Lecture 26: Algorithmic Game Theory

Michael Dinitz

December 2, 2021
601.433/633 Introduction to Algorithms

Introduction

Algorithmic game theory: (some) intersections of algorithms and game theory (or economics more broadly)

Three subareas:

- Computation of equilibria
- Inefficiency of equilibria
- Algorithmic mechanism design

Today: very fast examples of first two

- See 601.436/636 for a whole class on this!

Two-Player Zero-Sum Games: Penalty Kicks

Penalty kicks in soccer:

- Two players: goalie and kicker
- Too fast to react: both players have to guess.

Model as *matrix game*: matrix $\mathbf{M}$, each entry of form $\mathbf{(a, b)}$

- Kicker picks row and goalie picks column (simultaneously)
- $\mathbf{(a, b)}$: kicker (row player) gets "utility" $\mathbf{a}$, goalie (column player) gets "utility" $\mathbf{b}$
- "Zero-sum": $\mathbf{a + b = 0}$ (so usually just write first value: row player's utility)

Goalie

kicker	Left	Right
Left	(0,0)	(1,-1)
Right	(1, -1)	(0,0)

What should each player do?

Minimax

Two-player zero-sum matrix game: $\mathbf{M} \in \mathbb{R}^{\mathbf{n} \times \mathbf{m}}$, row player tries to maximize, column player tries to minimize.

Natural approach: assume other player knows you well, do as best as possible.

- Row player: choose *distribution* over rows, so that no matter what column player does (even if they know distribution), still get utility
- Penalty kicks:
  - Probability $\mathbf{1/2}$ for each direction. Even if goalie knows, still get utility $\mathbf{1}$ with probability $\mathbf{1/2}$!
  - If we bias at all, then goalie who knows this is more likely to block us: get utility less than $\mathbf{1/2}$ in expectation

Minimax

Two-player zero-sum matrix game: $\mathbf{M} \in \mathbb{R}^{\mathbf{n} \times \mathbf{m}}$, row player tries to maximize, column player tries to minimize.

Natural approach: assume other player knows you well, do as best as possible.

- Row player: choose *distribution* over rows, so that no matter what column player does (even if they know distribution), still get utility
- Penalty kicks:
  - Probability $\mathbf{1/2}$ for each direction. Even if goalie knows, still get utility $\mathbf{1}$ with probability $\mathbf{1/2}$!
  - If we bias at all, then goalie who knows this is more likely to block us: get utility less than $\mathbf{1/2}$ in expectation
- Choose *minimax* strategy: probability distribution $\mathbf{p}$ over $[\mathbf{n}]$ to maximize

$$\mathbf{V_p} = \min_{\mathbf{j} \in [\mathbf{m}]} \sum_{\mathbf{i} \in [\mathbf{n}]} \mathbf{p_i M_{ij}}$$

Minimax

Two-player zero-sum matrix game: $\mathbf{M} \in \mathbb{R}^{\mathbf{n} \times \mathbf{m}}$, row player tries to maximize, column player tries to minimize.

Natural approach: assume other player knows you well, do as best as possible.

- Row player: choose *distribution* over rows, so that no matter what column player does (even if they know distribution), still get utility
- Penalty kicks:
  - Probability $\mathbf{1/2}$ for each direction. Even if goalie knows, still get utility $\mathbf{1}$ with probability $\mathbf{1/2}$!
  - If we bias at all, then goalie who knows this is more likely to block us: get utility less than $\mathbf{1/2}$ in expectation
- Choose *minimax* strategy: probability distribution $\mathbf{p}$ over $[\mathbf{n}]$ to maximize

$$\mathbf{V_p} = \min_{\mathbf{j} \in [\mathbf{m}]} \sum_{\mathbf{i} \in [\mathbf{n}]} \mathbf{p_i M_{ij}}$$

$$\mathbf{V} = \max_{\substack{\text{probability distributions } \mathbf{p} \\ \text{over } [\mathbf{n}]}} \mathbf{V_p} = \max_{\substack{\text{probability distributions } \mathbf{p} \\ \text{over } [\mathbf{n}]}} \min_{\mathbf{j} \in [\mathbf{m}]} \sum_{\mathbf{i} \in [\mathbf{n}]} \mathbf{p_i M_{ij}}$$

Computing Minimax

How to compute minimax strategy?

