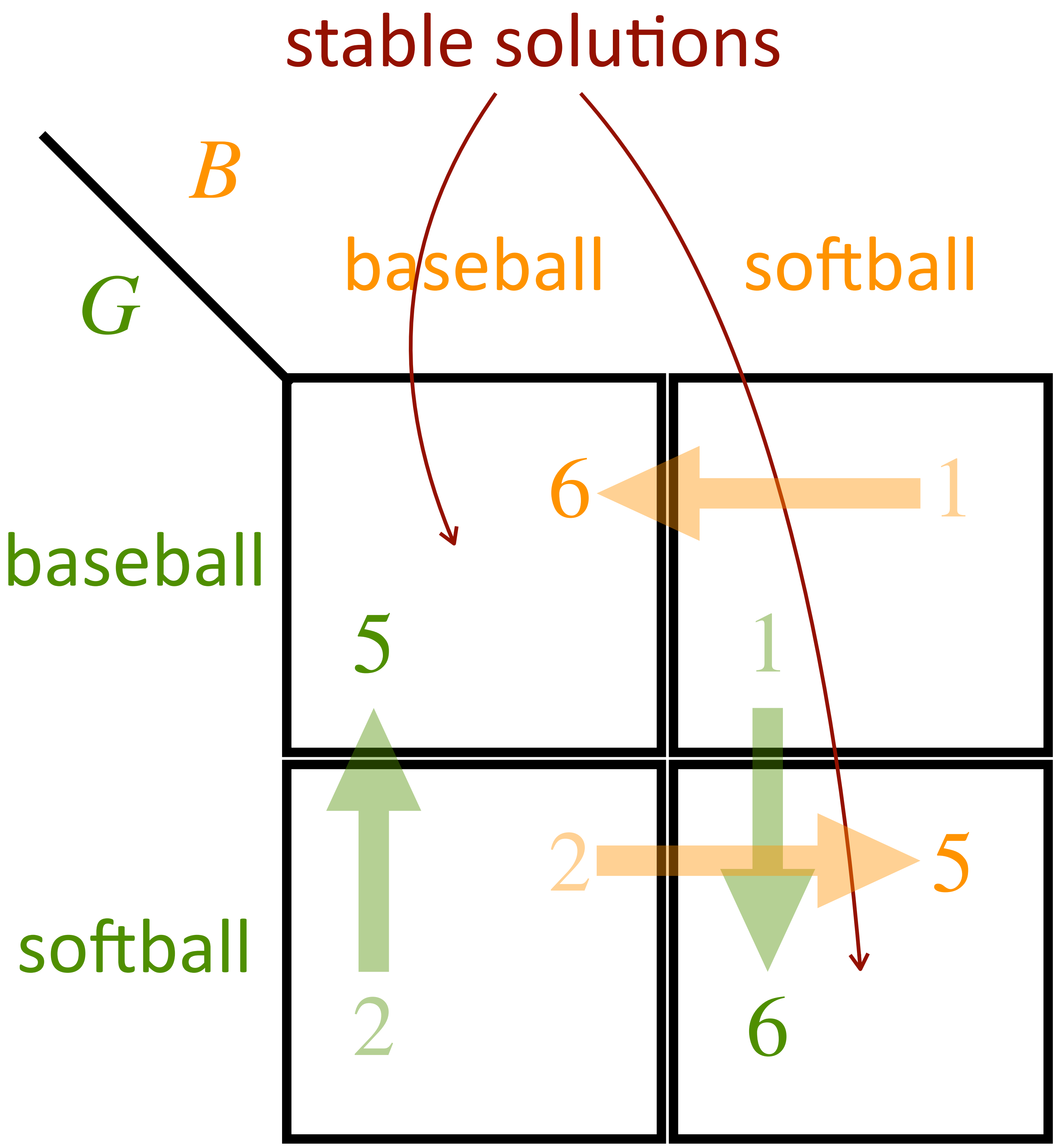
- Two players B and G are deciding on how to spend their evening
  - Possibilities: going to a baseball game or going to a softball game
    - B prefers baseball game and G prefers softball
    - But they both would like to spend the evening together rather than separately



G \ B	baseball	softball
baseball	6 5	1 1
softball	2 2	5 6

Evening Together

- Two players B and G are deciding on how to spend their evening
  - Possibilities: going to a baseball game or going to a softball game
    - B prefers baseball game and G prefers softball
    - But they both would like to spend the evening together rather than separately



stable solutions

G \ B	baseball	softball
baseball	6 5	1 1
softball	2 2	5 6

Matching Pennies

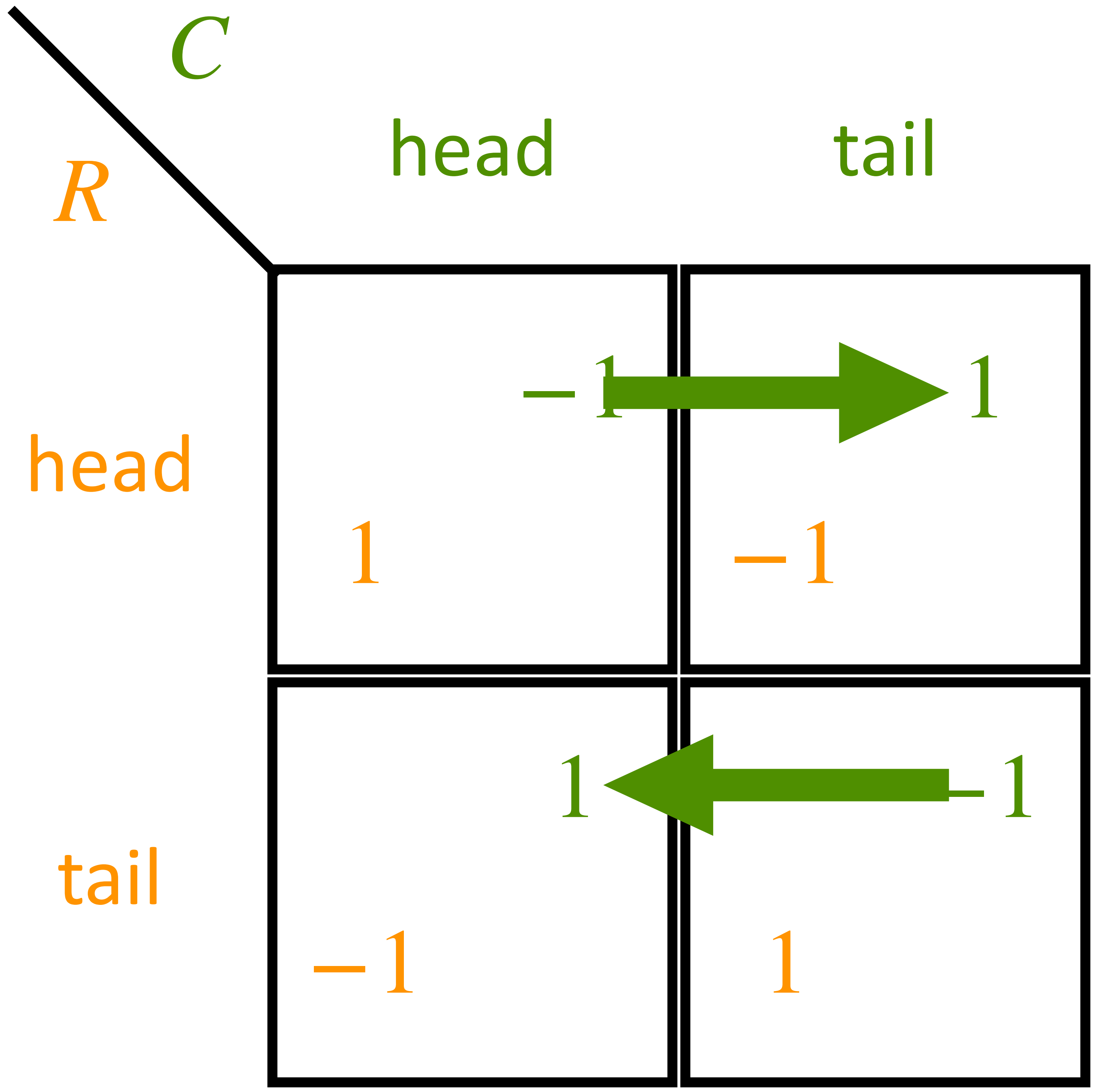
Matching Pennies

- Two players, each having a penny
  - Two strategies: head (H) or tail (T)
    - The row player wins if the two pennies match
    - The column player wins if the two pennies do not match

R \ C	head	tail
head	-1 1	1 -1
tail	1 -1	-1 1

Matching Pennies

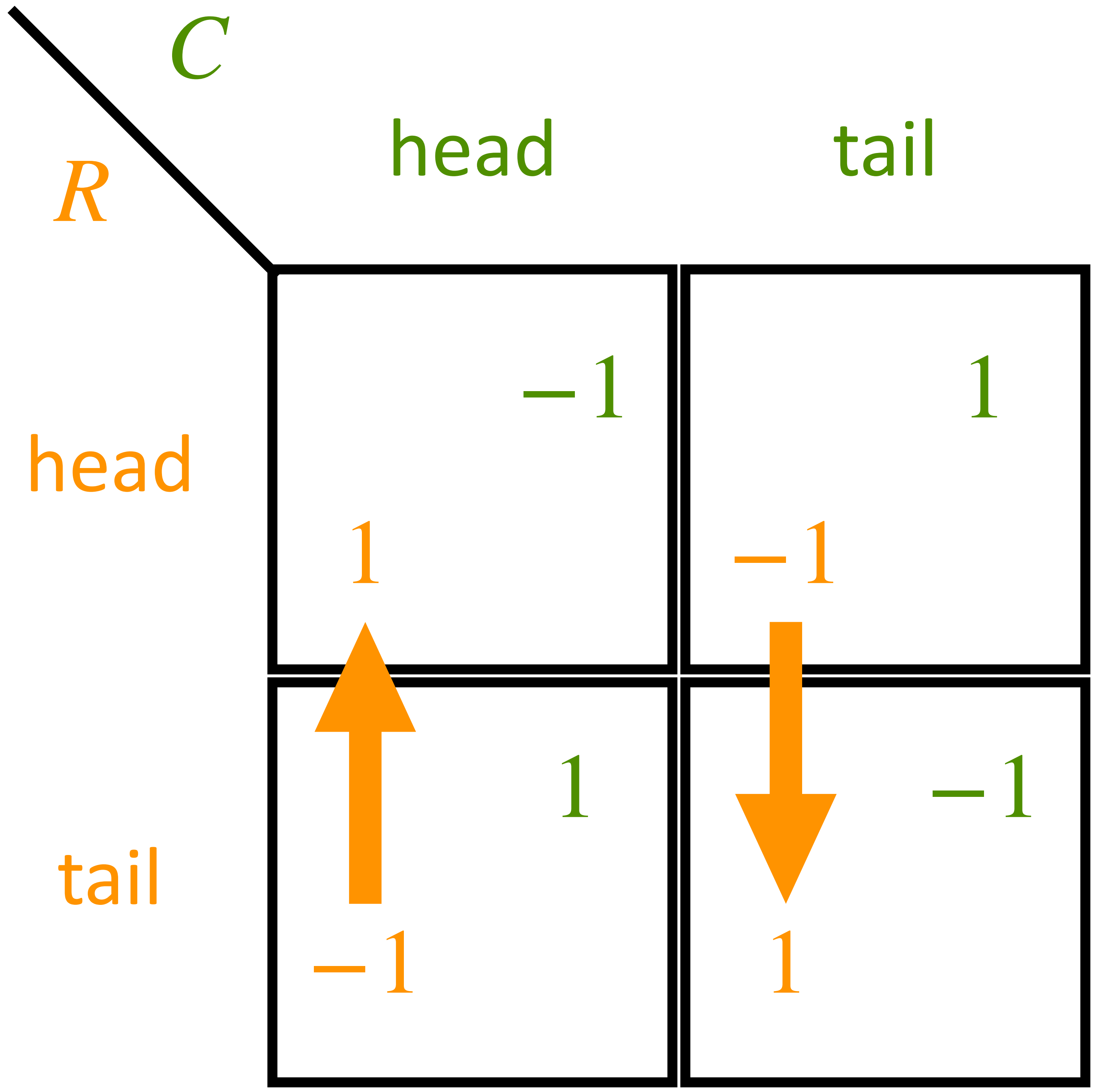
- Two players, each having a penny
  - Two strategies: head (H) or tail (T)
    - The row player wins if the two pennies match
    - The column player wins if the two pennies do not match



R \ C	head	tail
head	-1 1	1 -1
tail	1 -1	-1 1

Matching Pennies

- Two players, each having a penny
  - Two strategies: head (H) or tail (T)
    - The row player wins if the two pennies match
    - The column player wins if the two pennies do not match

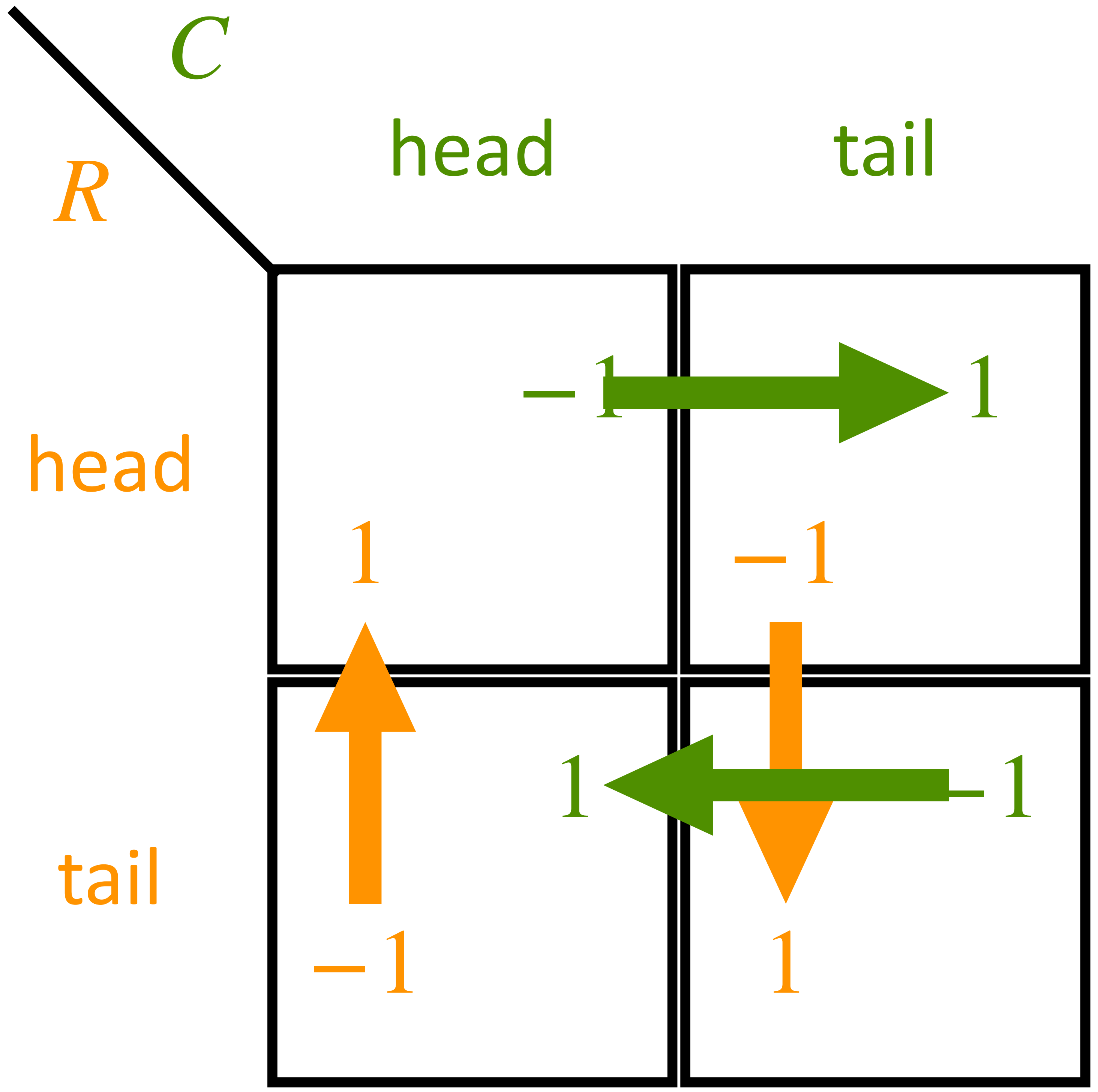


R \ C	head	tail
head	1, -1	-1, 1
tail	-1, 1	1, -1

Matching Pennies

- Two players, each having a penny
  - Two strategies: head (H) or tail (T)
    - The row player wins if the two pennies match
    - The column player wins if the two pennies do not match

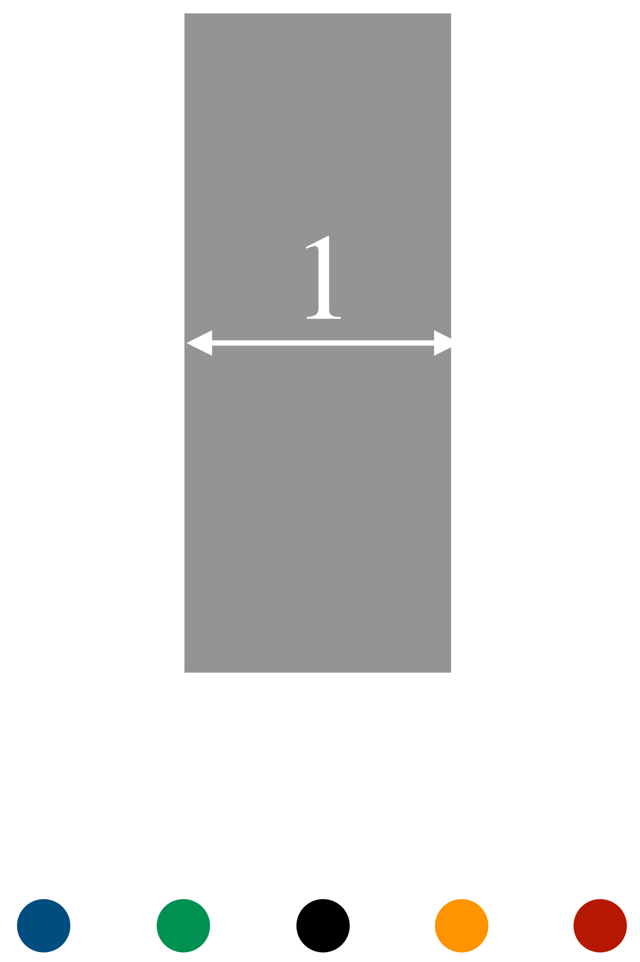
No stable solution!



Tragedy of Commons

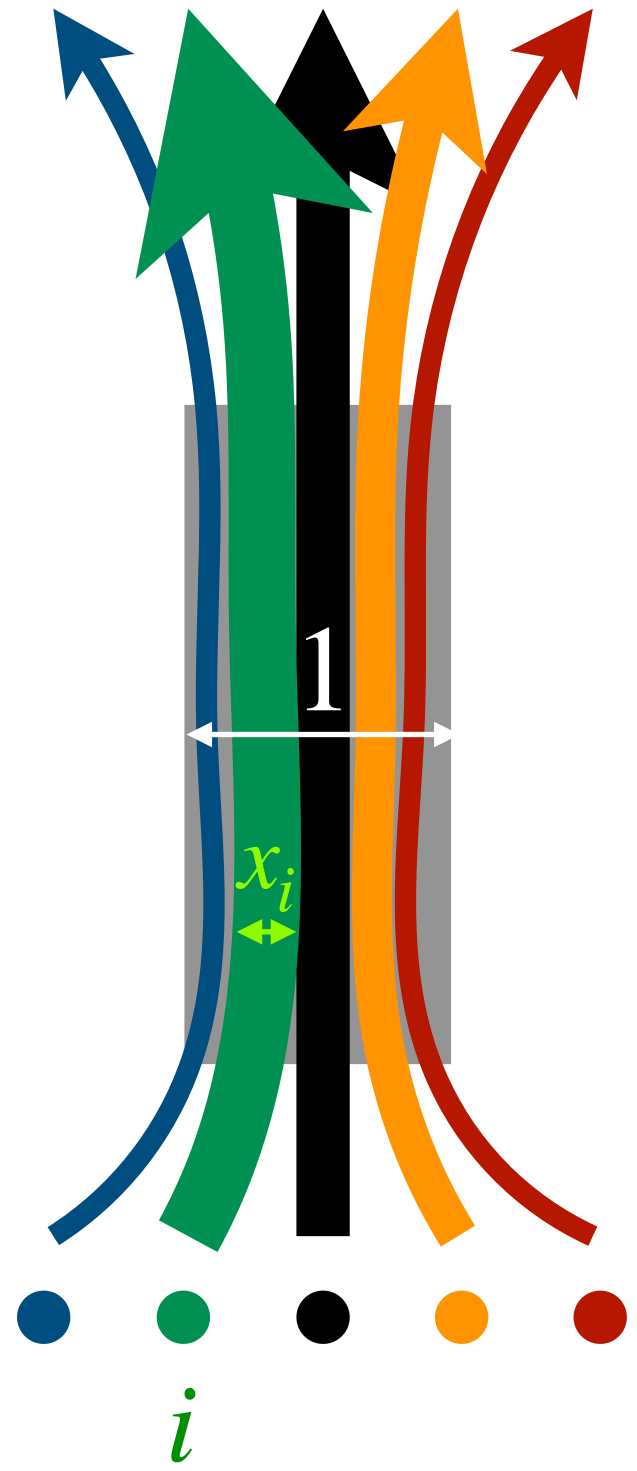
Tragedy of Commons

- n players want to have a part of a shared channel
    - The channel maximum capacity is 1



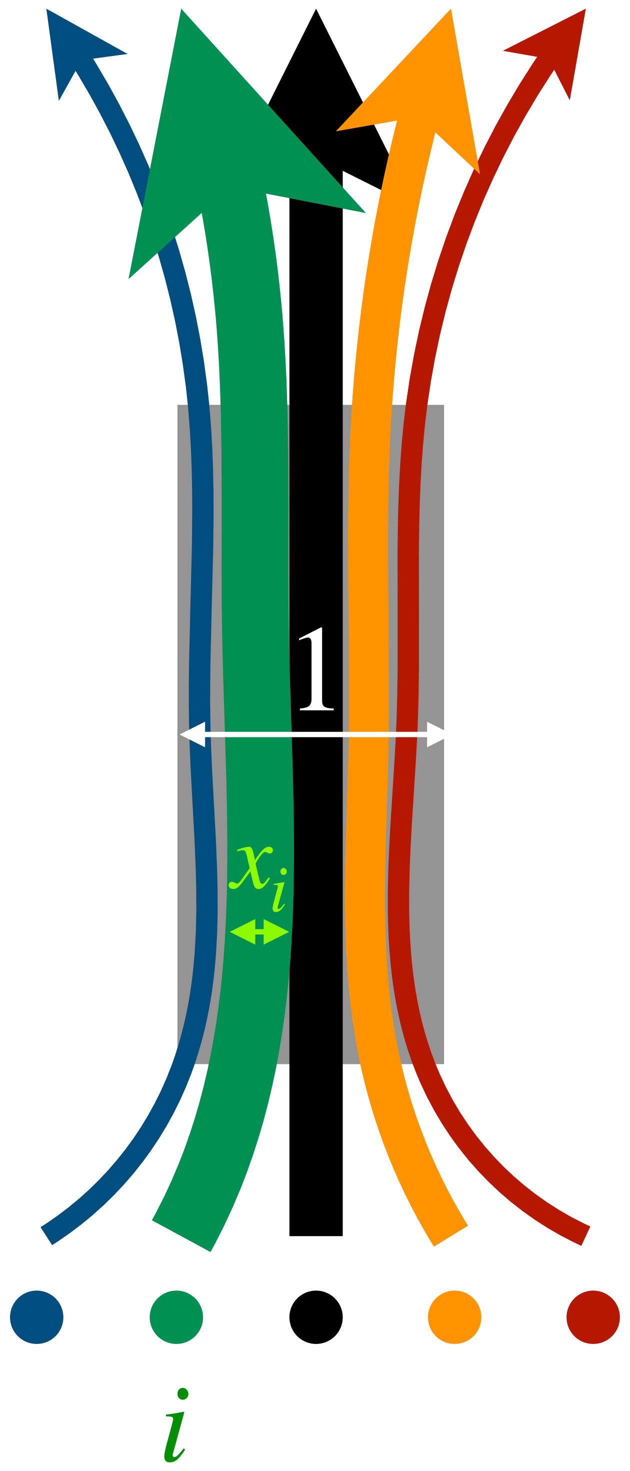
Tragedy of Commons

- n players want to have a part of a shared channel
  - The channel maximum capacity is 1
  - Each player has infinite set of strategies: sent x_i units of flow along the channel where $x_i \in [0,1]$



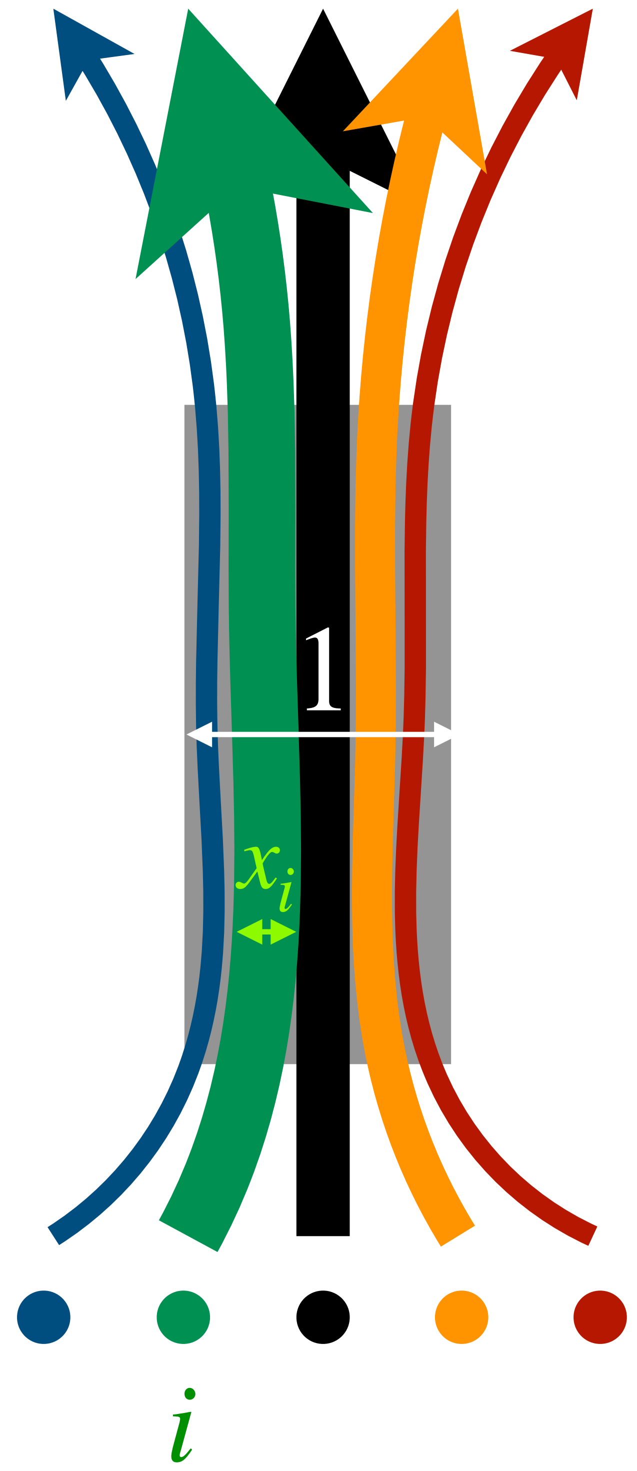
Tragedy of Commons

- n players want to have a part of a shared channel
  - The channel maximum capacity is 1, but the quality of the channel deteriorates with the total bandwidth used
  - Each player has infinite set of strategies: sent x_i units of flow along the channel where $x_i \in [0,1]$



Tragedy of Commons

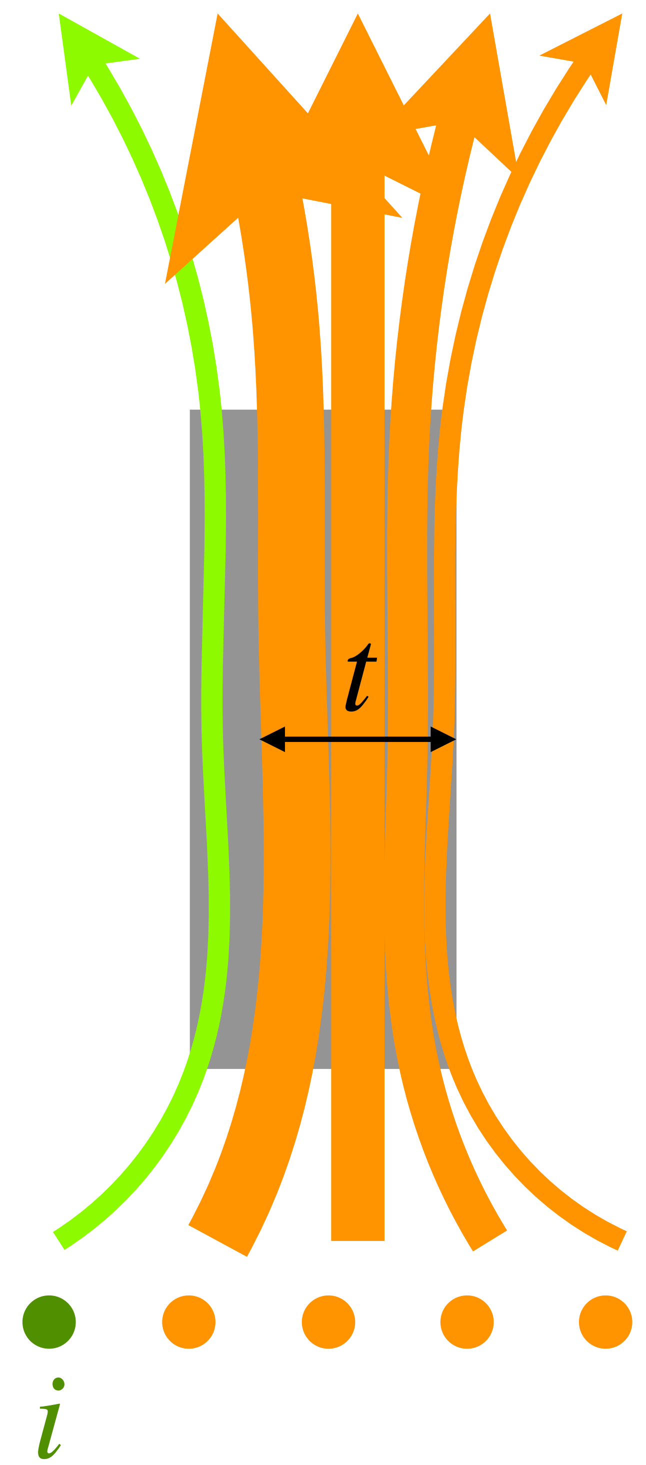
- n players want to have a part of a shared channel
  - The channel maximum capacity is 1, but the quality of the channel deteriorates with the total bandwidth used
  - Each player has infinite set of strategies: sent x_i units of flow along the channel where $x_i \in [0,1]$
    - If $\sum_j x_j \geq 1$, no player gets any benefit
    - If $\sum_j x_j < 1$, player i gets a value of $x_i\,(1 - \sum_j x_j)$



Tragedy of Commons — Stable solution

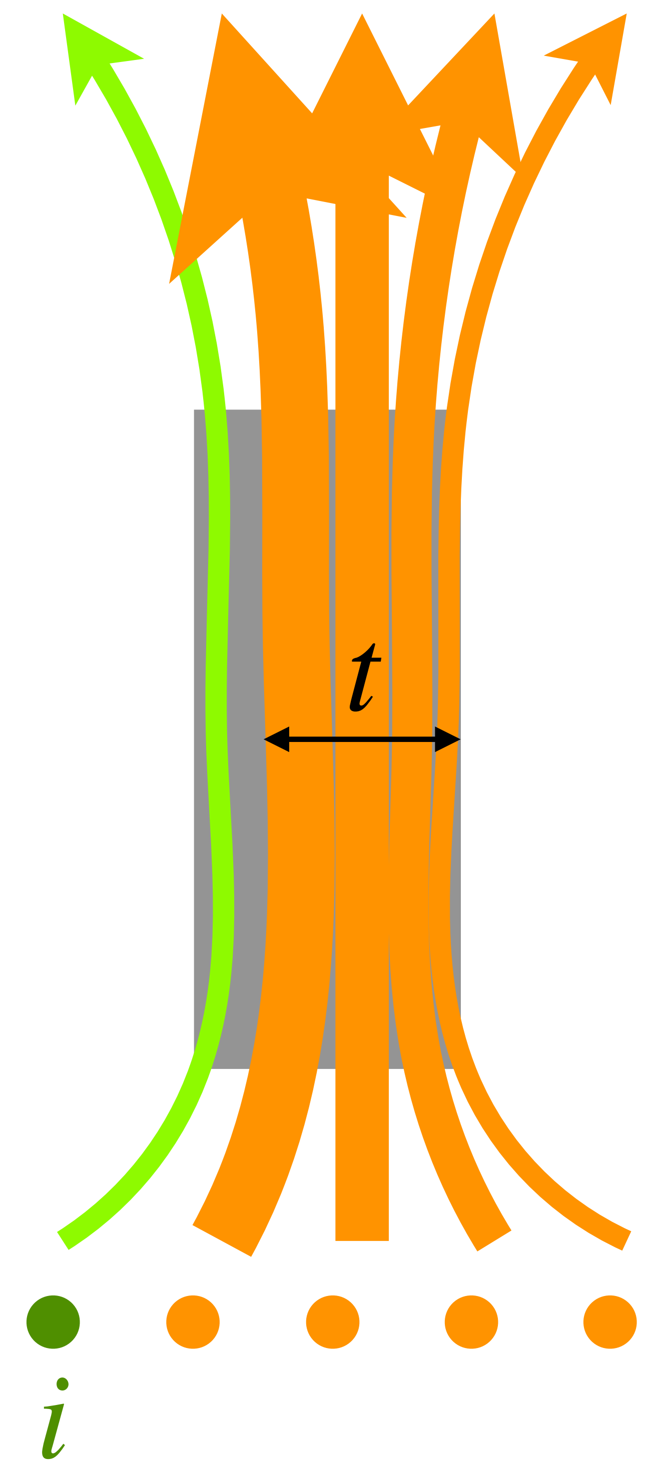
Tragedy of Commons — Stable solution

- Concentrate on player i. Let $t = \sum_{j \neq i} x_j < 1$ be the flow sent by all others



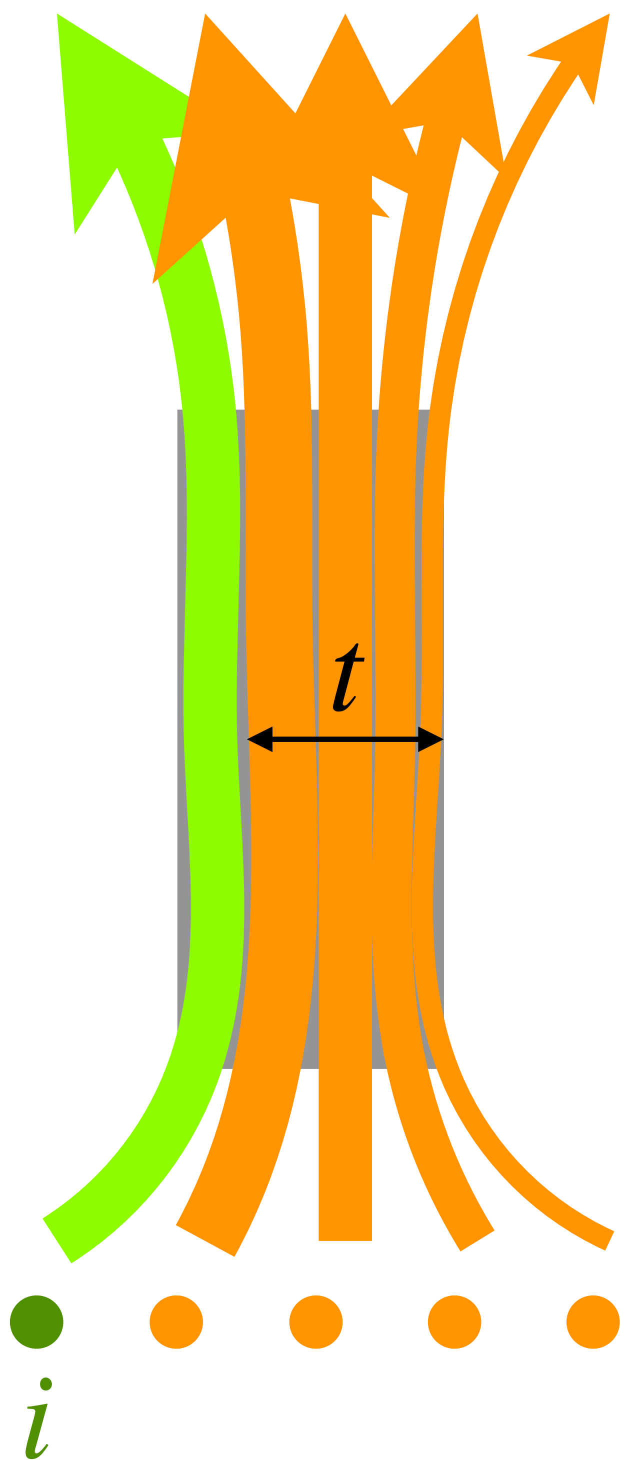
Tragedy of Commons — Stable solution

- Concentrate on player i. Let $t = \Sigma_{j \neq i} x_j < 1$ be the flow sent by all others
  - Player i's strategy is to maximize $x_i\,(1 - t - x_i)$



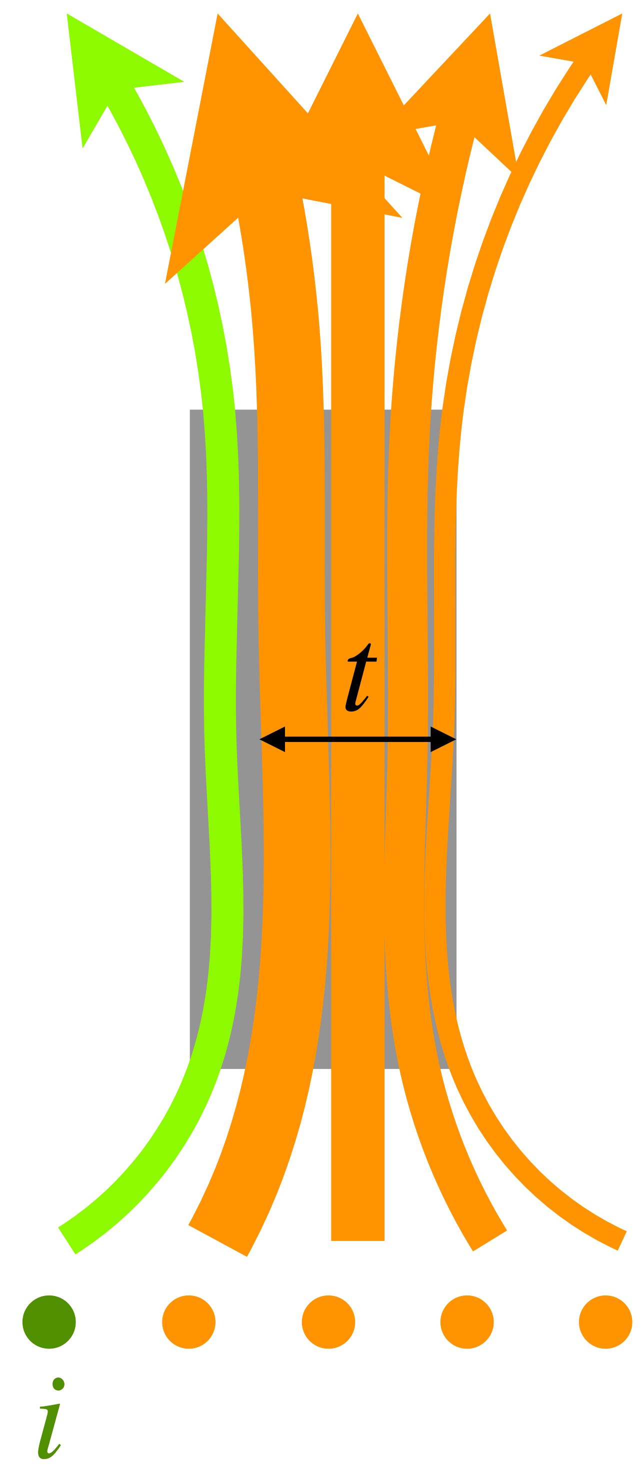
Tragedy of Commons — Stable solution

- Concentrate on player i. Let $t = \Sigma_{j \neq i} x_j < 1$ be the flow sent by all others
  - Player i's strategy is to maximize $x_i\,(1 - t - x_i)$



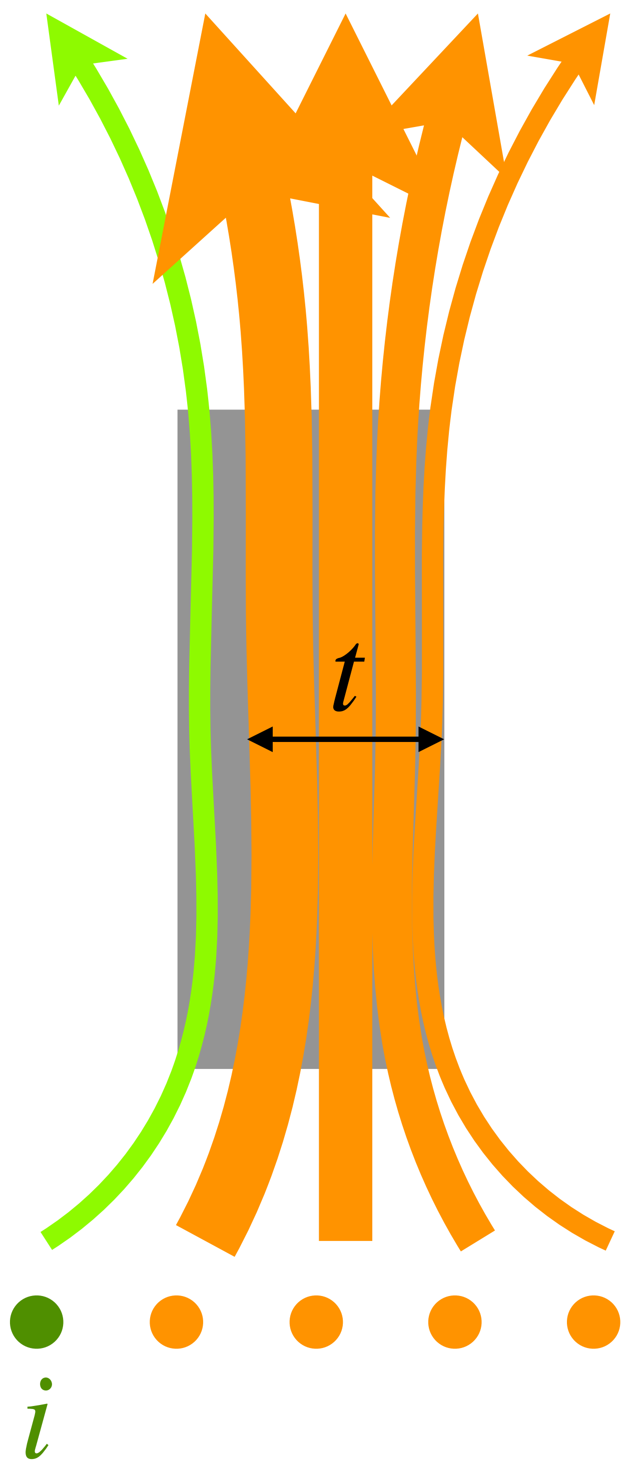
Tragedy of Commons — Stable solution

- Concentrate on player i. Let $t = \Sigma_{j \neq i} x_j < 1$ be the flow sent by all others
  - Player i's strategy is to maximize $x_i(1 - t - x_i)$



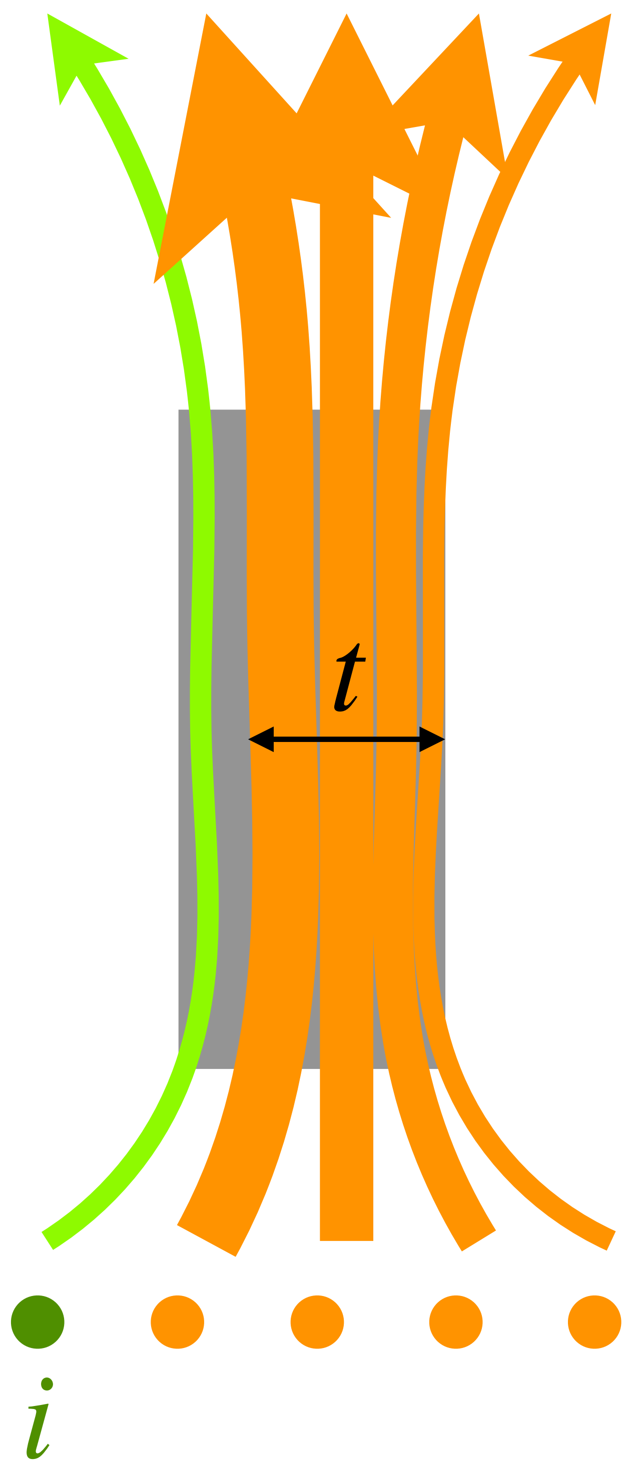
Tragedy of Commons — Stable solution

- Concentrate on player i. Let $t = \Sigma_{j \neq i} x_j < 1$ be the flow sent by all others
  - Player i's strategy is to maximize $x_i\,(1 - t - x_i) \Rightarrow x_i = \dfrac{1 - t}{2} = \dfrac{1 - \Sigma_{j \neq i} x_j}{2}$



