In Congestion Games, Taxes Achieve Optimal Approximation

Dario Paccagnan, Martin Gairing
**Problem:** minimum social cost in atomic congestion games
Problem: minimum social cost in atomic congestion games

Main result I: tight NP-hardness of approximation
Problem: minimum social cost in atomic congestion games

Main result I: tight NP-hardness of approximation
Problem: minimum social cost in atomic congestion games

Main result I: tight NP-hardness of approximation

Main result II: taxes achieve matching approximation
Problem: minimum social cost in atomic congestion games

Main result I: tight NP-hardness of approximation

Main result II: taxes achieve matching approximation
**Problem:** minimum social cost in atomic congestion games

**Main result I:** tight NP-hardness of approximation

**Main result II:** taxes achieve matching approximation
Problem: minimum social cost in atomic congestion games

Main result I: tight NP-hardness of approximation

Main result II: taxes achieve matching approximation
   \[ \Rightarrow \text{first poly algo optimal approx} \]
Problem: minimum social cost in atomic congestion games

Main result I: tight NP-hardness of approximation

Main result II: taxes achieve matching approximation
\[ \implies \text{first poly algo optimal approx} \]

Judiciously designed taxes achieve optimal approx, and no other tractable intervention can improve
Atomic congestion games
Atomic congestion games

- Set of resources $\mathcal{R}$
Atomic congestion games

- Set of resources $\mathcal{R}$
- Resource costs $\ell_r(\cdot)$
Atomic congestion games

- Set of resources $\mathcal{R}$
- Resource costs $\ell_r(\cdot)$
- Set of players $\{1, \ldots, N\}$
Atomic congestion games

- Set of resources $\mathcal{R}$
- Resource costs $\ell_r(\cdot)$
- Set of players $\{1, \ldots, N\}$
- Player $i$ feasible set $\mathcal{A}_i \subseteq 2^\mathcal{R}$
Atomic congestion games

- Set of resources $\mathcal{R}$
- Resource costs $\ell_r(\cdot)$
- Set of players $\{1, \ldots, N\}$
- Player $i$ feasible set $\mathcal{A}_i \subseteq 2^\mathcal{R}$
Atomic congestion games

- Set of resources $\mathcal{R}$
- Resource costs $\ell_r(\cdot)$

- Set of players $\{1, \ldots, N\}$
- Player $i$ feasible set $\mathcal{A}_i \subseteq 2^\mathcal{R}$
- Player $i$ cost $C_i(a) = \sum_{r \in a_i} \ell_r(|a|_r)$

Applications:
- routing, sensor allocation, scheduling, minimum power, . . .
Atomic congestion games

- Set of resources $\mathcal{R}$
- Resource costs $\ell_r(\cdot)$
- Set of players $\{1, \ldots, N\}$
- Player $i$ feasible set $A_i \subseteq 2^\mathcal{R}$
- Player $i$ cost $C_i(a) = \sum_{r \in a_i} \ell_r(|a|_r)$

System cost: $SC(a) = \sum_i C_i(a)$
Atomic congestion games

- Set of resources $\mathcal{R}$
- Resource costs $\ell_r(\cdot)$
- Set of players $\{1, \ldots, N\}$
- Player $i$ feasible set $A_i \subseteq 2^\mathcal{R}$
- Player $i$ cost $C_i(a) = \sum_{r \in a_i} \ell_r(|a|r)$

System cost: $SC(a) = \sum_i C_i(a)$

$SC = \ell_1(1) + \ell_2(1) + 2\ell_3(2) + \ell_4(1)$
Atomic congestion games

- Set of resources $\mathcal{R}$
- Resource costs $\ell_r(\cdot)$
- Set of players $\{1, \ldots, N\}$
- Player $i$ feasible set $\mathcal{A}_i \subseteq 2^\mathcal{R}$
- Player $i$ cost $C_i(a) = \sum_{r \in \mathcal{A}_i} \ell_r(|a|_r)$