Computing Minimax

How to compute minimax strategy?

$$\max \quad V$$

subject to

$$\sum_{i=1}^{n} p_i = 1$$

$$\sum_{i=1}^{n} p_i M_{ij} \geq V \qquad \forall j \in [m]$$

$0 \leq p_i$ ~~≤ 1~~ $\qquad \forall i \in [n]$

More Penalty Kicks

	Left	Right
Left	(0,0)	(1,-1)
Right	(1, -1)	(0,0)

Kicker (row) minimax:

- **1/2** on each direction
- Guarantees at least **1/2** utility in expectation

Goalie (column) minimax:

- **1/2** on each direction
- Guarantees at least **–1/2** utility in expectation (at most **1/2** loss)

More Penalty Kicks

	Left	Right
Left	(0,0)	(1,-1)
Right	(1, -1)	(0,0)

		1	0
		Left	Right
1	Left	$(\frac{1}{2}, -\frac{1}{2})$	(1,-1)
0	Right	(1, -1)	(0,0)

Kicker (row) minimax:

- **1/2** on each direction
- Guarantees at least **1/2** utility in expectation

Goalie (column) minimax:

- **1/2** on each direction
- Guarantees at least **−1/2** utility in expectation (at most **1/2** loss)

More Penalty Kicks

	Left	Right
Left	(0,0)	(1,-1)
Right	(1, -1)	(0,0)

Kicker (row) minimax:

- **1/2** on each direction
- Guarantees at least **1/2** utility in expectation

Goalie (column) minimax:

- **1/2** on each direction
- Guarantees at least **−1/2** utility in expectation (at most **1/2** loss)

	Left	Right
Left	$(\frac{1}{2}, -\frac{1}{2})$	(1,-1)
Right	(1, -1)	(0,0)

Kicker (row) minimax:

More Penalty Kicks

	Left	Right
Left	(0,0)	(1,-1)
Right	(1, -1)	(0,0)

Kicker (row) minimax:

- **1/2** on each direction
- Guarantees at least **1/2** utility in expectation

Goalie (column) minimax:

- **1/2** on each direction
- Guarantees at least **−1/2** utility in expectation (at most **1/2** loss)

		Left	Right
2/3	Left	$(\frac{1}{2}, -\frac{1}{2})$	(1,-1)
1/3	Right	(1, -1)	(0,0)

Kicker (row) minimax:

- $\mathbf{(2/3, 1/3)}$
- Guarantees at least **2/3** utility in expectation

More Penalty Kicks

	Left	Right
Left	(0,0)	(1,-1)
Right	(1, -1)	(0,0)

Kicker (row) minimax:

- **1/2** on each direction
- Guarantees at least **1/2** utility in expectation

Goalie (column) minimax:

- **1/2** on each direction
- Guarantees at least **−1/2** utility in expectation (at most **1/2** loss)

	Left	Right
Left	$(\frac{1}{2}, -\frac{1}{2})$	(1,-1)
Right	(1, -1)	(0,0)

Kicker (row) minimax:

- $\mathbf{(2/3, 1/3)}$
- Guarantees at least **2/3** utility in expectation

Goalie (column) minimax:

More Penalty Kicks

	Left	Right
Left	(0,0)	(1,-1)
Right	(1, -1)	(0,0)

	Left	Right
Left	$(\frac{1}{2}, -\frac{1}{2})$	(1,-1)
Right	(1, -1)	(0,0)

Kicker (row) minimax:

- **1/2** on each direction
- Guarantees at least **1/2** utility in expectation

Goalie (column) minimax:

- **1/2** on each direction
- Guarantees at least **−1/2** utility in expectation (at most **1/2** loss)