Tragedy of Commons — Stable solution

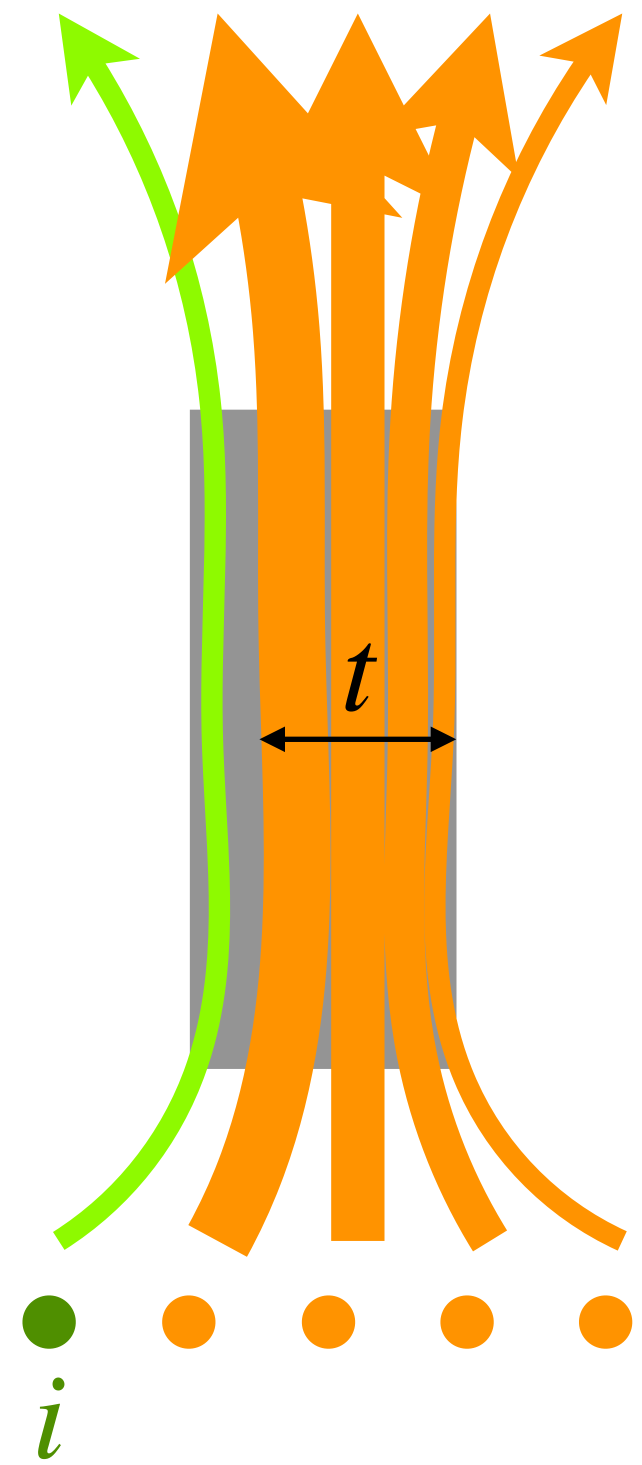
- Concentrate on player i. Let $t = \Sigma_{j \neq i} x_j < 1$ be the flow sent by all others
  - Player i's strategy is to maximize $x_i\,(1 - t - x_i) \Rightarrow x_i = \dfrac{1-t}{2} = \dfrac{1 - \Sigma_{j \neq i} x_j}{2}$
- A set of strategies is stable if all players are playing their optimal selfish strategy, given the strategies of all other players



Tragedy of Commons — Stable solution

- Concentrate on player i. Let $t = \Sigma_{j \neq i} x_j < 1$ be the flow sent by all others
  - Player i's strategy is to maximize $x_i\,(1 - t - x_i) \Rightarrow x_i = \dfrac{1-t}{2} = \dfrac{1 - \Sigma_{j \neq i} x_j}{2}$
- A set of strategies is stable if all players are playing their optimal selfish strategy, given the strategies of all other players

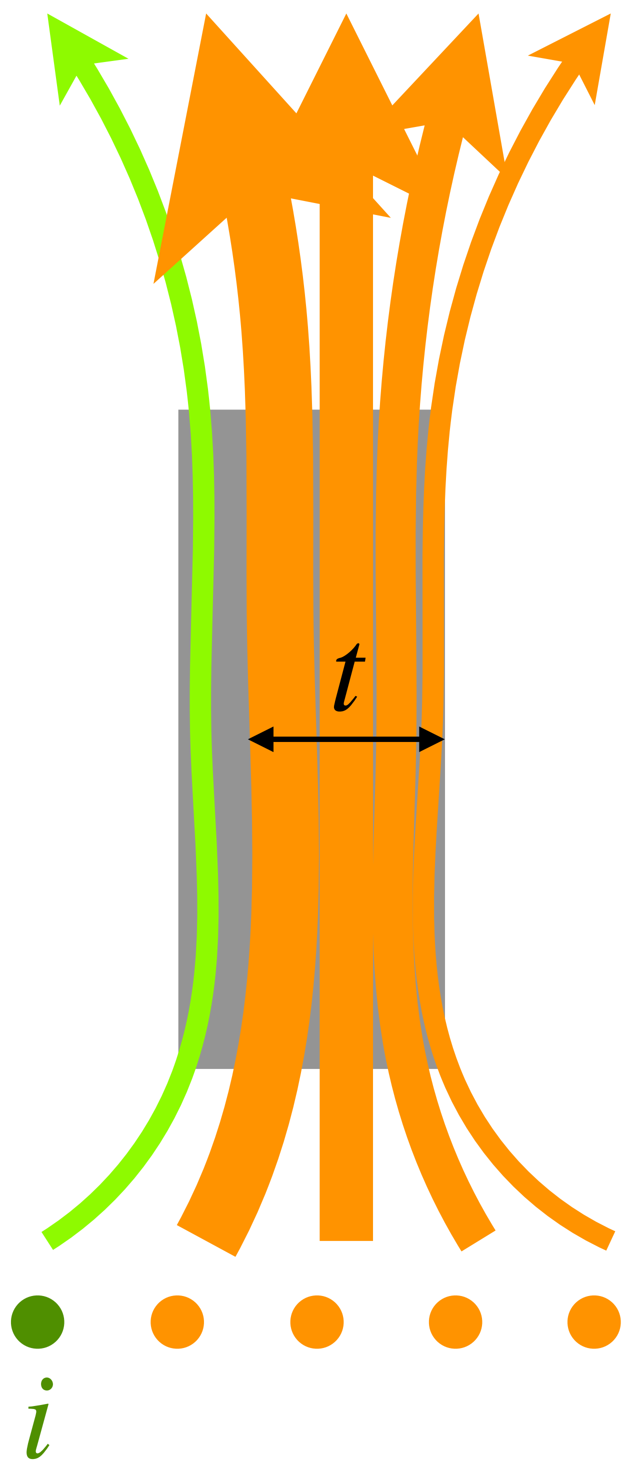
$$\Rightarrow x_i = \frac{1 - \Sigma_{j \neq i} x_j}{2} \text{ for all } i$$



Tragedy of Commons — Stable solution

- Concentrate on player i. Let $t = \Sigma_{j \neq i} x_j < 1$ be the flow sent by all others
  - Player i's strategy is to maximize $x_i\,(1 - t - x_i) \Rightarrow x_i = \dfrac{1-t}{2} = \dfrac{1 - \Sigma_{j \neq i} x_j}{2}$
- A set of strategies is stable if all players are playing their optimal selfish strategy, given the strategies of all other players

$$\Rightarrow x_i = \frac{1 - \Sigma_{j \neq i} x_j}{2} \text{ for all } i \quad \Rightarrow \Sigma_i x_i = \frac{n}{2} - \frac{n-1}{2} \cdot \Sigma_i x_i$$

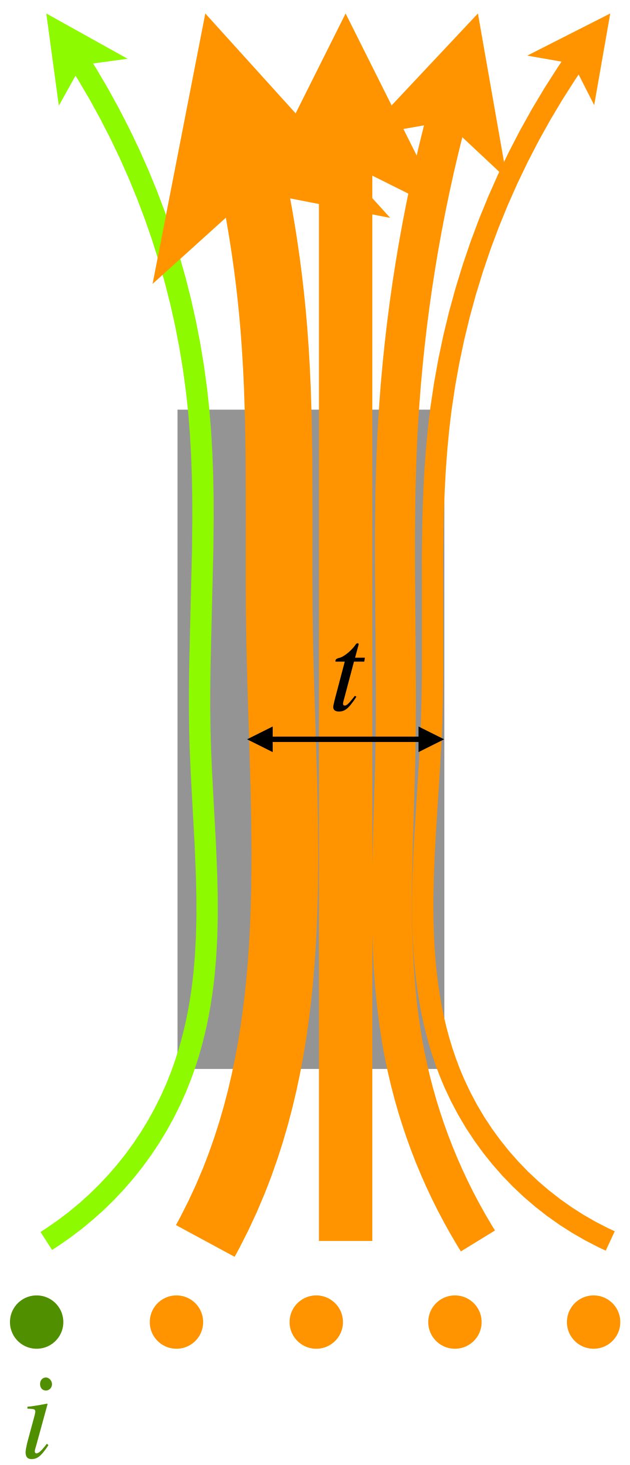


Tragedy of Commons — Stable solution

- Concentrate on player i. Let $t = \Sigma_{j \neq i}\, x_j < 1$ be the flow sent by all others
  - Player i's strategy is to maximize $x_i\,(1 - t - x_i) \Rightarrow x_i = \frac{1-t}{2} = \frac{1 - \Sigma_{j \neq i} x_j}{2}$
- A set of strategies is stable if all players are playing their optimal selfish strategy, given the strategies of all other players

$$\Rightarrow x_i = \frac{1 - \Sigma_{j \neq i} x_j}{2} \text{ for all } i \quad \Rightarrow \Sigma_i x_i = \frac{n}{2} - \frac{n-1}{2} \cdot \Sigma_i x_i$$

$$\Rightarrow x_i = \frac{1}{n+1} \text{ for all } i$$



Tragedy of Commons — Better solution

- Selfish strategy: $x_i = \dfrac{1}{n+1}$ for all i
  - Total bandwidth used is $\dfrac{n}{n+1}$
  - For each player, the value is $x_i(1 - \Sigma_j x_i) = \dfrac{1}{n+1}(1 - \dfrac{n}{n+1}) = \dfrac{1}{(n+1)^2}$

Tragedy of Commons — Better solution

- Selfish strategy: $x_i = \frac{1}{n+1}$ for all i
  - Total bandwidth used is $\frac{n}{n+1}$
  - For each player, the value is $x_i(1 - \Sigma_j x_i) = \frac{1}{n+1}(1 - \frac{n}{n+1}) = \frac{1}{(n+1)^2}$
- (Centralized) better strategy: if the total bandwidth used is $\frac{1}{2} \cdot \frac{n}{n+1}$:
  - $x_i = \frac{1}{2(n+1)}$ for each player i, and the value of each player is $\frac{1}{2(n+1)} \cdot (1 - \frac{n}{2(n+1)}) = \frac{n+2}{4(n+1)^2}$
  - The new value is $\frac{n+2}{4}$ times the old value (!!)

What happened

- Self-interested behavior in a decentralized environment can decrease the overall performance:
  - Agents are selfish (Prisoner's dilemma)
  - Agents cannot communicate (Evening together, tragedy of commons)

Outline

- Fundamental concepts
  - Game, players, strategies, payoffs/costs
- Nash Equilibrium
- Price of Anarchy
  - Selfish load balancing
- Mechanism design
  - Vickrey-Clarke-Groves mechanism

Games: Formal Definitions

- A game consists of a set of n self-interested *players*, $\{1,2,\cdots,n\}$
- Each player i selects a *strategy* s_i
- The *vector of strategies* $\vec{s} = (s_1, s_2, \cdots, s_n)$ selected by the players determine the outcome for each player
  - *payoff*/*utility* $u_i(s_1, s_2, \cdots, s_n) \in \mathbb{R}$
  - *cost* $c_i(s_1, s_2, \cdots, s_n) \in \mathbb{R}$

Prisoner’s Dilemma

- Two prisoners A and B are on trial for a crime
  - Each of them faces a choice of confessing to the crime or remaining silent
    - If both remain silent, they both serve a short prison term (2 years)
    - If only one of them confesses, his term will be reduced to 1 year and the other get a sentence of 5 years
    - If both confess, they both serve prison sentence of 4 years

B \ A	confess	silent
confess	4 4	5 1
silent	1 5	2 2

Prisoner’s Dilemma

- Two prisoners A and B are on trial for a crime
  - Each of them faces a choice of confessing to the crime or remaining silent
    - If both remain silent, they both serve a short prison term (2 years)
    - If only one of them confesses, his term will be reduced to 1 year and the other get a sentence of 5 years
    - If both confess, they both serve prison sentence of 4 years

B \ A	confess	silent
confess	4 4	5 1
silent	1 5	2 2

strategy: confess or silent

Prisoner’s Dilemma

- Two prisoners A and B are on trial for a crime
  - Each of them faces a choice of confessing to the crime or remaining silent
    - If both remain silent, they both serve a short prison term (2 years)
    - If only one of them confesses, his term will be reduced to 1 year and the other get a sentence of 5 years
    - If both confess, they both serve prison sentence of 4 years

B \ A	confess	silent
confess	4 4	5 1
silent	1 5	2 2

strategy: confess or silent

c_A(confess, silent) $= 1$

c_B(confess, silent) $= 5$

Games: Formal Definitions

- A game consists of a set of n self-interested *players*, $\{1,2,\cdots,n\}$
- Each player i selects a *strategy* s_i
- The *vector of strategies* $\vec{s} = (s_1, s_2, \cdots, s_n)$ selected by the players determine the outcome for each player
  - *payoff/utility* $u_i(s_1, s_2, \cdots, s_n) \in \mathbb{R}$
  - *cost* $c_i(s_1, s_2, \cdots, s_n) \in \mathbb{R}$

Evening Together

- Two players B and G are deciding on how to spend their evening
  - Possibilities: going to a baseball game or going to a softball game
    - B prefers baseball game and G prefers softball
    - But they both would like to spend the evening together rather than separately

B G	baseball	softball
baseball	6 5	1 1
softball	2 2	5 6

Evening Together

- Two players B and G are deciding on how to spend their evening
  - Possibilities: going to a baseball game or going to a softball game
    - B prefers baseball game and G prefers softball
    - But they both would like to spend the evening together rather than separately

B G	baseball	softball
baseball	6 5	1 1
softball	2 2	5 6

strategy: baseball or softball

Evening Together

- Two players B and G are deciding on how to spend their evening
  - Possibilities: going to a baseball game or going to a softball game
    - B prefers baseball game and G prefers softball
    - But they both would like to spend the evening together rather than separately

G \ B	baseball	softball
baseball	6 5	1 1
softball	2 2	5 6

strategy: baseball or softball

u_B(softball, softball) $= 5$

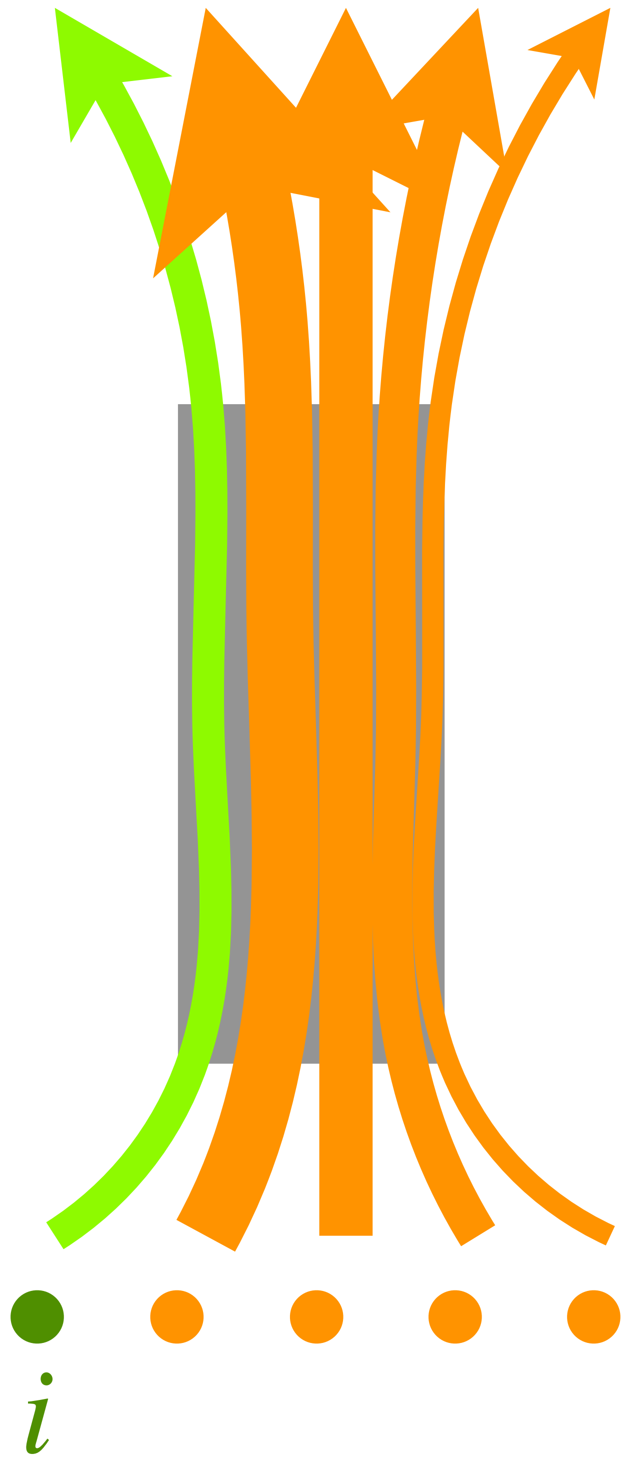
u_G(softball, softball) $= 6$

Games: Formal Definitions

- A game consists of a set of n self-interested *players*, $\{1,2,\cdots,n\}$
- Each player i selects a *strategy* s_i
- The *vector of strategies* $\vec{s} = (s_1, s_2, \cdots, s_n)$ selected by the players determine the outcome for each player
  - *payoff/utility* $u_i(s_1, s_2, \cdots, s_n) \in \mathbb{R}$
  - *cost* $c_i(s_1, s_2, \cdots, s_n) \in \mathbb{R}$

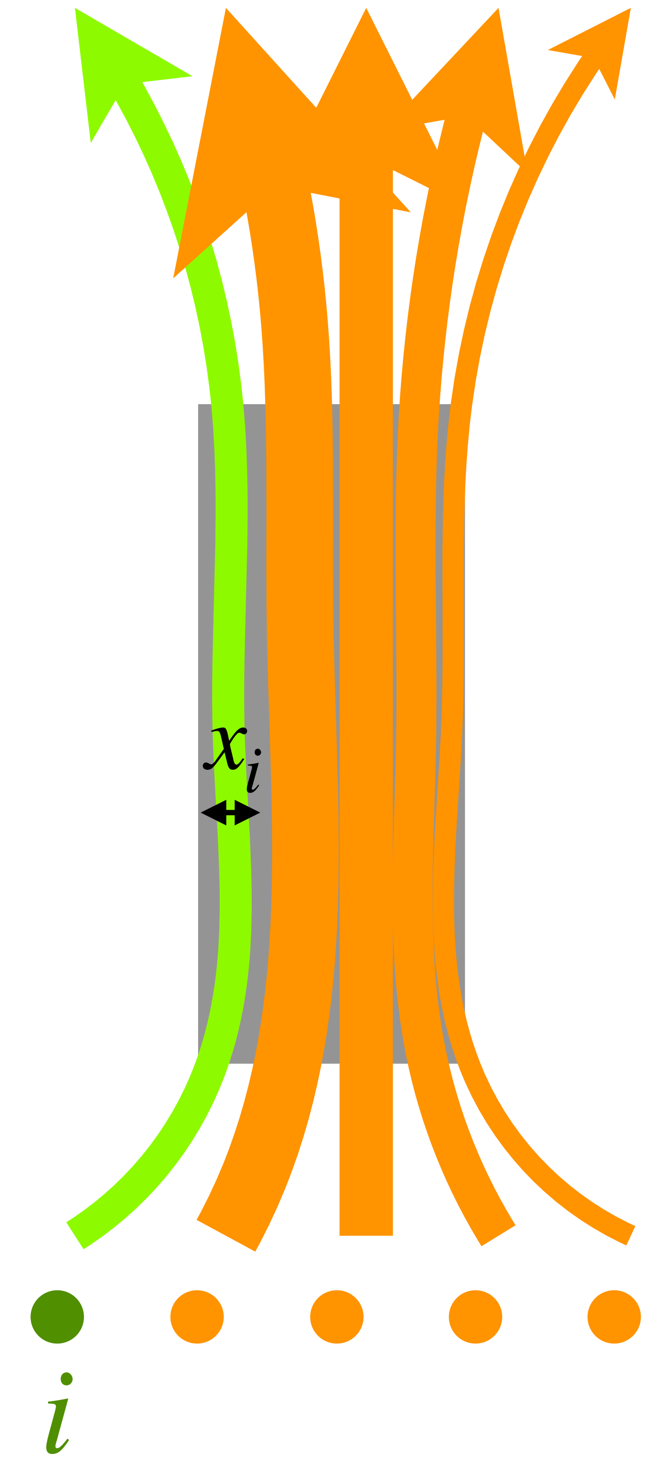
Tragedy of Commons

- n players want to have a part of a shared channel
  - The channel maximum capacity is 1, but the quality of the channel deteriorates with the total bandwidth used
  - Each player has infinite set of strategies: sent x_i units of flow along the channel where $x_i \in [0,1]$
    - If $\sum_j x_j \geq 1$, no player gets any benefit
    - If $\sum_j x_j < 1$, player i gets a value of $x_i\,(1 - \sum_j x_j)$



Tragedy of Commons

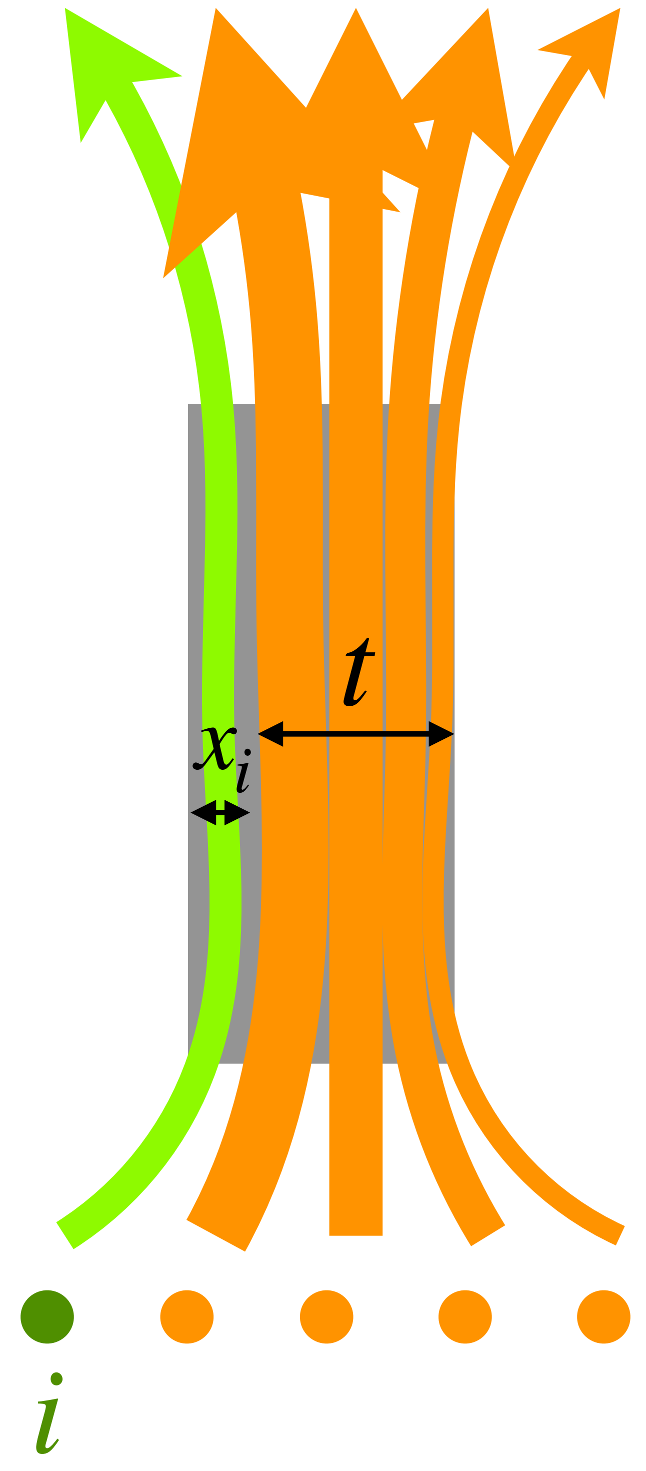
- n players want to have a part of a shared channel
  - The channel maximum capacity is 1, but the quality of the channel deteriorates with the total bandwidth used
  - Each player has infinite set of strategies: sent x_i units of flow along the channel where $x_i \in [0,1]$
    - If $\sum_j x_j \geq 1$, no player gets any benefit
    - If $\sum_j x_j < 1$, player i gets a value of $x_i (1 - \sum_j x_j)$



strategy: $x_i \in [0,1]$

Tragedy of Commons

- n players want to have a part of a shared channel
    - The channel maximum capacity is 1, but the quality of the channel deteriorates with the total bandwidth used
    - Each player has infinite set of strategies: sent x_i units of flow along the channel where $x_i \in [0,1]$
        - If $\sum_j x_j \geq 1$, no player gets any benefit
        - If $\sum_j x_j < 1$, player i gets a value of $x_i(1 - \sum_j x_j)$



strategy: $x_i \in [0,1]$

$u_i(x_i, t) = x_i(1 - t - x_i)$

Outline

- Fundamental concepts
    - Game, players, strategies, payoffs/costs
- Nash Equilibrium
- Price of Anarchy
    - Selfish load balancing
- Mechanism design
    - Auction
    - Vickrey-Clarke-Groves mechanism

Nash Equilibrium

- Player i *(weakly) prefers* $\overrightarrow{s_x}$ to $\overrightarrow{s_y}$ if i prefers $\overrightarrow{s_x}$ to $\overrightarrow{s_y}$ or considers them as equally good outcomes. That is, $u_i(\overrightarrow{s_x}) \geq u_i(\overrightarrow{s_y})$
  - $\overrightarrow{s_{-i}} = (s_1, s_2, \cdots, s_{i-1}, s_{i+1}, \cdots, s_n)$
  - $\vec{s} = (s_1, s_2, \cdots, s_{i-1}, s_i, s_{i+1}, \cdots, s_n) = (s_i, \overrightarrow{s_{-i}})$

$\vec{s} =$ (confess, silent)

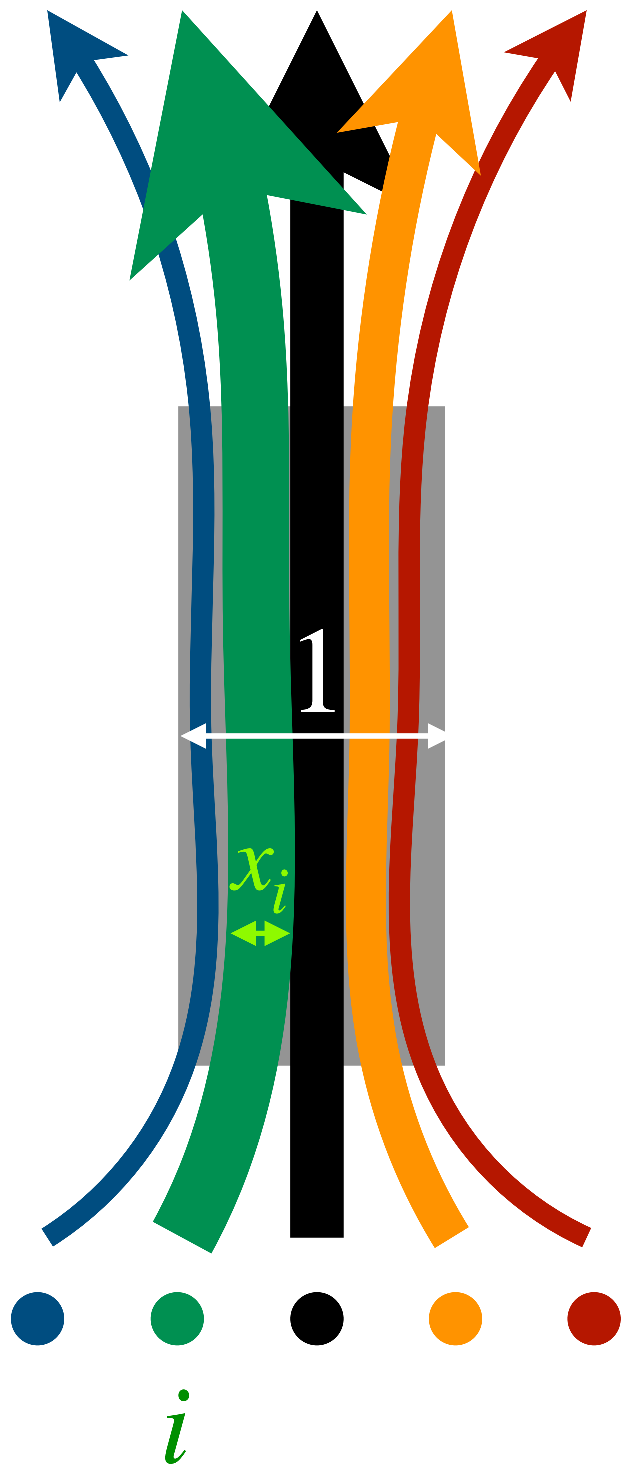
$\overrightarrow{s_{-A}} =$ (silent)

A B	confess	silent
confess	4 4	5 1
silent	1 5	2 2

Nash Equilibrium

- Player i *(weakly) prefers* $\overrightarrow{s_x}$ to $\overrightarrow{s_y}$ if i prefers $\overrightarrow{s_x}$ to $\overrightarrow{s_y}$ or considers them as equally good outcomes. That is, $u_i(\overrightarrow{s_x}) \geq u_i(\overrightarrow{s_y})$
  - $\overrightarrow{s_{-i}} = (s_1, s_2, \cdots, s_{i-1}, s_{i+1}, \cdots, s_n)$
  - $\vec{s} = (s_1, s_2, \cdots, s_{i-1}, s_i, s_{i+1}, \cdots, s_n) = (s_i, \overrightarrow{s_{-i}})$

$$\vec{s} = (0.08, 0.25, 0.2, 0.15, 0.08)$$
$$\overrightarrow{s_{-2}} = (0.08, 0.2, 0.15, 0.08)$$



Nash Equilibrium

- Player i *(weakly) prefers* $\overrightarrow{s_x}$ to $\overrightarrow{s_y}$ if i prefers $\overrightarrow{s_x}$ to $\overrightarrow{s_y}$ or considers them as equally good outcomes. That is, $u_i(\overrightarrow{s_x}) \geq u_i(\overrightarrow{s_y})$
  - $\overrightarrow{s_{-i}} = (s_1, s_2, \cdots, s_{i-1}, s_{i+1}, \cdots, s_n)$
  - $\vec{s} = (s_1, s_2, \cdots, s_{i-1}, s_i, s_{i+1}, \cdots, s_n) = (s_i, \overrightarrow{s_{-i}})$

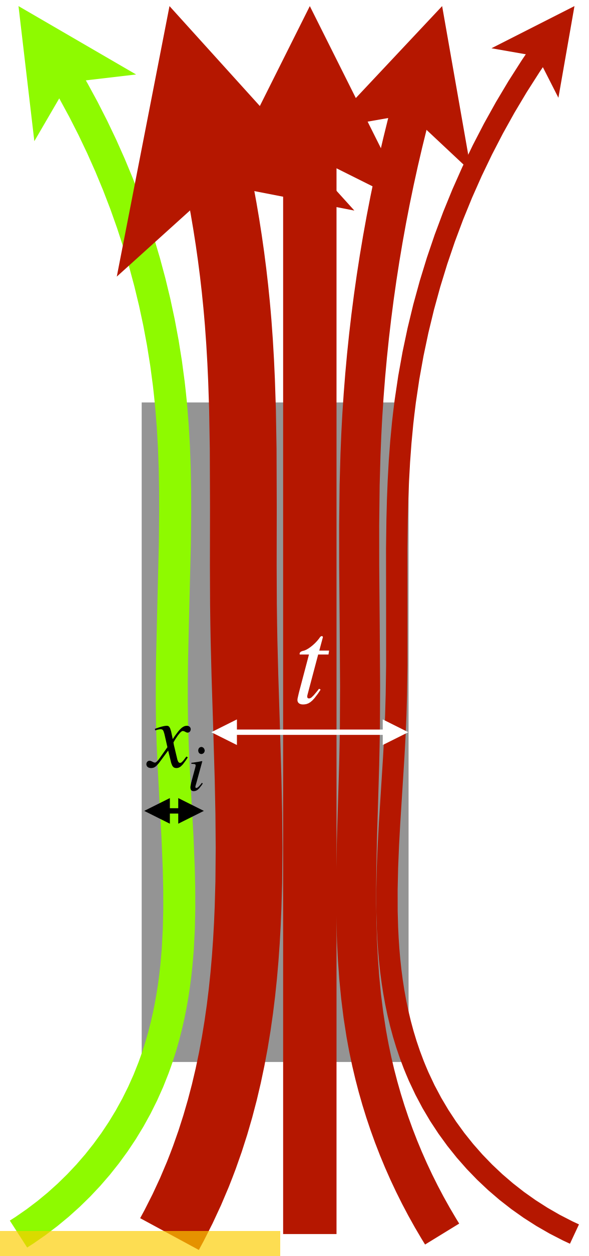
- A strategy vector $\vec{s}$ is a **Nash equilibrium** if

for all players i and each alternate strategy s_i':

$$u_i(s_i, \overrightarrow{s_{-i}}) \geq u_i(s_i', \overrightarrow{s_{-i}})$$

$$(\text{or, } c_i(s_i, \overrightarrow{s_{-i}}) \leq c_i(s_i', \overrightarrow{s_{-i}}))$$

Nash Equilibrium



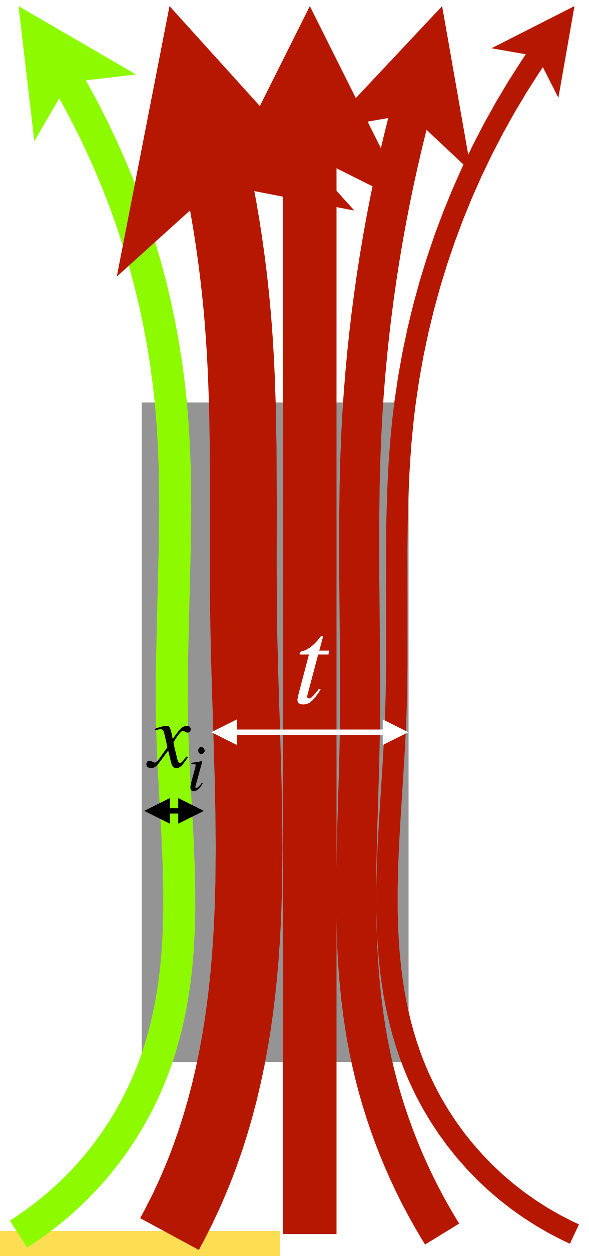
- Player i *(weakly) prefers* $\overrightarrow{s_x}$ to $\overrightarrow{s_y}$ if i prefers $\overrightarrow{s_x}$ to $\overrightarrow{s_y}$ or considers them as equally good outcomes. That is, $u_i(\overrightarrow{s_x}) \geq u_i(\overrightarrow{s_y})$
  - $\overrightarrow{s_{-i}} = (s_1, s_2, \cdots, s_{i-1}, s_{i+1}, \cdots, s_n)$
  - $\vec{s} = (s_1, s_2, \cdots, s_{i-1}, s_i, s_{i+1}, \cdots, s_n) = (s_i, \overrightarrow{s_{-i}})$
- A strategy vector $\vec{s}$ is a **Nash equilibrium** if

for all players i and each alternate strategy s_i':

$$u_i(s_i, \overrightarrow{s_{-i}}) \geq u_i(s_i', \overrightarrow{s_{-i}})$$

(or, $c_i(s_i, \overrightarrow{s_{-i}}) \leq c_i(s_i', \overrightarrow{s_{-i}})$)

Nash Equilibrium



- Player i *(weakly) prefers* $\overrightarrow{s_x}$ to $\overrightarrow{s_y}$ if i prefers $\overrightarrow{s_x}$ to $\overrightarrow{s_y}$ or considers them as equally good outcomes. That is, $u_i(\overrightarrow{s_x}) \geq u_i(\overrightarrow{s_y})$
  - $\overrightarrow{s_{-i}} = (s_1, s_2, \cdots, s_{i-1}, s_{i+1}, \cdots, s_n)$
  - $\vec{s} = (s_1, s_2, \cdots, s_{i-1}, s_i, s_{i+1}, \cdots, s_n) = (s_i, \overrightarrow{s_{-i}})$

- A strategy vector $\vec{s}$ is a **Nash equilibrium** if

for all players i and each alternate strategy s_i':

$$u_i(s_i, \overrightarrow{s_{-i}}) \geq u_i(s_i', \overrightarrow{s_{-i}})$$

(or, $c_i(s_i, \overrightarrow{s_{-i}}) \leq c_i(s_i', \overrightarrow{s_{-i}})$)

only change strategy from s_i to s_i' when s_i' is strictly better

What happened

- Nash equilibrium: The stable state that no player can improve its wellbeing by changing its own strategy (given others’ strategies don’t change)

Outline

- Fundamental concepts
    - Game, players, strategies, payoffs/costs
- Nash Equilibrium
- Price of Anarchy
    - Selfish load balancing
- Mechanism design
    - Auction
    - Vickrey-Clarke-Groves mechanism

Social Welfare/Cost

- Social choice: an aggregation of the preference of the different participants toward a single joint decision
- Let $\vec{s}$ be a preferences/strategies of the players
    - The social choice $f(\vec{s})$ is the action given $\vec{s}$, and it has a social welfare (or social cost)

Price of Anarchy (PoA)

- Measure the inefficiency of equilibria
- Given a game, let S_{NE} be its set of equilibria (all stable strategies), the **Price of Anarchy** is

$$\frac{\max_{\vec{s}} \text{Social Welfare}(\vec{s})}{\min_{\vec{s} \in S_{\text{NE}}} \text{Social Welfare}(\vec{s})}$$

or

$$\frac{\max_{\vec{s} \in S_{\text{NE}}} \text{Social Cost}(\vec{s})}{\min_{\vec{s}} \text{Social Cost}(\vec{s})}$$

Price of Anarchy (PoA)

- Measure the inefficiency of equilibria
- Given a game, let S_{NE} be its set of equilibria (all stable strategies), the **Price of Anarchy** is

$$\frac{\max_{\vec{s}} \text{Social Welfare}(\vec{s})}{\min_{\vec{s} \in S_{\text{NE}}} \text{Social Welfare}(\vec{s})}$$

or

$$\frac{\max_{\vec{s} \in S_{\text{NE}}} \text{Social Cost}(\vec{s})}{\min_{\vec{s}} \text{Social Cost}(\vec{s})}$$

	A	
B	confess	silent
confess	4 4	5 1
silent	1 5	2 2

Price of Anarchy (PoA)

- Measure the inefficiency of equilibria
- Given a game, let S_{NE} be its set of equilibria (all stable strategies), the **Price of Anarchy** is

$$\frac{\max_{\vec{s}} \text{Social Welfare}(\vec{s})}{\min_{\vec{s}\in S_{\text{NE}}} \text{Social Welfare}(\vec{s})}$$

or

$$\frac{\max_{\vec{s}\in S_{\text{NE}}} \text{Social Cost}(\vec{s})}{\min_{\vec{s}} \text{Social Cost}(\vec{s})}$$

B \ A	confess	silent
confess	4 4	5 1
silent	1 5	2 2

$$\text{PoA} = \frac{4+4}{2+2}$$

Price of Anarchy (PoA)

- Measure the inefficiency of equilibria
- Given a game, let S_{NE} be its set of equilibria (all stable strategies), the **Price of Anarchy** is

$$\frac{\max_{\vec{s}} \text{Social Welfare}(\vec{s})}{\min_{\vec{s} \in S_{\text{NE}}} \text{Social Welfare}(\vec{s})}$$

or

$$\frac{\max_{\vec{s} \in S_{\text{NE}}} \text{Social Cost}(\vec{s})}{\min_{\vec{s}} \text{Social Cost}(\vec{s})}$$

G \ B	baseball	softball
baseball	6 5	1 1
softball	2 2	5 6

Price of Anarchy (PoA)

- Measure the inefficiency of equilibria
- Given a game, let S_{NE} be its set of equilibria (all stable strategies), the **Price of Anarchy** is

$$\frac{\max_{\vec{s}} \text{Social Welfare}(\vec{s})}{\min_{\vec{s}\in S_{\text{NE}}} \text{Social Welfare}(\vec{s})}$$

or

$$\frac{\max_{\vec{s}\in S_{\text{NE}}} \text{Social Cost}(\vec{s})}{\min_{\vec{s}} \text{Social Cost}(\vec{s})}$$

B G	baseball	softball
baseball	6 5	1 1
softball	2 2	5 6

$$\text{PoA} = \frac{6+5}{5+6}$$

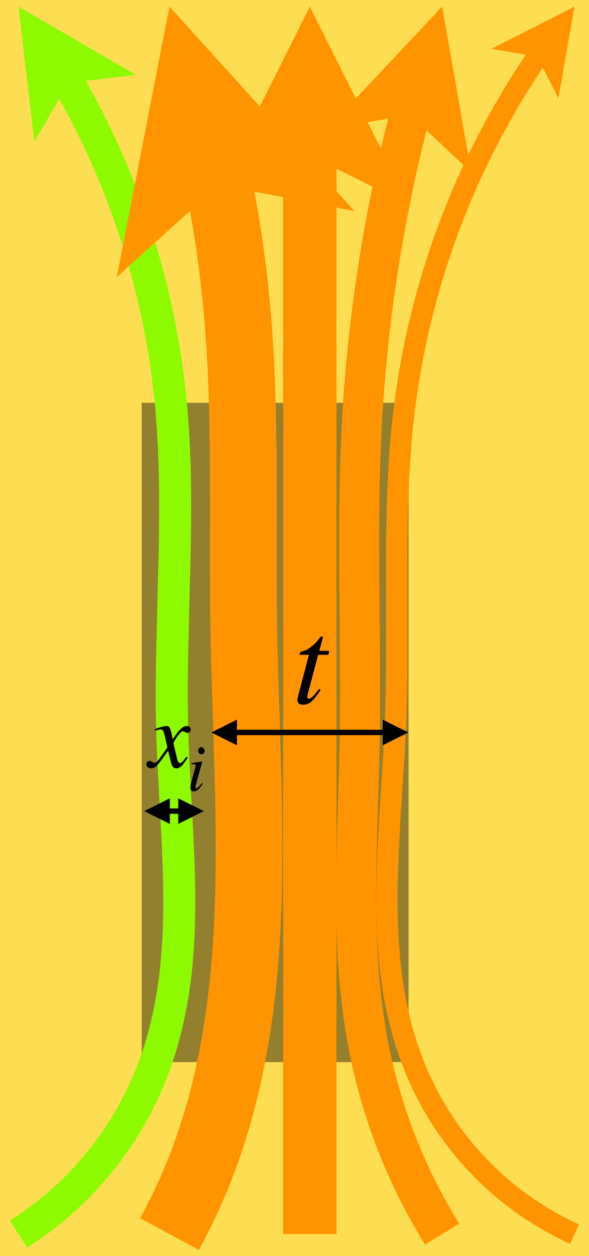
Price of Anarchy (PoA)

- Measure the inefficiency of equilibria
- Given a game, let S_{NE} be its set of equilibria (all stable strategies), the **Price of Anarchy** is

$$\frac{\max_{\vec{s}} \text{Social Welfare}(\vec{s})}{\min_{\vec{s}\in S_{\text{NE}}} \text{Social Welfare}(\vec{s})}$$

or

$$\frac{\max_{\vec{s}\in S_{\text{NE}}} \text{Social Cost}(\vec{s})}{\min_{\vec{s}} \text{Social Cost}(\vec{s})}$$



Tragedy of Commons — Better solution

- Selfish strategy: $x_i = \dfrac{1}{n+1}$ for all i
  - Total bandwidth used is $\dfrac{n}{n+1}$
  - For each player, the payoff is $x_i(1-\Sigma_j x_j) = \dfrac{1}{n+1}(1-\dfrac{n}{n+1}) = \dfrac{1}{(n+1)^2}$
- (Centralized) better strategy: if the total bandwidth used is $\dfrac{1}{2}\cdot\dfrac{n}{n+1}$:
  - $x_i = \dfrac{1}{2(n+1)}$ for each player i, and the payoff of each player is $\dfrac{1}{2(n+1)}\cdot(1-\dfrac{n}{2(n+1)}) = \dfrac{n+2}{4(n+1)^2}$
  - The new value is $\dfrac{n+2}{4}$ times the old value (!!)

