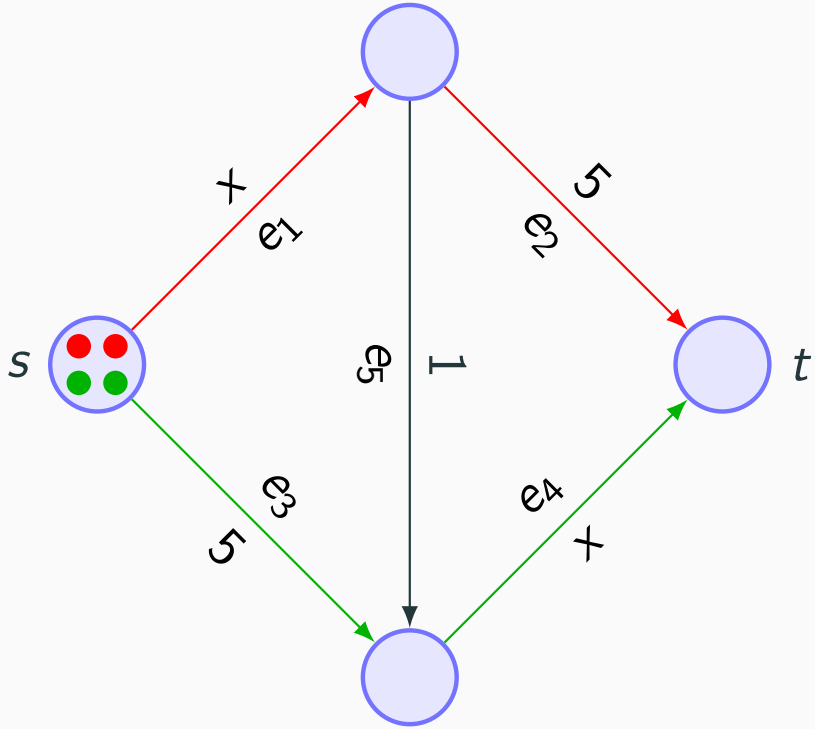
Dynamic Network Congestion Games

Joint work with Nathalie Bertrand, Nicolas Markey, and Ocan Sankur

Suman Sadhukhan, Inria Rennes

MOVEP, 23^{rd} June, 2020

Routing Games[1]



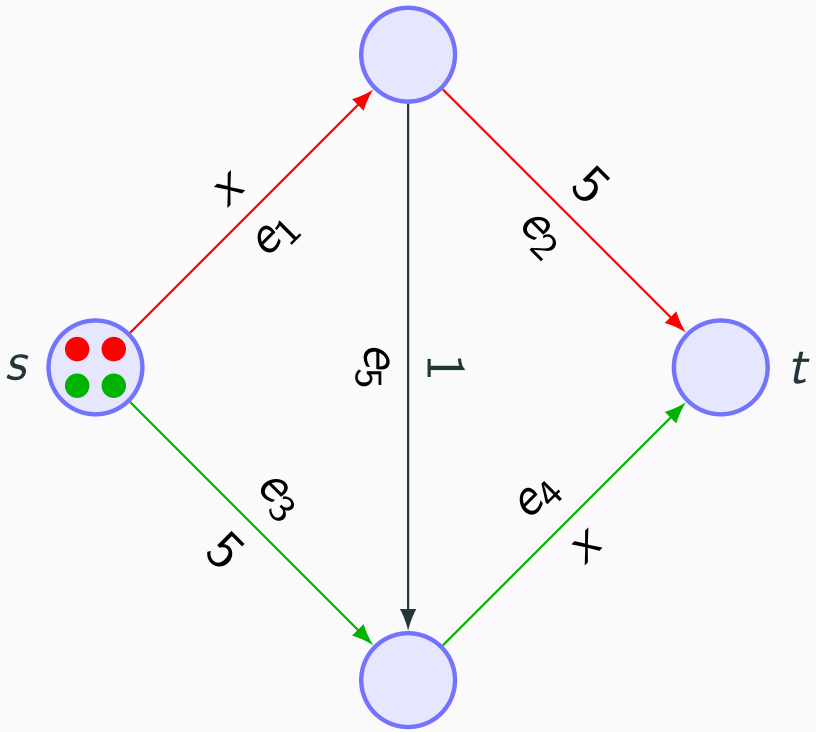
$$cost = 2 + 5 = 7$$

$$cost = 5 + 2 = 7$$

$$Total\ cost = 7 \times 2 + 7 \times 2 = 28$$

[1] Roughgarden 2005.