Kicker (row) minimax:

- $\mathbf{(2/3, 1/3)}$
- Guarantees at least **2/3** utility in expectation

Goalie (column) minimax:

- $\mathbf{(2/3, 1/3)}$
- Guarantees at least **−2/3** utility in expectation (at most **2/3** loss)

Minimax Theorem

Theorem (Minimax Theorem (von Neumann))

Every $\mathbf{2}$*-player zero-sum game has a unique value* $\mathbf{V}$ *such that the minimax strategy for the row player guarantees expected gain of at least* $\mathbf{V}$*, and the minimax strategy for the column player also guarantees expected loss of at most* $\mathbf{V}$*.*

Minimax Theorem

Theorem (Minimax Theorem (von Neumann))

Every $\mathbf{2}$-player zero-sum game has a unique value $\mathbf{V}$ such that the minimax strategy for the row player guarantees expected gain of at least $\mathbf{V}$, and the minimax strategy for the column player also guarantees expected loss of at most $\mathbf{V}$.

Proof outside the scope of the course, but not hard.

- Easiest proof: LP duality

General Games and Nash Equilibria

General (one-shot) games: allow more than $\mathbf{2}$ players, utilities don't have to add to $\mathbf{0}$.

- No longer a unique value!

Replace minimax strategies with *Nash equilibria*

- (Randomized) strategy for every player so that no one has incentive to deviate (knowing all other strategies)

Example

Example: two people walking down the sidewalk

	Left	Right
Left	(1,1)	(-1,-1)
Right	(-1, -1)	(1,1)

Example

Example: two people walking down the sidewalk

	Left	Right
Left	(1,1)	(-1,-1)
Right	(-1, -1)	(1,1)

Nash equilibria:

- Both left
- Both right

Example

Example: two people walking down the sidewalk

	Left	Right
Left	(1,1)	(-1,-1)
Right	(-1, -1)	(1,1)

Nash equilibria:

- Both left
- Both right
- Both $\mathbf{(1/2, 1/2)}$

Example

Example: two people walking down the sidewalk

	Left	Right
Left	(1,1)	(-1,-1)
Right	(-1, -1)	(1,1)

Nash equilibria:

- Both left
- Both right
- Both $\mathbf{(1/2, 1/2)}$
  - Row player: expected utility is $\mathbf{0}$

Example

Example: two people walking down the sidewalk

	Left ($\frac{1}{2}$)	Right ($\frac{1}{2}$)
p_L Left	(1,1)	(-1,-1)
p_R Right	(-1, -1)	(1,1)

Nash equilibria:

- Both left
- Both right
- Both $(1/2, 1/2)$
  - Row player: expected utility is 0
  - Suppose deviated to (p_L, p_R) (column player stays at $(1/2, 1/2)$):

$$\frac{1}{2}(1 \cdot p_L - 1 \cdot p_R) + \frac{1}{2}(-1 \cdot p_L + 1 \cdot p_R) = 0$$

Nash Equilibria

Theorem (Nash)

Every game has at least one Nash equilibrium.

Nash Equilibria

Theorem (Nash)

Every game has at least one Nash equilibrium.

The most important concept in game theory!

- Other definitions of equilibria either for special cases (minimax), or generalize Nash

Nash Equilibria

Theorem (Nash)

Every game has at least one Nash equilibrium.

The most important concept in game theory!

- Other definitions of equilibria either for special cases (minimax), or generalize Nash

Nash's proof: through Brouwer's fixed-point theorem

- "Every continuous function from a convex compact subset **K** of a Euclidean space to **K** itself has a fixed point."
- Famous and fundamental result in topology
- Non-constructive!

Nash Equilibria

Theorem (Nash)

Every game has at least one Nash equilibrium.

The most important concept in game theory!

- Other definitions of equilibria either for special cases (minimax), or generalize Nash

Nash's proof: through Brouwer's fixed-point theorem

- "Every continuous function from a convex compact subset **K** of a Euclidean space to **K** itself has a fixed point."
- Famous and fundamental result in topology
- Non-constructive!