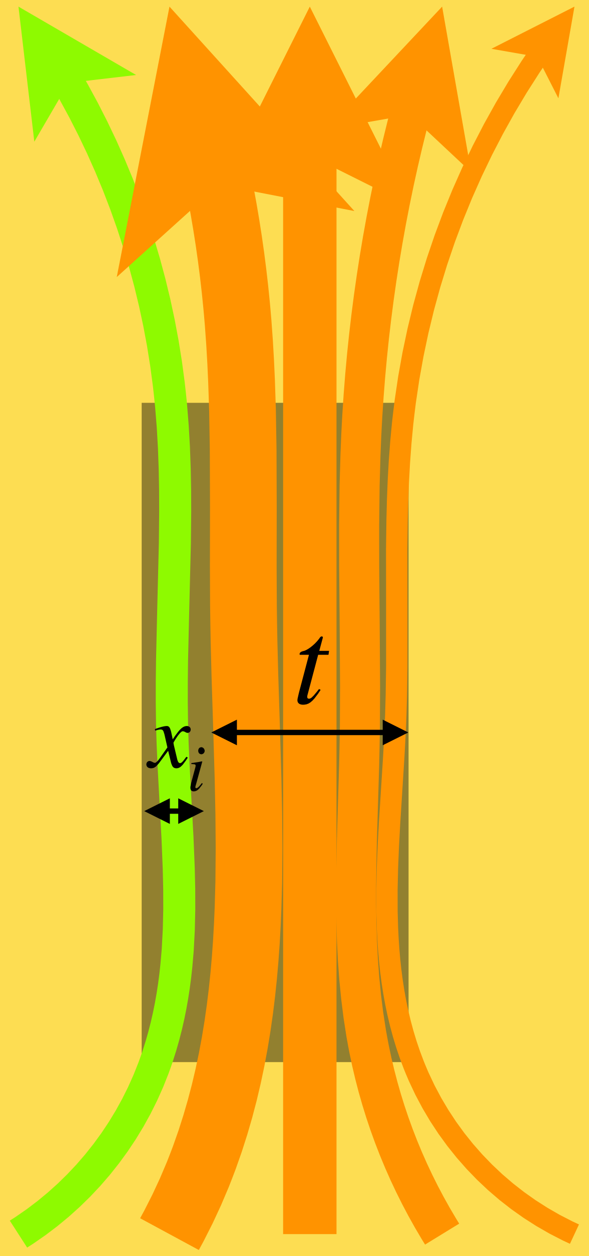
Price of Anarchy (PoA)

- Measure the inefficiency of equilibria
- Given a game, let S_{NE} be its set of equilibria (all stable strategies), the **Price of Anarchy** is

$$\frac{\max_{\vec{s}} \text{Social Welfare}(\vec{s})}{\min_{\vec{s} \in S_{\mathrm{NE}}} \text{Social Welfare}(\vec{s})}$$

or

$$\frac{\max_{\vec{s} \in S_{\mathrm{NE}}} \text{Social Cost}(\vec{s})}{\min_{\vec{s}} \text{Social Cost}(\vec{s})}$$



$$\text{PoA} \geq \frac{\frac{n(n+2)}{4(n+1)^2}}{\frac{n}{(n+1)^2}} = \frac{n+2}{4}$$

What happened

- Price of Anarchy measures the performance loss due to decentralization in the worst case

Outline

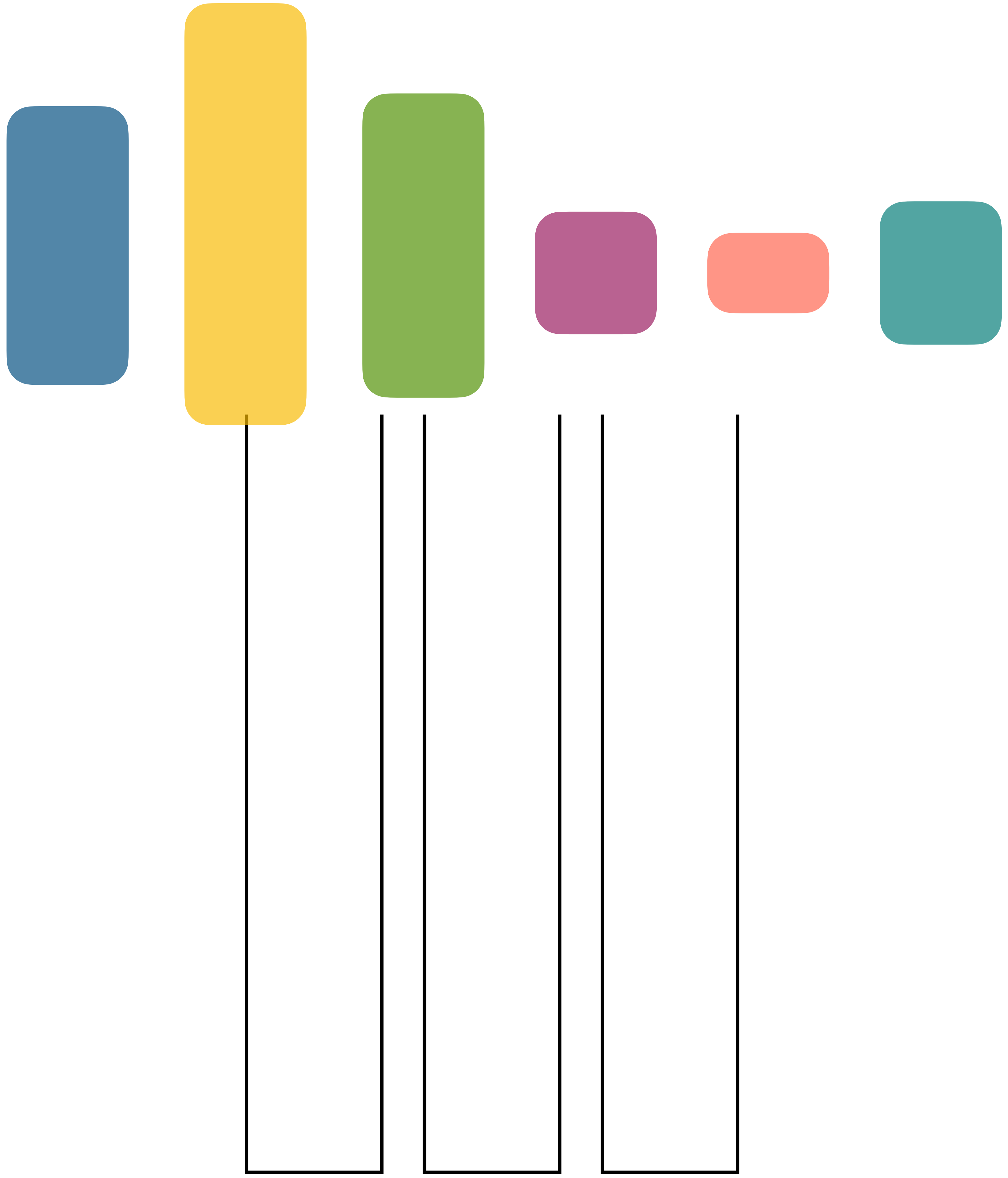
- Fundamental concepts
  - Game, players, strategies, payoffs/costs
- Nash Equilibrium
- Price of Anarchy
  - Selfish load balancing
- Mechanism design
  - Auction
  - Vickrey-Clarke-Groves mechanism

Load Balancing Game

- There are n jobs, each has processing time p_i and belongs to a self-interested player i

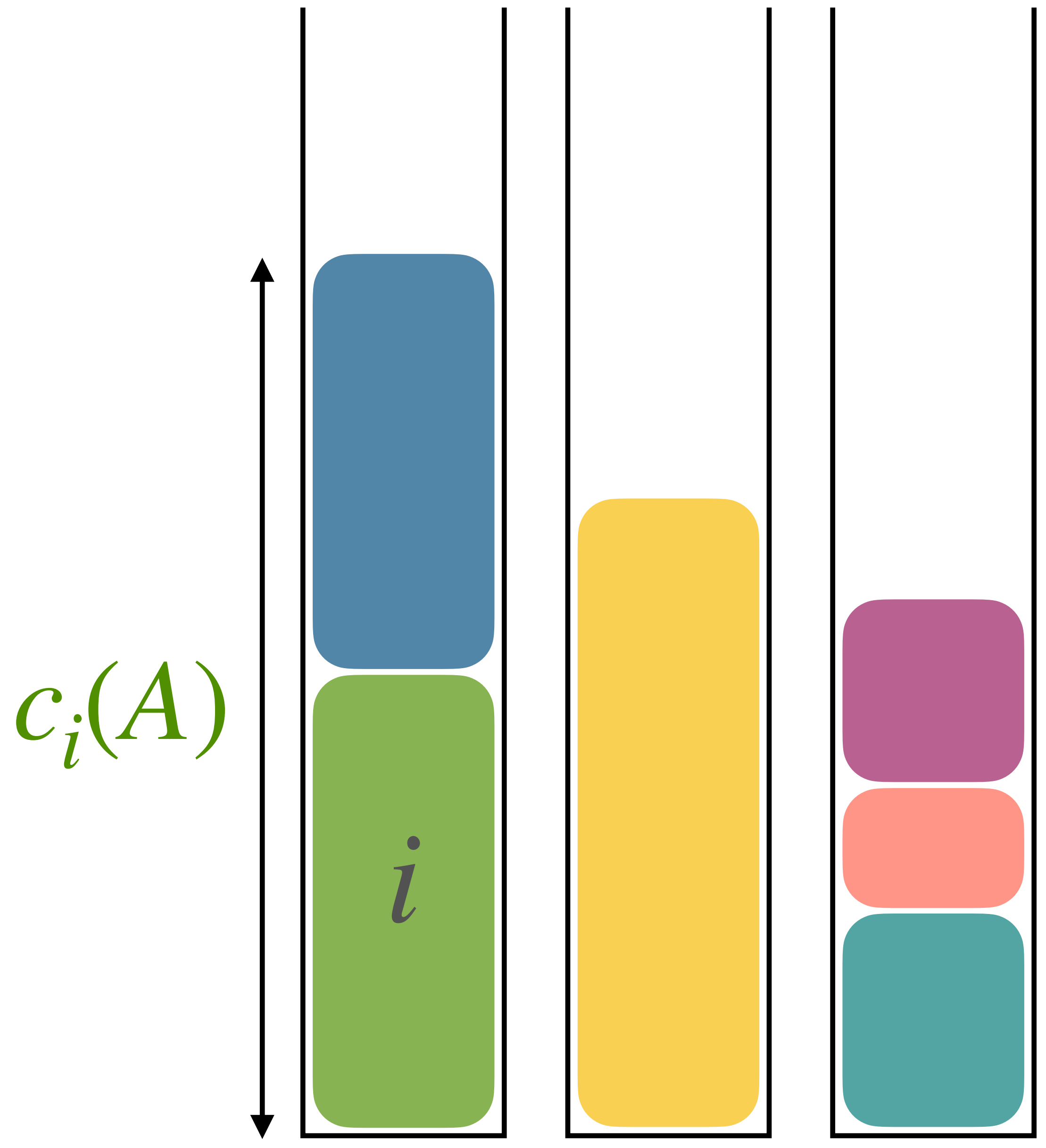
Load Balancing Game

- There are n jobs, each has processing time p_i and belongs to a self-interested player i
- There are m machines



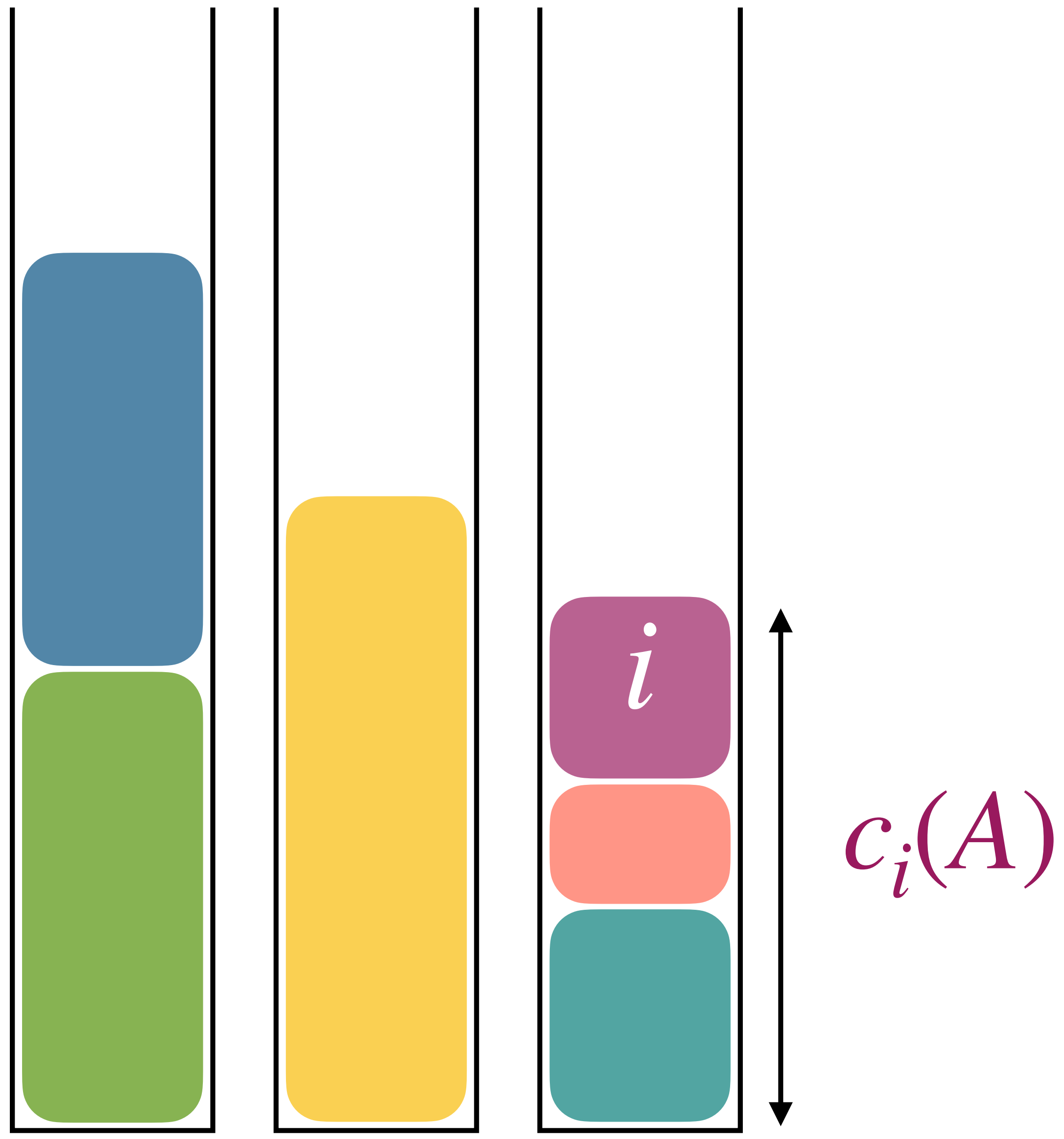
Load Balancing Game

- There are n jobs, each has processing time p_i and belongs to a self-interested player i
- There are m machines
- Game: the players want to schedule their jobs on the lowest-loaded machine
    - Load of machine k: $\ell_k = \Sigma_{j \text{ is assigned to machine } k}\, p_j$



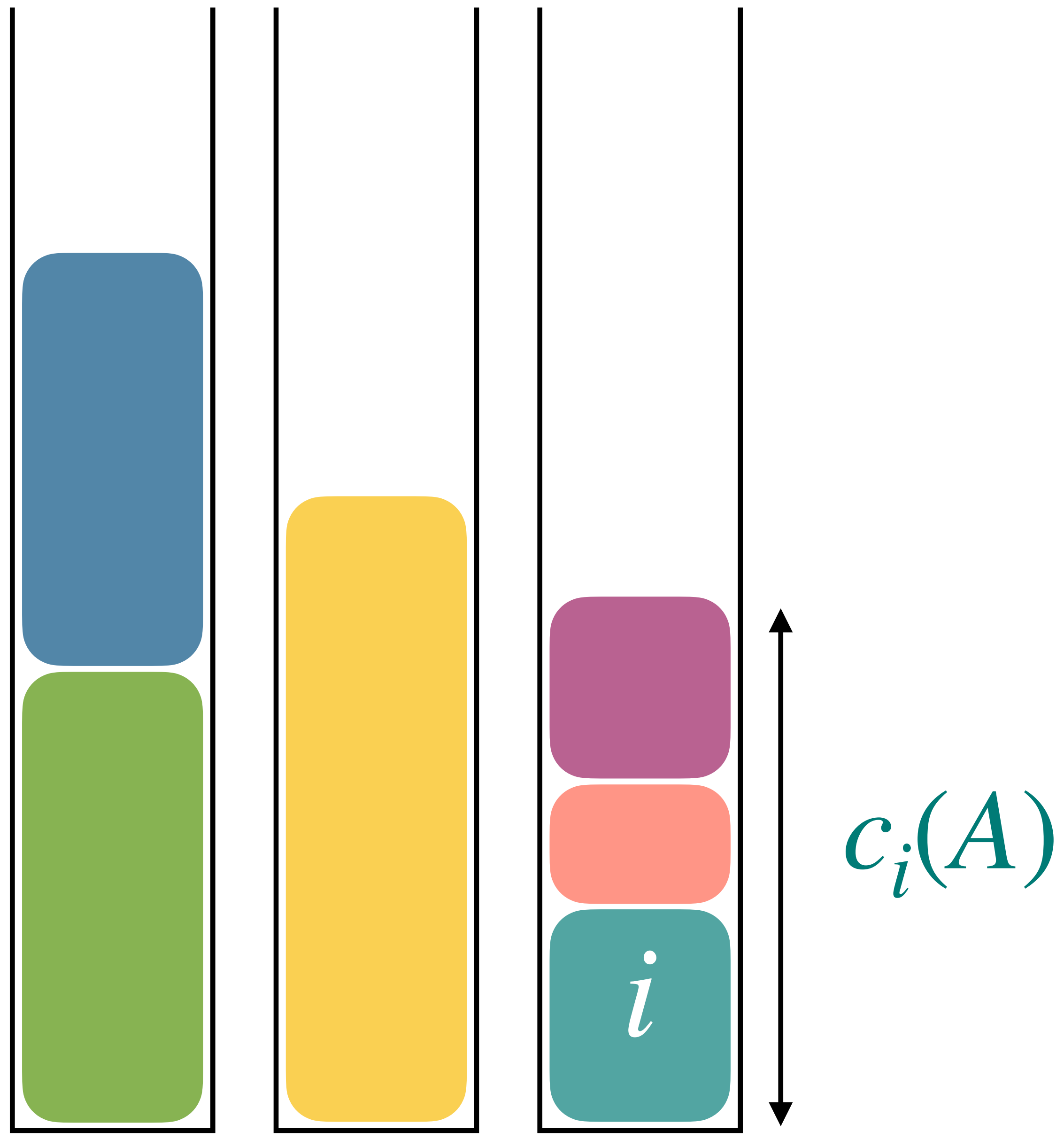
Load Balancing Game

- There are n jobs, each has processing time p_i and belongs to a self-interested player i
- There are m machines
- Game: the players want to schedule their jobs on the lowest-loaded machine
  - Load of machine k: $\ell_k = \Sigma_j$ is assigned to machine ${}_k p_j$



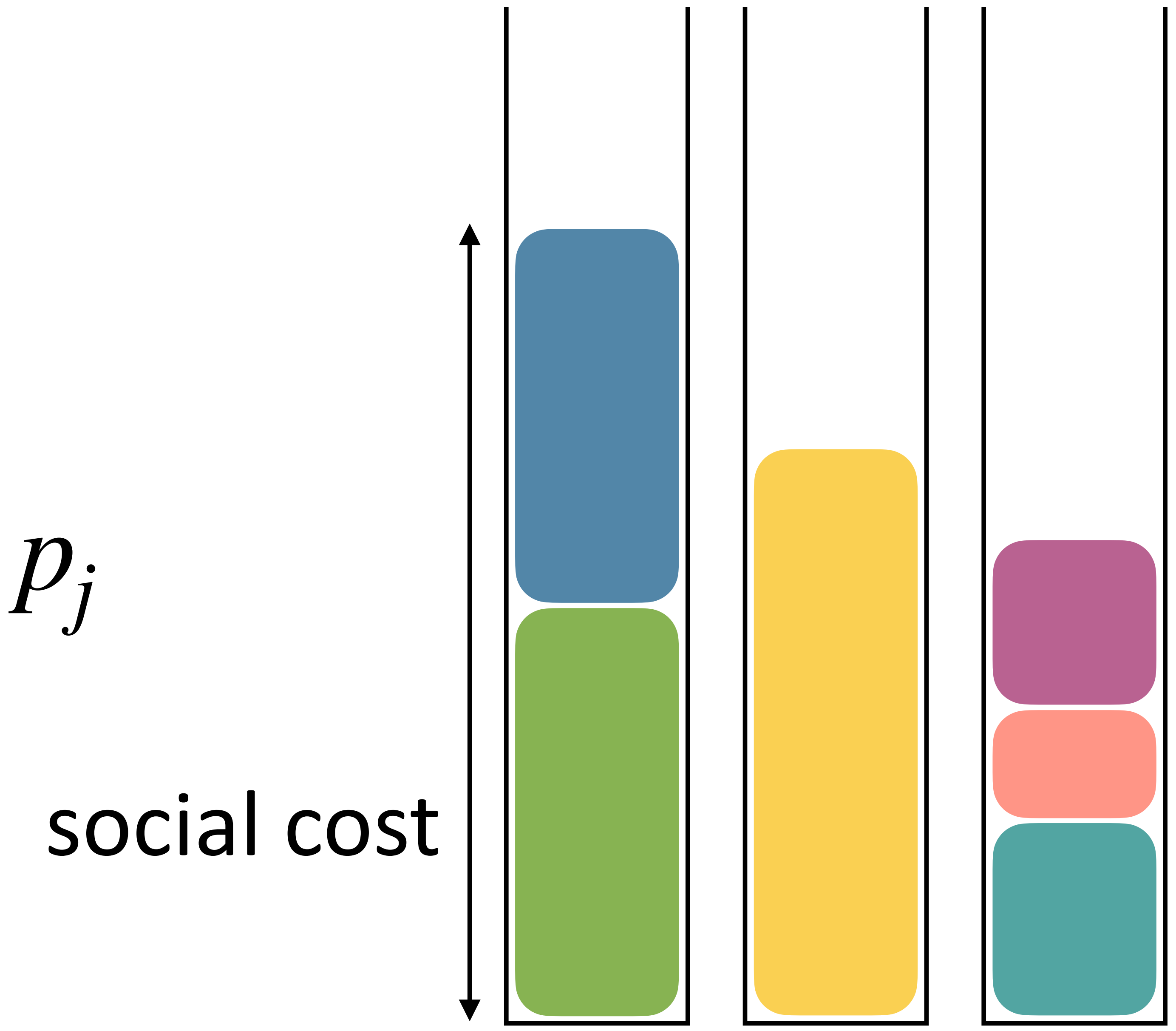
Load Balancing Game

- There are n jobs, each has processing time p_i and belongs to a self-interested player i
- There are m machines
- Game: the players want to schedule their jobs on the lowest-loaded machine
    - Load of machine k: $\ell_k = \Sigma_j$ is assigned to machine ${}_k p_j$



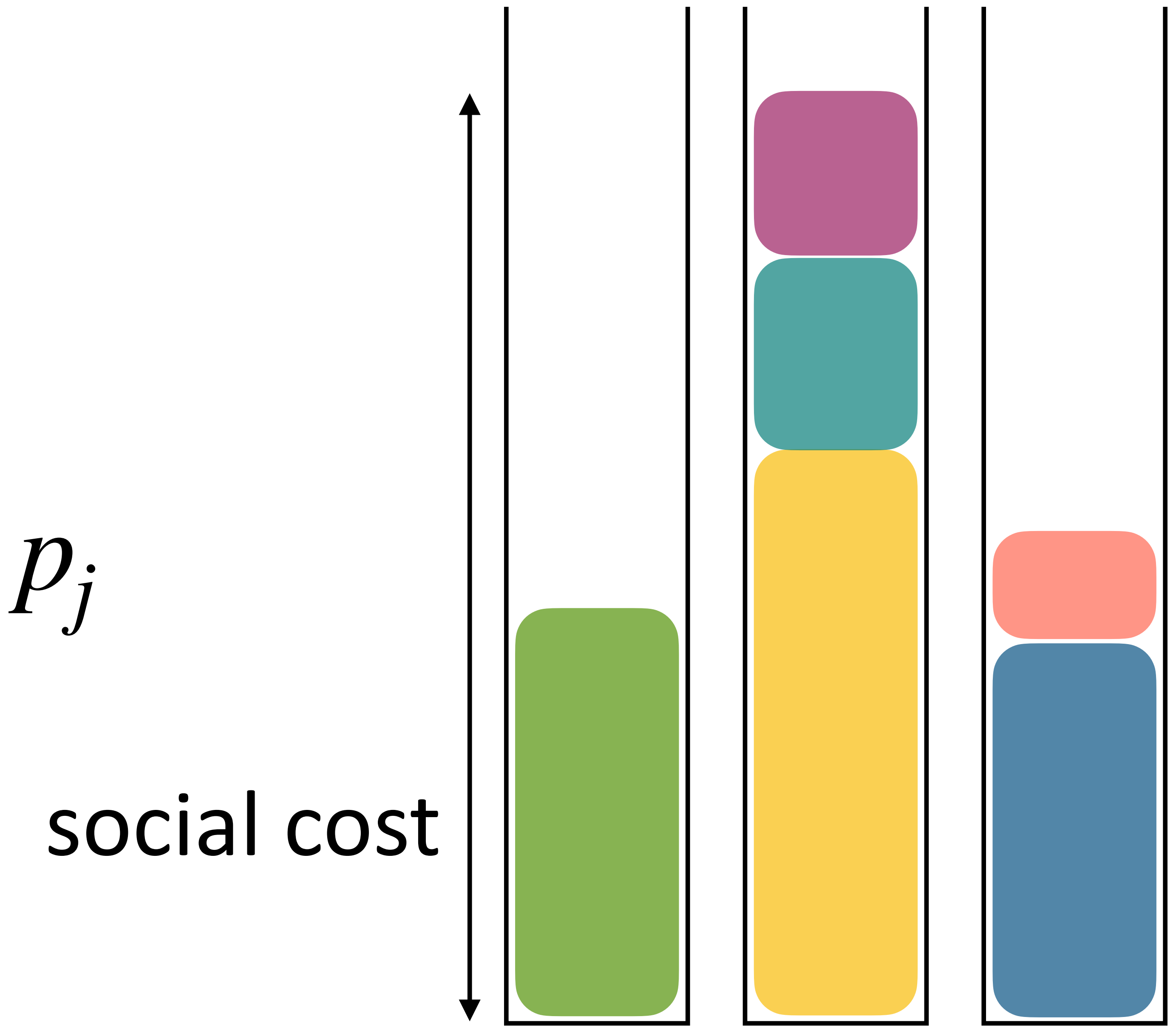
Load Balancing Game

- There are n jobs, each has processing time p_i and belongs to a self-interested player i
- There are m machines
- Game: the players want to schedule their jobs on the lowest-loaded machine
  - Load of machine k: $\ell_k = \Sigma_j$ is assigned to machine $_k p_j$
  - Social cost: $\max_k \ell_k$



Load Balancing Game

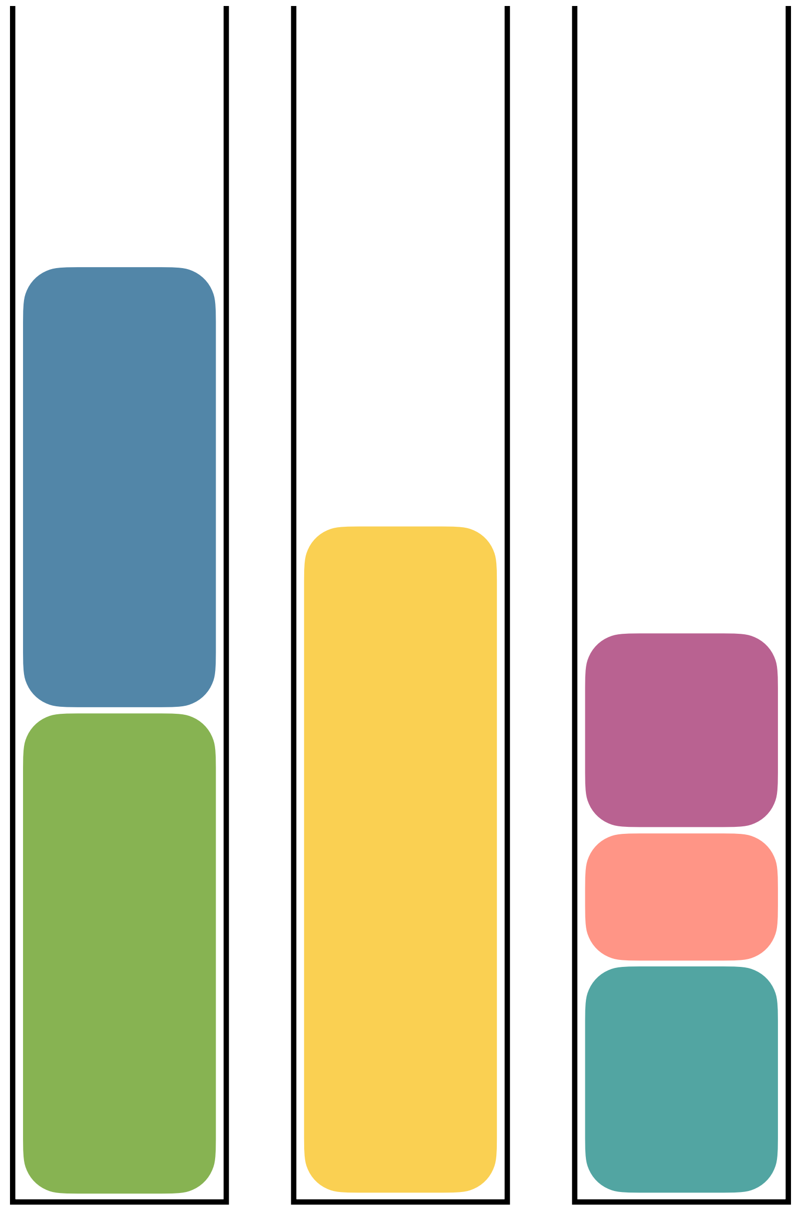
- There are n jobs, each has processing time p_i and belongs to a self-interested player i
- There are m machines
- Game: the players want to schedule their jobs on the lowest-loaded machine
    - Load of machine k: $\ell_k = \Sigma_j$ is assigned to machine $_k p_j$
    - Social cost: $\max_k \ell_k$



Load Balancing Game

- There are n jobs, each has processing time p_i and belongs to a self-interested player i
- There are m machines
- Game: the players want to schedule their jobs on the lowest-loaded machine
  - Load of machine k: $\ell_k = \Sigma_j$ is assigned to machine $_k p_j$
  - Social cost: $\max_k \ell_k$
- An assignment A is a Nash equilibrium if and only if

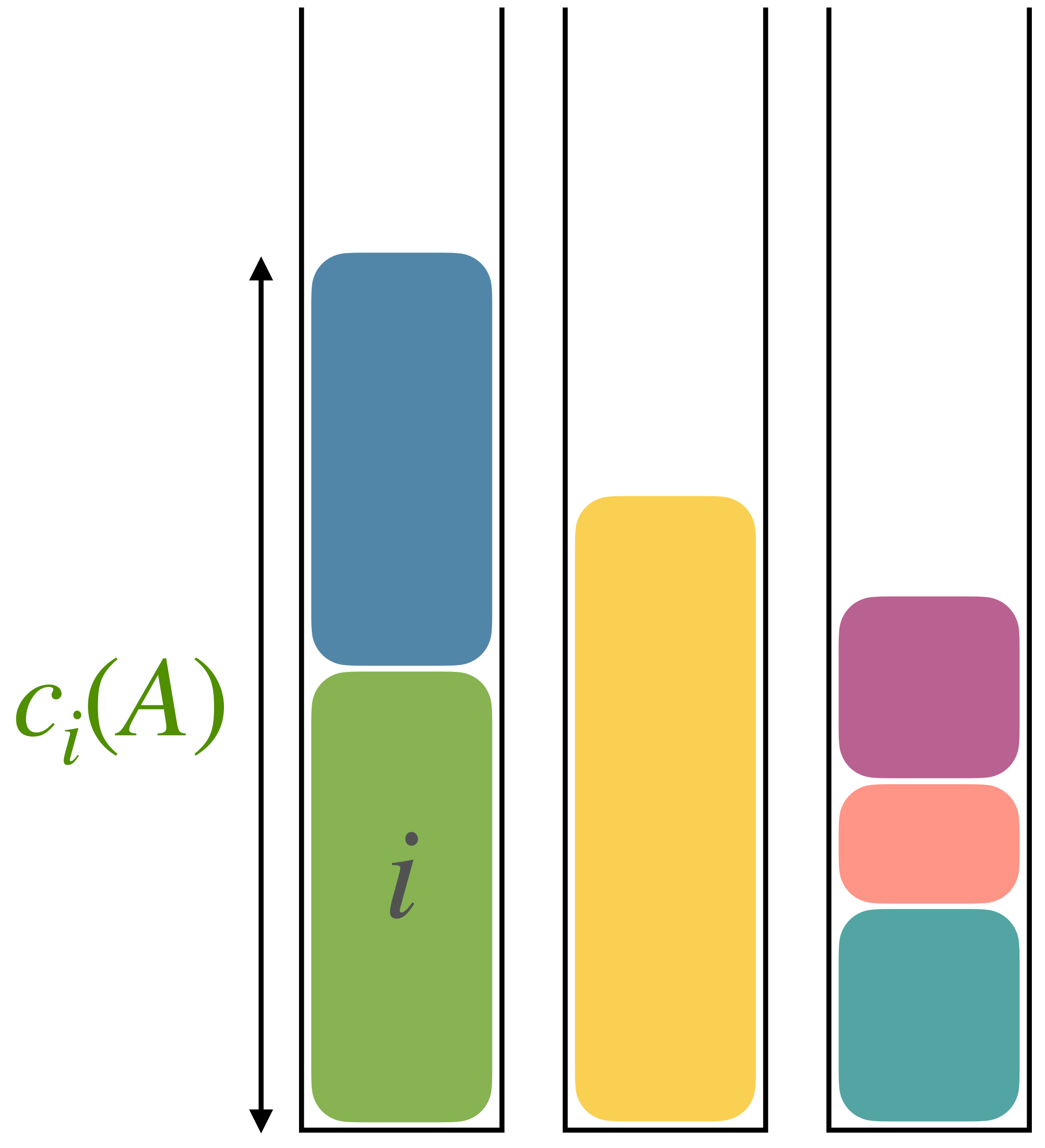
  for all i, $\ell_{A(i)} \leq \ell_k$ for any k



Load Balancing Game

- There are n jobs, each has processing time p_i and belongs to a self-interested player i
- There are m machines
- Game: the players want to schedule their jobs on the lowest-loaded machine
  - Load of machine k: $\ell_k = \Sigma_j$ is assigned to machine $_k\, p_j$
  - Social cost: $\max_k \ell_k$
- An assignment A is a Nash equilibrium if and only if

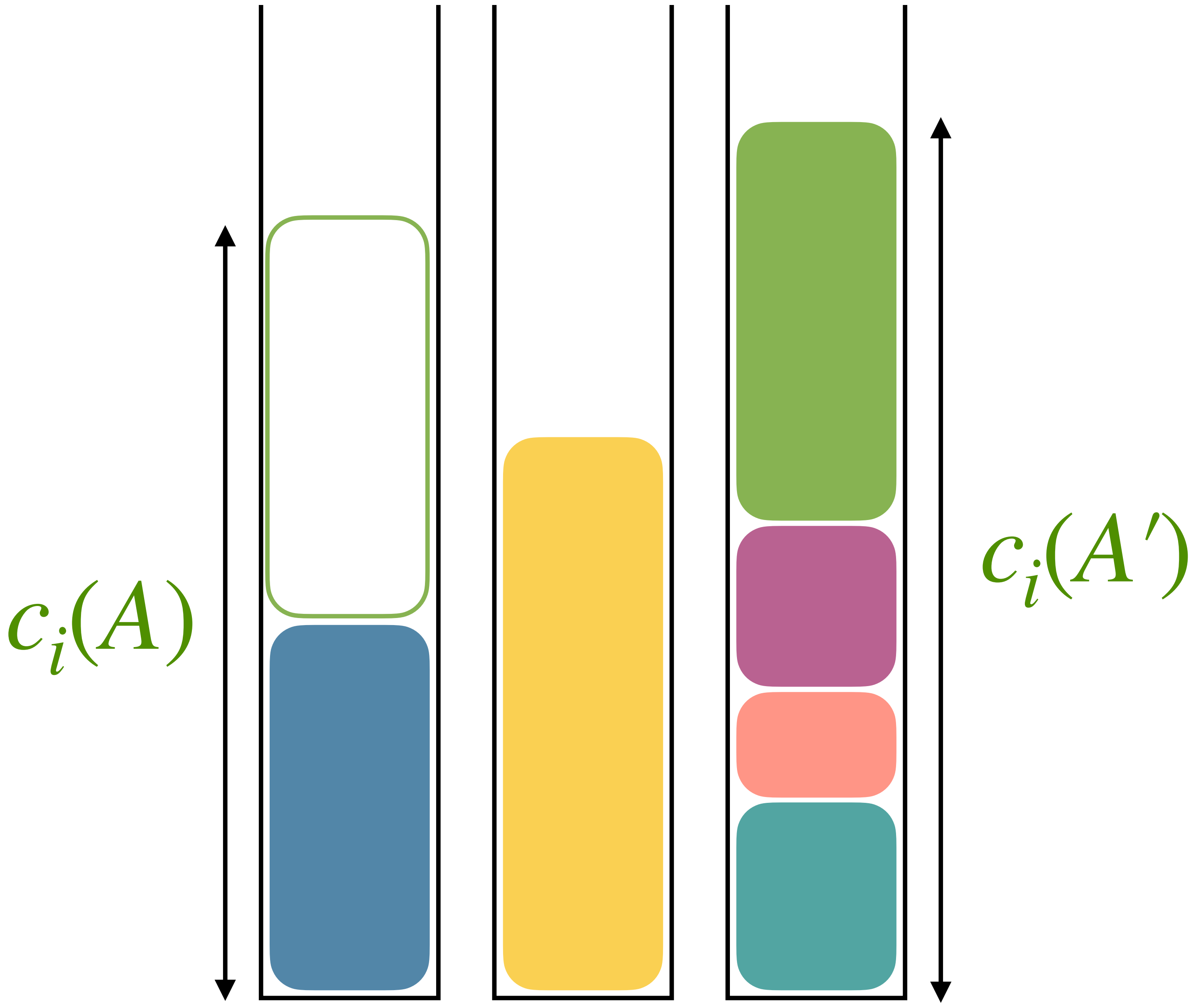
for all i, $\ell_{A(i)} \leq \ell_k$ for any k



Load Balancing Game

- There are n jobs, each has processing time p_i and belongs to a self-interested player i
- There are m machines
- Game: the players want to schedule their jobs on the lowest-loaded machine
  - Load of machine k: $\ell_k = \Sigma_j$ is assigned to machine $_k\, p_j$
  - Social cost: $\max_k \ell_k$
- An assignment A is a Nash equilibrium if and only if

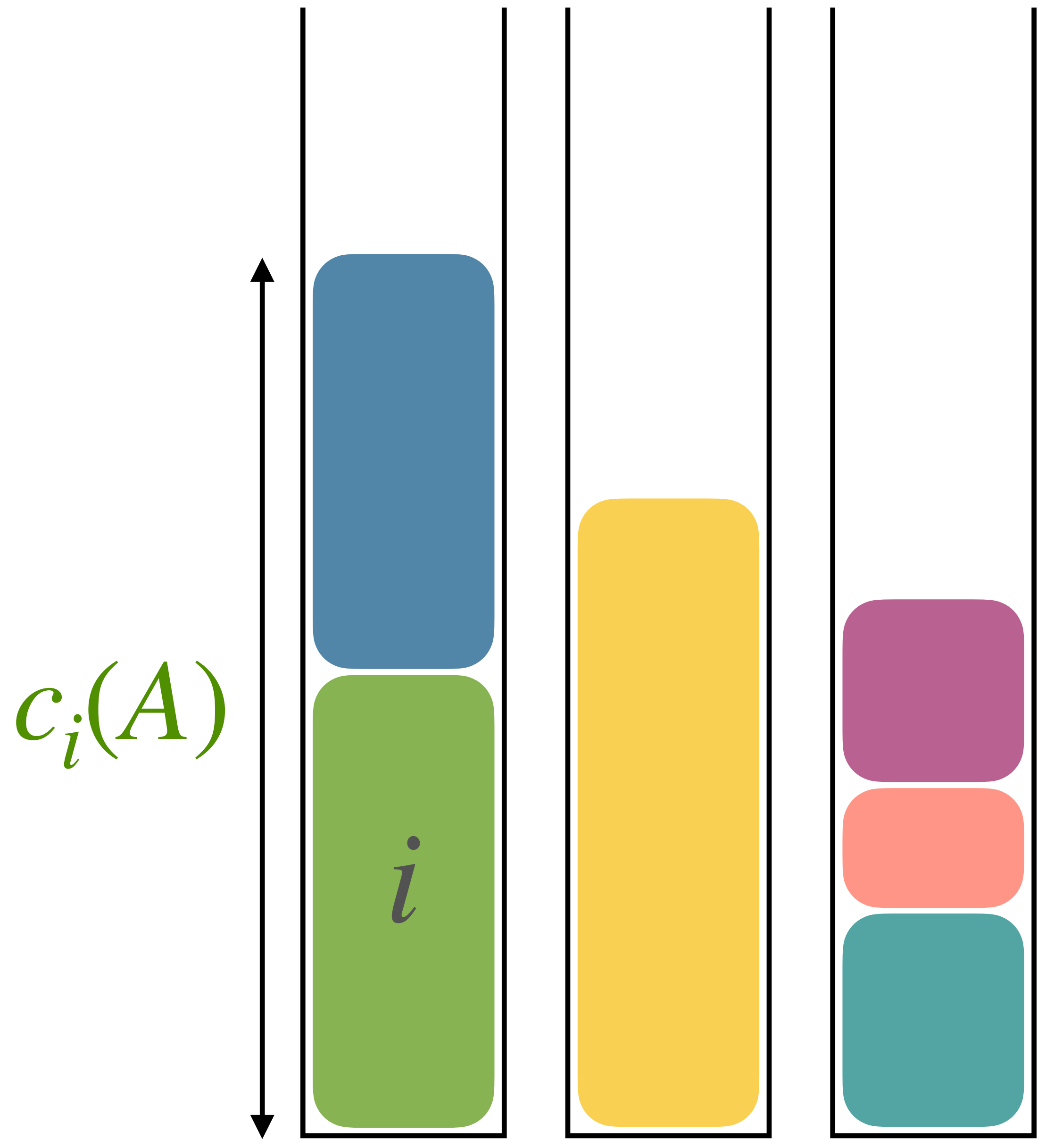
for all i, $\ell_{A(i)} \leq \ell_k$ for any k