System cost: $SC(a) = \sum_i C_i(a)$

Applications: routing, sensor allocation, scheduling, minimum power, ...
Problem: minimum social cost in atomic congestion games

Main result I: tight NP-hardness of approximation

Main result II: taxes achieve matching approximation \[\Rightarrow\] first poly algo optimal approx
Hardness of approximation – related work

\[ \text{MinSC} : \min_{a \in A} SC(a) \]
Hardness of approximation – related work

\[ \text{MinSC} : \min_{a \in A} SC(a) \]

* MinSC is NP-hard [Meyers/Schulz, Networks’12]

* MinSC is NP-hard if latencies are linear [Castiglioni/Celli/Marchesi/Gatti, ArXiv’20]
Hardness of approximation – related work

\[ \text{MinSC} : \min_{a \in A} SC(a) \]

* \text{MinSC} is NP-hard [Meyers/Schulz, Networks’12]

* \text{MinSC} is NP-hard if latencies are linear [Castiglioni/Celli/Marchesi/Gatti, ArXiv’20]

* If latencies are polynomial of degree \( \leq d \), then \text{MinSC} is NP-hard to approx within a factor \((\beta d)^{d/2}\), for some \( \beta > 0 \) [Roughgarden, FOCS’12]
Hardness of approximation – related work

\[
\text{MinSC} : \min_{a \in A} SC(a)
\]

* \text{MinSC} is NP-hard \cite{Meyers/Schulz, Networks’12}

* \text{MinSC} is NP-hard if latencies are linear \cite{Castiglioni/Celli/Marchesi/Gatti, ArXiv’20}

* If latencies are polynomial of degree \( \leq d \), then \text{MinSC} is NP-hard to approx within a factor \((\beta d)^{\frac{d}{2}}\), for some \( \beta > 0 \) \cite{Roughgarden, FOCS’12}

**Take-away:** so far no tight computational lower bound
Hardness of approximation – main result

Theorem:
In congestion games with resource costs identical to $b(\cdot)$,
MinSC is NP-hard to approximate within any factor smaller than
$\rho_b = \sup_{x \in \mathbb{N}} \mathcal{E}_{\mathcal{P}} \sim \mathcal{P}_{\mathcal{B}}(x) [\mathcal{P}b(\mathcal{P})] x b(\cdot)$

Extension to resource costs produced by non-negative combinations of functions $b_1, \ldots, b_m$ obtained replacing $\rho_b$ with $\max_j \rho_{b_j}$

Corollary:
In polynomial congestion games of max degree $d_{\max}$

For example $d = 1$ corresponds to $B(d+1) = 2$
$d = 2$ corresponds to $B(d+1) = 5$

...
Hardness of approximation – main result

**Theorem:** In congestion games with resource costs identical to $b(\cdot)$
Hardness of approximation – main result

**Theorem:** In congestion games with resource costs identical to $b(\cdot)$, MinSC is NP-hard to approximate within any factor smaller than

$$\rho_b = \sup_{x \in \mathbb{N}} \frac{\mathbb{E}_{P \sim Poi(x)}[Pb(P)]}{xb(x)}$$
**Theorem:** In congestion games with resource costs identical to \( b(\cdot) \), MinSC is NP-hard to approximate within any factor smaller than

\[
\rho_b = \sup_{x \in \mathbb{N}} \frac{\mathbb{E}_{P \sim \text{Poi}(x)}[Pb(P)]}{xb(x)}
\]

**Extension** to resource costs produced by non-negative combinations of functions \( b_1, \ldots, b_m \) obtained replacing \( \rho_b \) with \( \max_j \rho_{b_j} \)
Hardness of approximation – main result