Question: Can we *compute* Nash equilibria?

Computing Nash Equilibria

Somewhat tricky to formalize

Attempt 1: Is it **NP**-hard to compute a Nash equilibrium?

Computing Nash Equilibria

Somewhat tricky to formalize

Attempt 1: Is it **NP**-hard to compute a Nash equilibrium?

- Decision problem: YES if game has a Nash, NO otherwise. Always YES!
- Need some other complexity class that can deal with answer always being YES.

Computing Nash Equilibria

Somewhat tricky to formalize

Attempt 1: Is it **NP**-hard to compute a Nash equilibrium?

- Decision problem: YES if game has a Nash, NO otherwise. Always YES!
- Need some other complexity class that can deal with answer always being YES.

New complexity class: **PPAD** (Polynomial Parity Argument (Directed))

- Answer always YES, but (we think) it is hard to find solution.

Computing Nash Equilibria

Somewhat tricky to formalize

Attempt 1: Is it **NP**-hard to compute a Nash equilibrium?

- Decision problem: YES if game has a Nash, NO otherwise. Always YES!
- Need some other complexity class that can deal with answer always being YES.

New complexity class: **PPAD** (Polynomial Parity Argument (Directed))

- Answer always YES, but (we think) it is hard to find solution.

Theorem (Daskalakis, Goldberg, Papadimitriou)

Computing a Nash equilibrium is PPAD-complete.

Computing Nash Equilibria

Somewhat tricky to formalize

Attempt 1: Is it **NP**-hard to compute a Nash equilibrium?

- Decision problem: YES if game has a Nash, NO otherwise. Always YES!
- Need some other complexity class that can deal with answer always being YES.

New complexity class: **PPAD** (Polynomial Parity Argument (Directed))

- Answer always YES, but (we think) it is hard to find solution.

Theorem (Daskalakis, Goldberg, Papadimitriou)

Computing a Nash equilibrium is PPAD-complete.

Issue for game theory in economics! If hard to compute Nash, why do we expect markets / games to end up at Nash?

Computing Nash Equilibria

Somewhat tricky to formalize

Attempt 1: Is it **NP**-hard to compute a Nash equilibrium?

- Decision problem: YES if game has a Nash, NO otherwise. Always YES!
- Need some other complexity class that can deal with answer always being YES.

New complexity class: **PPAD** (Polynomial Parity Argument (Directed))

- Answer always YES, but (we think) it is hard to find solution.

Theorem (Daskalakis, Goldberg, Papadimitriou)

Computing a Nash equilibrium is PPAD-complete.

Issue for game theory in economics! If hard to compute Nash, why do we expect markets / games to end up at Nash?

- Other equilibria (e.g., *coarse correlated equilibria*) can be computed efficiently:

Computing Nash Equilibria

Somewhat tricky to formalize

Attempt 1: Is it **NP**-hard to compute a Nash equilibrium?

- Decision problem: YES if game has a Nash, NO otherwise. Always YES!
- Need some other complexity class that can deal with answer always being YES.

New complexity class: **PPAD** (Polynomial Parity Argument (Directed))

- Answer always YES, but (we think) it is hard to find solution.

Theorem (Daskalakis, Goldberg, Papadimitriou)

Computing a Nash equilibrium is PPAD-complete.

Issue for game theory in economics! If hard to compute Nash, why do we expect markets / games to end up at Nash?

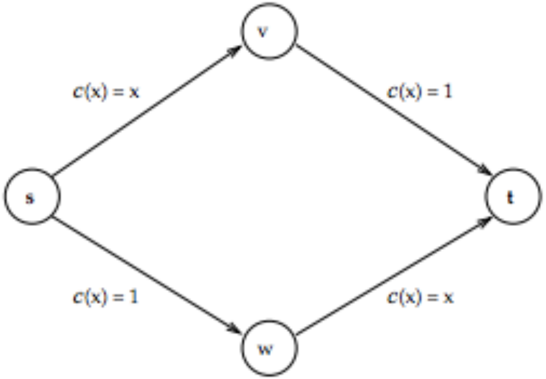
- Other equilibria (e.g., *coarse correlated equilibria*) can be computed efficiently: online learning!