Load Balancing Game

- There are n jobs, each has processing time p_i and belongs to a self-interested player i
- There are m machines
- Game: the players want to schedule their jobs on the lowest-loaded machine
  - Load of machine k: $\ell_k = \Sigma_j$ is assigned to machine $_k p_j$
  - Social cost: $\max_k \ell_k$
- An assignment A is a Nash equilibrium if and only if

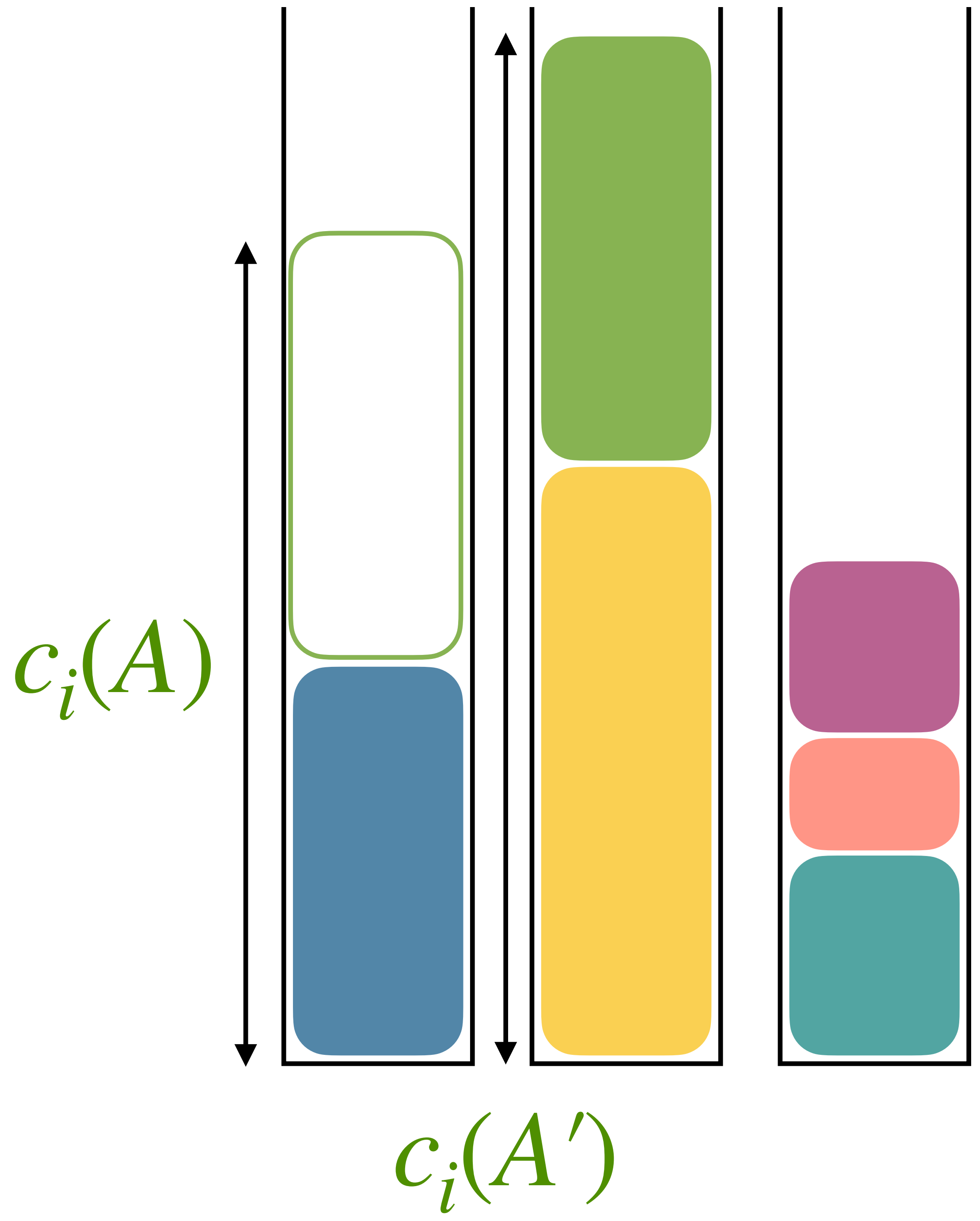
for all i, $\ell_{A(i)} \leq \ell_k$ for any k



Load Balancing Game

- There are n jobs, each has processing time p_i and belongs to a self-interested player i
- There are m machines
- Game: the players want to schedule their jobs on the lowest-loaded machine
  - Load of machine k: $\ell_k = \Sigma_j$ is assigned to machine ${}_k p_j$
  - Social cost: $\max_k \ell_k$
- An assignment A is a Nash equilibrium if and only if

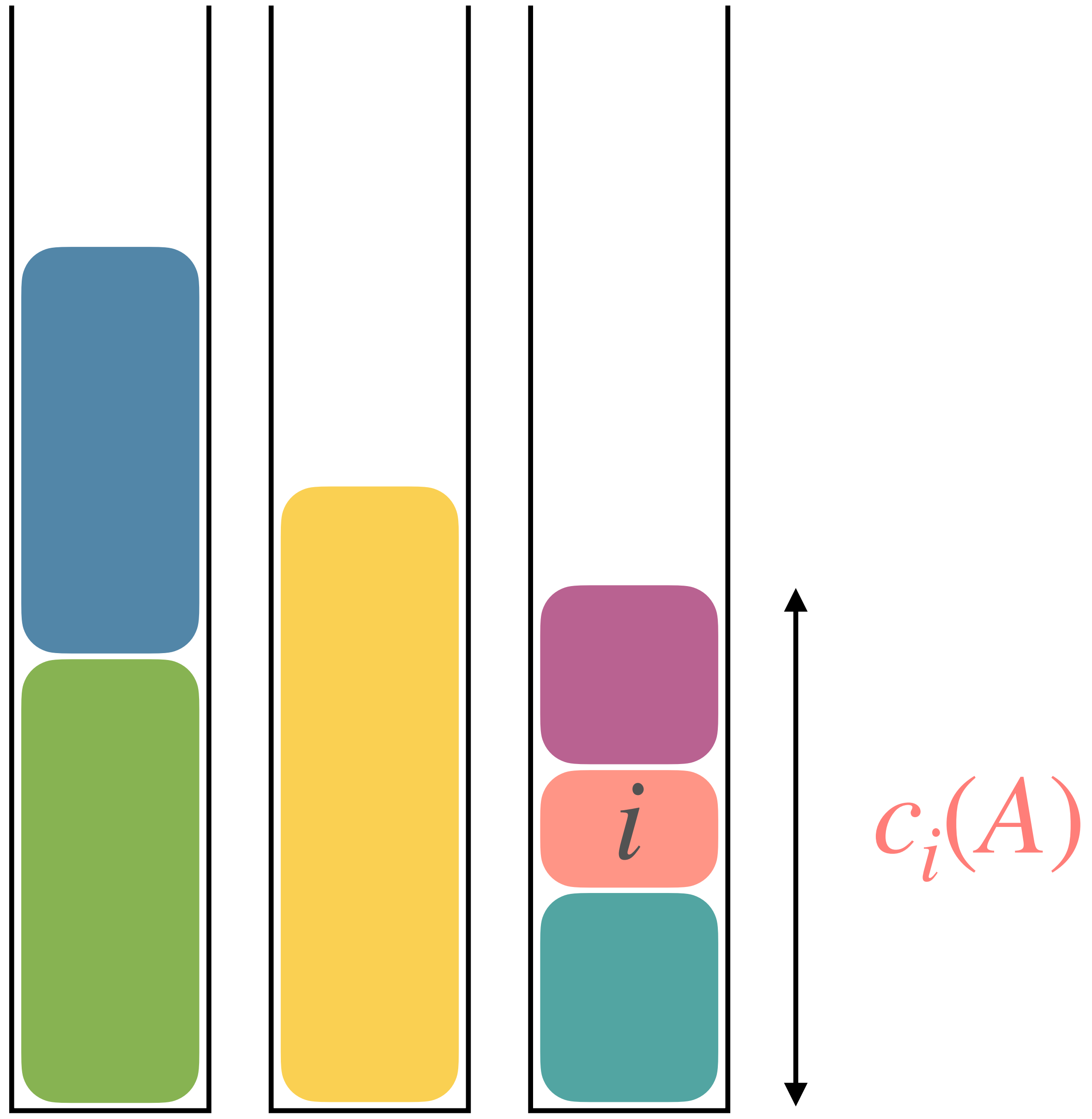
for all i, $\ell_{A(i)} \leq \ell_k$ for any k



Load Balancing Game

- There are n jobs, each has processing time p_i and belongs to a self-interested player i
- There are m machines
- Game: the players want to schedule their jobs on the lowest-loaded machine
  - Load of machine k: $\ell_k = \Sigma_j$ is assigned to machine $_k\, p_j$
  - Social cost: $\max_k \ell_k$
- An assignment A is a Nash equilibrium if and only if

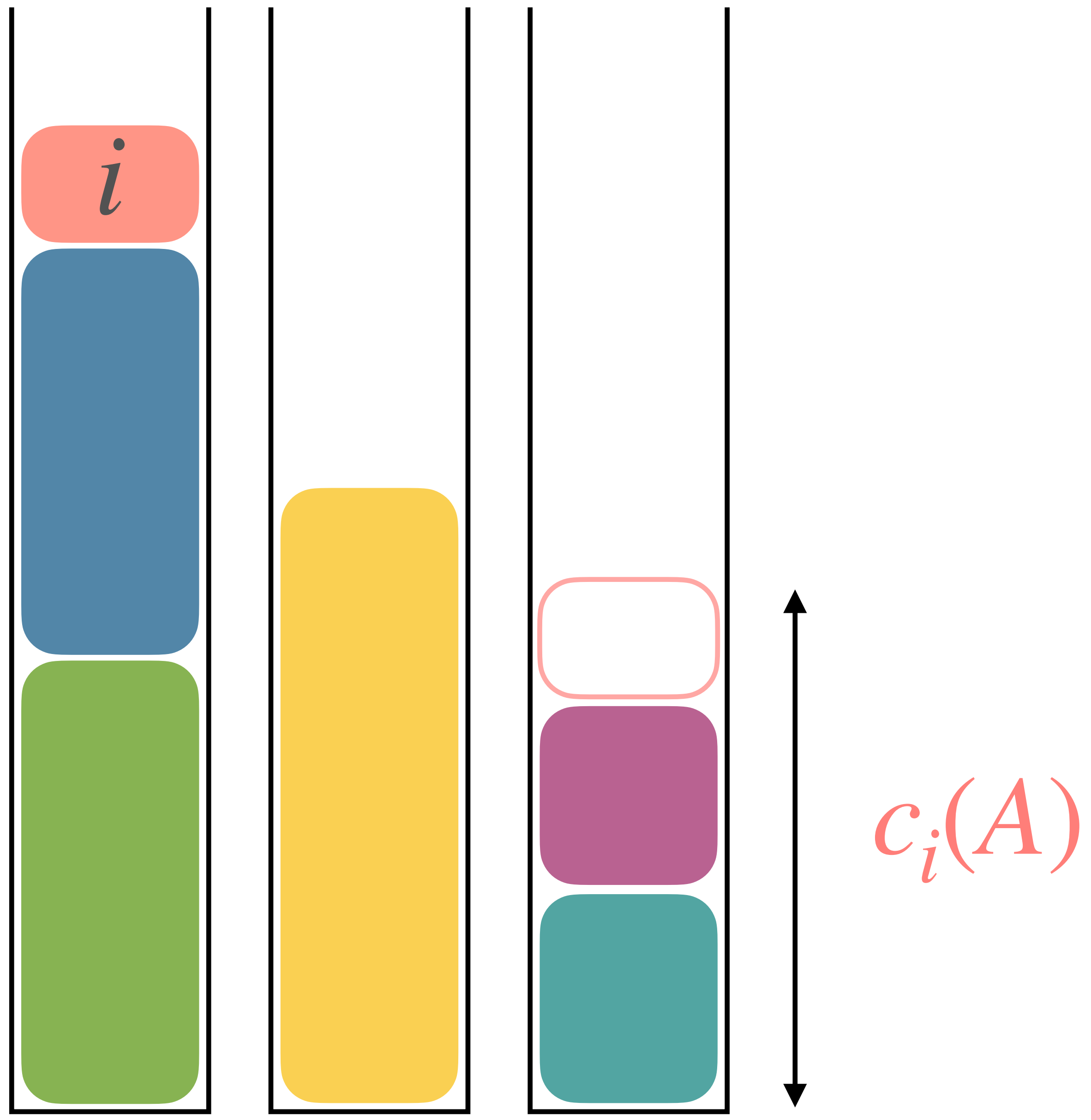
for all i, $\ell_{A(i)} \leq \ell_k$ for any k



Load Balancing Game

- There are n jobs, each has processing time p_i and belongs to a self-interested player i
- There are m machines
- Game: the players want to schedule their jobs on the lowest-loaded machine
  - Load of machine k: $\ell_k = \Sigma_j$ is assigned to machine $_k p_j$
  - Social cost: $\max_k \ell_k$
- An assignment A is a Nash equilibrium if and only if

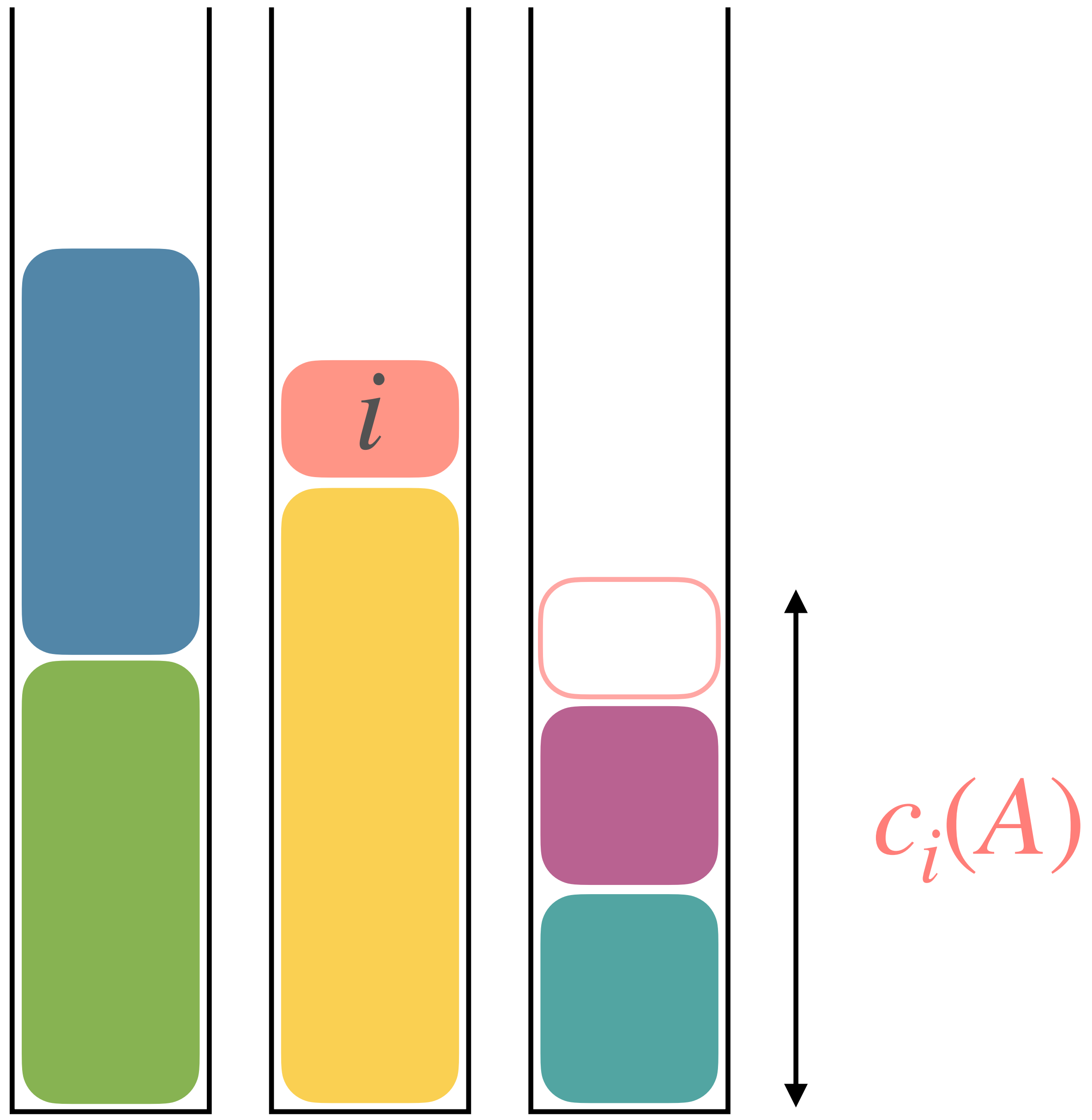
for all i, $\ell_{A(i)} \leq \ell_k$ for any k



Load Balancing Game

- There are n jobs, each has processing time p_i and belongs to a self-interested player i
- There are m machines
- Game: the players want to schedule their jobs on the lowest-loaded machine
  - Load of machine k: $\ell_k = \Sigma_j$ is assigned to machine $_k p_j$
  - Social cost: $\max_k \ell_k$
- An assignment A is a Nash equilibrium if and only if

  for all i, $\ell_{A(i)} \leq \ell_k$ for any k



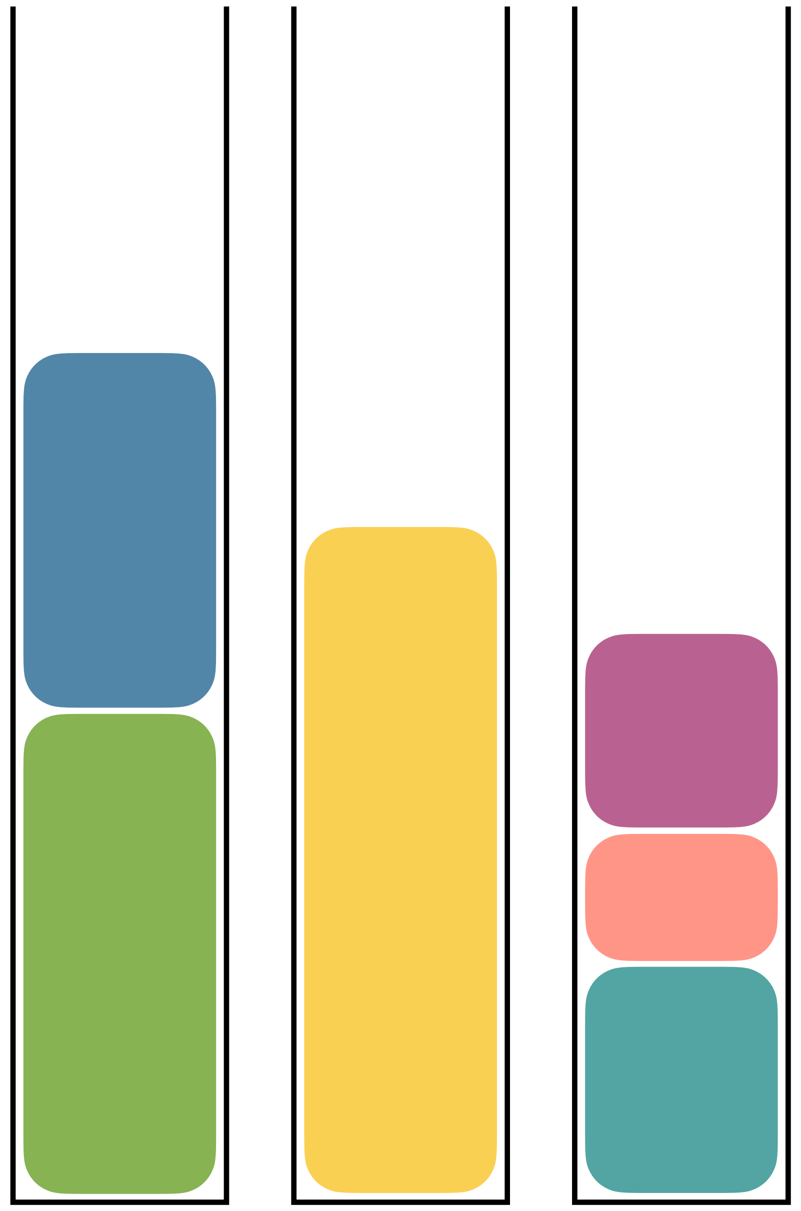
PoA of the Load Balancing Game

- Consider any instance of the load balancing game with n jobs of processing time $p_1, \cdots, p_n$ and m machines. Let A denote any Nash equilibrium assignment. Then, it holds that

$$\frac{cost(A)}{cost(\text{OPT})} \leq 2 - \frac{2}{m+1}$$

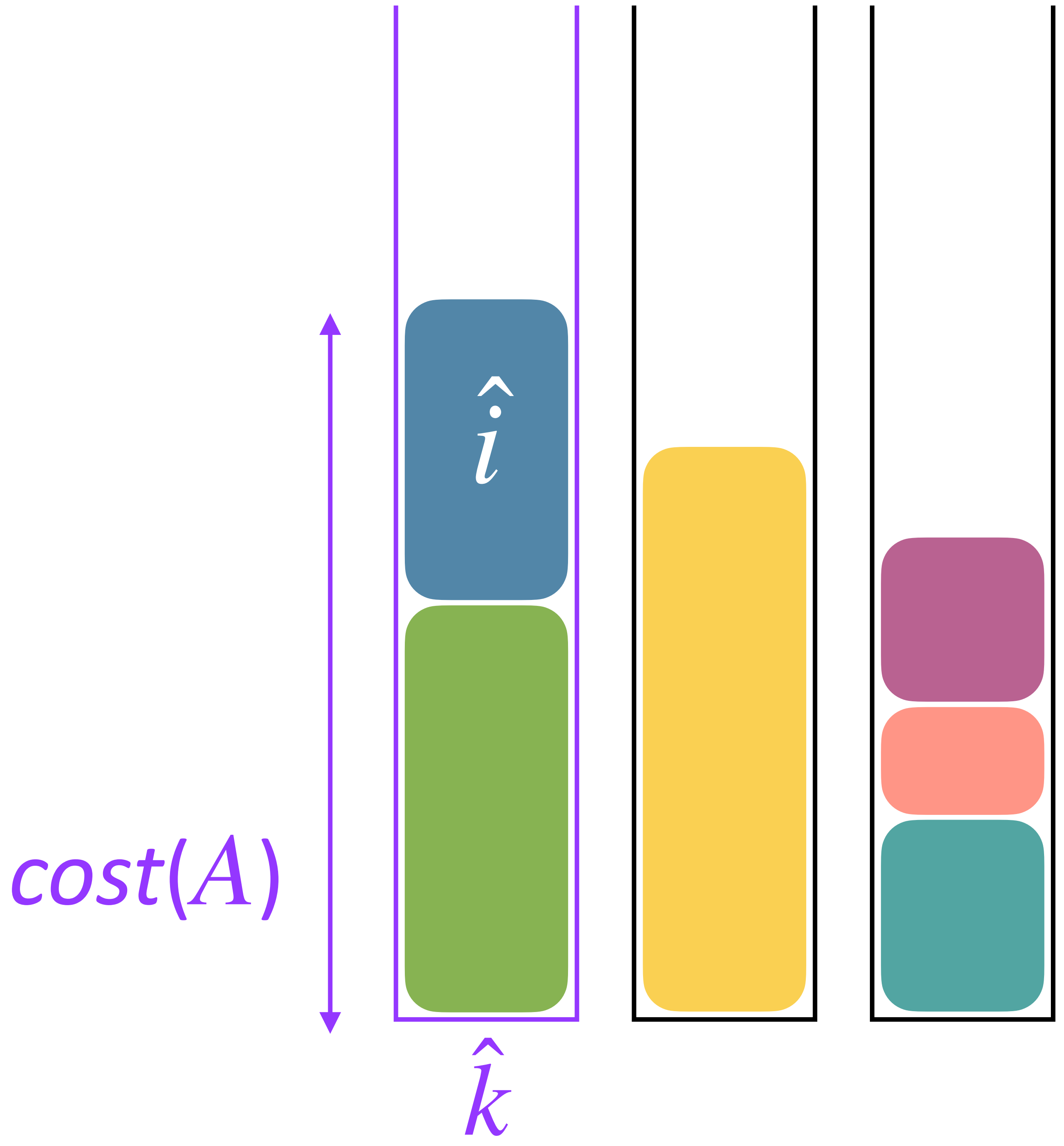
PoA of the Load Balancing Game

- Let $\hat{k}$ be the machine with the highest load under assignment A and $\hat{i}$ is the smallest job on $\hat{k}$



PoA of the Load Balancing Game

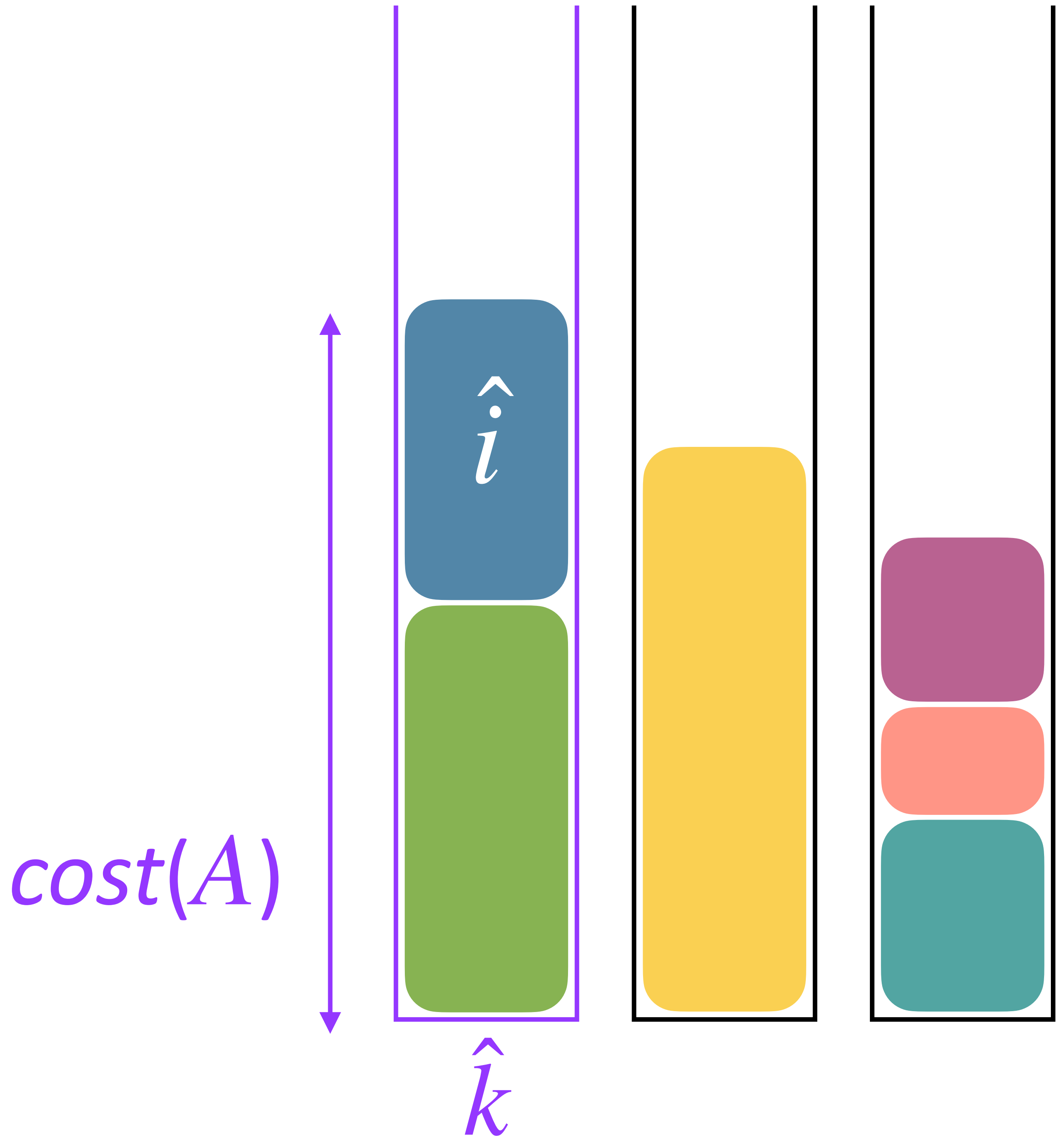
- Let $\hat{k}$ be the machine with the highest load under assignment A and $\hat{i}$ is the smallest job on $\hat{k}$



PoA of the Load Balancing Game

- Let $\hat{k}$ be the machine with the highest load under assignment A and $\hat{i}$ is the smallest job on $\hat{k}$
  - Without loss of generality, there are at least two tasks on $\hat{k}$ (otherwise, *cost*(OPT) $=$ *cost*(A) and the theorem is proven)

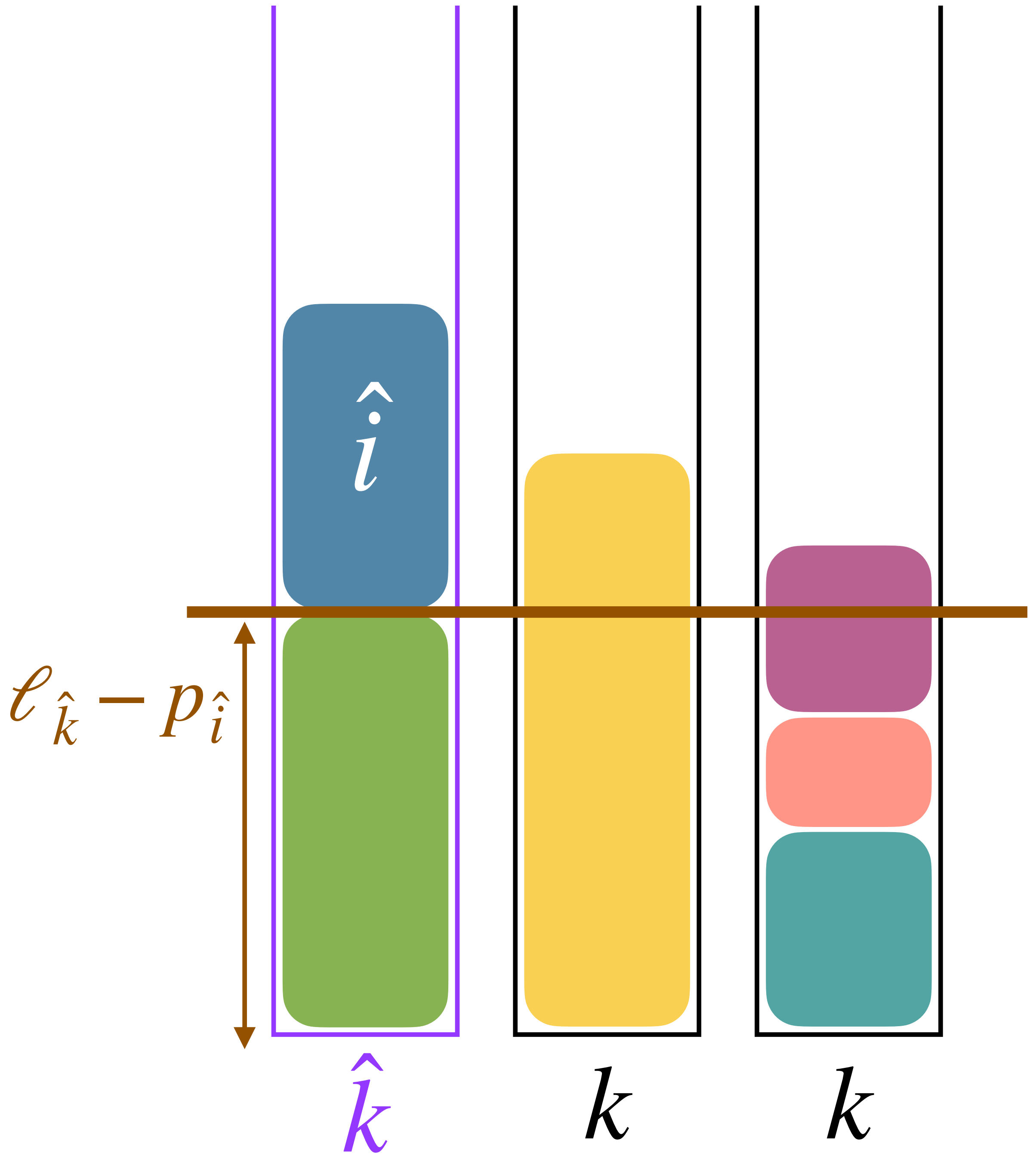
    $\Rightarrow p_{\hat{i}} \leq \dfrac{cost(A)}{2}$



PoA of the Load Balancing Game

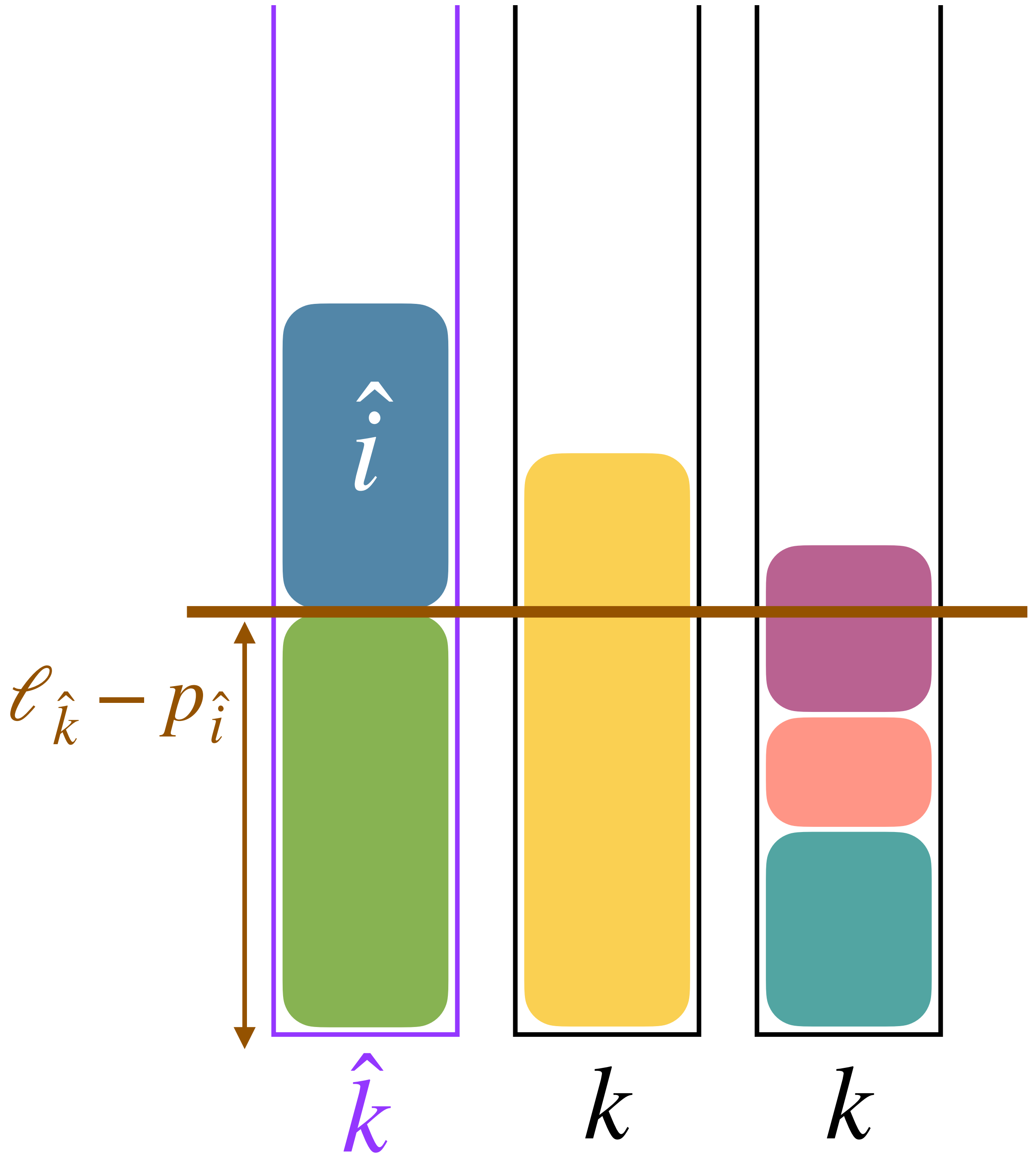
- Let $\hat{k}$ be the machine with the highest load under assignment A and $\hat{i}$ is the smallest job on $\hat{k}$
  - Without loss of generality, there are at least two tasks on $\hat{k}$ (otherwise, $cost(\text{OPT}) = cost(A)$ and the theorem is proven)

    $\Rightarrow p_{\hat{i}} \leq \dfrac{cost(A)}{2}$
  - Suppose there is a machine $k \neq \hat{k}$ with load less than $\ell_{\hat{k}} - p_{\hat{i}}$. Then, moving job $\hat{i}$ from $\hat{k}$ to k would decrease the cost of the agent



PoA of the Load Balancing Game

- Let $\hat{k}$ be the machine with the highest load under assignment A and $\hat{i}$ is the smallest job on $\hat{k}$
  - Without loss of generality, there are at least two tasks on $\hat{k}$ (otherwise, *cost*(OPT) $=$ *cost*(A) and the theorem is proven) $\Rightarrow p_{\hat{i}} \leq \dfrac{cost(A)}{2}$
  - Suppose there is a machine $k \neq \hat{k}$ with load less than $\ell_{\hat{k}} - p_{\hat{i}}$. Then, moving job $\hat{i}$ from $\hat{k}$ to k would decrease the cost of the agent
    - Since A is a Nash equilibrium, this cannot happen

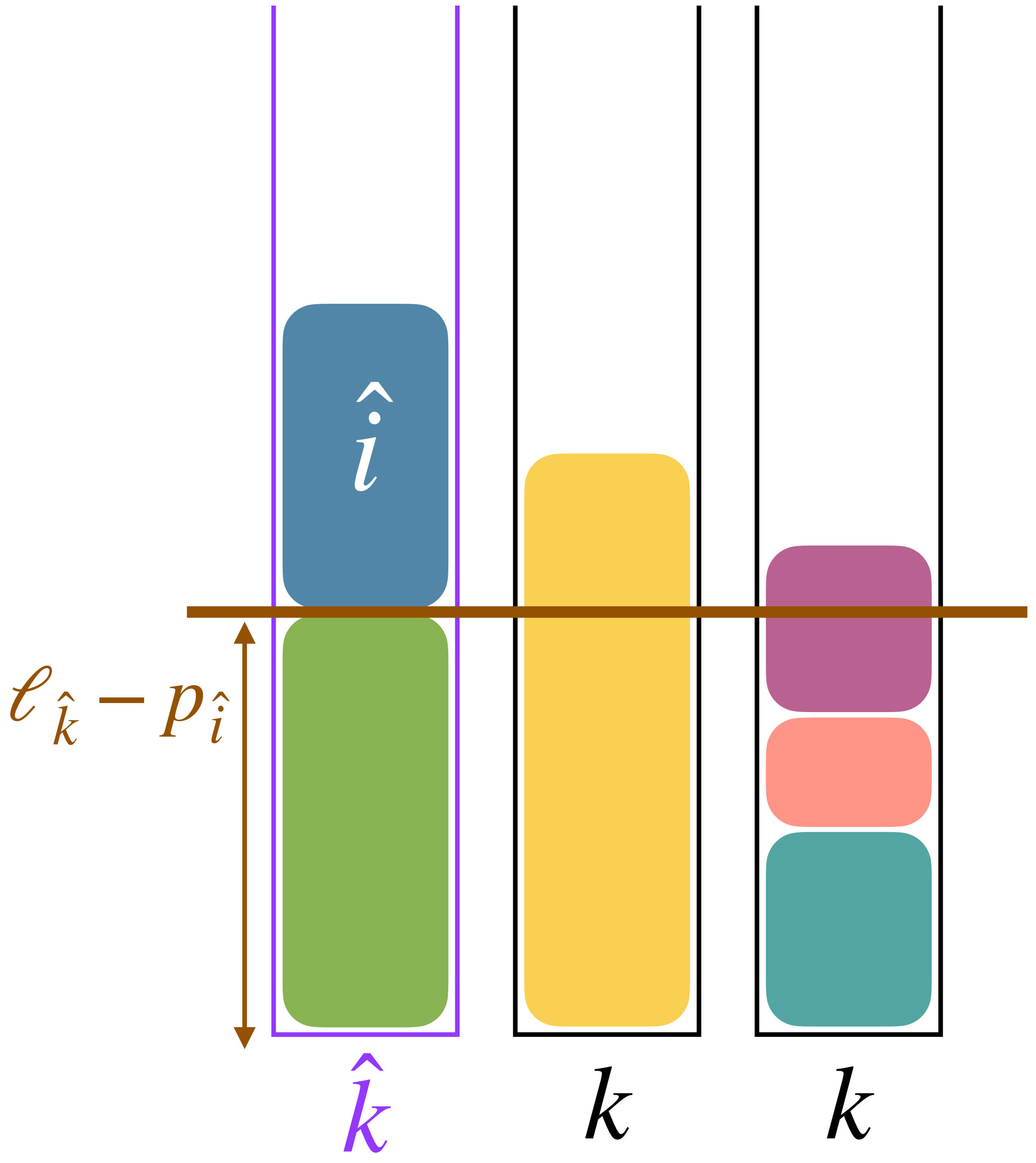


PoA of the Load Balancing Game

- Let $\hat{k}$ be the machine with the highest load under assignment A and $\hat{i}$ is the smallest job on $\hat{k}$
  - Without loss of generality, there are at least two tasks on $\hat{k}$ (otherwise, *cost*(OPT) $=$ *cost*(A) and the theorem is proven)

    $\Rightarrow p_{\hat{i}} \leq \dfrac{cost(A)}{2}$
  - Suppose there is a machine $k \neq \hat{k}$ with load less than $\ell_{\hat{k}} - p_{\hat{i}}$. Then, moving job $\hat{i}$ from $\hat{k}$ to k would decrease the cost of the agent
    - Since A is a Nash equilibrium, this cannot happen

      $\Rightarrow \ell_k \geq \ell_{\hat{k}} - p_{\hat{i}}$



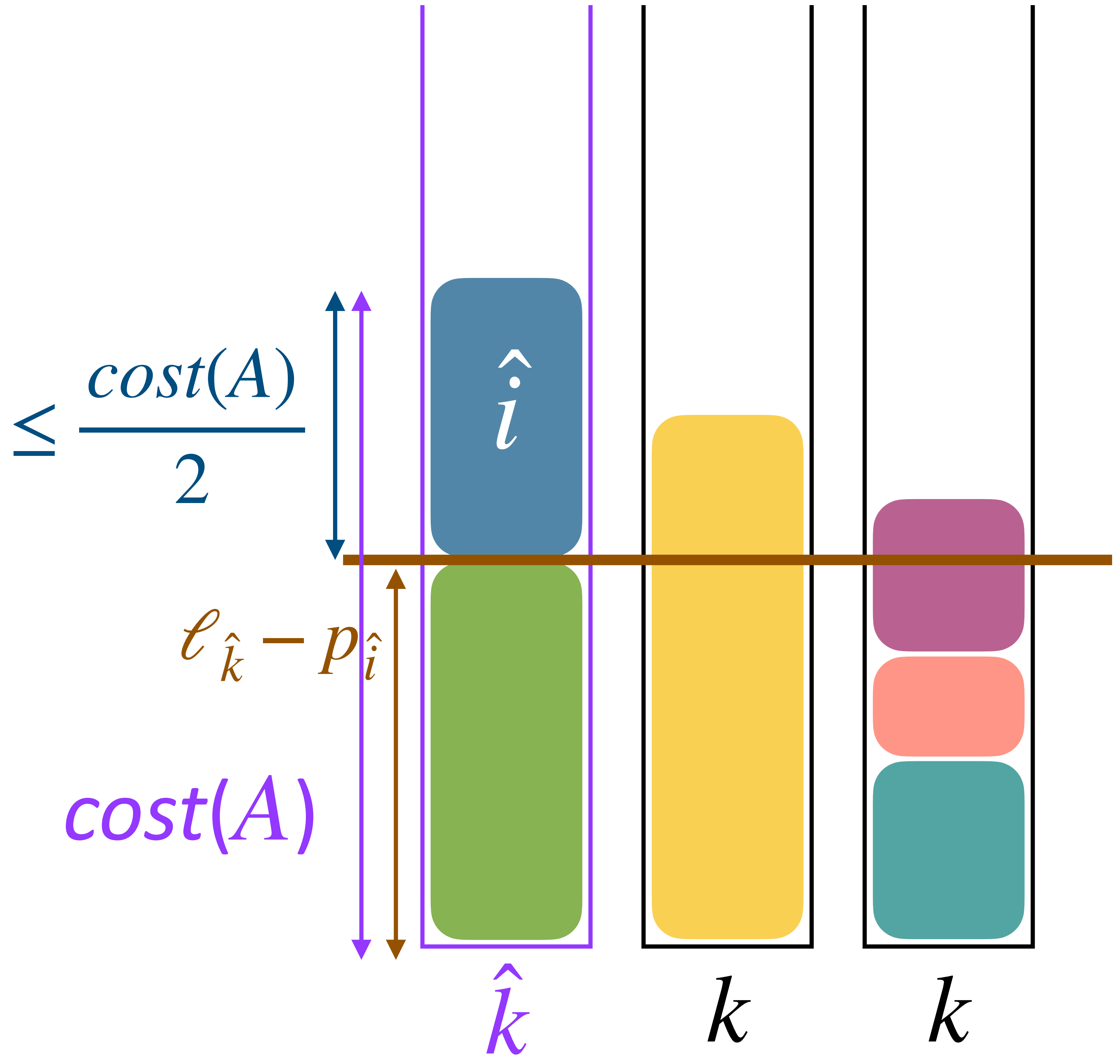
PoA of the Load Balancing Game

- Let $\hat{k}$ be the machine with the highest load under assignment A and $\hat{i}$ is the smallest job on $\hat{k}$
  - Without loss of generality, there are at least two tasks on $\hat{k}$ (otherwise, *cost*(OPT) $=$ *cost*(A) and the theorem is proven)

    $$\Rightarrow p_{\hat{i}} \leq \frac{cost(A)}{2}$$

  - Suppose there is a machine $k \neq \hat{k}$ with load less than $\ell_{\hat{k}} - p_{\hat{i}}$. Then, moving job $\hat{i}$ from $\hat{k}$ to k would decrease the cost of the agent
    - Since A is a Nash equilibrium, this cannot happen

      $$\Rightarrow \ell_k \geq \ell_{\hat{k}} - p_{\hat{i}} \geq cost(A) - \frac{cost(A)}{2} = \frac{cost(A)}{2}$$