**Theorem:** In congestion games with resource costs identical to \( b(\cdot) \), \( \text{MinSC} \) is NP-hard to approximate within any factor smaller than

\[
\rho_b = \sup_{x \in \mathbb{N}} \frac{\mathbb{E}_{P \sim \text{Poi}(x)}[Pb(P)]}{xb(x)}
\]

**Extension** to resource costs produced by non-negative combinations of functions \( b_1, \ldots, b_m \) obtained replacing \( \rho_b \) with \( \max_j \rho_{b_j} \)

**Corollary:** In polynomial congestion games of max degree \( d \)

\[
\max_j \rho_{b_j} = (d + 1)\text{'st Bell number}
\]
Hardness of approximation – main result

**Theorem:** In congestion games with resource costs identical to $b(\cdot)$, MinSC is NP-hard to approximate within any factor smaller than

$$
\rho_b = \sup_{x \in \mathbb{N}} \frac{\mathbb{E}_{P \sim \text{Poi}(x)}[Pb(P)]}{xb(x)}
$$

**Extension** to resource costs produced by non-negative combinations of functions $b_1, \ldots, b_m$ obtained replacing $\rho_b$ with $\max_j \rho_{b_j}$

**Corollary:** In polynomial congestion games of max degree $d$

$$
\max_j \rho_{b_j} = (d + 1)'st \text{ Bell number}
$$

For example $d = 1$ corresponds to $B(d + 1) = 2$
Hardness of approximation – main result

**Theorem:** In congestion games with resource costs identical to $b(\cdot)$, MinSC is NP-hard to approximate within any factor smaller than

$$\rho_b = \sup_{x \in \mathbb{N}} \frac{E_{P \sim \text{Poi}(x)}[Pb(P)]}{xb(x)}$$

*Extension* to resource costs produced by non-negative combinations of functions $b_1, \ldots, b_m$ obtained replacing $\rho_b$ with $\max_j \rho_{b_j}$

**Corollary:** In polynomial congestion games of max degree $d$

$$\max_j \rho_{b_j} = (d + 1)'st \ Bell \ number$$

For example $d = 1$ corresponds to $\mathcal{B}(d + 1) = 2$

$d = 2$ corresponds to $\mathcal{B}(d + 1) = 5$

\vdots
Proof Ideas
Proof Ideas

Reduction from **Gap-label-cover** to CG using **partitioning system**
Proof Ideas

Reduction from **Gap-label-cover** to CG using **partitioning system**

**Gap-label-cover** [Feige JACM’98; Dudycz/Manurangsi/Marcinkowski/Sornat IJCAI'20]

- bi-partite graph

![Bi-partite Graph](image-url)
Proof Ideas

Reduction from **Gap-label-cover** to CG using **partitioning system**

**Gap-label-cover** [Feige JACM’98; Dudycz/Manurangsi/Marcinkowski/Sornat IJCAI’20]

- bi-partite graph

- palette of colors
Proof Ideas

Reduction from \textbf{Gap-label-cover} to \textit{CG} using \textit{partitioning system}

\textbf{Gap-label-cover} [Feige JACM’98; Dudycz/Manurangsi/Marcinkowski/Sornat IJCAI’20]

- bi-partite graph
- palette of colors
- set of constraints
Proof Ideas

Reduction from **Gap-label-cover** to CG using **partitioning system**

**Gap-label-cover** [Feige JACM’98; Dudycz/Manurangsi/Marcinkowski/Sornat IJCAI’20]

- bi-partite graph
- palette of colors
- set of constraints
Proof Ideas

Reduction from **Gap-label-cover** to CG using **partitioning system**

**Gap-label-cover** [Feige JACM’98; Dudyciz/Manurangsi/Marcinkowski/Sornat IJCAI’20]

- bi-partite graph

- palette of colors

- set of constraints

**Partitioning system** generalizes [Feige, JACM’98], used in [Barman/Fawzi/Fermé, STACS'21]

- resources $\mathcal{R}$, cost $b(\cdot)$
Proof Ideas

Reduction from **Gap-label-cover** to CG using **partitioning system**

**Gap-label-cover** [Feige JACM’98; Dudycz/Manurangsi/Marcinkowski/Sornat IJCAI’20]
- bi-partite graph
- palette of colors
- set of constraints