Braess's Paradox

Nash equilibria can behave strangely. Example: *Braess's Paradox* in *routing games*.

Braess's Paradox

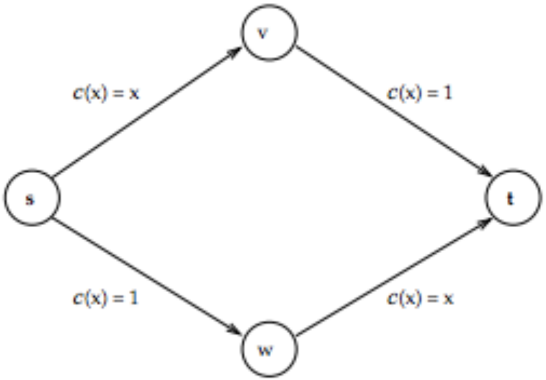
Nash equilibria can behave strangely. Example: *Braess's Paradox* in *routing games*.



- Huge number of players ($\mathbf{1}/\epsilon$) trying to get from **s** to **t**, each controls ϵ traffic
- Actions: path from **s** to **t**
- Cost of edge: $\mathbf{c(x)}$ where $\mathbf{x}$ is fraction of flow through edge
- Cost of a path (action): sum of costs of edges

Braess's Paradox

Nash equilibria can behave strangely. Example: *Braess's Paradox* in *routing games*.

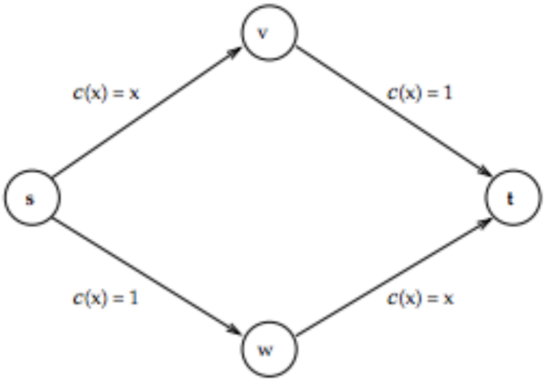


- Huge number of players ($\mathbf{1}/\epsilon$) trying to get from **s** to **t**, each controls ϵ traffic
- Actions: path from **s** to **t**
- Cost of edge: $\mathbf{c(x)}$ where $\mathbf{x}$ is fraction of flow through edge
- Cost of a path (action): sum of costs of edges

Nash equilibria:

Braess's Paradox

Nash equilibria can behave strangely. Example: *Braess's Paradox* in *routing games*.

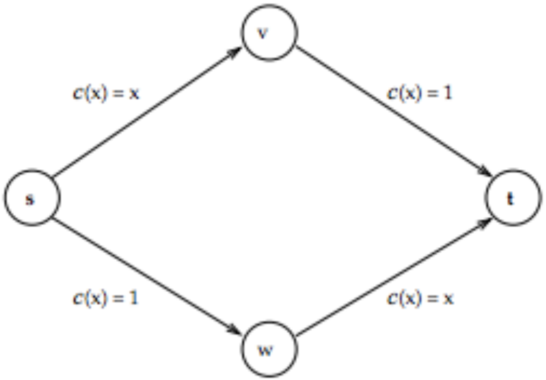


- Huge number of players ($\mathbf{1/\epsilon}$) trying to get from **s** to **t**, each controls ϵ traffic
- Actions: path from **s** to **t**
- Cost of edge: $\mathbf{c(x)}$ where **x** is fraction of flow through edge
- Cost of a path (action): sum of costs of edges

Nash equilibria: $\mathbf{1/2}$ use top path, $\mathbf{1/2}$ use bottom

Braess's Paradox

Nash equilibria can behave strangely. Example: *Braess's Paradox* in *routing games*.