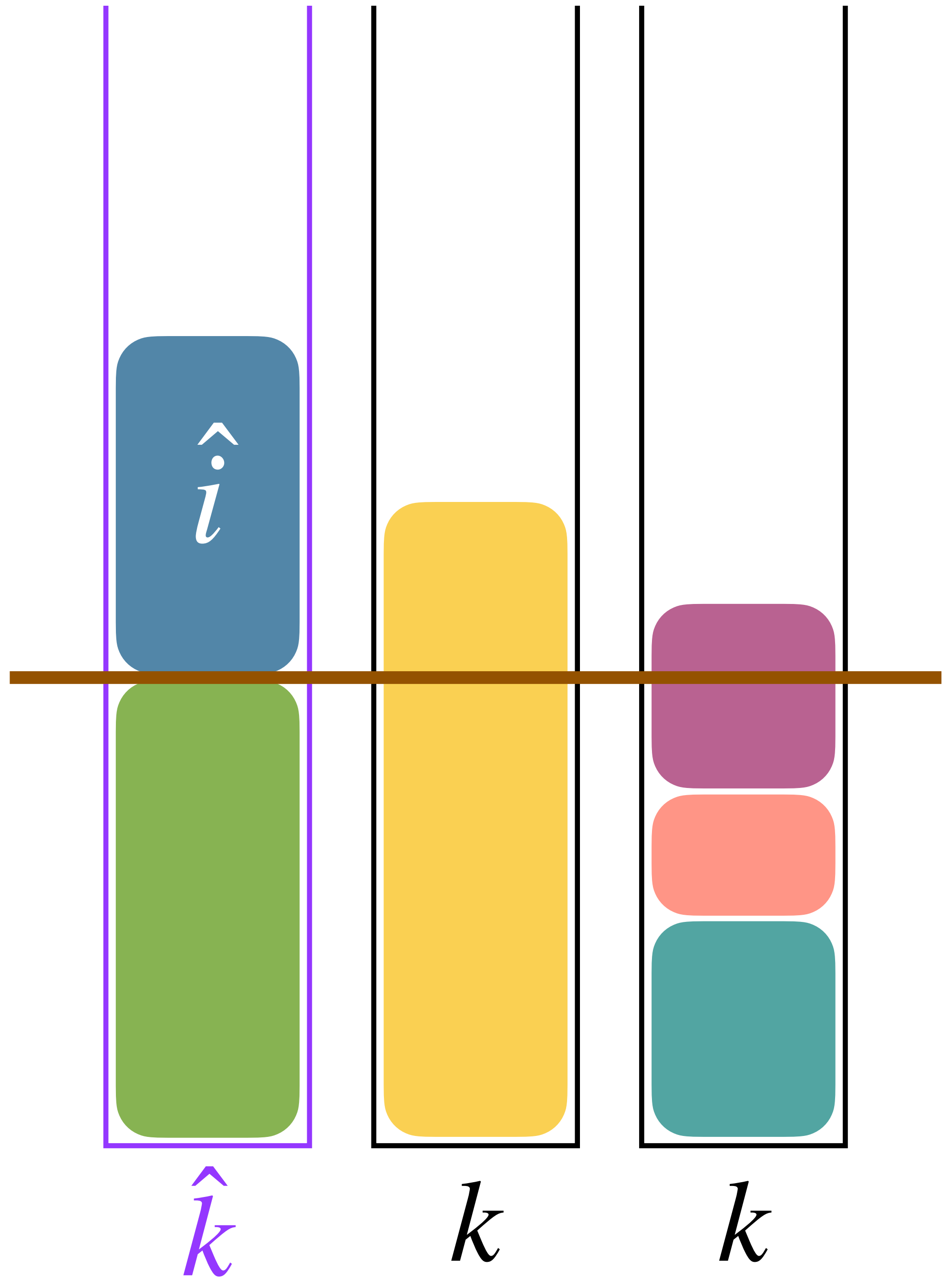


PoA of the Load Balancing Game

- Let $\hat{k}$ be the machine with the highest load under assignment A and $\hat{i}$ is the smallest job on $\hat{k}$
  - Without loss of generality, there are at least two tasks on $\hat{k}$ (otherwise, *cost*(OPT) $=$ *cost*(A) and the theorem is proven)

    $\Rightarrow p_{\hat{i}} \leq \dfrac{cost(A)}{2}$
  - Suppose there is a machine $k \neq \hat{k}$ with load less than $\ell_{\hat{k}} - p_{\hat{i}}$. Then, moving job $\hat{i}$ from $\hat{k}$ to k would decrease the cost of the agent
    - Since A is a Nash equilibrium, this cannot happen

      $\Rightarrow \ell_k \geq \ell_{\hat{k}} - p_{\hat{i}} \geq cost(A) - \dfrac{cost(A)}{2} = \dfrac{cost(A)}{2}$
  - From the average bound, *cost*(OPT) $\geq \dfrac{\Sigma_i p_i}{m} = \dfrac{\Sigma_k \ell_k}{m} \geq \dfrac{cost(A) + (m-1) \cdot \frac{cost(A)}{2}}{m} = \dfrac{m+1}{2m} \cdot cost(A)$



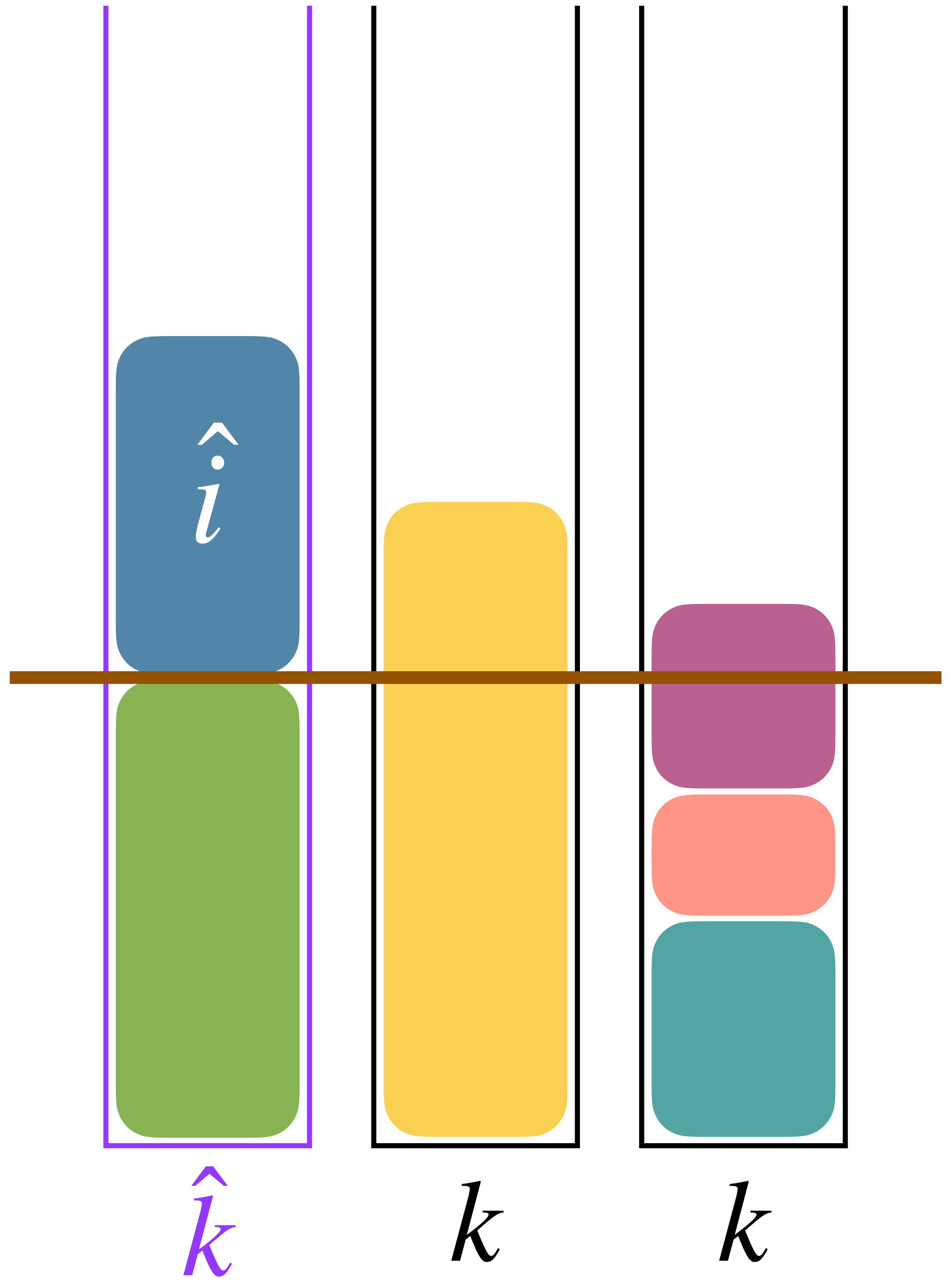
PoA of the Load Balancing Game

- Let $\hat{k}$ be the machine with the highest load under assignment A and $\hat{i}$ is the smallest job on $\hat{k}$
  - Without loss of generality, there are at least two tasks on $\hat{k}$ (otherwise, *cost*(OPT) $=$ *cost*(A) and the theorem is proven)

    $$\Rightarrow p_{\hat{i}} \leq \frac{cost(A)}{2}$$
  - Suppose there is a machine $k \neq \hat{k}$ with load less than $\ell_{\hat{k}} - p_{\hat{i}}$. Then, moving job $\hat{i}$ from $\hat{k}$ to k would decrease the cost of the agent
    - Since A is a Nash equilibrium, this cannot happen

      $$\Rightarrow \ell_k \geq \ell_{\hat{k}} - p_{\hat{i}} \geq cost(A) - \frac{cost(A)}{2} = \frac{cost(A)}{2}$$
  - From the average bound, *cost*(OPT) $\geq \frac{\Sigma_i p_i}{m} = \frac{\Sigma_k \ell_k}{m} \geq \frac{cost(A) + (m-1)\cdot\frac{cost(A)}{2}}{m} = \frac{m+1}{2m}\cdot cost(A)$

    $$\Rightarrow \frac{cost(A)}{cost(\text{OPT})} \leq \frac{2m}{m+1} = 2 - \frac{2}{m+1}$$



What happened

- The PoA of the selfish load balancing game is at most $2 - \dfrac{2}{m+1}$

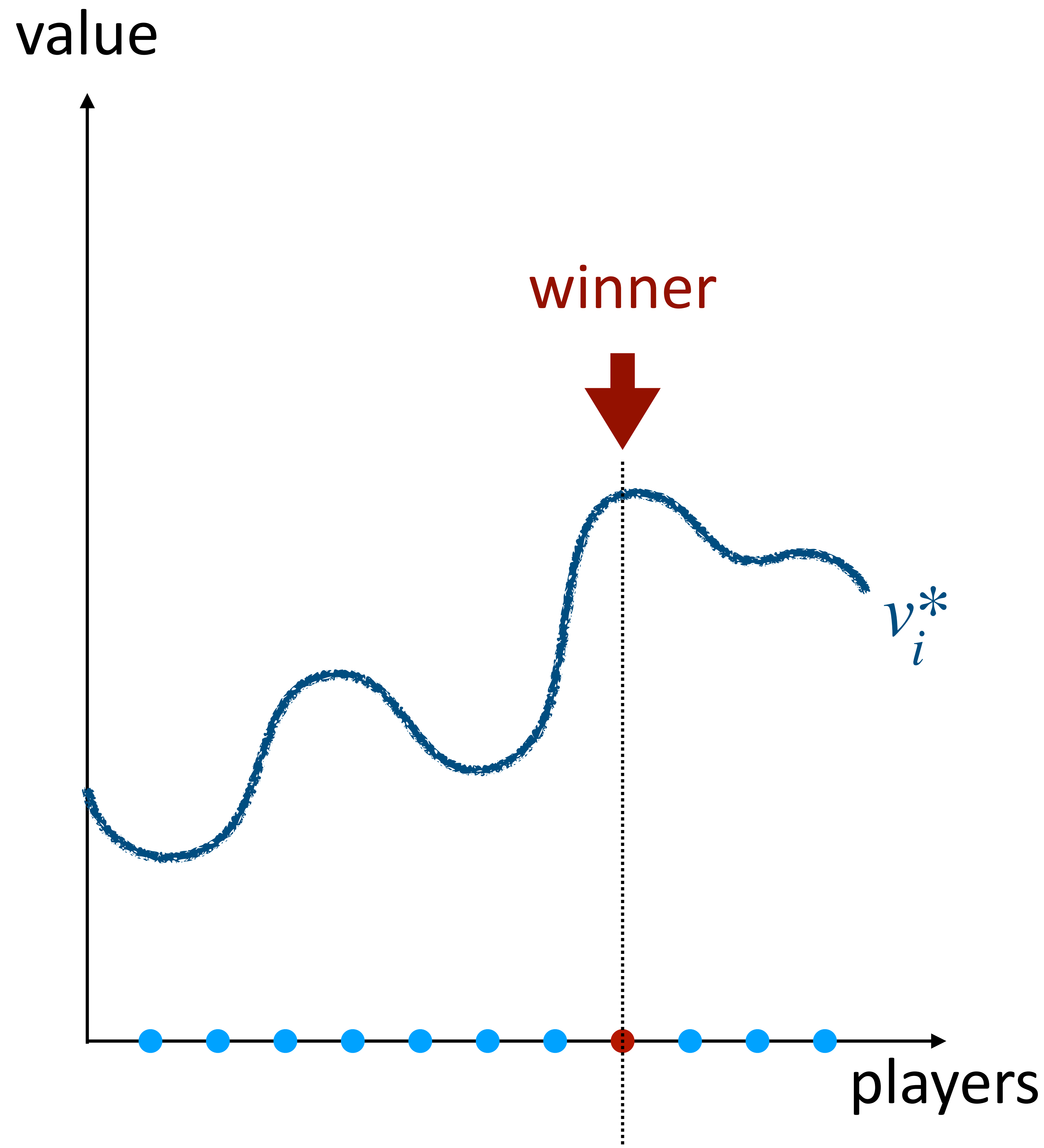
Outline

- Fundamental concepts
  - Game, players, strategies, payoffs/costs
- Nash Equilibrium
- Price of Anarchy
  - Selfish load balancing
- Mechanism design
  - Auction
  - Vickrey-Clarke-Groves mechanism

Auction Game

- Game: There is a valuable item.

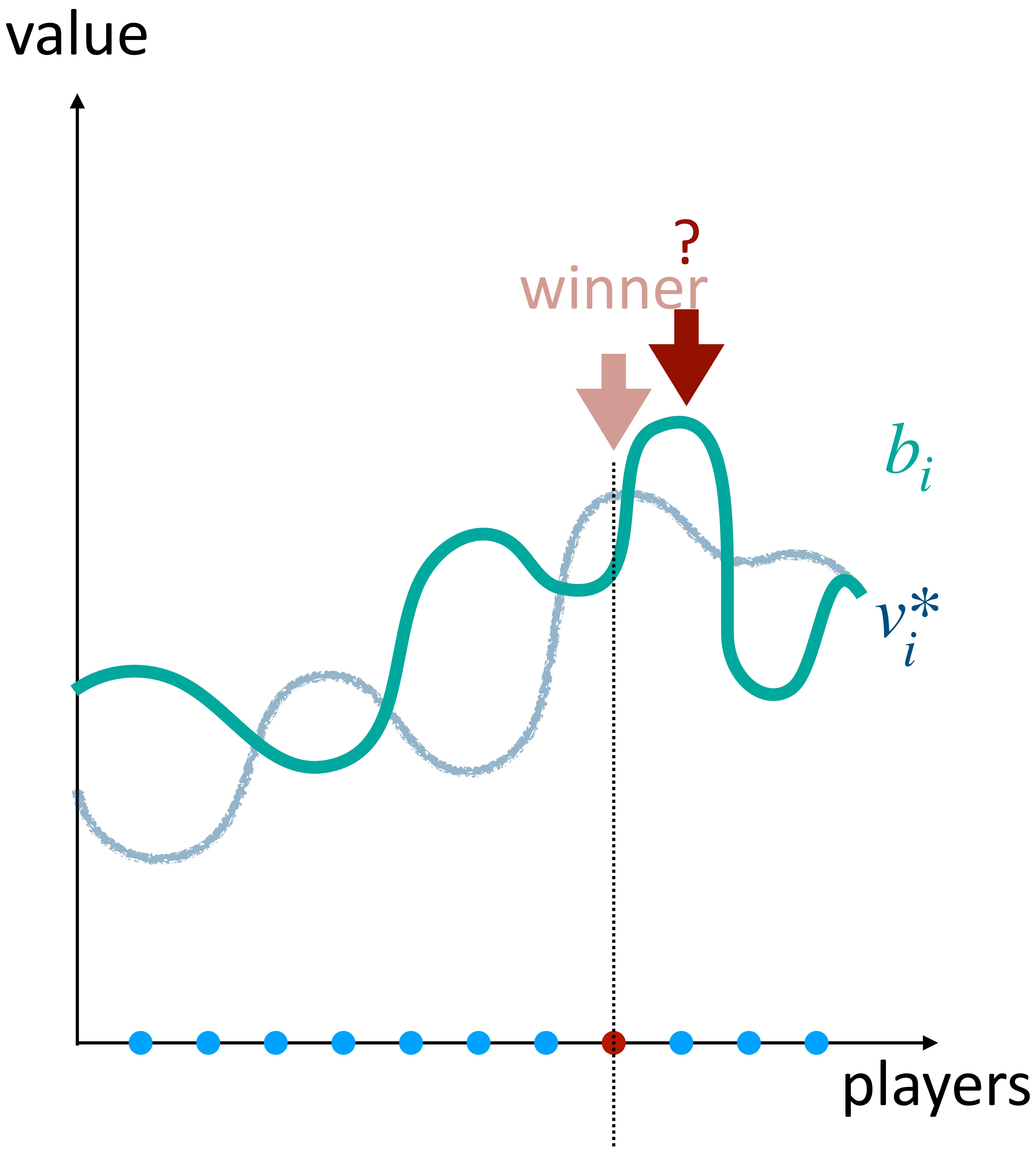
- Each player (bidder) i has a value v_i^* for the good that he is “willing to pay” for the item and private to himself



ι

Auction Game

- Game: There is a valuable item. All players submit their bids in sealed envelopes to the seller, and the seller picks one winner. The winner has to pay some price $p \geq 0$
  - Each player (bidder) i has a value v_i^* for the good that he is “willing to pay” for the item and private to himself

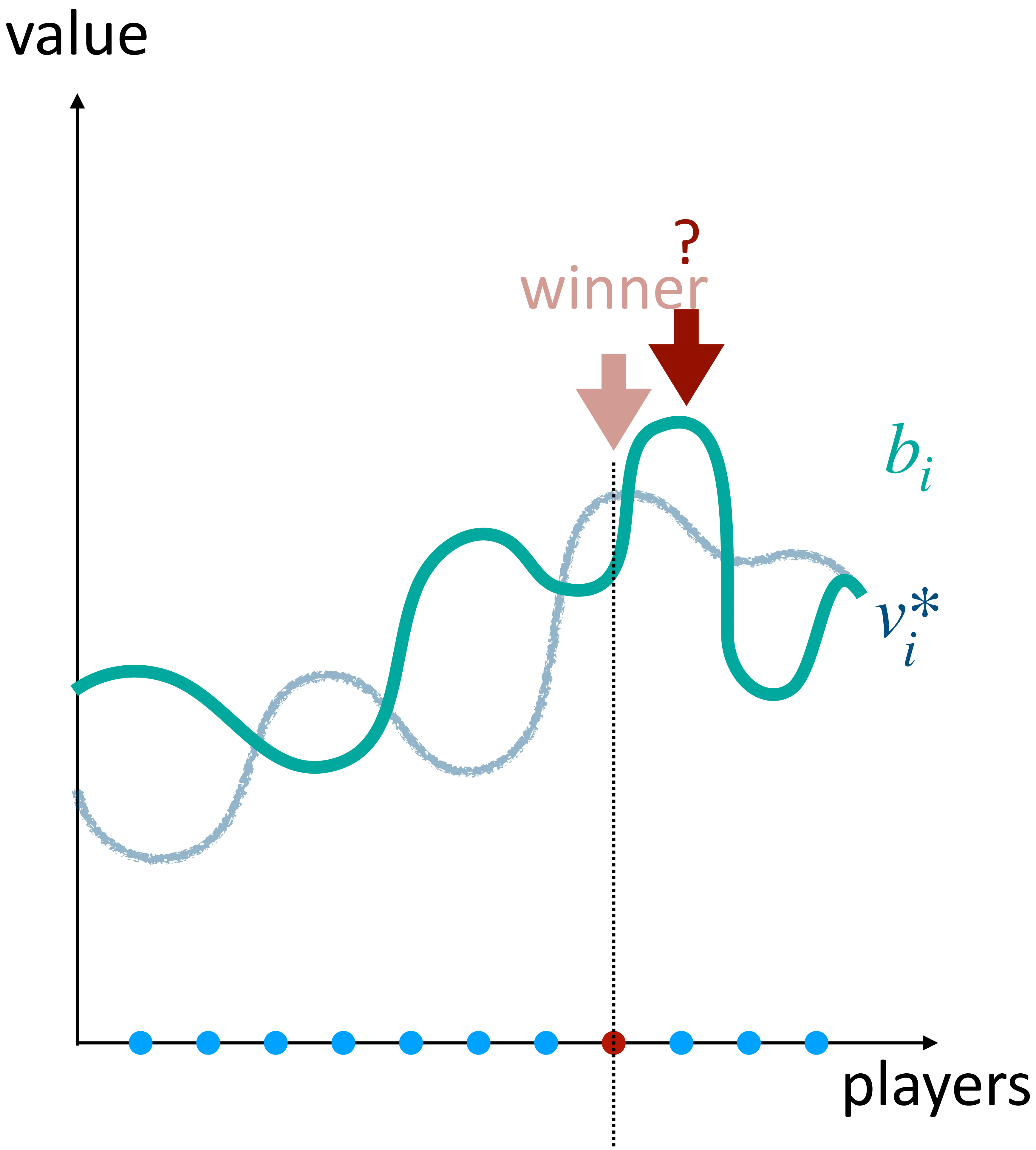


ι

Auction Game

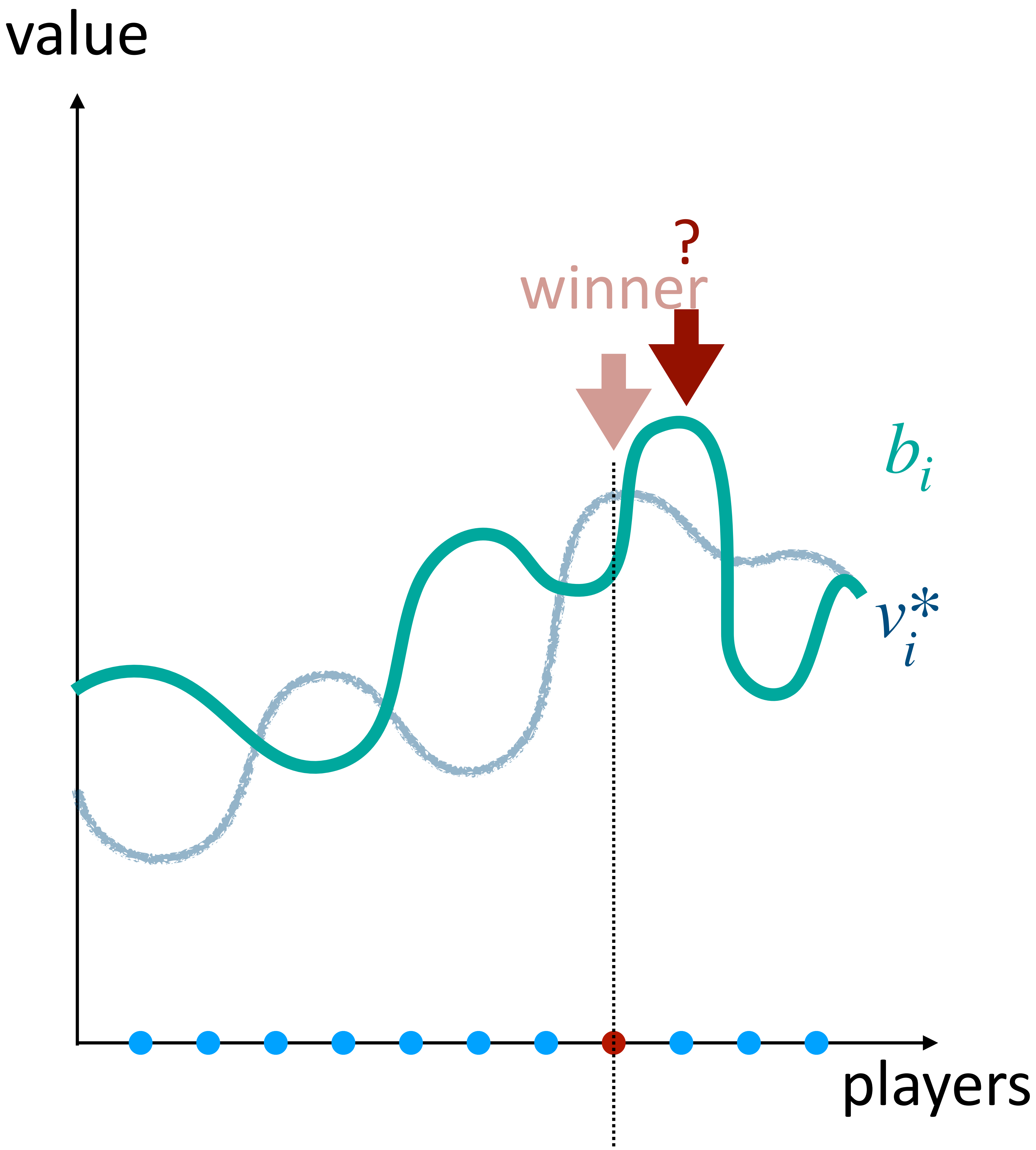
- Game: There is a valuable item. All players submit their bids in sealed envelopes to the seller, and the seller picks one winner. The winner has to pay some price $p \geq 0$
    - Each player (bidder) i has a value v_i^* for the good that he is "willing to pay" for the item and private to himself
        - Strategy of player i: bid b_i

ι



Auction Game

- Game: There is a valuable item. All players submit their bids in sealed envelopes to the seller, and the seller picks one winner. The winner has to pay some price $p \geq 0$
  - Each player (bidder) i has a value v_i^* for the good that he is "willing to pay" for the item and private to himself
    - Strategy of player i: bid b_i
    - *Utility* of player i is 0 if he does not win, and $v_i^* - p$ if he wins at a price of p



Truthfulness?

- Given the bids $b_1, b_2, \cdots, b_n$

Truthfulness?

- Given the bids $b_1, b_2, \cdots, b_n$, the *outcome* of the game is $f(b_1, b_2, \cdots, b_n)$

Truthfulness?

- Given the bids $b_1, b_2, \cdots, b_n$, the *outcome* of the game is $f(b_1, b_2, \cdots, b_n)$

player i wins

Truthfulness?

- Given the bids $b_1, b_2, \cdots, b_n$, the *outcome* of the game is $f(b_1, b_2, \cdots, b_n)$
  - Given the bids, player i has utility $u_i(f(b_1, b_2, \cdots, b_n)) = u_i(f(\overrightarrow{b}))$

Truthfulness?

- Given the bids $b_1, b_2, \cdots, b_n$, the *outcome* of the game is $f(b_1, b_2, \cdots, b_n)$
    - Given the bids, player i has utility $u_i(f(b_1, b_2, \cdots, b_n)) = u_i(f(\overrightarrow{b}))$

If player i wins, $u_i(f(\overrightarrow{b})) = v_i^* - p$

If player i loses, $u_i(f(\overrightarrow{b})) = 0$

Truthfulness?

- Given the bids $b_1, b_2, \cdots, b_n$, the *outcome* of the game is $f(b_1, b_2, \cdots, b_n)$
    - Given the bids, player i has utility $u_i(f(b_1, b_2, \cdots, b_n)) = u_i(f(\overrightarrow{b}))$
- To maximize the social welfare, the seller wants to maximize $\Sigma_i u_i(f(b_1, b_2, \cdots, b_n))$ = maximize $u_{\text{choice}} - p$ = maximize $v^*_{\text{choice}} - p$

Truthfulness?

- Given the bids $b_1, b_2, \cdots, b_n$, the *outcome* of the game is $f(b_1, b_2, \cdots, b_n)$
  - Given the bids, player i has utility $u_i(f(b_1, b_2, \cdots, b_n)) = u_i(f(\overrightarrow{b}))$
- To maximize the social welfare, the seller wants to maximize $\Sigma_i u_i(f(b_1, b_2, \cdots, b_n))$ = maximize $u_{\text{choice}} - p$ = maximize $v^*_{\text{choice}} - p$
  - That is, the seller wants to find the maximum v_i^* among all players i

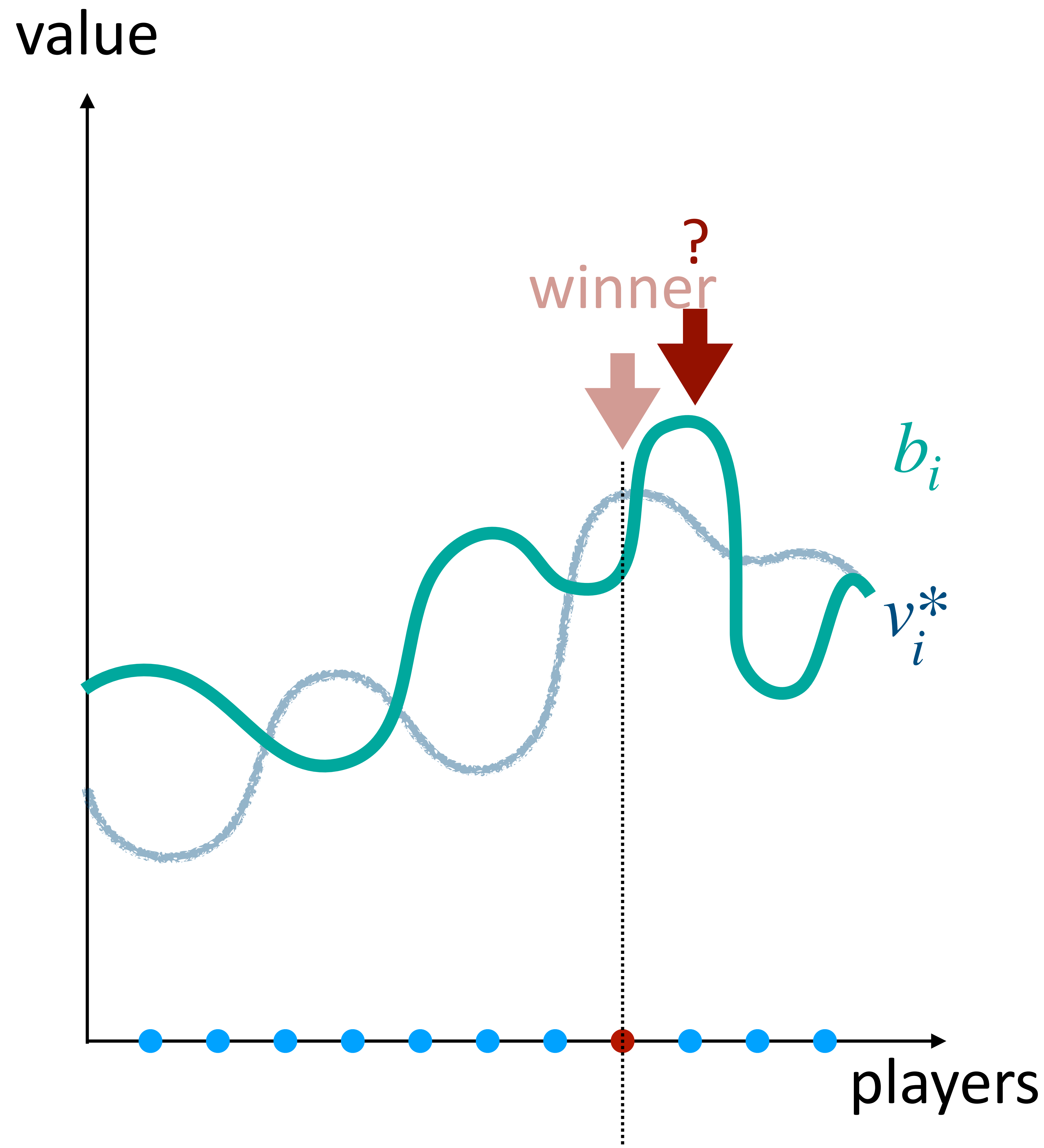
Truthfulness?

- Given the bids $b_1, b_2, \cdots, b_n$, the *outcome* of the game is $f(b_1, b_2, \cdots, b_n)$
    - Given the bids, player i has utility $u_i(f(b_1, b_2, \cdots, b_n)) = u_i(f(\overrightarrow{b}))$
- To maximize the social welfare, the seller wants to maximize $\Sigma_i u_i(f(b_1, b_2, \cdots, b_n)) =$ maximize $u_{\text{choice}} - p =$ maximize $v^*_{\text{choice}} - p$
    - That is, the seller wants to find the maximum v_i^* among all players i
    - However, v_i^* is a private value, and the seller doesn't know them

Truthfulness?

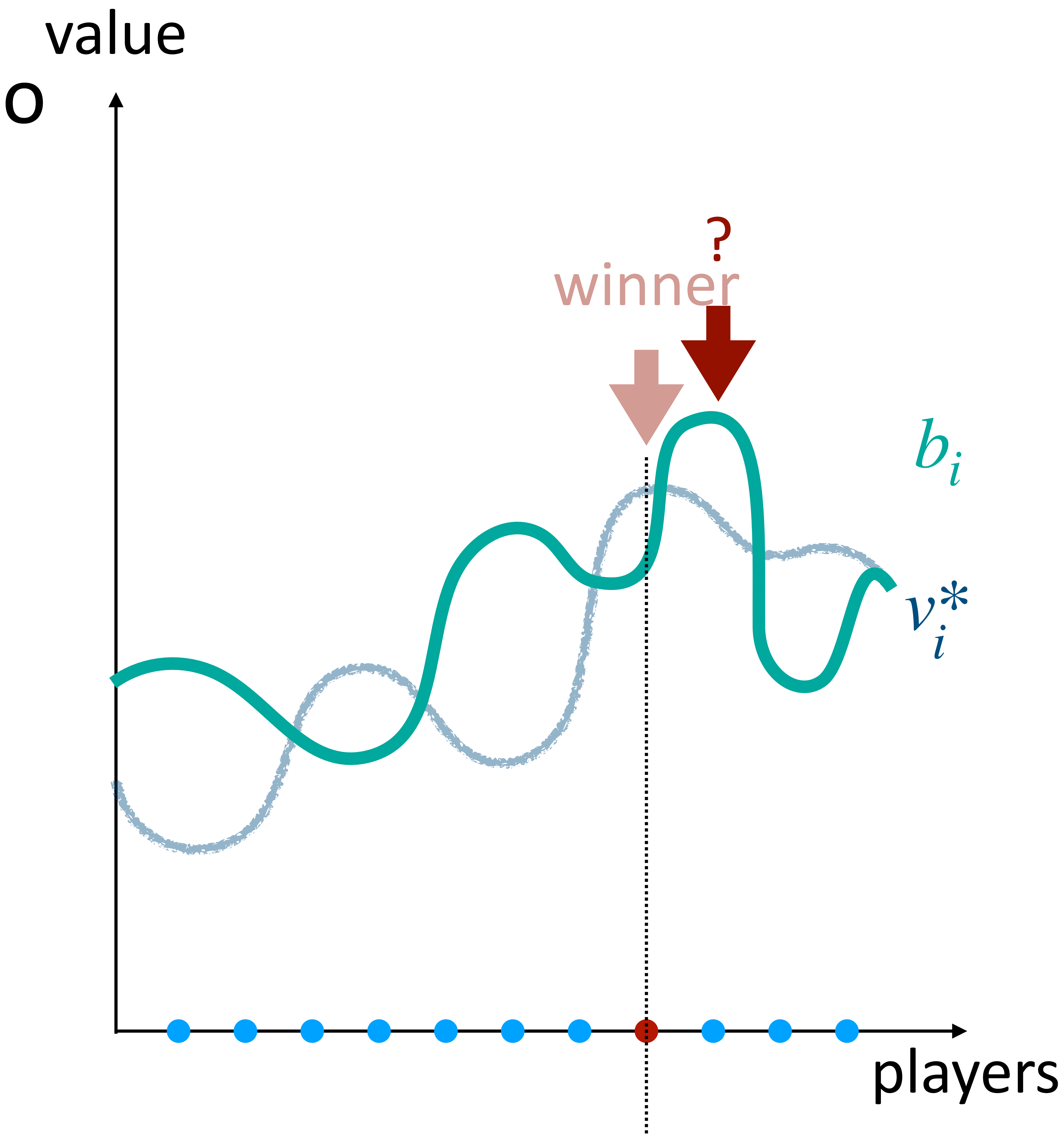
- Given the bids $b_1, b_2, \cdots, b_n$, the *outcome* of the game is $f(b_1, b_2, \cdots, b_n)$
  - Given the bids, player i has utility $u_i(f(b_1, b_2, \cdots, b_n)) = u_i(f(\overrightarrow{b}))$
- To maximize the social welfare, the seller wants to maximize $\Sigma_i u_i(f(b_1, b_2, \cdots, b_n)) =$ maximize $u_{\text{choice}} - p =$ maximize $v^*_{\text{choice}} - p$
  - That is, the seller wants to find the maximum v_i^* among all players i
  - However, v_i^* is a private value, and the seller doesn't know them
    - Can players *strategically manipulate* the game?

Truthfulness?



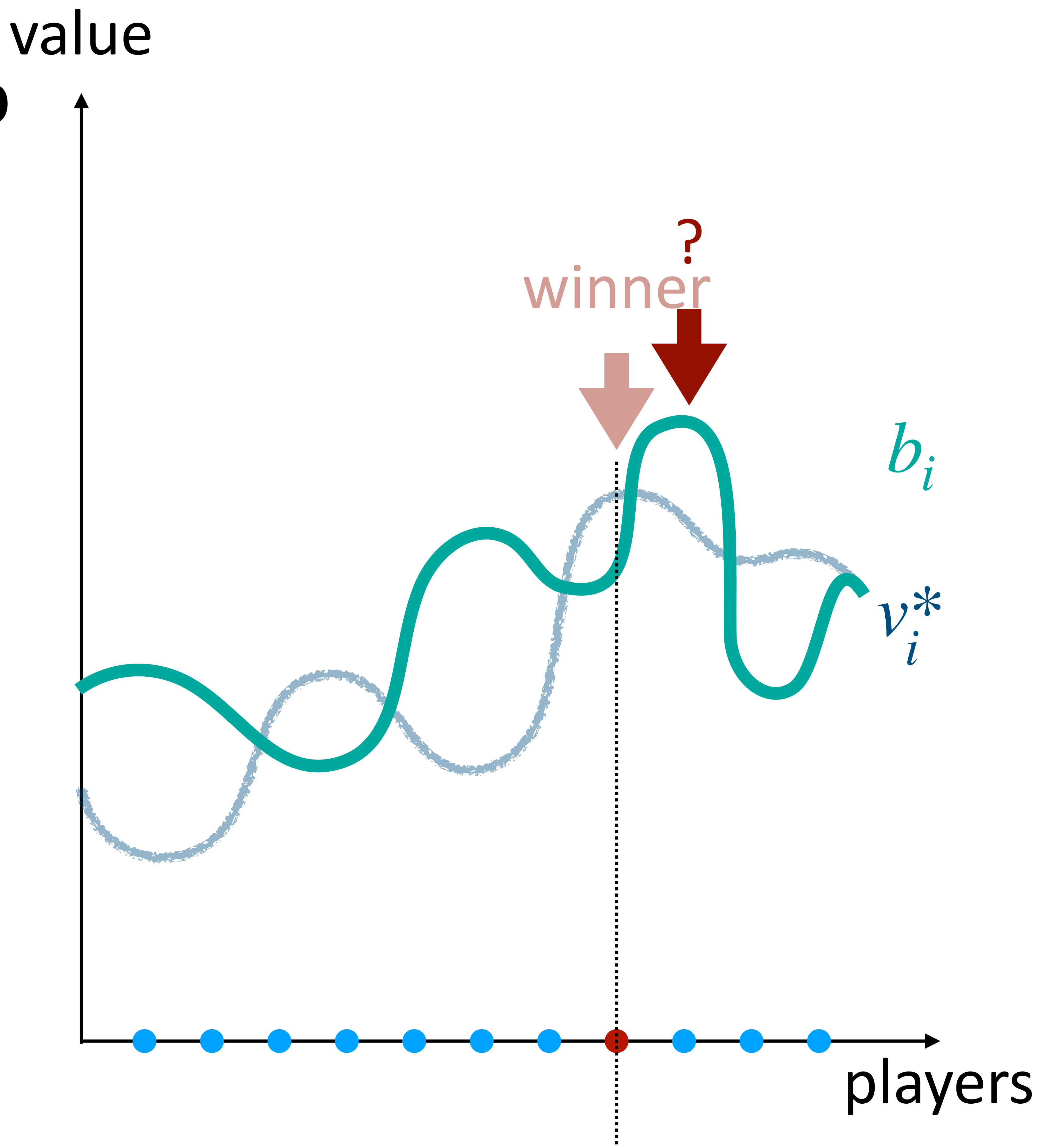
Truthfulness?

- No payment ($p = 0$): We give the item for free to the player with highest v_i^*



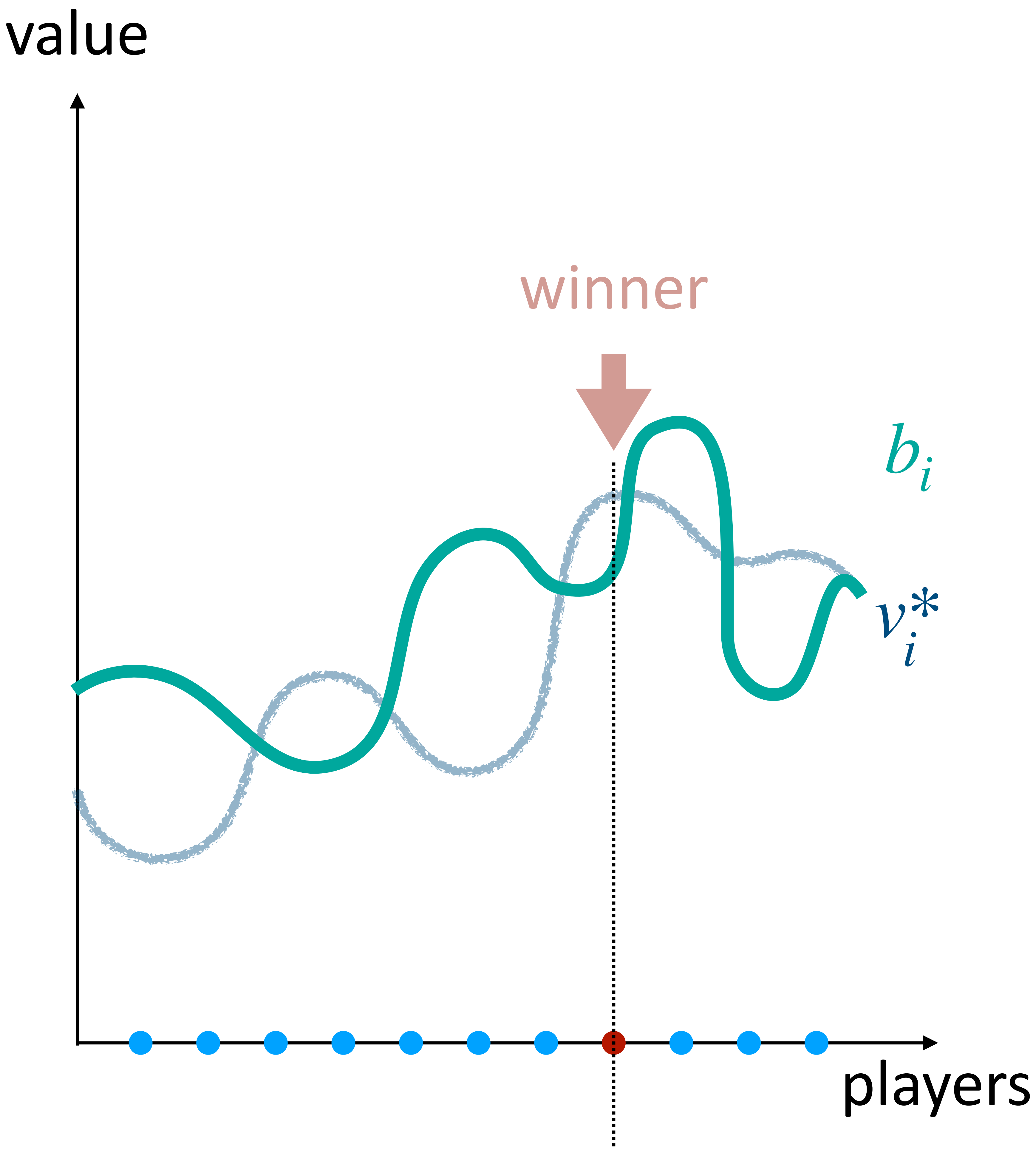
Truthfulness?