**Partitioning system** generalizes [Feige, JACM’98], used in [Barman/Fawzi/Fermé, STACS’21]
- resources \( \mathcal{R} \), cost \( b(\cdot) \)
- subsets \( P_{i,j} \subseteq \mathcal{R} \)
Proof Ideas

Reduction from **Gap-label-cover** to CG using **partitioning system**

**Gap-label-cover** [Feige JACM’98; Dudycz/Manurangsi/Marcinkowski/Sornat IJCAI’20]

- bi-partite graph
- palette of colors
- set of constraints

**Partitioning system** generalizes [Feige, JACM'98], used in [Barman/Fawzi/Fermé, STACS'21]

- resources $\mathcal{R}$, cost $b(\cdot)$
- subsets $P_{i,j} \subseteq \mathcal{R}$
- $SC(\text{row})$
Proof Ideas

Reduction from **Gap-label-cover** to CG using **partitioning system**

**Gap-label-cover** [Feige JACM’98; Dudycz/Manurangsi/Marcinkowski/Sornat IJCAI’20]

- bi-partite graph
- palette of colors
- set of constraints

**Partitioning system** generalizes [Feige, JACM’98], used in [Barman/Fawzi/Fermé, STACS’21]

- resources $\mathcal{R}$, cost $b(\cdot)$
- subsets $P_{i,j} \subseteq \mathcal{R}$
- $SC(\text{row})$, $SC(\text{scr})$
Proof Ideas

Reduction from **Gap-label-cover** to CG using **partitioning system**

**Gap-label-cover** [Feige JACM’98; Dudycz/Manurangsi/Marcinkowski/Sornat IJCAI’20]
- bi-partite graph
- palette of colors
- set of constraints

**Partitioning system** generalizes [Feige, JACM’98], used in [Barman/Fawzi/Fermé, STACS’21]
- resources $\mathcal{R}$, cost $b(\cdot)$
- subsets $P_{i,j} \subseteq \mathcal{R}$
- $SC(\text{row})$, $SC(\text{scr})$ satisfy
  \[
  \frac{SC(\text{scr})}{SC(\text{row})} \approx \rho b
  \]
**Problem:** minimum social cost in atomic congestion games

**Main result I:** tight NP-hardness of approximation

**Main result II:** taxes achieve matching approximation

⇒ first poly algo optimal approx
Poly-time algorithm based on taxes
Poly-time algorithm based on taxes

**Background:**

- price of anarchy measures equilibrium quality, e.g., $\frac{SC(a^{NE})}{SC(a^{OPT})}$
  
  [Koutsoupias/Papadimitriou STACS’99; Christodoulou/Koutsoupias STOC’05; Aland/Dumrauf/Gairing/Monien/Schoppmann, STACS’06; Roughgarden JACM’15]
Poly-time algorithm based on taxes

Background:
– price of anarchy measures equilibrium quality, e.g., $SC(a^{NE})/SC(a^{OPT})$
  [Koutsoupias/Papadimitriou STACS’99; Christodoulou/Koutsoupias STOC’05; Aland/Dumrauf/Gairing/Monien/Schopppmann, STACS’06; Roughgarden JACM’15]
– efficient computation of CE/CCE
  [Papadimitriou/Roughgarden, JACM’08; Xin Jiang/Leyton-Brown, GEB’15; Hart/Mas-Colell, Econometrica’00; Blum/Hajiaghayi/Ligett/Roth, STOC’08]
Poly-time algorithm based on taxes