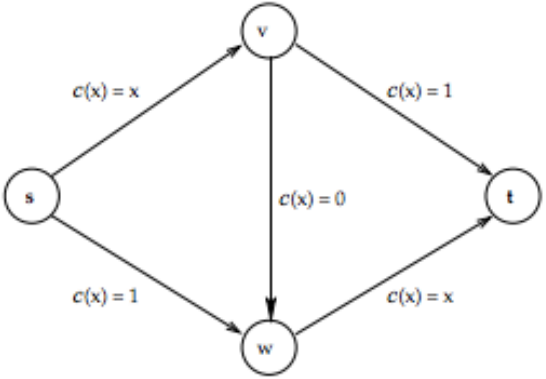
- Huge number of players ($\mathbf{1/\epsilon}$) trying to get from $\mathbf{s}$ to $\mathbf{t}$, each controls ϵ traffic
- Actions: path from $\mathbf{s}$ to $\mathbf{t}$
- Cost of edge: $\mathbf{c(x)}$ where $\mathbf{x}$ is fraction of flow through edge
- Cost of a path (action): sum of costs of edges

Nash equilibria: $\mathbf{1/2}$ use top path, $\mathbf{1/2}$ use bottom

- Each player pays $\mathbf{3/2}$. If any player deviates, pays more than $\mathbf{3/2}$

Braess's Paradox

Nash equilibria can behave strangely. Example: *Braess's Paradox* in *routing games*.



- Huge number of players ($1/\epsilon$) trying to get from $\mathbf{s}$ to $\mathbf{t}$, each controls ϵ traffic
- Actions: path from $\mathbf{s}$ to $\mathbf{t}$
- Cost of edge: $\mathbf{c(x)}$ where $\mathbf{x}$ is fraction of flow through edge
- Cost of a path (action): sum of costs of edges

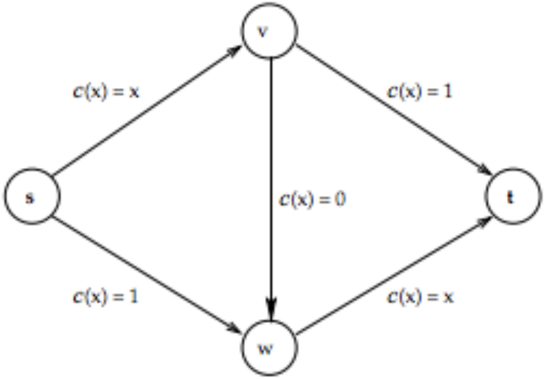
Nash equilibria: $\mathbf{1/2}$ use top path, $\mathbf{1/2}$ use bottom

- Each player pays $\mathbf{3/2}$. If any player deviates, pays more than $\mathbf{3/2}$

Nash equilibria:

Braess's Paradox

Nash equilibria can behave strangely. Example: *Braess's Paradox* in *routing games*.



- Huge number of players ($1/\epsilon$) trying to get from $\mathbf{s}$ to $\mathbf{t}$, each controls ϵ traffic
- Actions: path from $\mathbf{s}$ to $\mathbf{t}$
- Cost of edge: $\mathbf{c(x)}$ where $\mathbf{x}$ is fraction of flow through edge
- Cost of a path (action): sum of costs of edges

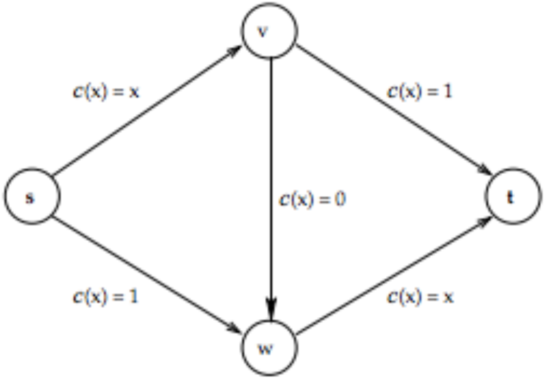
Nash equilibria: $\mathbf{1/2}$ use top path, $\mathbf{1/2}$ use bottom

- Each player pays $\mathbf{3/2}$. If any player deviates, pays more than $\mathbf{3/2}$

Nash equilibria: Everyone uses diagonal path, pays $\mathbf{2}$

Braess's Paradox

Nash equilibria can behave strangely. Example: *Braess's Paradox* in *routing games*.