- No payment ($p = 0$): We give the item for free to the player with highest v_i^*
  - This method is easily manipulated: player can benefit by exaggerating his v_i^* by reporting bid $b_i \gg v_i^*$



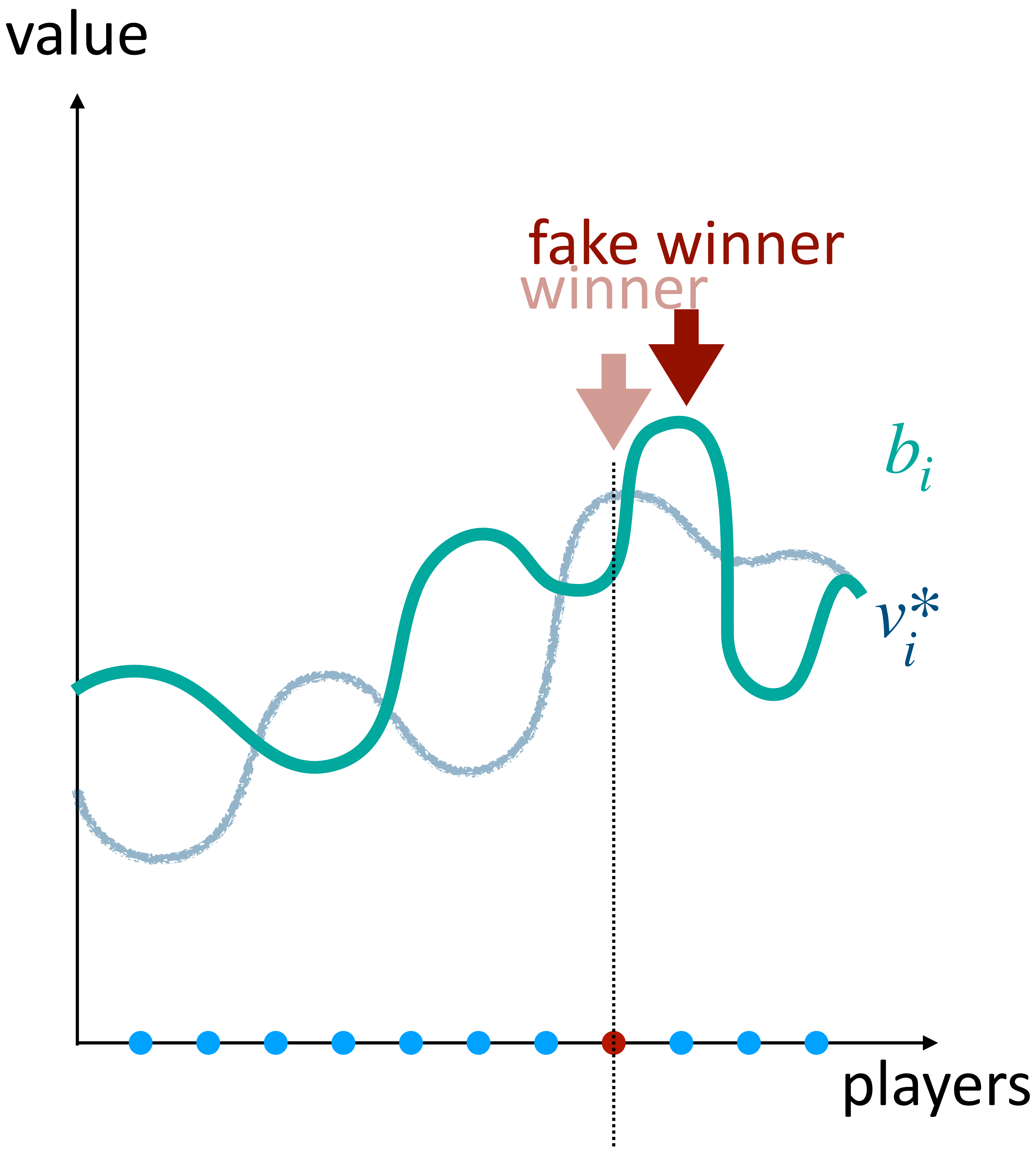
Truthfulness?

- Pay your bid ($p = b_w$, where w is the winner):



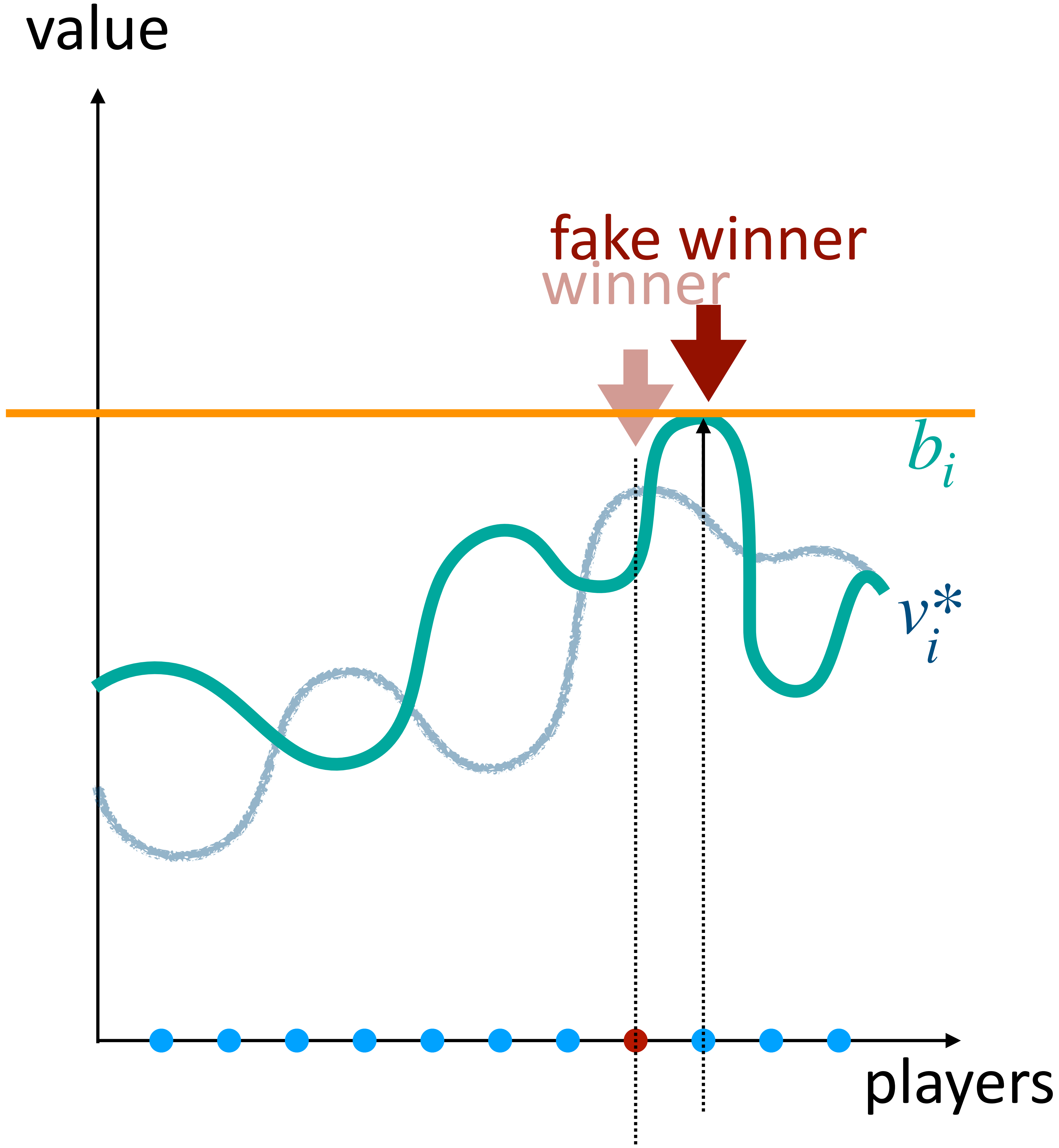
Truthfulness?

- Pay your bid ($p = b_w$, where w is the winner):
  - Fake winner i has utility $u_i = v_i^* - b_i < 0$



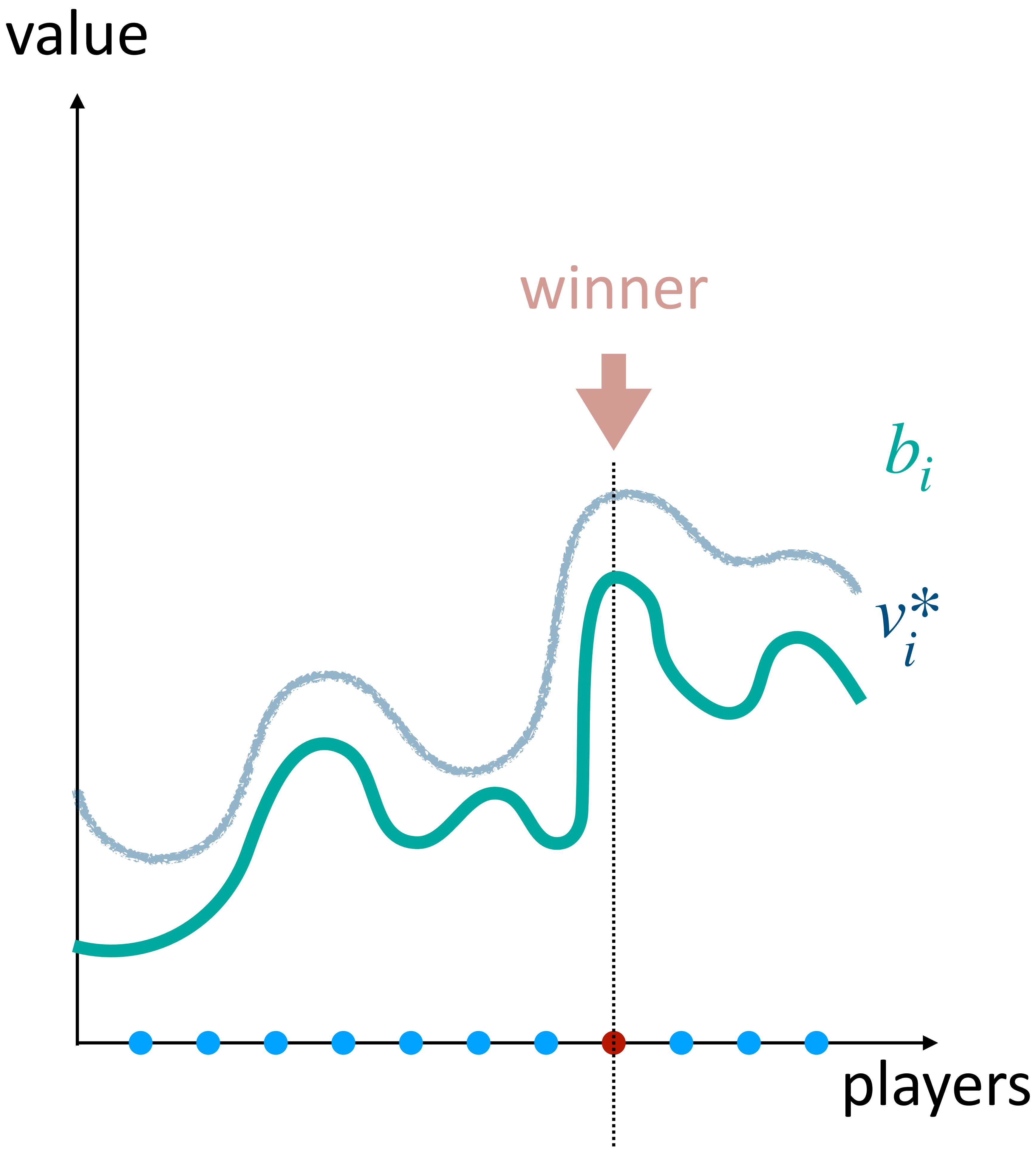
Truthfulness?

- Pay your bid ($p = b_w$, where w is the winner):
  - Fake winner i has utility $u_i = v_i^* - b_i < 0$



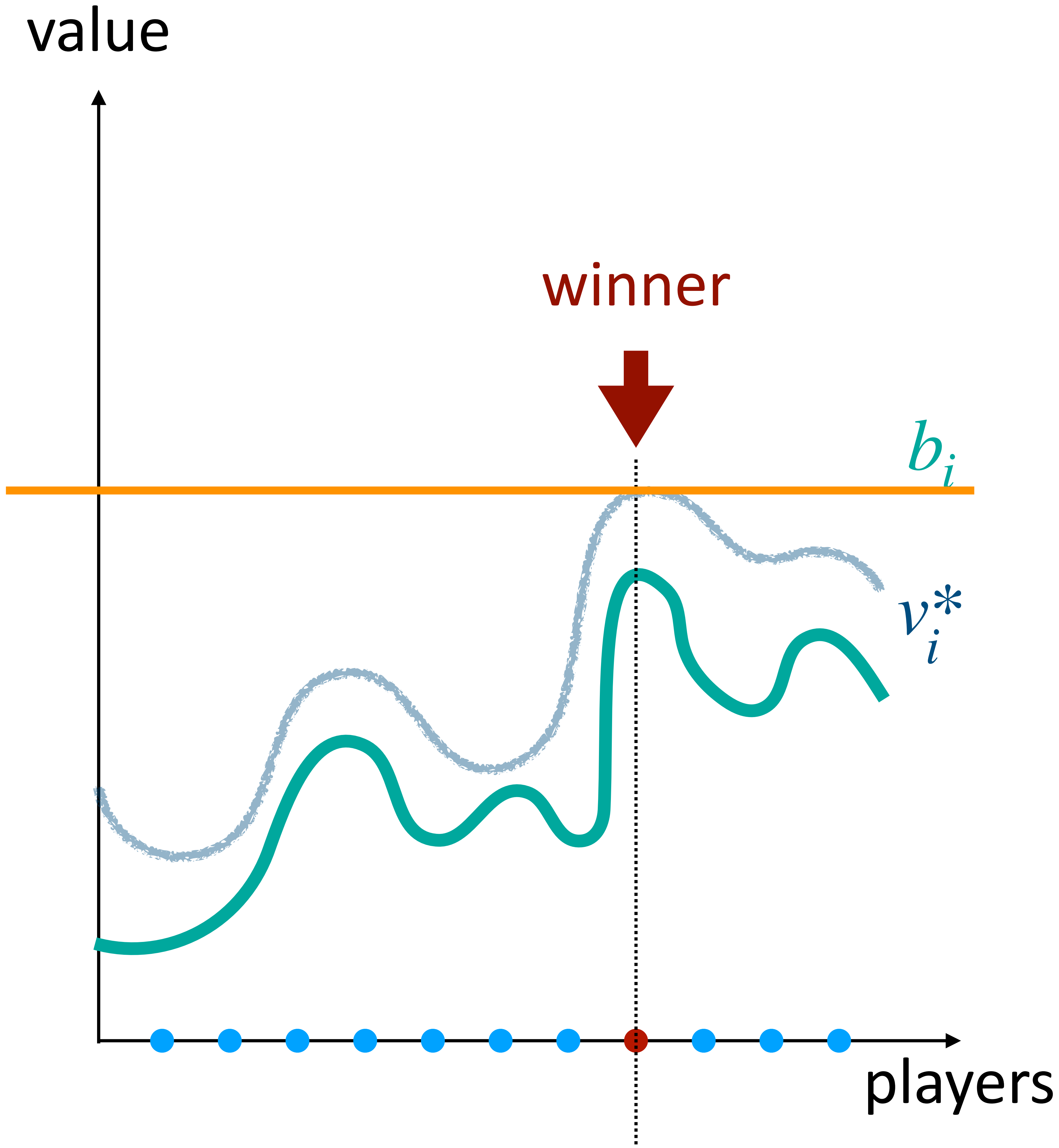
Truthfulness?

- Pay your bid ($p = b_w$, where w is the winner):
  - Fake winner i has utility $u_i = v_i^* - b_i < 0$



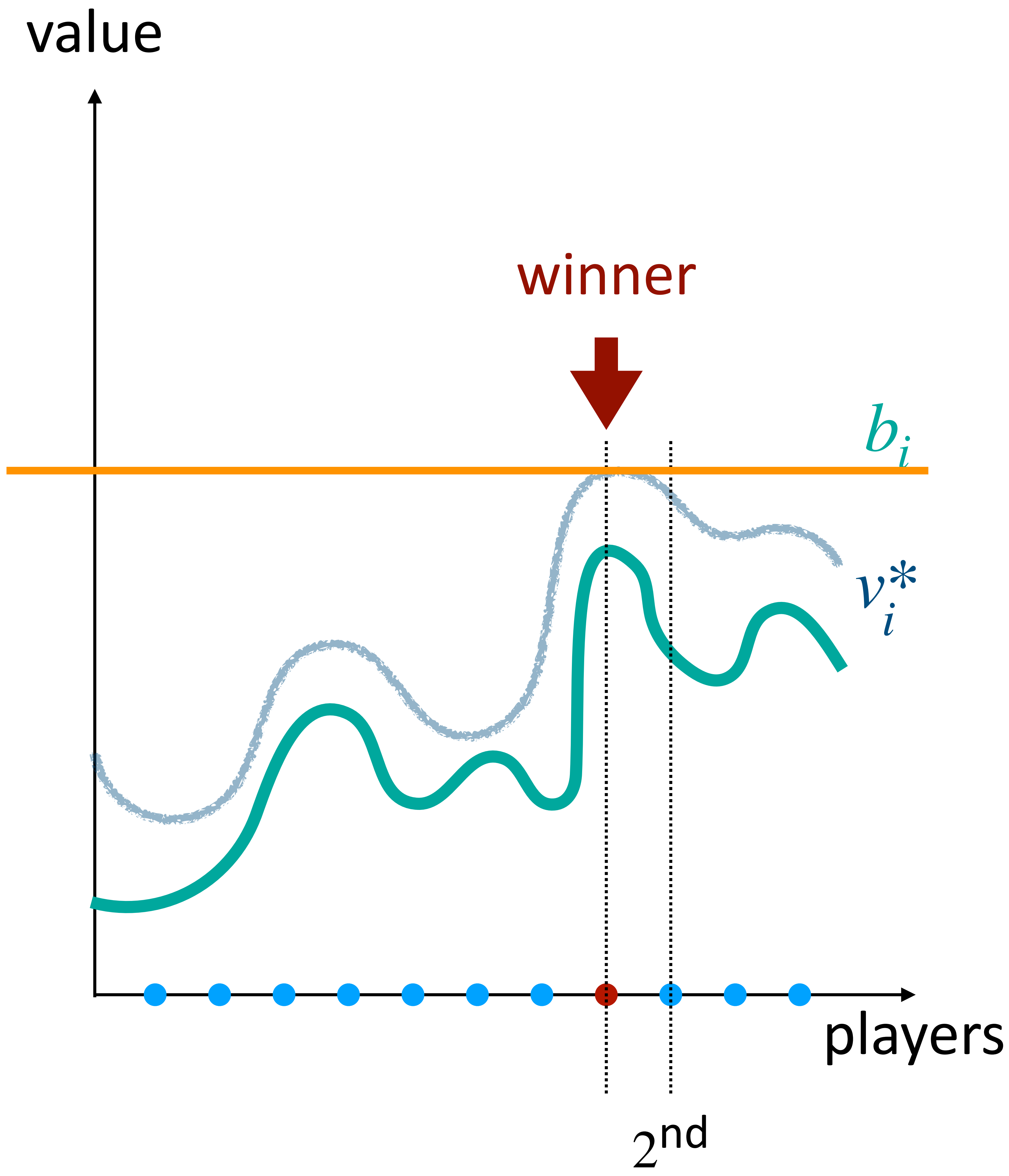
Truthfulness?

- Pay your bid ($p = b_w$, where w is the winner):
  - Fake winner i has utility $u_i = v_i^* - b_i < 0$
  - Winner w has utility $u_w = v_w^* - b_w = 0$



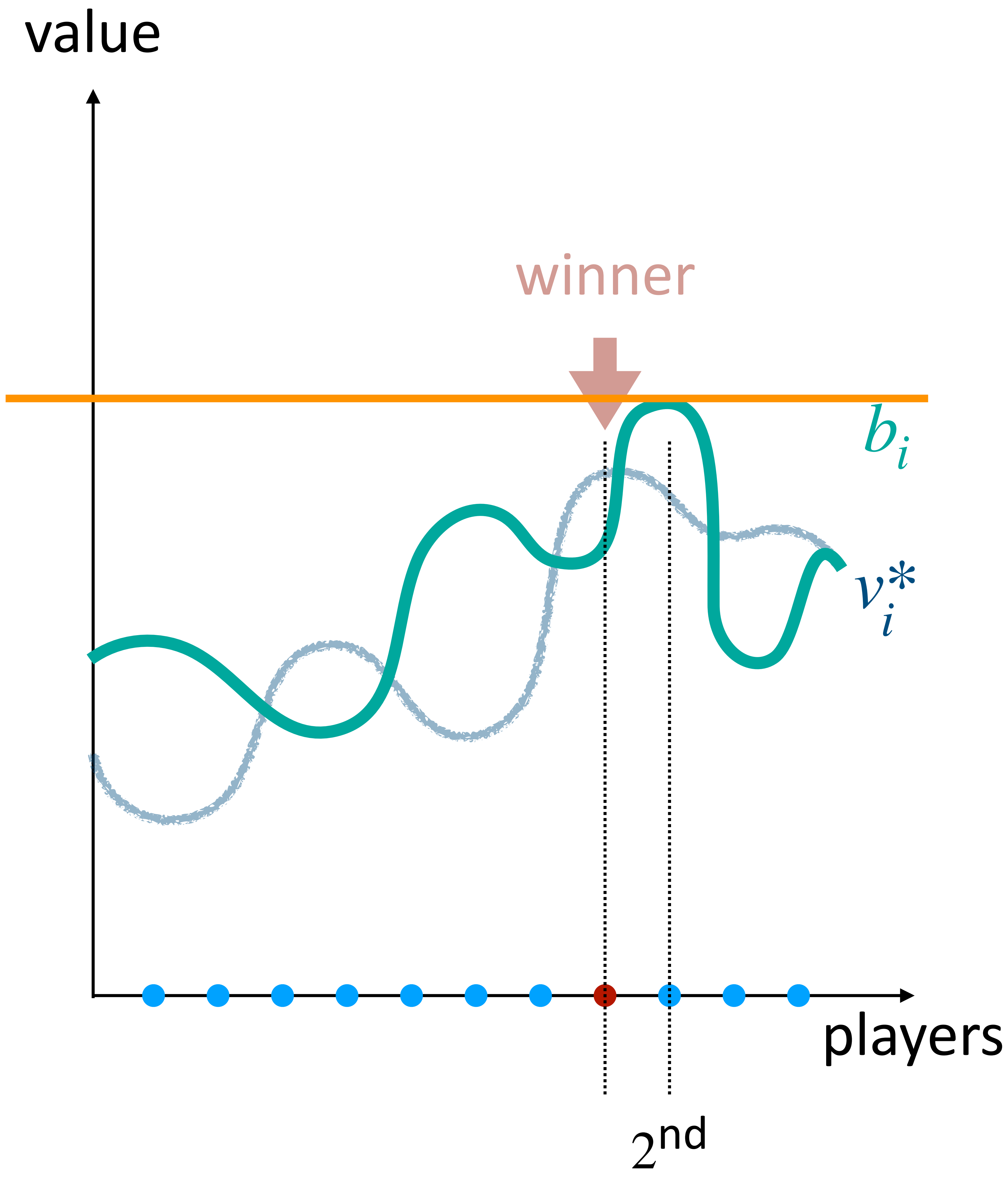
Truthfulness?

- Pay your bid ($p = b_w$, where w is the winner):
  - Fake winner i has utility $u_i = v_i^* - b_i < 0$
  - Winner w has utility $u_w = v_w^* - b_w = 0$
    - He should attempt declaring a lower value $b_w < v_w^*$ and gets new utility $u_w' > u_w$



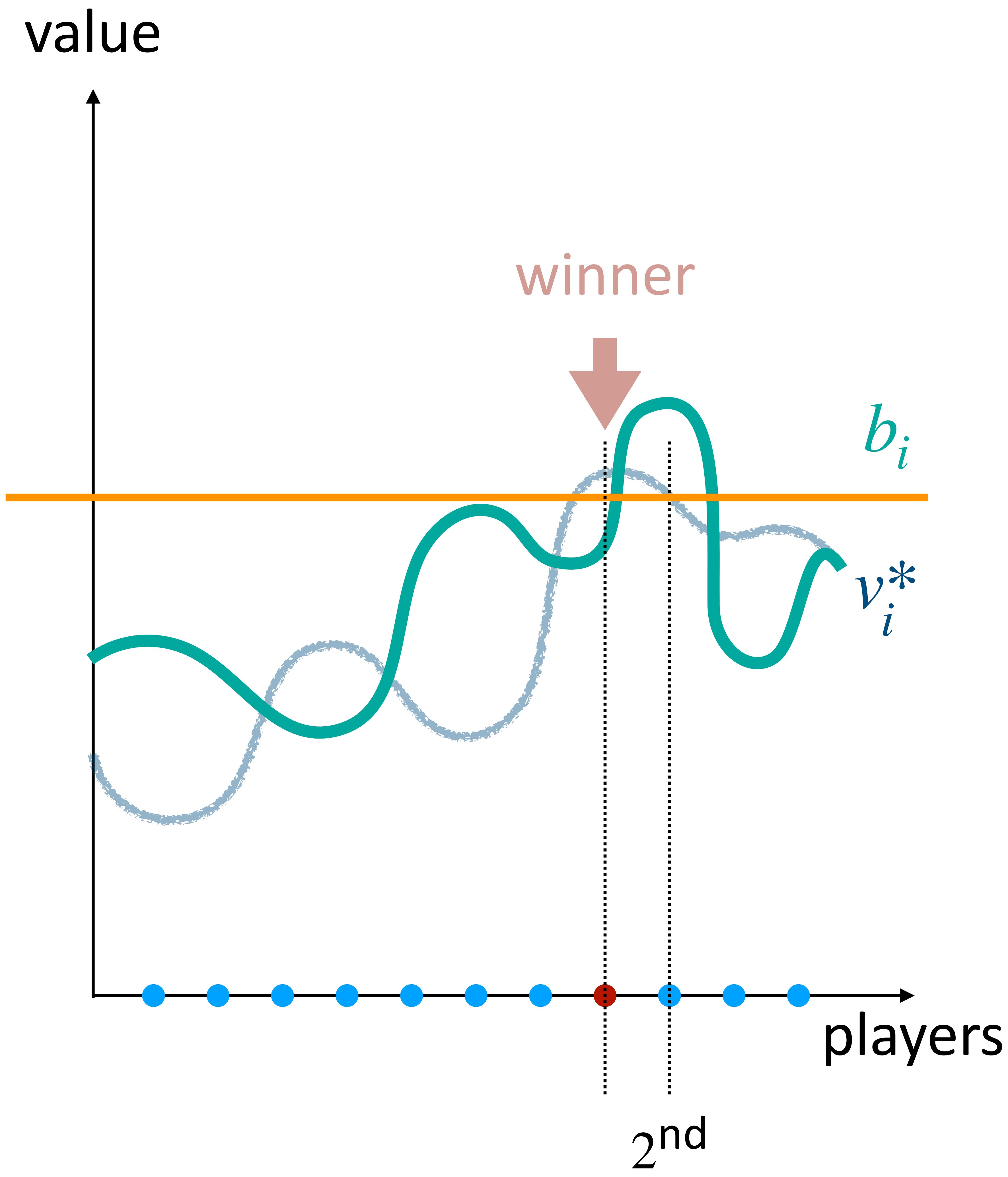
Truthfulness?

- Pay your bid ($p = b_w$, where w is the winner):
  - Fake winner i has utility $u_i = v_i^* - b_i < 0$
  - Winner w has utility $u_w = v_w^* - b_w = 0$
    - He should attempt declaring a lower value $b_w < v_w^*$ and gets new utility $u_w' > u_w$
      - The better scenario is that he knows the second-highest bid and make b_w a bit larger than it



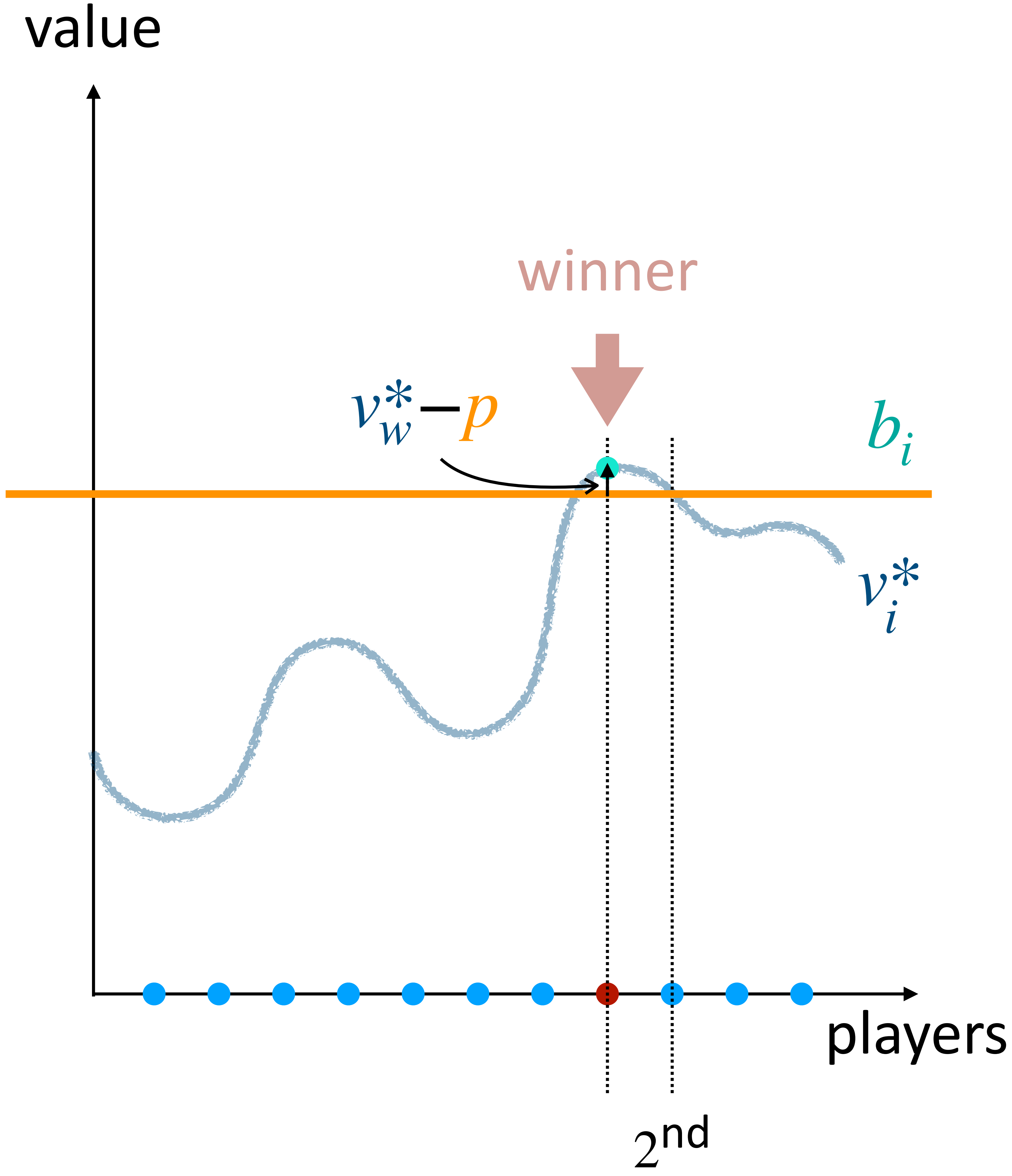
Vickrey's Second Price Auction

- Let the winner be the player i with the highest declared value of b_i, and let i pay the second highest declared bid
  - That is, $p = \max\limits_{j \neq i} b_j$



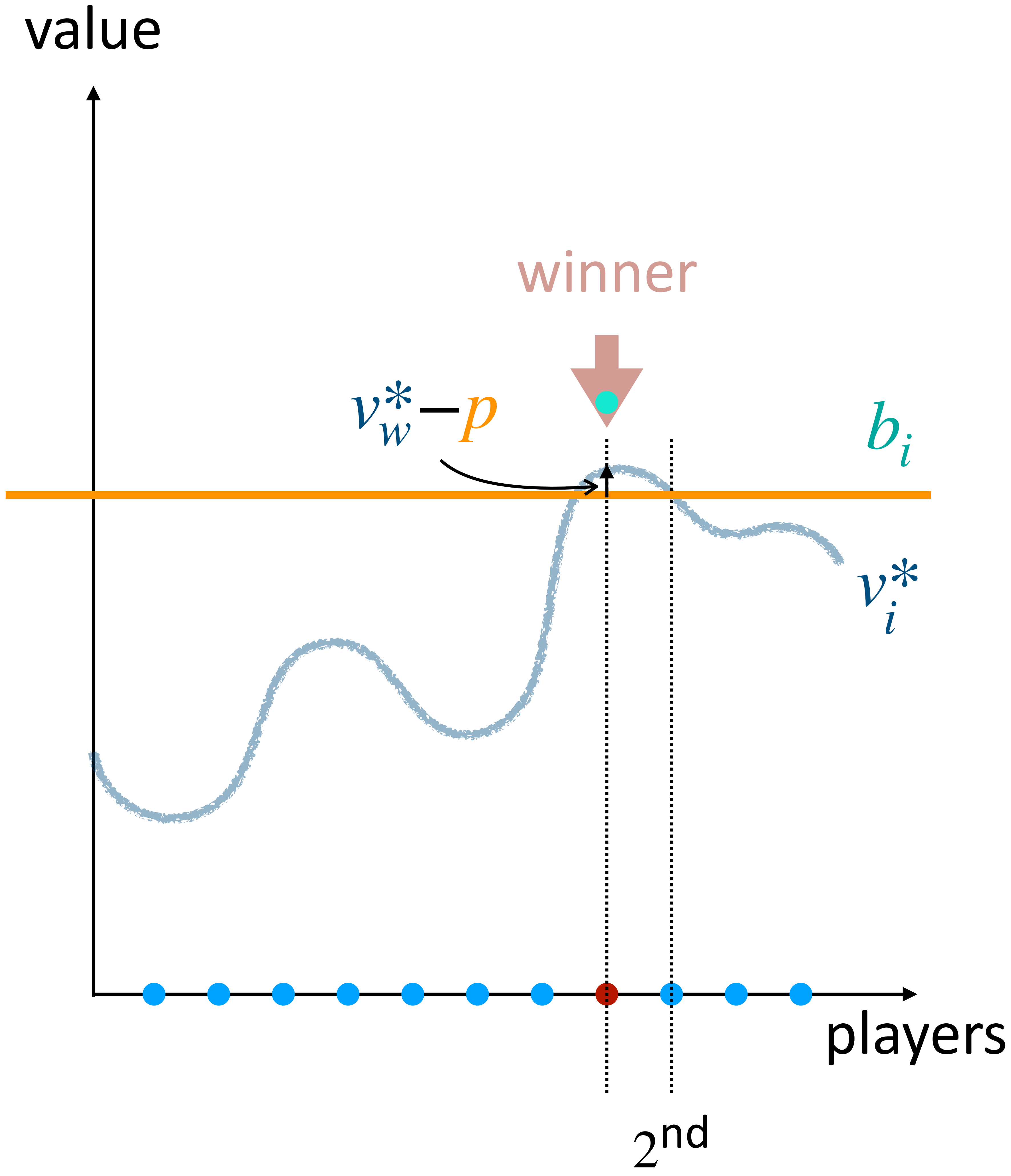
Vickrey’s Second Price Auction

- Let the winner be the player i with the highest declared value of b_i, and let i pay the second highest declared bid. That is, $p = \max_{j \neq i} b_j$
- For every $b_1, b_2, \cdots, b_n$ and v_i^*, let u_i^* be i's utility if it bids v_i^* and u_i his utility if he bids b_i. Then, $u_i^* \geq u_i$.
  - If i is the winner:
    - If $b_i > v_i^*$ or $v_i^* > b_i > p$: player i still wins, $u_i = v_i^* - p = u_i^*$
    - If $b_i < v_i^*$ and $p > b_i$: player i loses, $u_i = 0 \leq u_i^*$



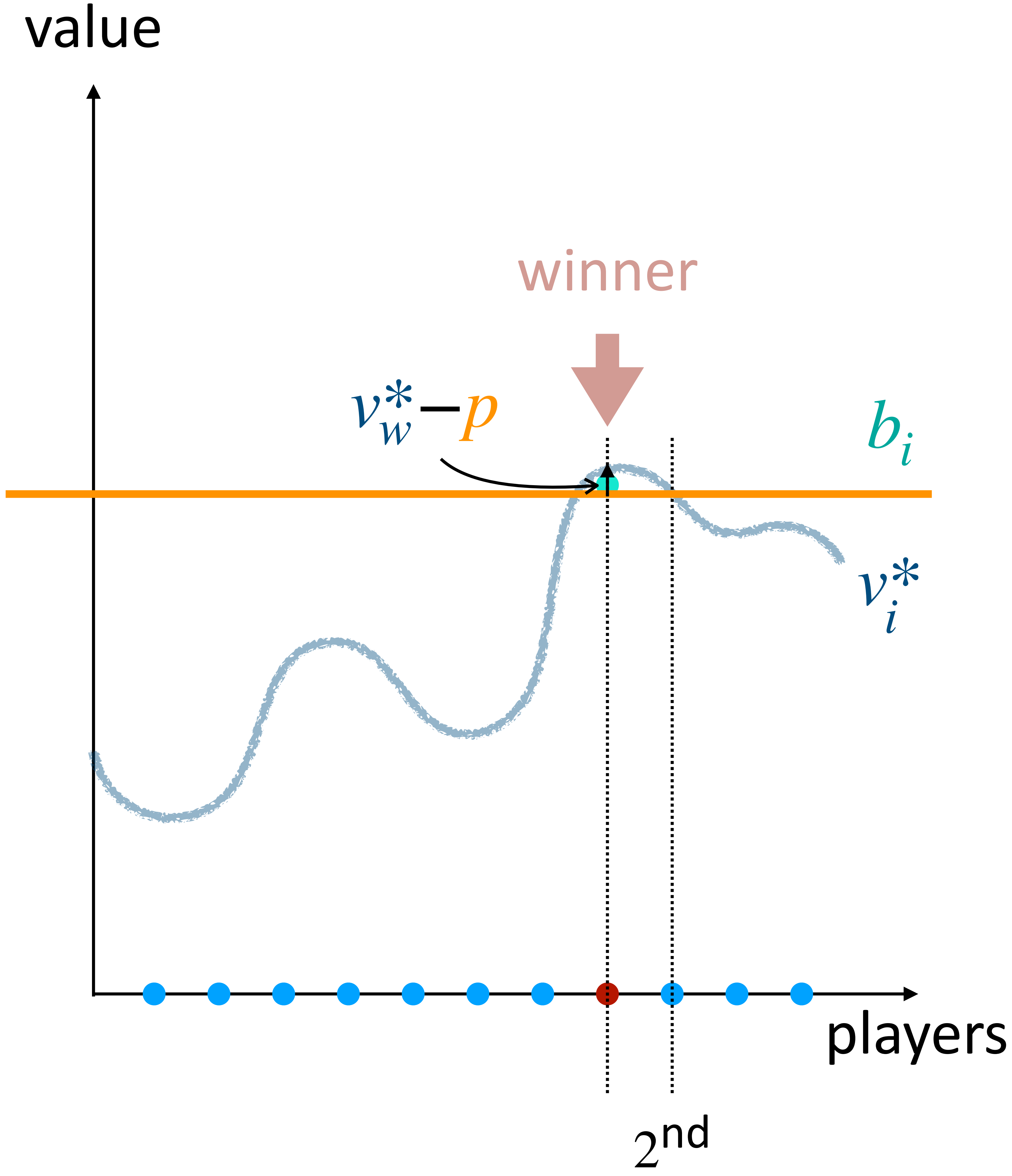
Vickrey's Second Price Auction

- Let the winner be the player i with the highest declared value of b_i, and let i pay the second highest declared bid. That is, $p = \max_{j \neq i} b_j$
- For every $b_1, b_2, \cdots, b_n$ and v_i^*, let u_i^* be i's utility if it bids v_i^* and u_i his utility if he bids b_i. Then, $u_i^* \geq u_i$.
    - If i is the winner:
        - If $b_i > v_i^*$ or $v_i^* > b_i > p$: player i still wins, $u_i = v_i^* - p = u_i^*$
        - If $b_i < v_i^*$ and $p > b_i$: player i loses, $u_i = 0 \leq u_i^*$



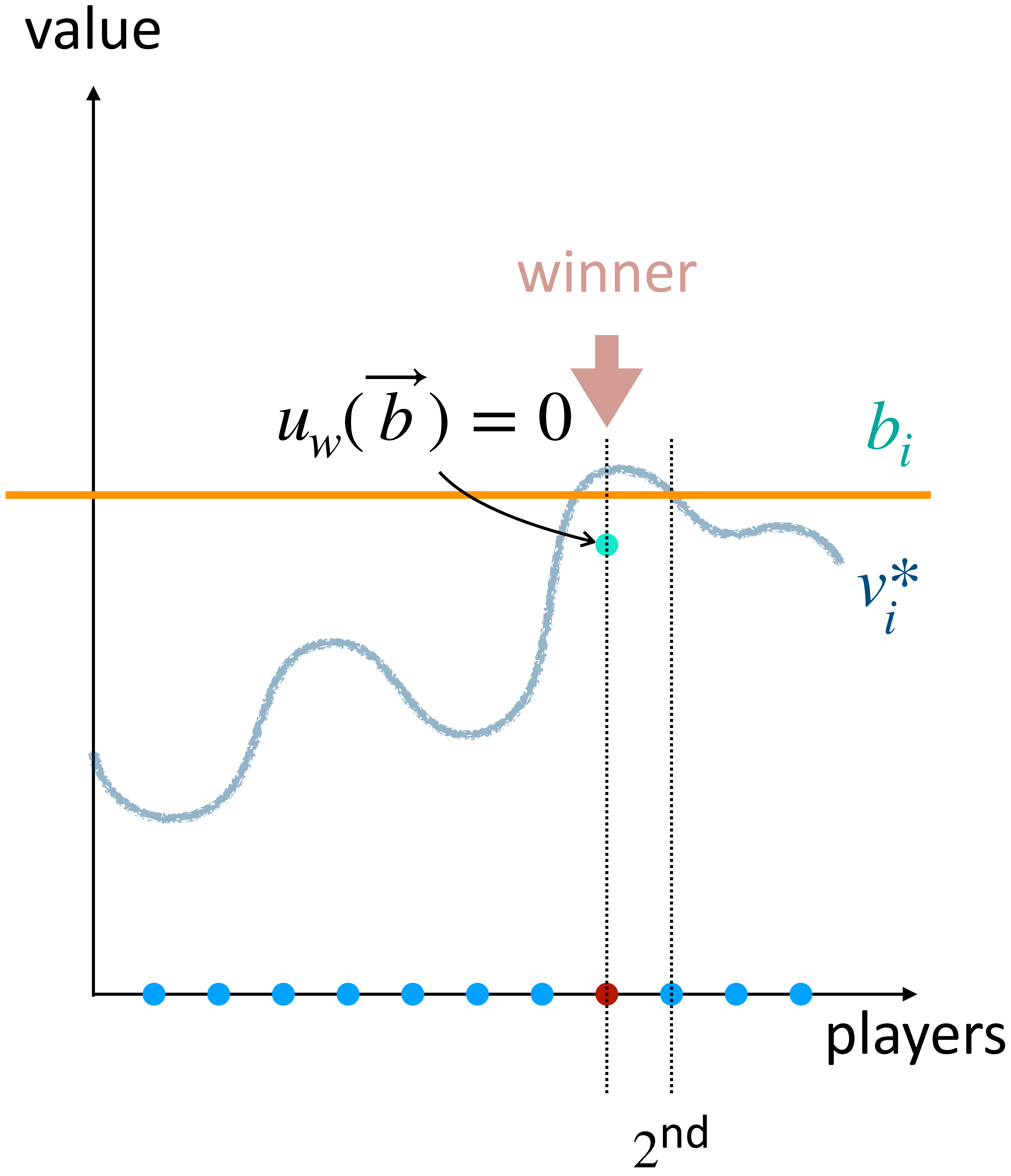
Vickrey’s Second Price Auction

- Let the winner be the player i with the highest declared value of b_i, and let i pay the second highest declared bid. That is, $p = \max_{j \neq i} b_j$
- For every $b_1, b_2, \cdots, b_n$ and v_i^*, let u_i^* be i's utility if it bids v_i^* and u_i his utility if he bids b_i. Then, $u_i^* \geq u_i$.
  - If i is the winner:
    - If $b_i > v_i^*$ or $v_i^* > b_i > p$: player i still wins, $u_i = v_i^* - p = u_i^*$
    - If $b_i < v_i^*$ and $p > b_i$: player i loses, $u_i = 0 \leq u_i^*$