**Background:**

- price of anarchy measures equilibrium quality, e.g., $SC(a^{NE})/SC(a^{OPT})$
  
  [Koutsoupias/Papadimitriou STACS’99; Christodoulou/Koutsoupias STOC’05; Aland/Dumrauf/Gairing/Monien/Schopppmann, STACS’06; Roughgarden JACM’15]

- efficient computation of CE/CCE
  
  [Papadimitriou/Roughgarden, JACM’08; Xin Jiang/Leyton-Brown, GEB’15; Hart/Mas-Colell, Econometrica’00; Blum/Hajiaghayi/Ligett/Roth, STOC’08]

Price of anarchy as approximation ratio
Poly-time algorithm based on taxes

**Background:**

- Price of anarchy measures equilibrium quality, e.g., $SC(a^{NE})/SC(a^{OPT})$
  
  [Koutsoupias/Papadimitriou STACS’99; Christodoulou/Koutsoupias STOC’05; Aland/Dumrauf/Gairing/Monien/Schopppmann, STACS’06; Roughgarden JACM’15]

- Efficient computation of CE/CCE
  
  [Papadimitriou/Roughgarden, JACM’08; Xin Jiang/Leyton-Brown, GEB’15; Hart/Mas-Colell, Econometrica’00; Blum/Hajiaghayi/Ligett/Roth, STOC’08]

Price of anarchy as approximation ratio

$\leadsto$ **Q**: How to improve PoA?
Poly-time algorithm based on taxes

Background:
– price of anarchy measures equilibrium quality, e.g., \( \frac{SC(a^{NE})}{SC(a^{OPT})} \)
  [Koutsoupias/Papadimitriou STACS’99; Christodoulou/Koutsoupias STOC’05;
   Aland/Dumrauf/Gairing/Monien/Schoppmann, STACS’06; Roughgarden JACM’15]

– efficient computation of CE/CCE
  [Papadimitriou/Roughgarden, JACM’08; Xin Jiang/Leyton-Brown, GEB’15;
   Hart/Mas-Colell, Econometrica’00; Blum/Hajiaghayi/Ligett/Roth, STOC’08]

Price of anarchy as approximation ratio
\( \rightsquigarrow \textbf{Q: How to improve PoA?} \)

* coordination mechanisms: [Christodolou/Koutsoupias/Nanavati, ICALP’04 …]
* Stackelberg strategies: [Fotakis, ESA’04; Swamy, SODA’07 …]
* information provision: [Bhaskar/Cheng/Kun Ko/ Swamy, EC’16; Nachbar/Xu ArXiv’20 …]
* cost sharing: [Gkatzelis/Kollias/Roughgarden, WINE’14; Chen/Roughgarden/Valiant, J Comput’10 …]
* taxes: [Caragiannis/Kaklamanis/Kanellopoulos, Trans Alg’10; Bilò/Vinci, EC’16 …]
Poly-time algorithm based on taxes

Background:

– price of anarchy measures equilibrium quality, e.g., $\frac{SC(a^{NE})}{SC(a^{OPT})}$
  [Koutsoupias/Papadimitriou STACS’99; Christodoulou/Koutsoupias STOC’05;
   Aland/Dumrauf/Gairing/Monien/Schopppmann, STACS’06; Roughgarden JACM’15]

– efficient computation of CE/CCE
  [Papadimitriou/Roughgarden, JACM’08; Xin Jiang/Leyton-Brown, GEB’15;
   Hart/Mas-Colell, Econometrica’00; Blum/Hajiaghayi/Ligett/Roth, STOC’08]

Price of anarchy as approximation ratio

$\rightsquigarrow Q$: How to improve PoA?