- Huge number of players ($1/\epsilon$) trying to get from **s** to **t**, each controls ϵ traffic
- Actions: path from **s** to **t**
- Cost of edge: $\mathbf{c(x)}$ where $\mathbf{x}$ is fraction of flow through edge
- Cost of a path (action): sum of costs of edges

Nash equilibria: $\mathbf{1/2}$ use top path, $\mathbf{1/2}$ use bottom

- Each player pays **3/2**. If any player deviates, pays more than **3/2**

Nash equilibria: Everyone uses diagonal path, pays **2**

- So *improved* edges leads to *worse* performance!

Price of Anarchy

Braess's paradox $\implies$ sometime Nash are not "optimal"

- Approximation and online algorithms: compare algorithmic solutions to OPT
- Natural from a TCS point of view: compare Nash to OPT!

Price of Anarchy

Braess's paradox $\implies$ sometime Nash are not "optimal"

- Approximation and online algorithms: compare algorithmic solutions to OPT
- Natural from a TCS point of view: compare Nash to OPT!

Let $\mathbf{OPT}$ denote "cost" of best solution, for each Nash $\mathbf{s}$ let $\mathbf{W(s)}$ denote "cost" of $\mathbf{s}$, let $\mathcal{S}$ denote all Nash.

Definition

The *price of anarchy* of a minimization game is $\max_{\mathbf{s} \in \mathcal{S}} \mathbf{W(s)}/\mathbf{OPT}$.

Price of Anarchy

Braess's paradox $\implies$ sometime Nash are not "optimal"

- Approximation and online algorithms: compare algorithmic solutions to OPT
- Natural from a TCS point of view: compare Nash to OPT!

Let $\mathbf{OPT}$ denote "cost" of best solution, for each Nash $\mathbf{s}$ let $\mathbf{W(s)}$ denote "cost" of $\mathbf{s}$, let $\mathcal{S}$ denote all Nash.

Definition

The *price of anarchy* of a minimization game is $\max_{s \in \mathcal{S}} \mathbf{W(s)/OPT}$.

Routing game example: $\mathbf{OPT} = \mathbf{3/2}$, only one Nash, has cost $\mathbf{2}$.

$\implies$ Price of Anarchy $= \mathbf{2/(3/2)} = \mathbf{4/3}$

Price of Anarchy

Braess's paradox $\implies$ sometime Nash are not "optimal"

- Approximation and online algorithms: compare algorithmic solutions to OPT
- Natural from a TCS point of view: compare Nash to OPT!

Let **OPT** denote "cost" of best solution, for each Nash **s** let **W(s)** denote "cost" of **s**, let $\mathcal{S}$ denote all Nash.

Definition

The *price of anarchy* of a minimization game is $\max_{s \in \mathcal{S}} W(s)/OPT$.

Routing game example: **OPT** = **3/2**, only one Nash, has cost **2**.

$\implies$ Price of Anarchy = $2/(3/2) = 4/3$

Theorem (Roughgarden)

The price of anarchy in any routing game with linear edge costs is at most **4/3**

Conclusion

Algorithmic Game Theory:

- Can we compute equilibria?
- How good are equilibria compare to optimal?
- (Mechanism Design) Can we design games with nice properties?

Conclusion

Algorithmic Game Theory:

- Can we compute equilibria?
- How good are equilibria compare to optimal?
- (Mechanism Design) Can we design games with nice properties?

Hope you enjoyed the class, and good luck on the final!