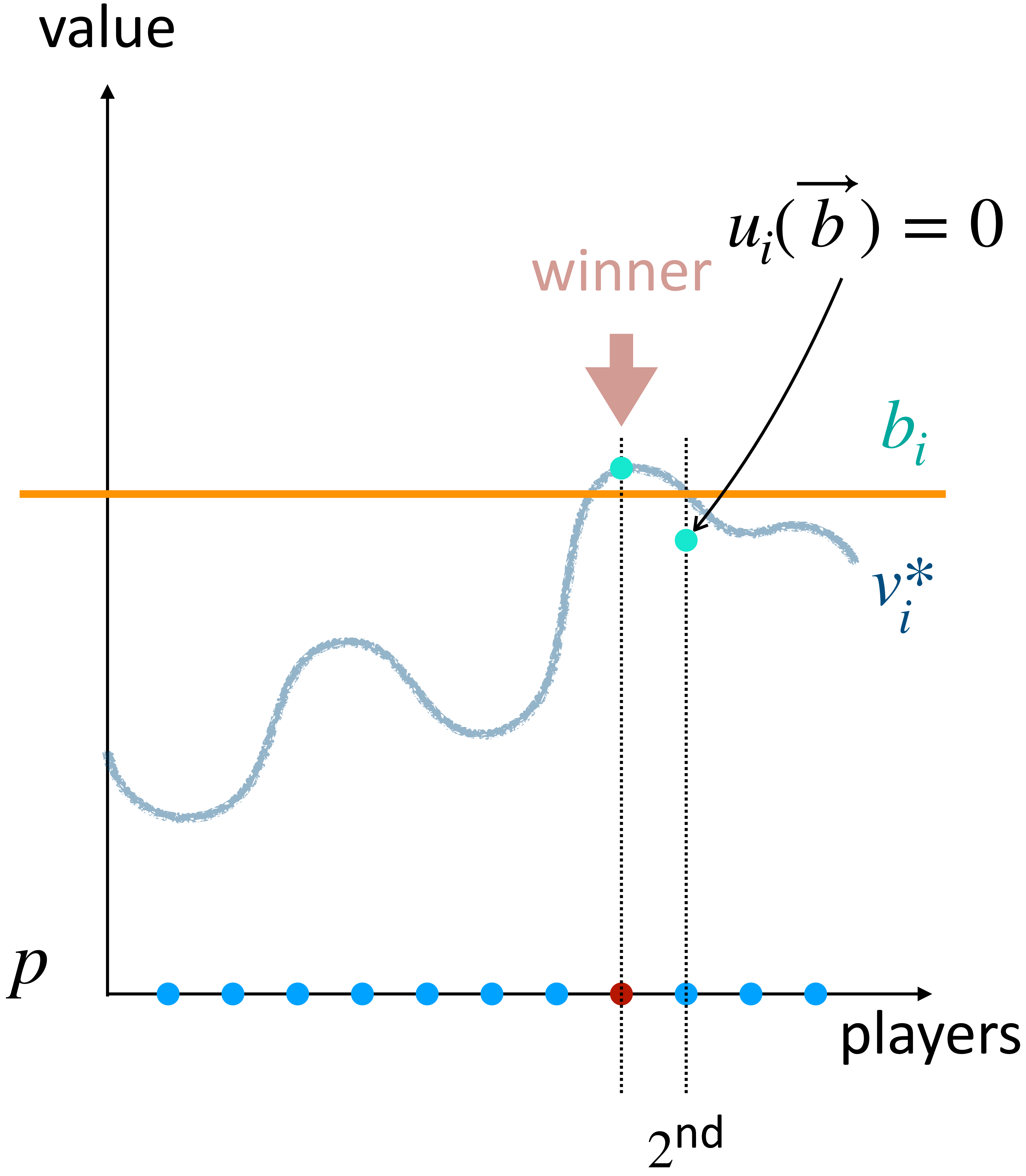
Vickrey’s Second Price Auction

- Let the winner be the player i with the highest declared value of b_i, and let i pay the second highest declared bid. That is, $p = \max_{j \neq i} b_j$
- For every $b_1, b_2, \cdots, b_n$ and v_i^*, let u_i^* be i's utility if it bids v_i^* and u_i his utility if he bids b_i. Then, $u_i^* \geq u_i$.
  - If i is the winner:
    - If $b_i > v_i^*$ or $v_i^* > b_i > p$: player i still wins, $u_i = v_i^* - p = u_i^*$
    - If $b_i < v_i^*$ and $p > b_i$: player i loses, $u_i = 0 \leq u_i^*$



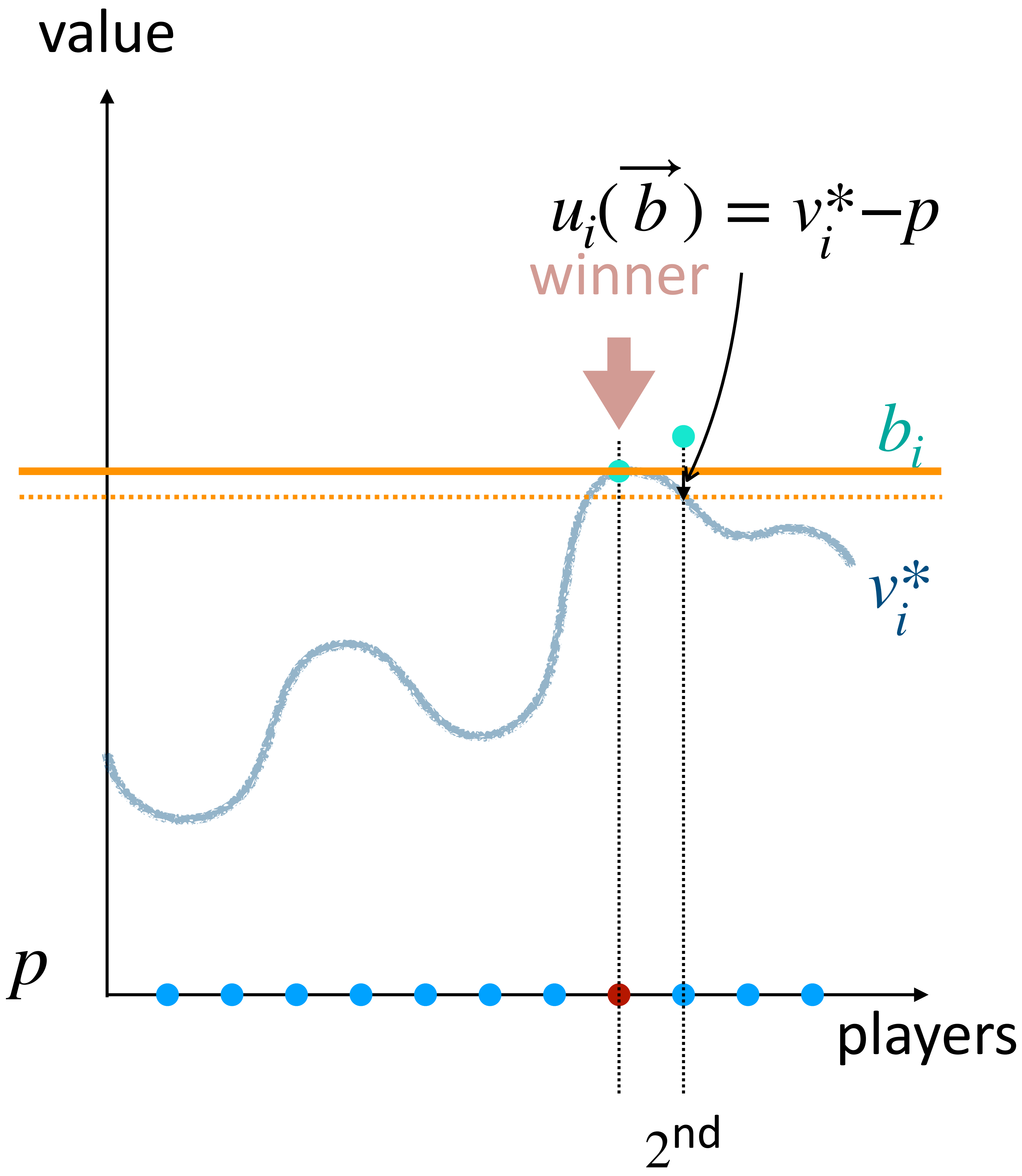
Vickrey's Second Price Auction

- Let the winner be the player i with the highest declared value of b_i, and let i pay the second highest declared bid. That is, $p = \max_{j \neq i} b_j$
- For every $b_1, b_2, \cdots, b_n$ and v_i^*, let u_i^* be i's utility if it bids v_i^* and u_i his utility if he bids b_i. Then, $u_i^* \geq u_i$.
  - If i is a loser (given that winner is honest):
    - If $b_i < v_i^*$: player i still loses, $u_i = 0 = u_i^*$
    - If $b_i > v_i^*$: player i wins, $u_i = v_i^* - p \leq 0 = u_i^*$ (since $p \geq v_w^* \geq v_i^*$)



Vickrey’s Second Price Auction

- Let the winner be the player i with the highest declared value of b_i, and let i pay the second highest declared bid. That is, $p = \max_{j \neq i} b_j$
- For every $b_1, b_2, \cdots, b_n$ and v_i^*, let u_i^* be i's utility if it bids v_i^* and u_i his utility if he bids b_i. Then, $u_i^* \geq u_i$.
  - If i is a loser (given that winner is honest):
    - If $b_i < v_i^*$: player i still loses, $u_i = 0 = u_i^*$
    - If $b_i > v_i^*$: player i wins, $u_i = v_i^* - p \leq 0 = u_i^*$ (since $p \geq v_w^* \geq v_i^*$)



What happened

- In the auction game, Vickrey's second price auction (letting the winner pays the second-highest bid) is strategy-proof

Outline

- Fundamental concepts
  - Game, players, strategies, payoffs/costs
- Nash Equilibrium
- Price of Anarchy
  - Selfish load balancing
- Mechanism design
  - Auction
  - Vickrey-Clarke-Groves mechanism

Mechanism Design

- Social choice: an aggregation of the preference of the different participants toward a single joint decision

Mechanism Design

- Social choice: an aggregation of the preference of the different participants toward a single joint decision
- Mechanism Design attempts implementing desired social choices in a strategic setting

Mechanism Design

- Social choice: an aggregation of the preference of the different participants toward a single joint decision
- Mechanism Design attempts implementing desired social choices in a strategic setting
    - The different members of society act *rationally* in a game theoretic sense

Mechanism Design

- Social choice: an aggregation of the preference of the different participants toward a single joint decision
- Mechanism Design attempts implementing desired social choices in a strategic setting
    - The different members of society act *rationally* in a game theoretic sense
    - The preference of the participants are private

Mechanism Design

- Social choice: an aggregation of the preference of the different participants toward a single joint decision
- Mechanism Design attempts implementing desired social choices in a strategic setting
  - The different members of society act *rationally* in a game theoretic sense
  - The preference of the participants are private
  - Examples: elections, markets, auctions, government policy, etc

Outline

- Fundamental concepts
  - Game, players, strategies, payoffs/costs
- Nash Equilibrium
- Price of Anarchy
  - Selfish load balancing
- Mechanism design
  - Auction
  - Vickrey-Clarke-Groves mechanism

Vickrey-Clarke-Groves Mechanism

- A mechanism $(f, p_1, p_2, \cdots, p_n)$ is called a **Vickrey-Clarke-Groves (VCG) mechanism** if
  - $f(\vec{s}) \in \arg\max_{a \in A} \Sigma_i \, s_i(a)$, and
  - for some functions $h_1, h_2, \cdots, h_n$, where h_i is a function of $\overrightarrow{s_{-i}}$, we have that for all i: $p_i(\vec{s}) = h_i(\overrightarrow{s_{-i}}) - \Sigma_{j \neq i} \, s_j(f(\vec{s}))$

A: all possible outcome actions by the mechanism

$s_i(a)$: player i's (reported) utility/cost given the action a

Vickrey-Clarke-Groves Mechanism

- A mechanism $(f, p_1, p_2, \cdots, p_n)$ is called a **Vickrey-Clarke-Groves (VCG) mechanism** if
  - $f(\vec{s}) \in \arg\max_{a \in A} \Sigma_i \, s_i(a)$, and
  - for some functions $h_1, h_2, \cdots, h_n$, where h_i is a function of $\overrightarrow{s_{-i}}$, we have that for all i: $p_i(\vec{s}) = h_i(\overrightarrow{s_{-i}}) - \Sigma_{j \neq i} \, s_j(f(\vec{s}))$

A: all possible outcome actions by the mechanism
$s_i(a)$: player i's (reported) utility/cost given the action a

Vickrey-Clarke-Groves Mechanism

- A mechanism $(f, p_1, p_2, \cdots, p_n)$ is called a **Vickrey-Clarke-Groves (VCG) mechanism** if
  - $f(\vec{s}) \in \arg\max_{a \in A} \Sigma_i \, s_i(a)$, and
  - for some functions $h_1, h_2, \cdots, h_n$, where h_i is a function of $\overrightarrow{s_{-i}}$, we have that for all i: $p_i(\vec{s}) = \underbrace{h_i(\overrightarrow{s_{-i}})}_{\text{has nothing to do with } s_i} - \Sigma_{j \neq i} \, s_j(f(\vec{s}))$

A: all possible outcome actions by the mechanism

$s_i(a)$: player i's (reported) utility/cost given the action a

VCG Mechanism is strategy-proof

- Fix i, $\overrightarrow{s_{-i}}$, v_i^*, and s_i

VCG Mechanism is strategy-proof

- Fix i, $\overrightarrow{s_{-i}}$, v_i^*, and s_i. We need to show that for player i with (true) valuation v_i^*, the utility when declaring v_i^* is not less than the utility when declaring s_i

VCG Mechanism is strategy-proof

- Fix i, $\overrightarrow{s_{-i}}$, v_i^*, and s_i. We need to show that for player i with (true) valuation v_i^*, the utility when declaring v_i^* is not less than the utility when declaring s_i
- Let $a^* = f(v_i^*, \overrightarrow{s_{-i}})$ and $a = f(s_i, \overrightarrow{s_{-i}})$

VCG Mechanism is strategy-proof

- Fix i, $\overrightarrow{s_{-i}}$, v_i^*, and s_i. We need to show that for player i with (true) valuation v_i^*, the utility when declaring v_i^* is not less than the utility when declaring s_i
- Let $a^* = f(v_i^*, \overrightarrow{s_{-i}})$ and $a = f(s_i, \overrightarrow{s_{-i}})$ others' declaration do not change

outcome when declaring truthfully outcome when declaring strategically

VCG Mechanism is strategy-proof

- Fix i, $\overrightarrow{s_{-i}}$, v_i^*, and s_i. We need to show that for player i with (true) valuation v_i^*, the utility when declaring v_i^* is not less than the utility when declaring s_i
- Let $a^* = f(v_i^*, \overrightarrow{s_{-i}})$ and $a = f(s_i, \overrightarrow{s_{-i}})$
  - Utility of i when (truthfully) declaring v_i^* is

$$v_i^*(a^*) - p_i(a^*)$$

VCG Mechanism is strategy-proof

- Fix i, $\overrightarrow{s_{-i}}$, v_i^*, and s_i. We need to show that for player i with (true) valuation v_i^*, the utility when declaring v_i^* is not less than the utility when declaring s_i
- Let $a^* = f(v_i^*, \overrightarrow{s_{-i}})$ and $a = f(s_i, \overrightarrow{s_{-i}})$
  - Utility of i when (truthfully) declaring v_i^* is

$$v_i^*(a^*) - p_i(a^*) \underset{\overset{\uparrow}{\text{VCG}}}{=} v_i^*(a^*) - h_i(\overrightarrow{s_{-i}}) + \Sigma_{j \neq i}\, v_j^*(a^*)$$

VCG Mechanism is strategy-proof

- Fix i, $\overrightarrow{s_{-i}}$, v_i^*, and s_i. We need to show that for player i with (true) valuation v_i^*, the utility when declaring v_i^* is not less than the utility when declaring s_i
- Let $a^* = f(v_i^*, \overrightarrow{s_{-i}})$ and $a = f(s_i, \overrightarrow{s_{-i}})$
  - Utility of i when (truthfully) declaring v_i^* is

$$v_i^*(a^*) - p_i(a^*) = v_i^*(a^*) - h_i(\overrightarrow{s_{-i}}) + \Sigma_{j\neq i}\, v_j^*(a^*)$$

  - Utility of i when (strategically) declaring s_i is

$$v_i^*(a) - p_i(a) = v_i^*(a) - h_i(\overrightarrow{s_{-i}}) + \Sigma_{j\neq i}\, v_j^*(a)$$

VCG Mechanism is strategy-proof

- Fix i, $\overrightarrow{s_{-i}}$, v_i^*, and s_i. We need to show that for player i with (true) valuation v_i^*, the utility when declaring v_i^* is not less than the utility when declaring s_i
- Let $a^* = f(v_i^*, \overrightarrow{s_{-i}})$ and $a = f(s_i, \overrightarrow{s_{-i}})$
  - Utility of i when (truthfully) declaring v_i^* is

$$v_i^*(a^*) - p_i(a^*) = v_i^*(a^*) - h_i(\overrightarrow{s_{-i}}) + \Sigma_{j\neq i}\, v_j^*(a^*)$$

  - Utility of i when (strategically) declaring s_i is

$$v_i^*(a) - p_i(a) = v_i^*(a) - h_i(\overrightarrow{s_{-i}}) + \Sigma_{j\neq i}\, v_j^*(a)$$

  - Since social welfare of $a^* = v_i^*(a^*) + \Sigma_{j\neq i} s_i(a^*) \geq v_i^*(a') + \Sigma_{j\neq i} s_i(a')$ for any a'

$$v_i^*(a^*) - p_i(a^*) \geq v_i^*(a) - p_i(a)$$

Vickrey-Clarke-Groves Mechanism

- A mechanism $(f, p_1, p_2, \cdots, p_n)$ is called a **Vickrey-Clarke-Groves (VCG) mechanism** if
  - $f(\vec{s}) \in \arg\max_{a \in A} \Sigma_i \, s_i(a)$, and
  - for some functions $h_1, h_2, \cdots, h_n$, where h_i is a function of $\overrightarrow{s_{-i}}$, we have that for all i: $p_i(\vec{s}) = h_i(\overrightarrow{s_{-i}}) - \Sigma_{j \neq i} \, s_j(f(\vec{s}))$

Vickrey-Clarke-Groves Mechanism

- A mechanism $(f, p_1, p_2, \cdots, p_n)$ is called a **Vickrey-Clarke-Groves (VCG) mechanism** if
  - $f(\vec{s}) \in \underset{a \in A}{\arg\max}\ \Sigma_i\, s_i(a)$, and
  - for some functions $h_1, h_2, \cdots, h_n$, where h_i is a function of $\overrightarrow{s_{-i}}$, we have that for all i: $p_i(\vec{s}) = h_i(\overrightarrow{s_{-i}}) - \Sigma_{j \neq i}\, s_j(f(\vec{s}))$

- **Clarke pivot rule**: $p_i(\vec{s}) = \underset{a \in A}{\max}\, \Sigma_{j \neq i} s_i(a) - \Sigma_{j \neq i}\, s_j(f(\vec{s}))$

Vickrey-Clarke-Groves Mechanism

- A mechanism $(f, p_1, p_2, \cdots, p_n)$ is called a **Vickrey-Clarke-Groves (VCG) mechanism** if
  - $f(\vec{s}) \in \arg\max_{a \in A} \Sigma_i\, s_i(a)$, and
  - for some functions $h_1, h_2, \cdots, h_n$, where h_i is a function of $\overrightarrow{s_{-i}}$, we have that for all i: $p_i(\vec{s}) = h_i(\overrightarrow{s_{-i}}) - \Sigma_{j \neq i}\, s_j(f(\vec{s}))$

- **Clarke pivot rule**: $p_i(\vec{s}) = \underbrace{\max_{a \in A} \Sigma_{j \neq i} s_i(a)}_{\text{others' social welfare without } i} - \underbrace{\Sigma_{j \neq i}\, s_j(f(\vec{s}))}_{\text{others' social welfare with } i}$

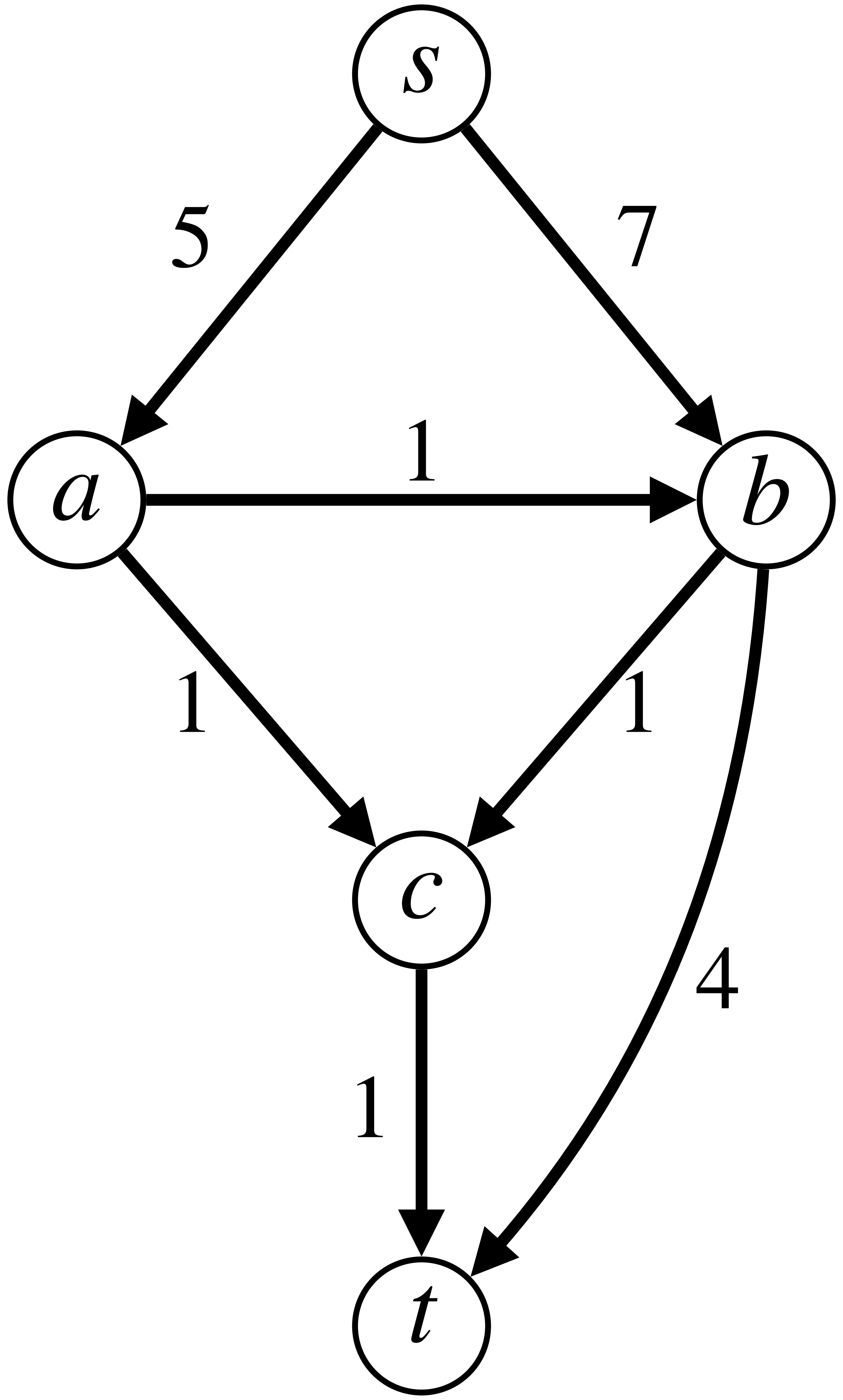
What happened

- The **Vickrey-Clarke-Groves (VCG) mechanism** is strategy-proof
  - As long as a mechanism is a VCG, it is strategy-proof

- **Clarke pivot rule**: $p_i(\vec{s}) = \max_{a \in A} \Sigma_{j \neq i} s_i(a) - \Sigma_{j \neq i}\, s_j(f(\vec{s}))$

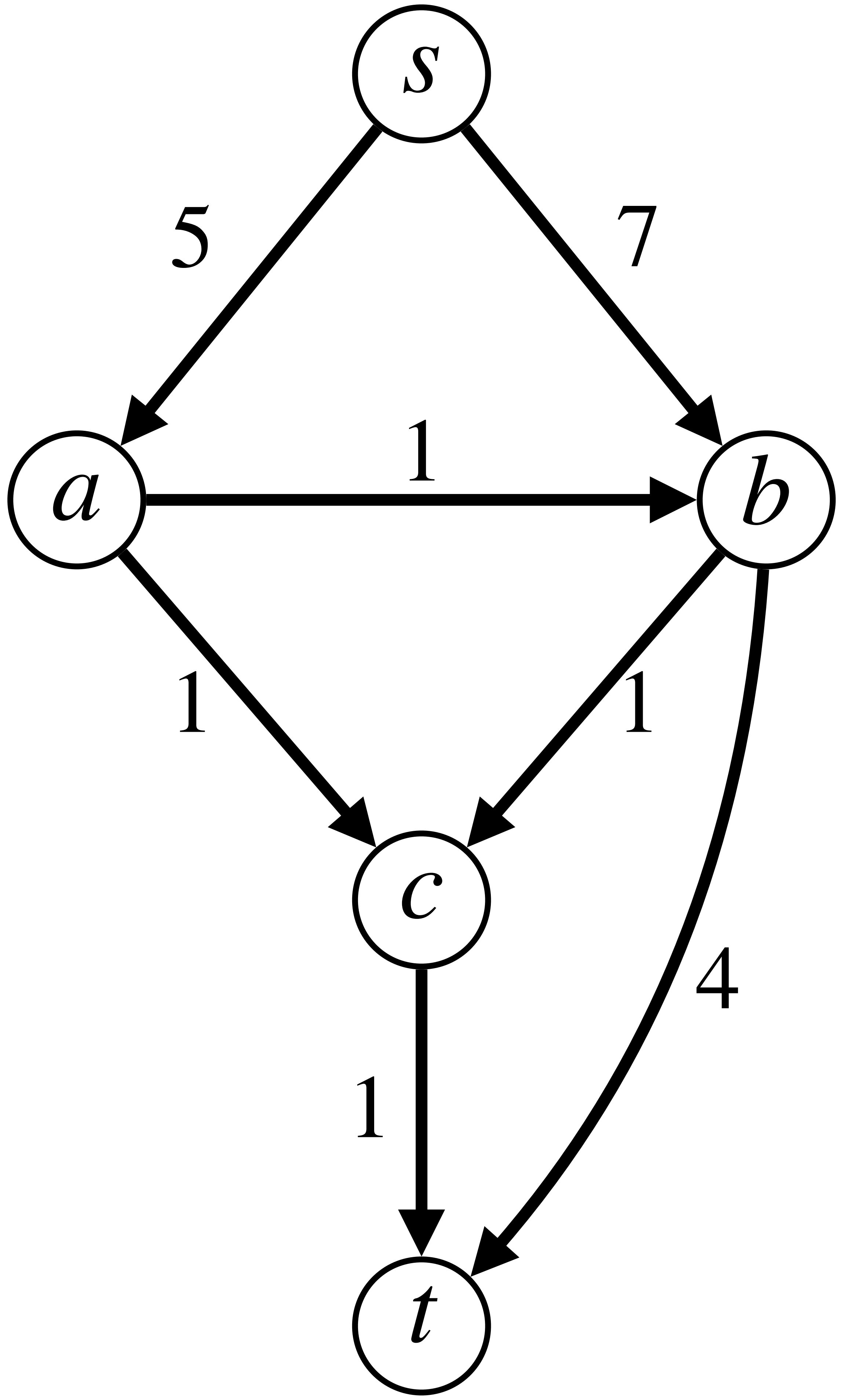
Shortest Path

- Given a directed graph $G = (V, E)$, where each edge $e \in E$ is owned by a player. The player e has a (private) value v_e



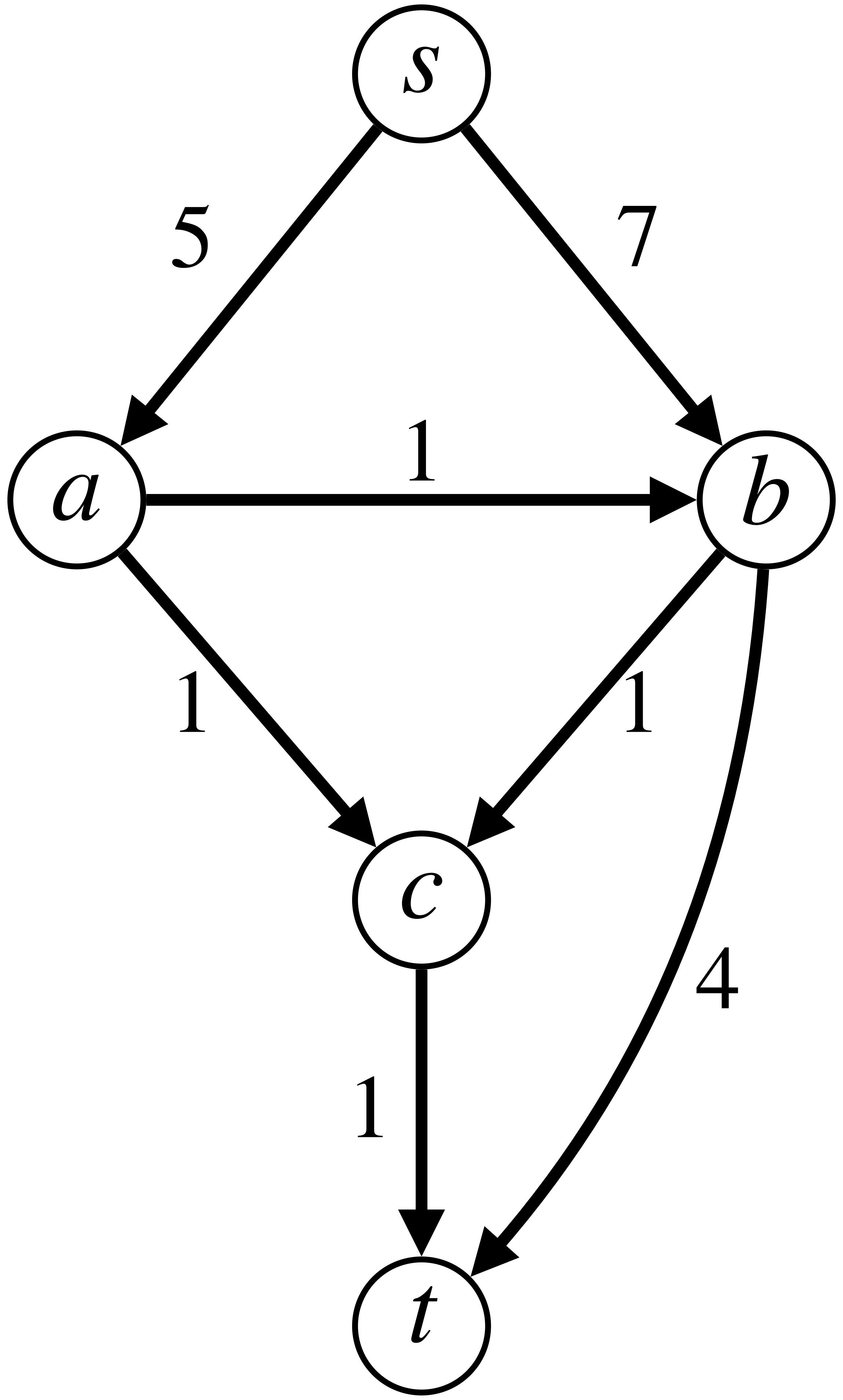
Shortest Path

- Given a directed graph $G = (V, E)$, where each edge $e \in E$ is owned by a player. The player e has a (private) value v_e
- The government wants to expropriate some edges to build a path from s to t, where $s, t \in V$



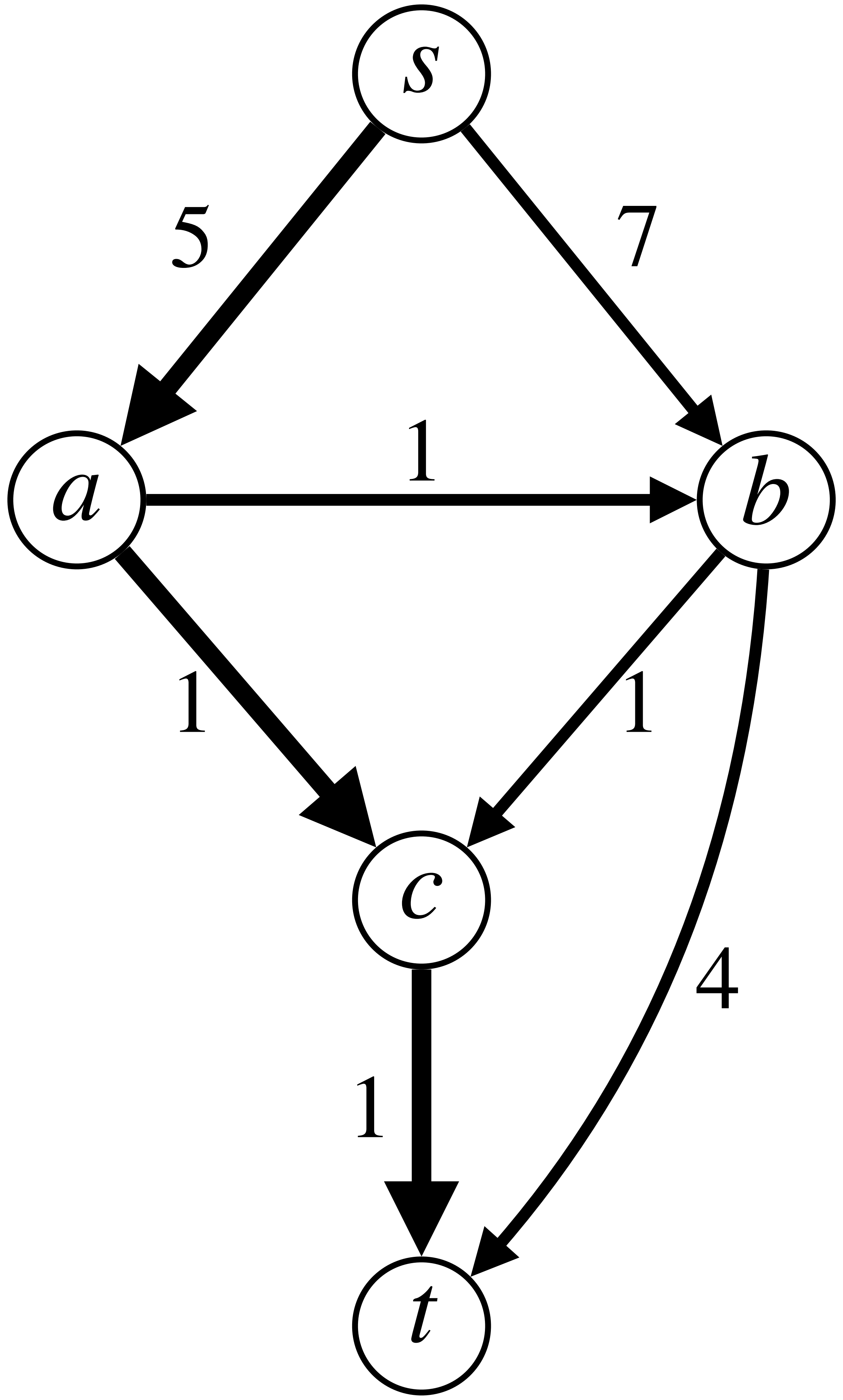
Shortest Path

- Given a directed graph $G = (V, E)$, where each edge $e \in E$ is owned by a player. The player e has a (private) value v_e
- The government wants to expropriate some edges to build a path from s to t, where $s, t \in V$
  - If an edge e is used, the corresponding player considers to lose a cost of v_e
  - To minimize the social cost, which is Σ_e is expropriated $-v_e$, how should the government set a price for each player?



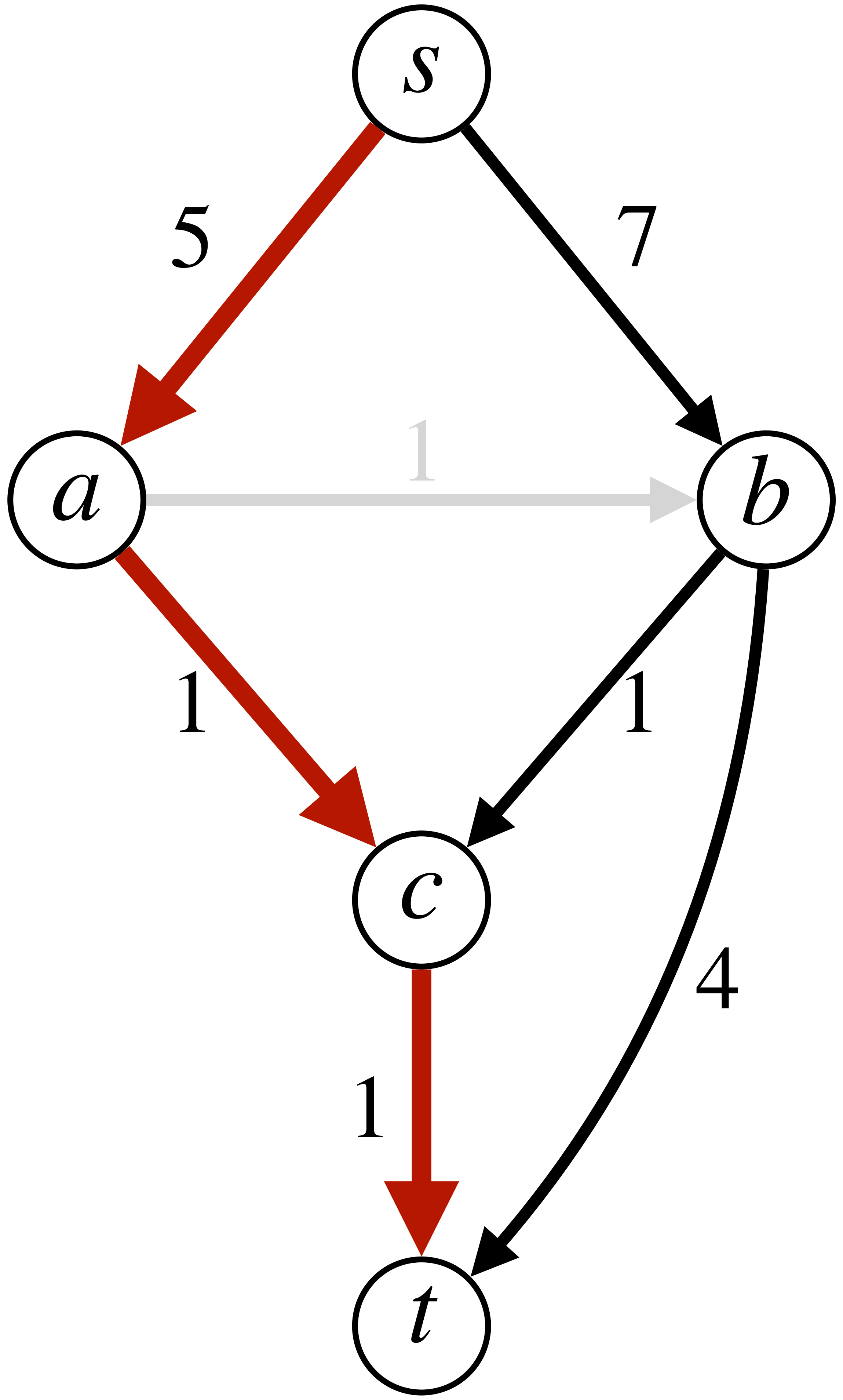
Shortest Path

- Given a directed graph $G = (V, E)$, where each edge $e \in E$ is owned by a player. The player e has a (private) value v_e
- The government wants to expropriate some edges to build a path from s to t, where $s, t \in V$
  - If an edge e is used, the corresponding player considers to lose a cost of v_e
  - To minimize the social cost, which is Σ_e is expropriated $-v_e$, how should the government set a price for each player?



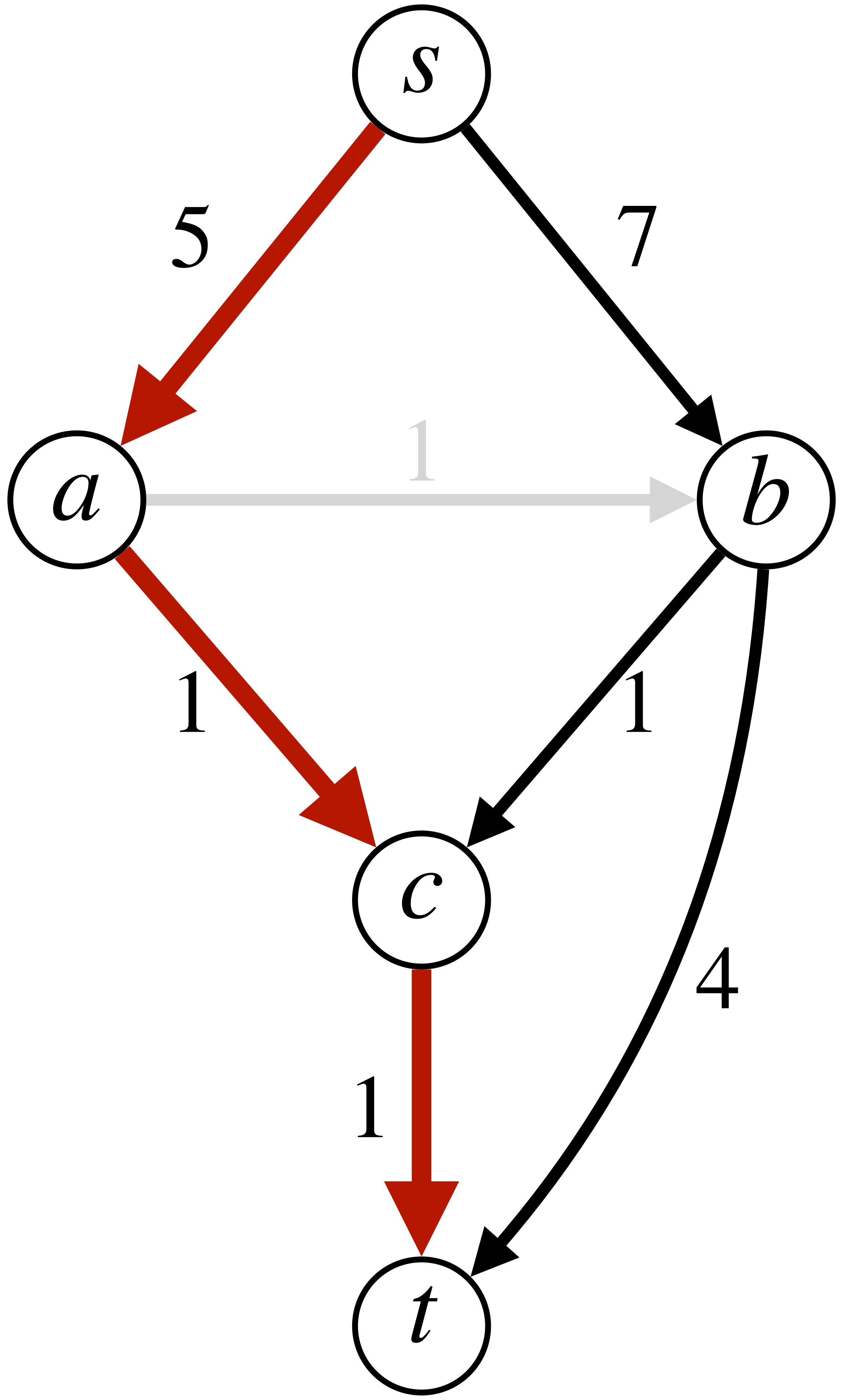
Shortest Path — Using VCG

- Given a directed graph $G = (V, E)$, where each edge $e \in E$ is owned by a player. The player e has a (private) value v_e
- $p_e(\hat{P}) = \max_{\text{path } P} \Sigma_{j \neq e \text{ and } j \in P}(-v_j) - \Sigma_{j \neq e}(-v_j(\hat{P}))$



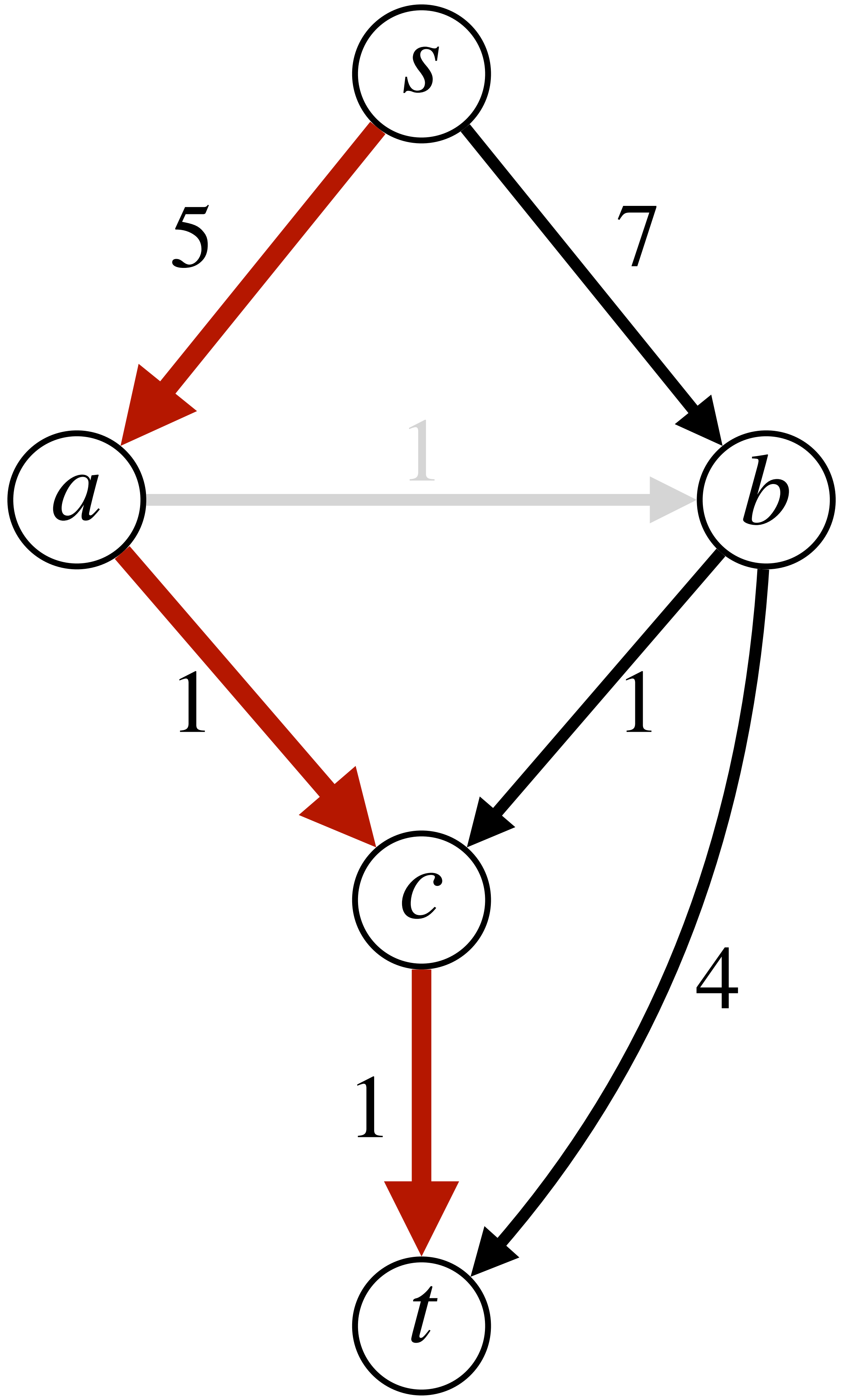
Shortest Path — Using VCG

- Given a directed graph $G = (V, E)$, where each edge $e \in E$ is owned by a player. The player e has a (private) value v_e
- $p_e(\hat{P}) = \max\limits_{\text{path } P} \Sigma_{j \neq e \text{ and } j \in P}(-v_j) - \Sigma_{j \neq e}(-v_j(\hat{P}))$
  - If $e \notin \hat{P}$, $p_e(\hat{P}) = \max\limits_{P' \text{ in } G'} \Sigma_{j \neq e \text{ and } j \in P'}(-v_j) - \Sigma_{j \neq e}(-v_j(\hat{P})) = 0$