* coordination mechanisms: [Christodolou/Koutsoupias/Nanavati, ICALP’04 ...]
* Stackelberg strategies: [Fotakis, ESA’04; Swamy, SODA’07 ...]
* information provision: [Bhaskar/Cheng/Kun Ko/Swamy, EC’16; Nachbar/Xu ArXiv’20 ...]
* cost sharing: [Gkatzelis/Kollias/Roughgarden, WINE’14; Chen/Roughgarden/Valiant, J Comput’10 ...]
* taxes: [Caragiannis/Kaklamanis/Kannelopoulos, Trans Alg’10; Bilò/Vinci, EC’16 ...]
Polynomial time algorithms – related work

* Best-known approx (LP + rounding)
  [Makarychev/Sviridenko, JACM'18]

\[ x \in \mathbb{R} > 0 \]
\[ \mathcal{E} \sim \text{Poi}(1) \]
\[ (\frac{x}{\mathcal{E}})^b (\frac{x}{\mathcal{E}}) \geq \text{NP-hardness factor} \]

* For polynomial costs, taxes achieve \( \text{PoA} = \mathcal{B}(d+1) \)
  [Caragiannis/Kaklamanis/Kannellopoulos, Trans Alg'10; Bilò/Vinci, EC'16]

Take-away: so far no matching approx in general
Polynomial time algorithms – related work

* Best-known approx (LP + rounding) [Makarychev/Sviridenko, JACM’18]

\[
\sup_{x \in \mathbb{R}_{>0}} \frac{\mathbb{E}_{P \sim \text{Poi}(1)}[(xP)b(xP)]}{xb(x)}
\]
Polynomial time algorithms – related work

* Best-known approx (LP + rounding) [Makarychev/Sviridenko, JACM’18]

\[
\sup_{x \in \mathbb{R}_{>0}} \frac{\mathbb{E}_{P \sim Poi(1)}[(xP)b(xP)]}{xb(x)} \geq \text{NP-hardness factor}
\]
Polynomial time algorithms – related work

* Best-known approx (LP + rounding) [Makarychev/Sviridenko, JACM’18]

\[
\sup_{x \in \mathbb{R}_{>0}} \frac{\mathbb{E}_{P \sim \text{Poi}(1)}[(xP)b(xP)]}{xb(x)} \geq \text{NP-hardness factor}
\]

* For polynomial costs, taxes achieve \( \text{PoA} = B(d + 1) \)

[Caragiannis/Kaklamanis/Kanellopoulos, Trans Alg’10; Bilò/Vinci, EC’16]
Polynomial time algorithms – related work

* Best-known approx (LP + rounding) \[\text{[Makarychev/Sviridenko, JACM’18]}\]

\[
\sup_{x \in \mathbb{R}^+} \frac{\mathbb{E}_{P \sim \text{Poi}(1)}[(xP)b(xP)]}{xb(x)} \geq \text{NP-hardness factor}
\]

* For polynomial costs, taxes achieve \(\text{PoA} = B(d + 1)\) \[\text{[Caragiannis/Kaklamanis/Kannellopoulos, Trans Alg’10; Bilò/Vinci, EC’16]}\]

Take-away: so far no matching approx in general
Theorem:
Consider congestion games where all resource costs are equal to $b \cdot \cdot$, positive, non-decreasing, semi-convex. For any $\epsilon > 0$, it is possible to efficiently compute a taxation mechanism so that $\text{PoA}_{\text{CCE}} \leq \rho b + \epsilon$.

Extends to resource costs obtained by non-negative combo of $b_1, \ldots, b_m$.

Corollary:
For any $\epsilon > 0$, there exists a polynomial time algorithm producing an allocation $a^*$ with cost $\text{SC}(a^*) \leq (\max_j \rho b_j + \epsilon) \cdot \text{OPT}$.
Theorem: Consider congestion games where all resource costs are equal to $b(\cdot)$, positive, non-decreasing, semi-convex.
Theorem: Consider congestion games where all resource costs are equal to $b(\cdot)$, positive, non-decreasing, semi-convex. For any $\varepsilon > 0$, it is possible to efficiently compute a taxation mechanism so that

$$\text{PoA}_{\text{CCE}} \leq \rho_b + \varepsilon$$
Matching polynomial time algorithm – main result

**Theorem:** Consider congestion games where all resource costs are equal to \( b(\cdot) \), positive, non-decreasing, semi-convex. For any \( \varepsilon > 0 \), it is possible to efficiently compute a taxation mechanism so that

\[
\text{PoA}_{\text{CCE}} \leq \rho_b + \varepsilon
\]

*Extends* to resource costs obtained by non-negative combo of \( b_1, \ldots, b_m \).
Theorem: Consider congestion games where all resource costs are equal to $b(\cdot)$, positive, non-decreasing, semi-convex. For any $\varepsilon > 0$, it is possible to efficiently compute a taxation mechanism so that

$$\text{PoA}_{\text{CCE}} \leq \rho_b + \varepsilon$$

*Extends* to resource costs obtained by non-negative combo of $b_1, \ldots, b_m$

**Corollary:** For any $\varepsilon > 0$, there exists a polynomial time algorithm producing an allocation $a^*$ with cost

$$SC(a^*) \leq \left( \max_j \rho_{b_j} + \varepsilon \right) \cdot OPT$$
Matching polynomial time algorithm - Proof Ideas
Matching polynomial time algorithm - Proof Ideas

\[
SC(a) = \sum_{r \in a} |a|_r b(|a|_r) \\
SC_P(a) = \sum_{r \in a} E_{P \sim \text{Poi}(|a|_r)}[Pb(P)]
\]
Matching polynomial time algorithm - Proof Ideas

\[
SC(a) = \sum_{r \in a} |a_r| b(|a_r|) \quad SC_P(a) = \sum_{r \in a} \mathbb{E}_{P \sim \text{Poi}(|a_r|)}[Pb(P)]
\]
Matching polynomial time algorithm - Proof Ideas

\[ SC(a) = \sum_{r \in a} |a|_r b(|a|_r) \quad SC_P(a) = \sum_{r \in a} \mathbb{E}_{P \sim \text{Poi}(|a|_r)}[Pb(P)] \]

Key ingredients:

P1: \( \bar{b}(x; \nu) \) solves crucial recursion
Matching polynomial time algorithm - Proof Ideas

\[ SC(a) = \sum_{r \in a} |a_r| b(|a_r|) \quad SC_P(a) = \sum_{r \in a} \mathbb{E}_{P \sim \text{Poi}(|a_r|)} [Pb(P)] \]

Key ingredients:

P1: \( \bar{b}(x; \nu) \) solves crucial recursion
P2: \( \bar{\nu} \) solves continuous relaxation of \( \min SC_P(a) \)
Matching polynomial time algorithm - Proof Ideas

\[
SC(a) = \sum_{r \in a} |a_r| b(|a_r|) \quad \quad SC_P(a) = \sum_{r \in a} \mathbb{E}_{P \sim \text{Poi}(|a_r|)} [Pb(P)]
\]

Key ingredients:

P1: \( \bar{b}(x; \nu) \) solves crucial recursion
P2: \( \nu \) solves continuous relaxation of \( \min SC_P(a) \)

\[
SC(a^{\text{NE}})^{P1} \leq SC_P(\nu)
\]
Matching polynomial time algorithm - Proof Ideas

\[ SC(a) = \sum_{r \in a} |a_r| b(|a_r|) \]

\[ SC_P(a) = \sum_{r \in a} \mathbb{E}_{P \sim \text{Poi}(|a_r|)} [Pb(P)] \]

Key ingredients:

P1: \( \tilde{b}(x; \nu) \) solves crucial recursion

P2: \( \nu \) solves continuous relaxation of min \( SC_P(a) \)

\[ SC(a^{\text{NE}}) \overset{P_1}{\leq} SC_P(\nu) \overset{P_2}{\leq} SC_P(a^{\text{OPT}}) \]
Matching polynomial time algorithm - Proof Ideas

\[ SC(a) = \sum_{r \in a} |a_r| b(|a_r|) \]

\[ SC_P(a) = \sum_{r \in a} \mathbb{E}_{P \sim \text{Poi}(|a_r|)}[Pb(P)] \]