Shortest Path — Using VCG

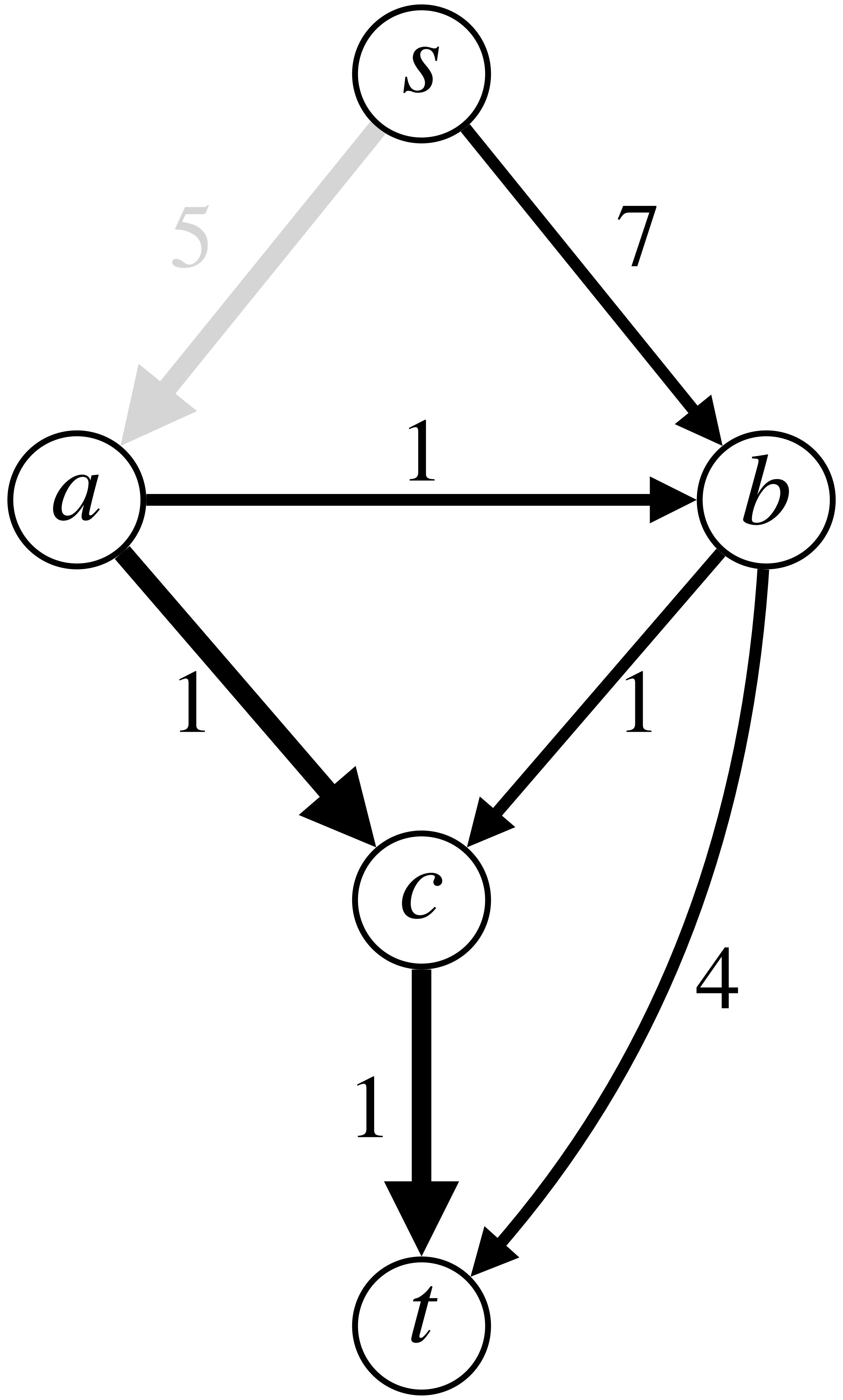
- Given a directed graph $G = (V, E)$, where each edge $e \in E$ is owned by a player. The player e has a (private) value v_e
- $p_e(\hat{P}) = \max_{\text{path } P} \Sigma_{j \neq e \text{ and } j \in P}(-v_j) - \Sigma_{j \neq e}(-v_j(\hat{P}))$
  - If $e \notin \hat{P}$, $p_e(\hat{P}) = \max_{P' \text{ in } G'} \Sigma_{j \neq e \text{ and } j \in P'}(-v_j) - \Sigma_{j \neq e}(-v_j(\hat{P})) = 0$



$$p_{(a,b)} = \max_{P' \text{ in } G'} \Sigma_{e \in P'}(-v_e) - \Sigma_{(a,b) \neq e \in P}(-v_e)$$
$$= -7 - (-7) \;= 0$$

Shortest Path — Using VCG

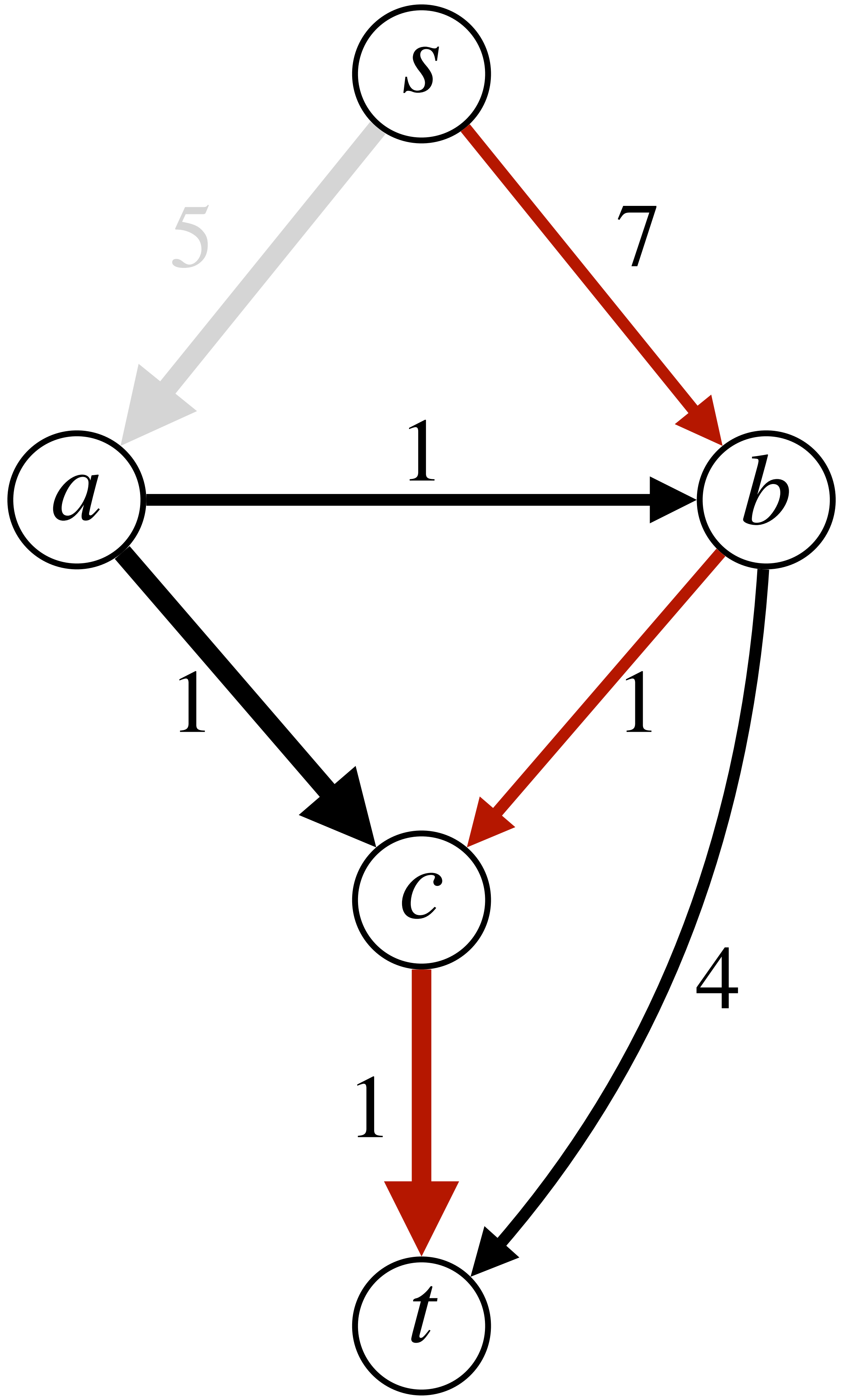
- Given a directed graph $G = (V, E)$, where each edge $e \in E$ is owned by a player. The player e has a (private) value v_e
- $p_e(\hat{P}) = \max\limits_{\text{path } P} \Sigma_{j \neq e \text{ and } j \in P}(-v_j) - \Sigma_{j \neq e}\,(-v_j(\hat{P}))$
  - If $e \notin \hat{P}$, $p_e(\hat{P}) = \max\limits_{P' \text{ in } G'} \Sigma_{j \neq e \text{ and } j \in P'}(-v_j) - \Sigma_{j \neq e}\,(-v_j(\hat{P})) = 0$
  - If $e \in \hat{P}$, $p_e(\hat{P}) = \max\limits_{P' \text{ in } G'} \Sigma_{j \neq e \text{ and } j \in P'}(-v_j) + \Sigma_{j \neq e}\, v_j(\hat{P})$



Shortest Path — Using VCG

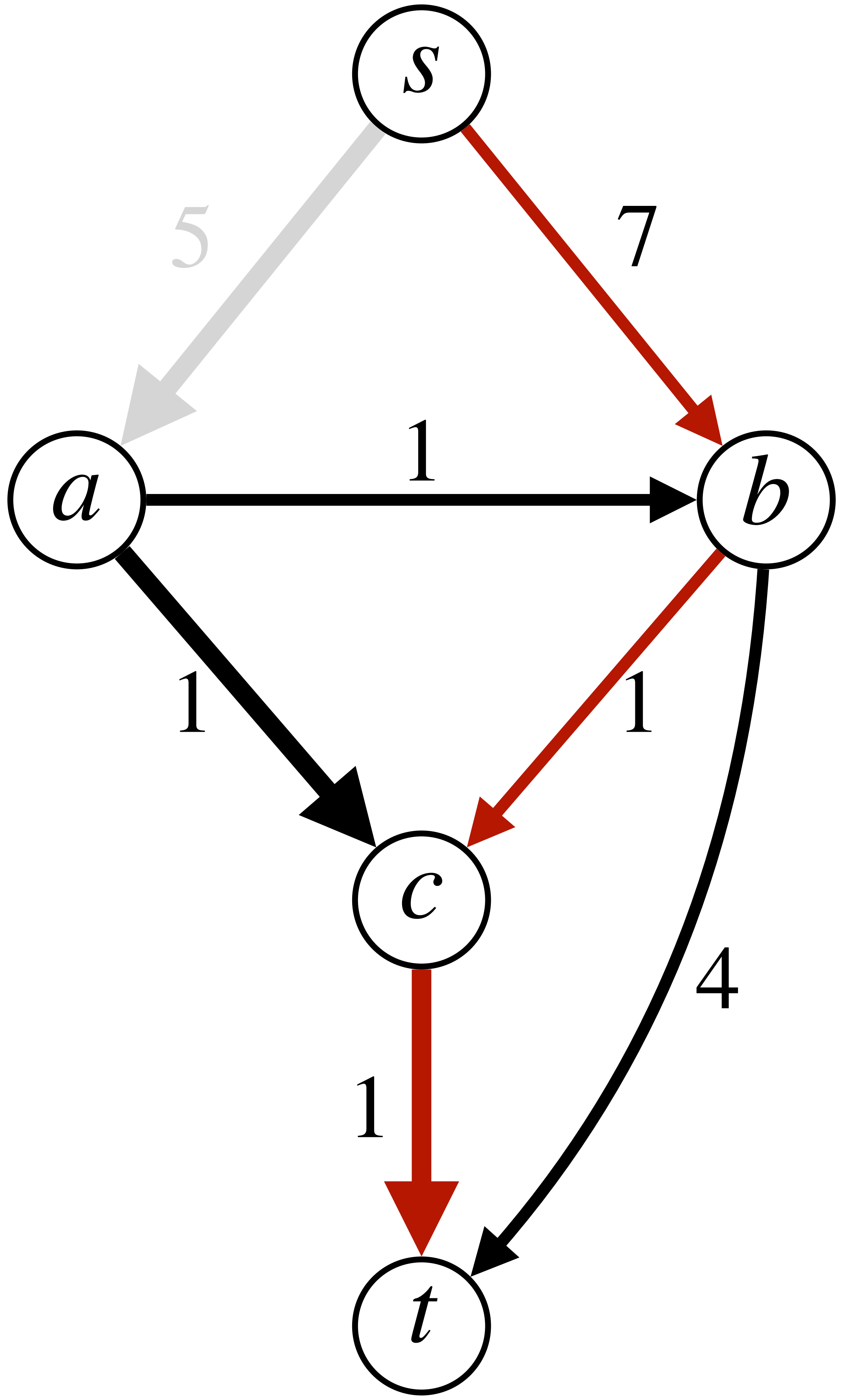
- Given a directed graph $G = (V, E)$, where each edge $e \in E$ is owned by a player. The player e has a (private) value v_e

- $p_e(\hat{P}) = \max\limits_{\text{path } P} \Sigma_{j \neq e \text{ and } j \in P}(-v_j) - \Sigma_{j \neq e}\,(-v_j(\hat{P}))$
  - If $e \notin \hat{P}$, $p_e(\hat{P}) = \max\limits_{P' \text{ in } G'} \Sigma_{j \neq e \text{ and } j \in P'}(-v_j) - \Sigma_{j \neq e}\,(-v_j(\hat{P})) = 0$
  - If $e \in \hat{P}$, $p_e(\hat{P}) = \max\limits_{P' \text{ in } G'} \Sigma_{j \neq e \text{ and } j \in P'}(-v_j) + \Sigma_{j \neq e}\, v_j(\hat{P})$



Shortest Path — Using VCG

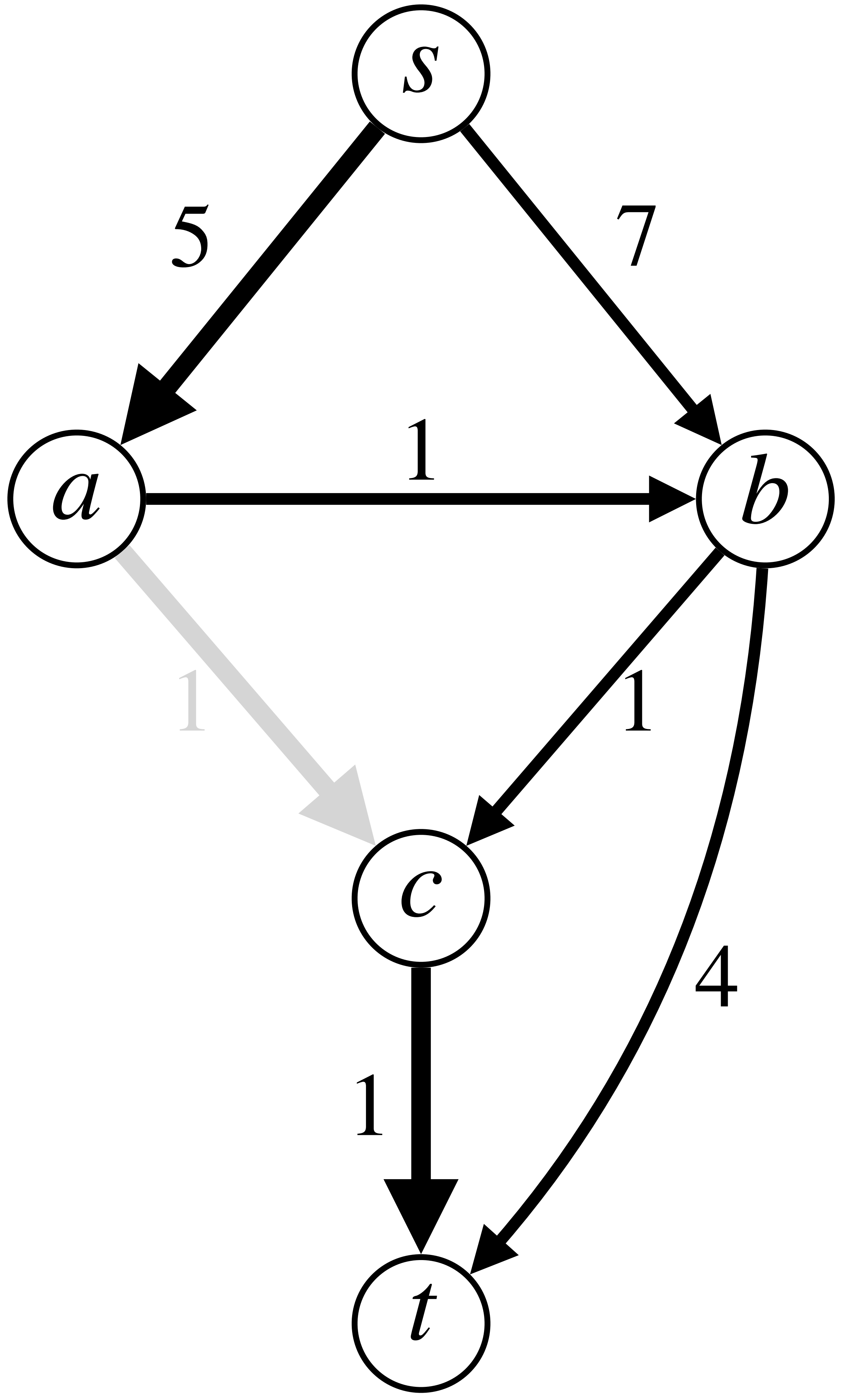
- Given a directed graph $G = (V, E)$, where each edge $e \in E$ is owned by a player. The player e has a (private) value v_e
- $p_e(\hat{P}) = \max\limits_{\text{path } P} \Sigma_{j \neq e \text{ and } j \in P}(-v_j) - \Sigma_{j \neq e}(-v_j(\hat{P}))$
  - If $e \notin \hat{P}$, $p_e(\hat{P}) = \max\limits_{P' \text{ in } G'} \Sigma_{j \neq e \text{ and } j \in P'}(-v_j) - \Sigma_{j \neq e}(-v_j(\hat{P})) = 0$
  - If $e \in \hat{P}$, $p_e(\hat{P}) = \max\limits_{P' \text{ in } G'} \Sigma_{j \neq e \text{ and } j \in P'}(-v_j) + \Sigma_{j \neq e} v_j(\hat{P})$



$$p_{(s,a)} = \max_{P' \text{ in } G'} \Sigma_{e \in P'}(-v_e) - \Sigma_{(s,a) \neq e \in P}(-v_e)$$
$$= -9 - (-2) \ = -7$$

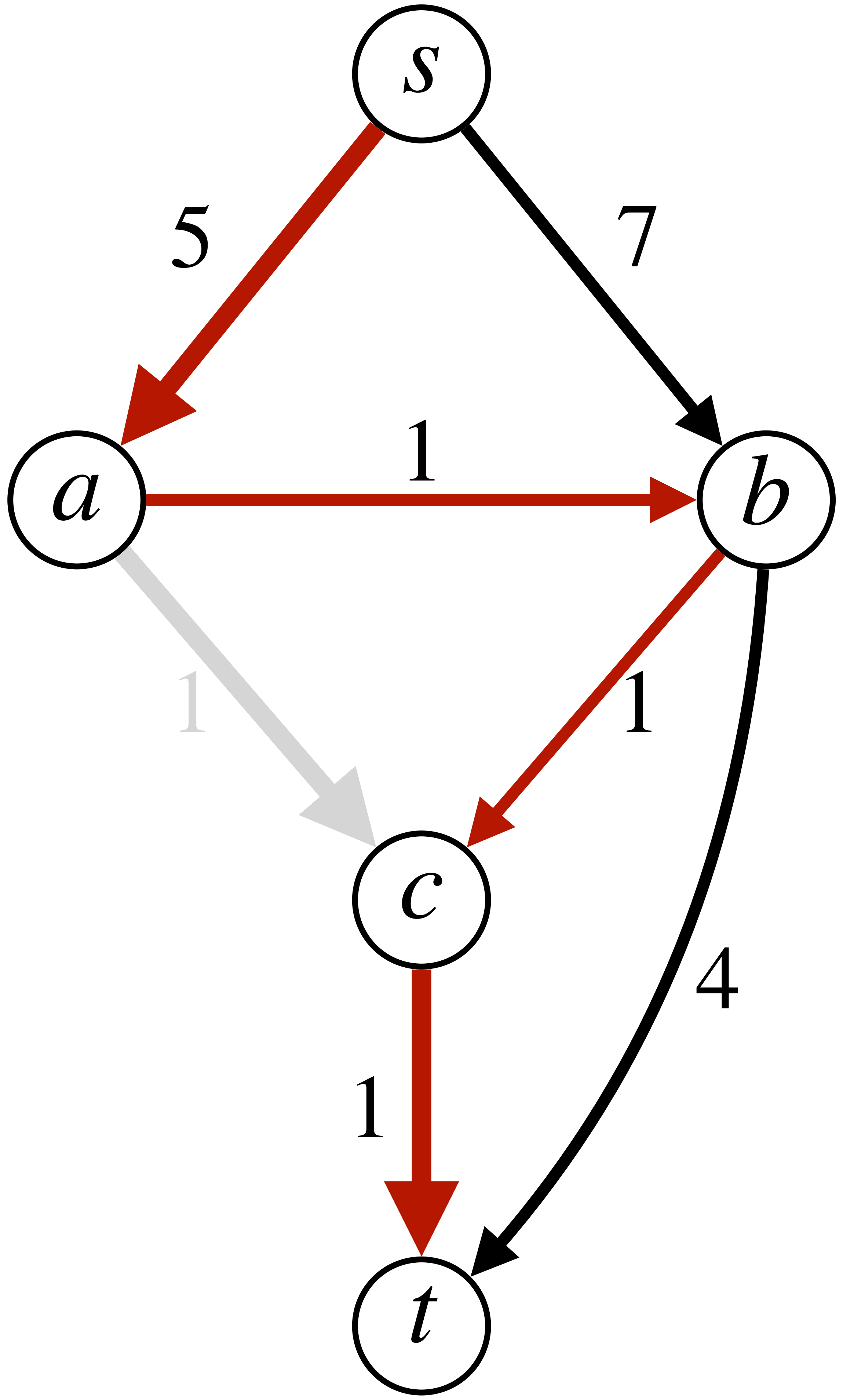
Shortest Path — Using VCG

- Given a directed graph $G = (V, E)$, where each edge $e \in E$ is owned by a player. The player e has a (private) value v_e
- $p_e(\hat{P}) = \max_{\text{path } P} \Sigma_{j \neq e \text{ and } j \in P}(-v_j) - \Sigma_{j \neq e}\,(-v_j(\hat{P}))$
  - If $e \notin \hat{P}$, $p_e(\hat{P}) = \max_{P' \text{ in } G'} \Sigma_{j \neq e \text{ and } j \in P'}(-v_j) - \Sigma_{j \neq e}\,(-v_j(\hat{P})) = 0$
  - If $e \in \hat{P}$, $p_e(\hat{P}) = \max_{P' \text{ in } G'} \Sigma_{j \neq e \text{ and } j \in P'}(-v_j) + \Sigma_{j \neq e}\, v_j(\hat{P})$



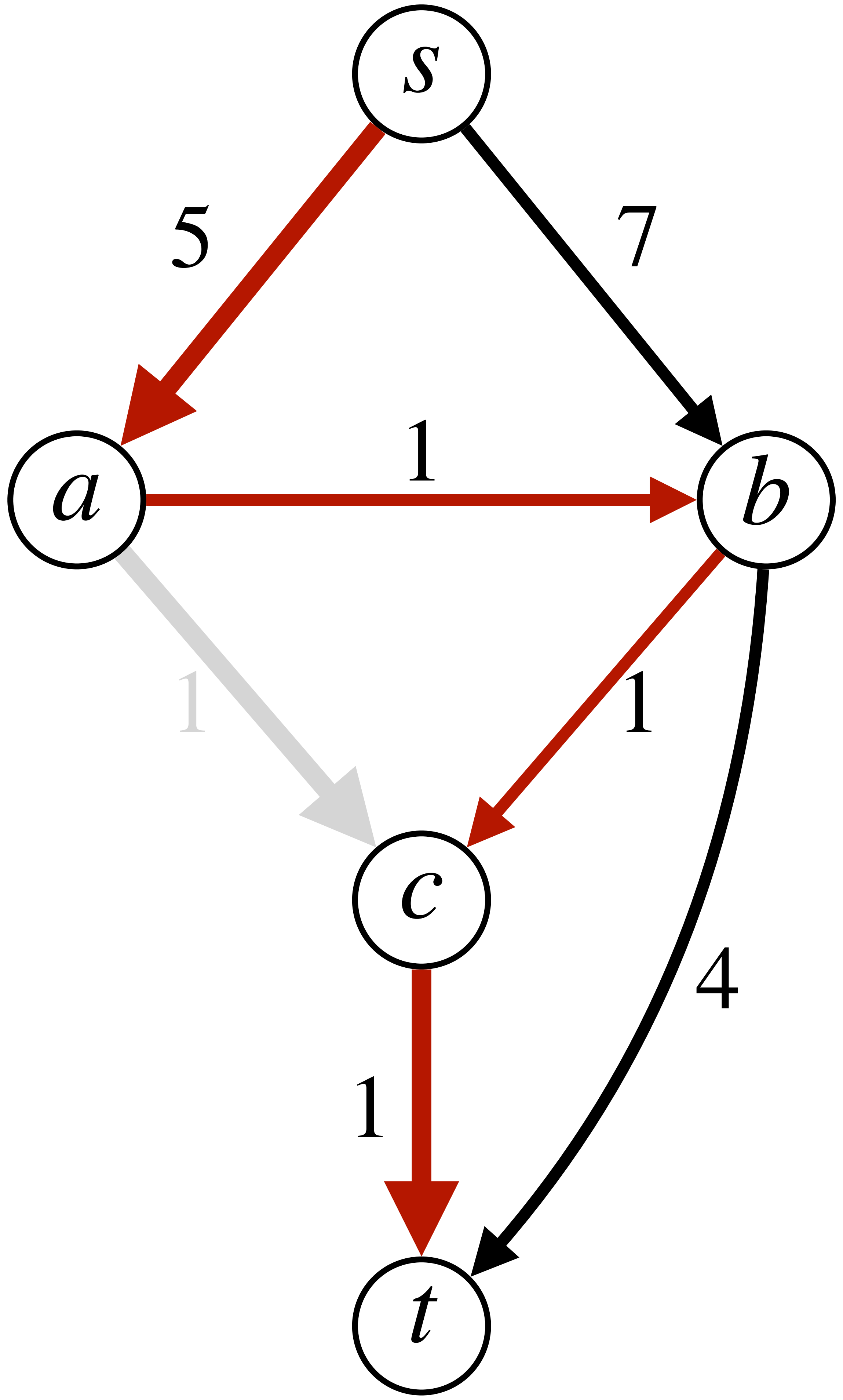
Shortest Path — Using VCG

- Given a directed graph $G = (V, E)$, where each edge $e \in E$ is owned by a player. The player e has a (private) value v_e
- $p_e(\hat{P}) = \max\limits_{\text{path } P} \Sigma_{j \neq e \text{ and } j \in P}(-v_j) - \Sigma_{j \neq e}(-v_j(\hat{P}))$
  - If $e \notin \hat{P}$, $p_e(\hat{P}) = \max\limits_{P' \text{ in } G'} \Sigma_{j \neq e \text{ and } j \in P'}(-v_j) - \Sigma_{j \neq e}(-v_j(\hat{P})) = 0$
  - If $e \in \hat{P}$, $p_e(\hat{P}) = \max\limits_{P' \text{ in } G'} \Sigma_{j \neq e \text{ and } j \in P'}(-v_j) + \Sigma_{j \neq e} v_j(\hat{P})$



Shortest Path — Using VCG

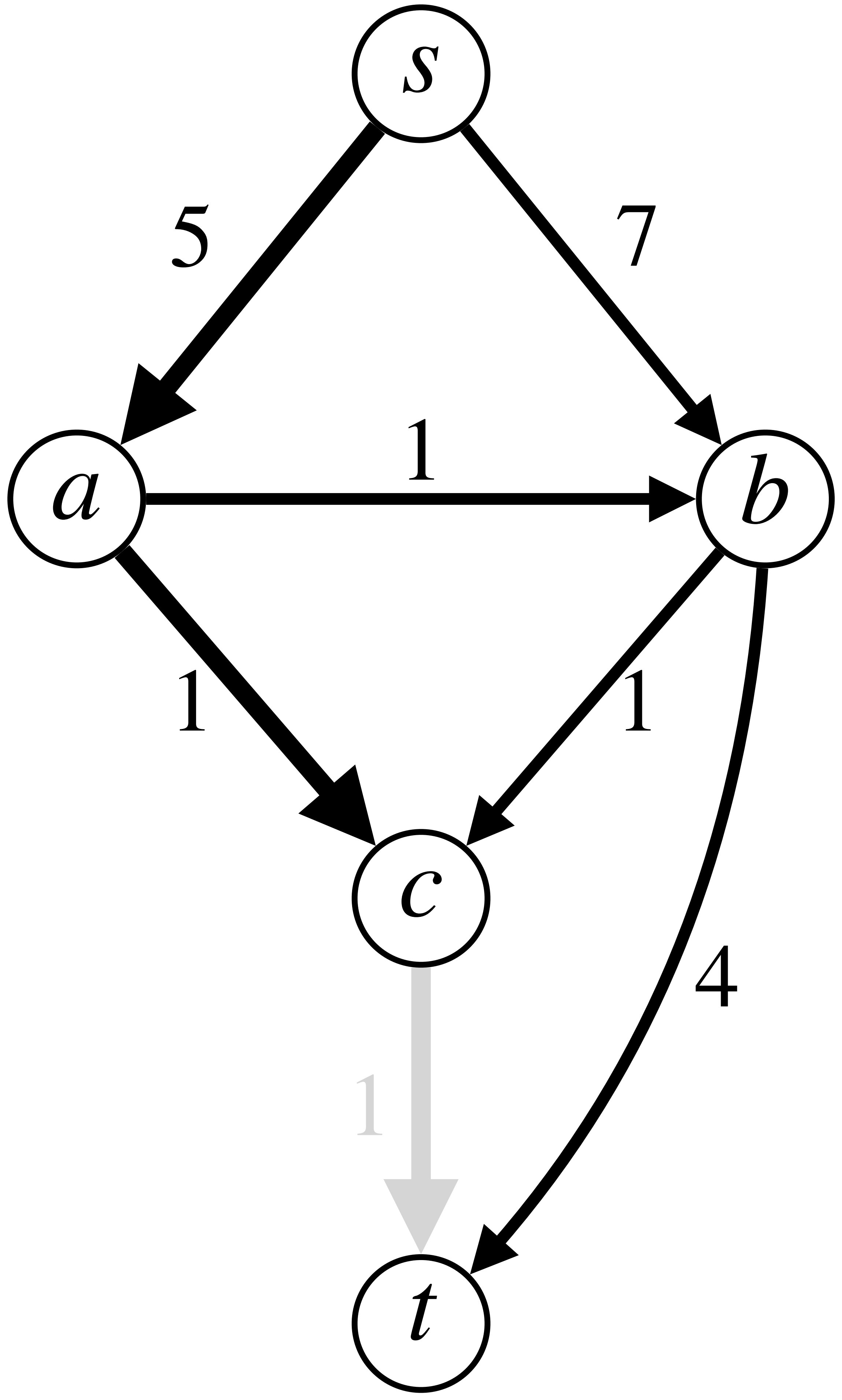
- Given a directed graph $G = (V, E)$, where each edge $e \in E$ is owned by a player. The player e has a (private) value v_e
- $p_e(\hat{P}) = \max_{\text{path } P} \Sigma_{j \neq e \text{ and } j \in P}(-v_j) - \Sigma_{j \neq e}(-v_j(\hat{P}))$
  - If $e \notin \hat{P}$, $p_e(\hat{P}) = \max_{P' \text{ in } G'} \Sigma_{j \neq e \text{ and } j \in P'}(-v_j) - \Sigma_{j \neq e}(-v_j(\hat{P})) = 0$
  - If $e \in \hat{P}$, $p_e(\hat{P}) = \max_{P' \text{ in } G'} \Sigma_{j \neq e \text{ and } j \in P'}(-v_j) + \Sigma_{j \neq e} v_j(\hat{P})$



$$p_{(a,c)} = \max_{P' \text{ in } G'} \Sigma_{e \in P'}(-v_e) - \Sigma_{(a,c) \neq e \in P}(-v_e)$$
$$= -8 - (-6) \;\; = -2$$

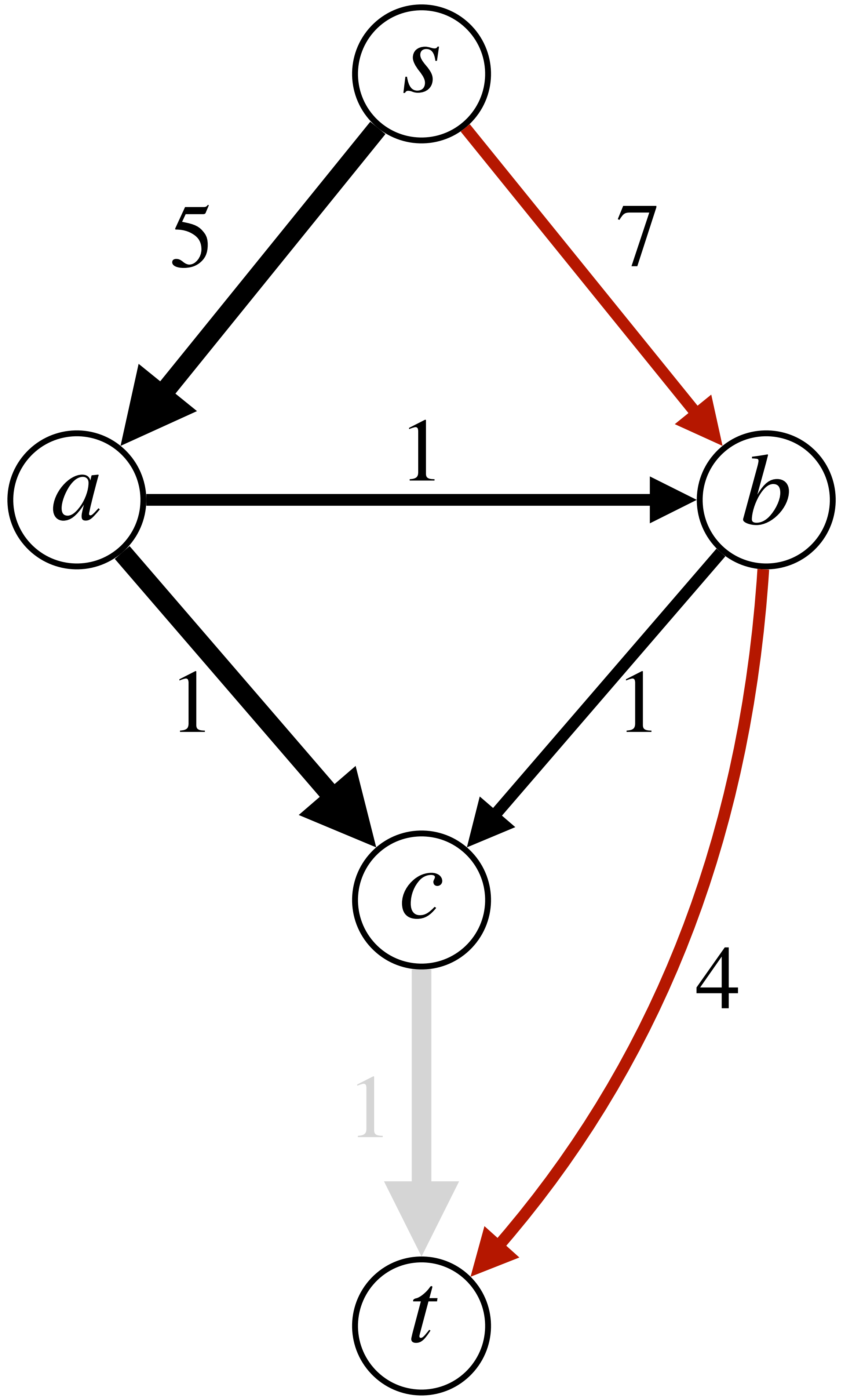
Shortest Path — Using VCG

- Given a directed graph $G = (V, E)$, where each edge $e \in E$ is owned by a player. The player e has a (private) value v_e
- $p_e(\hat{P}) = \max_{\text{path } P} \Sigma_{j \neq e \text{ and } j \in P}(-v_j) - \Sigma_{j \neq e}(-v_j(\hat{P}))$
  - If $e \notin \hat{P}$, $p_e(\hat{P}) = \max_{P' \text{ in } G'} \Sigma_{j \neq e \text{ and } j \in P'}(-v_j) - \Sigma_{j \neq e}(-v_j(\hat{P})) = 0$
  - If $e \in \hat{P}$, $p_e(\hat{P}) = \max_{P' \text{ in } G'} \Sigma_{j \neq e \text{ and } j \in P'}(-v_j) + \Sigma_{j \neq e} v_j(\hat{P})$



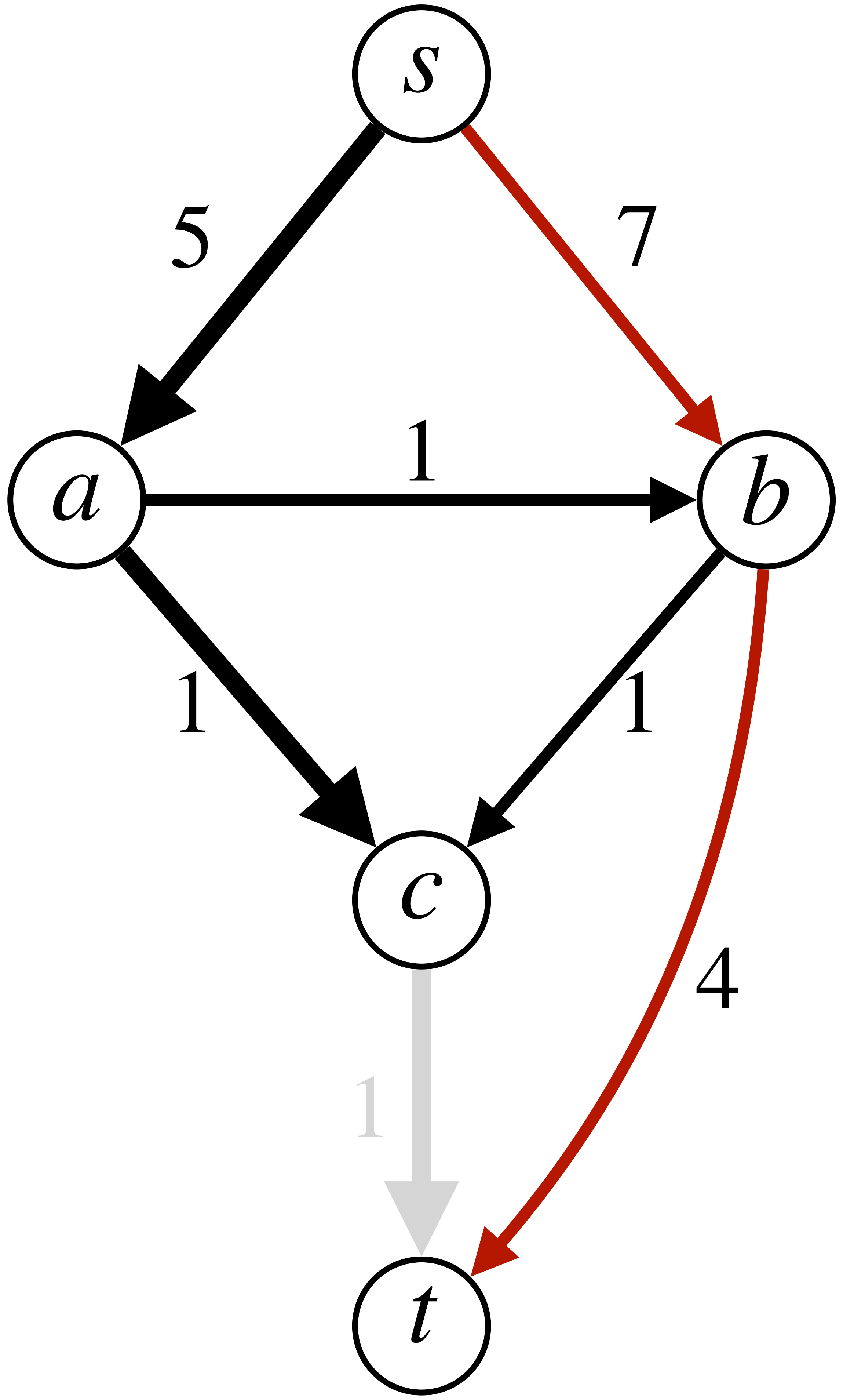
Shortest Path — Using VCG

- Given a directed graph $G = (V, E)$, where each edge $e \in E$ is owned by a player. The player e has a (private) value v_e
- $p_e(\hat{P}) = \max\limits_{\text{path } P} \Sigma_{j \neq e \text{ and } j \in P}(-v_j) - \Sigma_{j \neq e}(-v_j(\hat{P}))$
  - If $e \notin \hat{P}$, $p_e(\hat{P}) = \max\limits_{P' \text{ in } G'} \Sigma_{j \neq e \text{ and } j \in P'}(-v_j) - \Sigma_{j \neq e}(-v_j(\hat{P})) = 0$
  - If $e \in \hat{P}$, $p_e(\hat{P}) = \max\limits_{P' \text{ in } G'} \Sigma_{j \neq e \text{ and } j \in P'}(-v_j) + \Sigma_{j \neq e} v_j(\hat{P})$



Shortest Path — Using VCG

- Given a directed graph $G = (V, E)$, where each edge $e \in E$ is owned by a player. The player e has a (private) value v_e
- $p_e(\hat{P}) = \max_{\text{path } P} \Sigma_{j \neq e \text{ and } j \in P}(-v_j) - \Sigma_{j \neq e}(-v_j(\hat{P}))$
  - If $e \notin \hat{P}$, $p_e(\hat{P}) = \max_{P' \text{ in } G'} \Sigma_{j \neq e \text{ and } j \in P'}(-v_j) - \Sigma_{j \neq e}(-v_j(\hat{P})) = 0$
  - If $e \in \hat{P}$, $p_e(\hat{P}) = \max_{P' \text{ in } G'} \Sigma_{j \neq e \text{ and } j \in P'}(-v_j) + \Sigma_{j \neq e} v_j(\hat{P})$



$$p_{(c,t)} = \max_{P' \text{ in } G'} \Sigma_{e \in P'}(-v_e) - \Sigma_{(c,t) \neq e \in P}(-v_e)$$
$$= -11 - (-6) = -5$$

Trade

- Seller has an item that costs v_s, and a potential buyer values it at v_b
  - If $v_b > v_s$, there is a grade. Otherwise ($v_b \leq v_s$), there is no trade

- How to make a price for the buyer (to pay) and a price for the seller (to receive) so the buyers and sellers report their value truthfully?

Trade — Using VCG mechanism

- Seller has an item that costs v_s, and a potential buyer values it at v_b
  - If $v_b > v_s$, there is a grade. Otherwise ($v_b \leq v_s$), there is no trade

Clarke pivot rule: $p_i(\vec{s}) = \max\limits_{a \in A} \Sigma_{j \neq i} s_i(a) - \Sigma_{j \neq i} s_j(f(\vec{s}))$

- if $v_b > v_s$: $p_s(v_s, v_b) = \max\limits_{d \in \{0,1\}} v_b(d) - v_b = v_b - v_b = 0$

- if $v_b \leq v_s$: $p_s(v_s, v_b) = \max\limits_{d \in \{0,1\}} v_b(d) - 0 = v_b - 0 = v_b$

Trade — Using VCG mechanism

- Seller has an item that costs v_s, and a potential buyer values it at v_b
  - If $v_b > v_s$, there is a grade. Otherwise ($v_b \leq v_s$), there is no trade

Clarke pivot rule: $p_i(\vec{s}) = \max_{a \in A} \Sigma_{j \neq i} s_i(a) - \Sigma_{j \neq i} s_j(f(\vec{s}))$

- if $v_b > v_s$: $p_s(v_s, v_b) = \max_{d \in \{0,1\}} v_b(d) - v_b = v_b - v_b = 0$

$$p_b(v_s, v_b) = \max_{d \in \{0,1\}} - v_s(d) - (-v_s) = 0 + v_s = v_s$$

- if $v_b \leq v_s$: $p_s(v_s, v_b) = \max_{d \in \{0,1\}} v_b(d) - 0 = v_b - 0 = v_b$

$$p_b(v_s, v_b) = \max_{d \in \{0,1\}} - v_s(d) - 0 = 0 - 0 = 0$$