Key ingredients:

P1: \( \bar{b}(x; \nu) \) solves crucial recursion
P2: \( \nu \) solves continuous relaxation of \( \min SC_P(a) \)

\[ SC(a^{\text{NE}}) \overset{P_1}{\leq} SC_P(\nu) \overset{P_2}{\leq} SC_P(a^{\text{OPT}}) \overset{\text{def}}{=} \rho_b SC(a^{\text{OPT}}) \]
Conclusion and open questions

Problem: minimum social cost in atomic congestion games

Main result I: tight NP-hardness of approximation

Main result II: taxes achieve matching approximation \( \Rightarrow \) first poly algo optimal approx

Remarks:
* Competitive decision making + incentives = best-centralized
* Surprising that "taxes are enough"
* Poly-time algo requires centralized solution of cvx opt

\[ \text{[Paccagnan/Chandan/Ferguson/Marden, TEAC'21]} \]

13
Conclusion and open questions

<table>
<thead>
<tr>
<th>Problem: minimum social cost in atomic congestion games</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main result I: tight NP-hardness of approximation</td>
</tr>
<tr>
<td>Main result II: taxes achieve matching approximation</td>
</tr>
<tr>
<td>$\implies$ first poly algo optimal approx</td>
</tr>
</tbody>
</table>
Conclusion and open questions

**Problem:** minimum social cost in atomic congestion games

**Main result I:** tight NP-hardness of approximation

**Main result II:** taxes achieve matching approximation

\[ \implies \text{first poly algo optimal approx} \]

**Remarks:**

* Competitive decision making + incentives = best-centralized
Conclusion and open questions

**Problem:** minimum social cost in atomic congestion games

**Main result I:** tight NP-hardness of approximation

**Main result II:** taxes achieve matching approximation
   \[ \Rightarrow \text{first poly algo optimal approx} \]

**Remarks:**

* Competitive decision making + incentives = best-centralized
* Surprising that “taxes are enough”
Conclusion and open questions

**Problem:** minimum social cost in atomic congestion games

**Main result I:** tight NP-hardness of approximation

**Main result II:** taxes achieve matching approximation

⇒ first poly algo optimal approx

**Remarks:**

* Competitive decision making + incentives = best-centralized

* Surprising that “taxes are enough”

* Poly-time algo requires centralized solution of cvx opt
Conclusion and open questions

**Problem:** minimum social cost in atomic congestion games

**Main result I:** tight NP-hardness of approximation

**Main result II:** taxes achieve matching approximation
\[ \Rightarrow \text{first poly algo optimal approx} \]

**Remarks:**

* Competitive decision making + incentives = best-centralized
* Surprising that “taxes are enough”

* Poly-time algo requires centralized solution of cvx opt
  If undesirable \[ \sim \] optimal local tax [Paccagnan/Chandan/Ferguson/Marden, TEAC’21]
  very little performance loss, e.g., 2.012 vs 2 for affine
Problem: minimum social cost in atomic congestion games

Main result I: tight NP-hardness of approximation

Main result II: taxes achieve matching approximation

\[ \rightarrow \text{first poly algo optimal approx} \]

Remarks:

* Competitive decision making + incentives = best-centralized

* Surprising that “taxes are enough”

* Poly-time algo requires centralized solution of cvx opt
  If undesirable \[ \rightsquigarrow \text{optimal local tax} \]
  [Paccagnan/Chandan/Ferguson/Marden, TEAC’21]
  very little performance loss, e.g., 2.012 vs 2 for affine

* Main result II extends to network CG
“Judiciously designed taxes achieve optimal approximation, and no other tractable intervention can improve upon this result”