Routing Games[1]

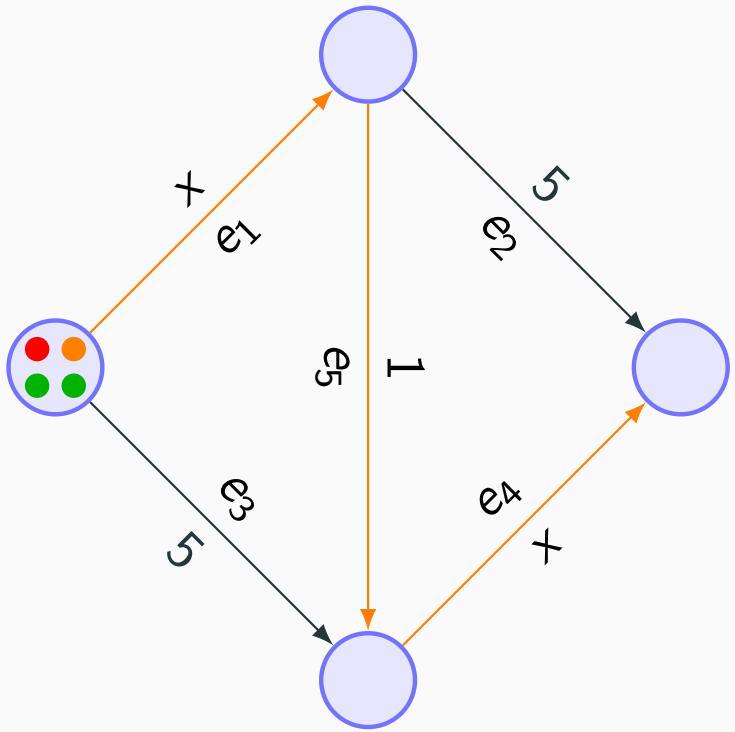


"centralized"

$cost = 7$

$cost = 7$

$Total\ cost = 7 \times 2 + 7 \times 2 = 28$



"selfish"

$cost = 7$

$cost = 5 + 3 = 8$

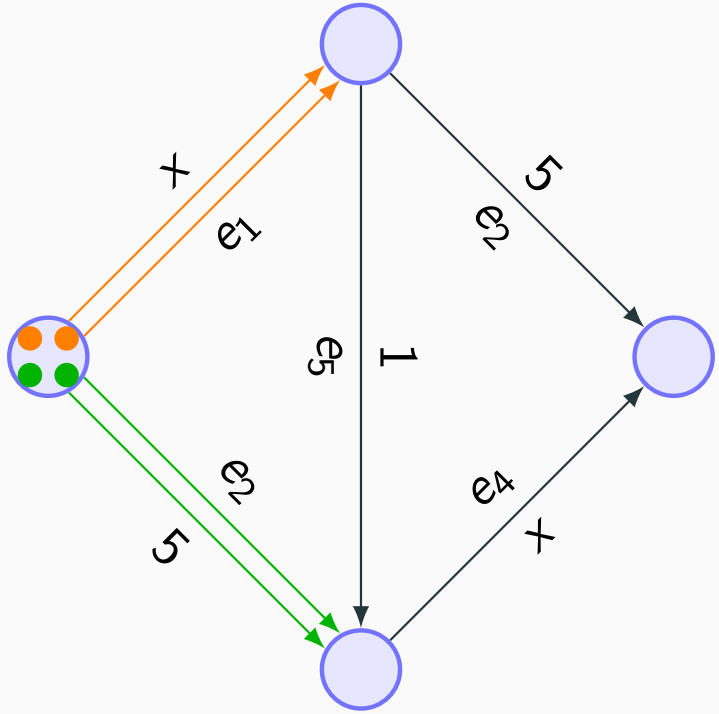
$cost = 2 + 1 + 3 = 6$

$Total\ cost = 7 + 8 \times 2 + 6 = 29$

[1] Roughgarden 2005.

Where we differ?

- Synchronicity: congestion cost only if players take an edge simultaneously
- Dynamic strategies: Strategies are not just paths from source to target

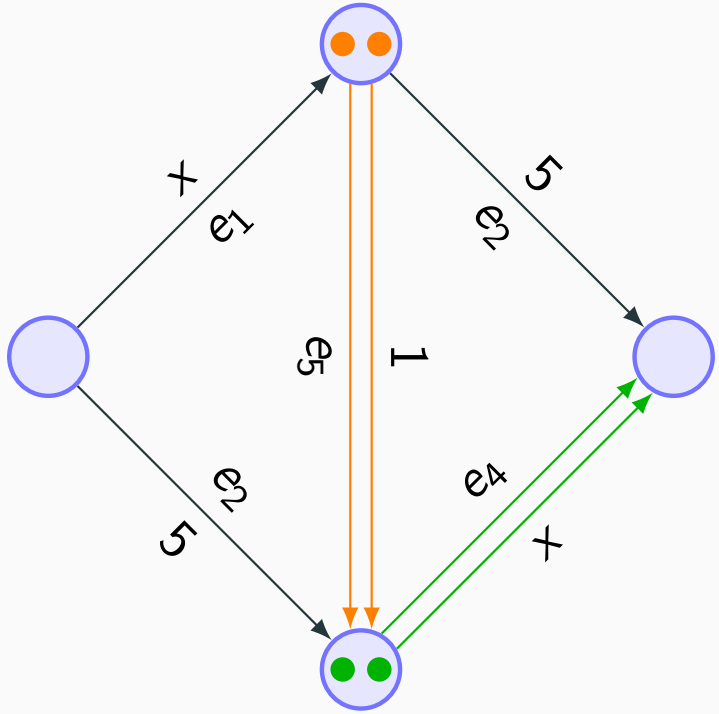


$cost = 2+$

$cost = 5+$

Where we differ?

- Synchronicity: congestion cost only if players take an edge simultaneously
- Dynamic strategies: Strategies are not just paths from source to target

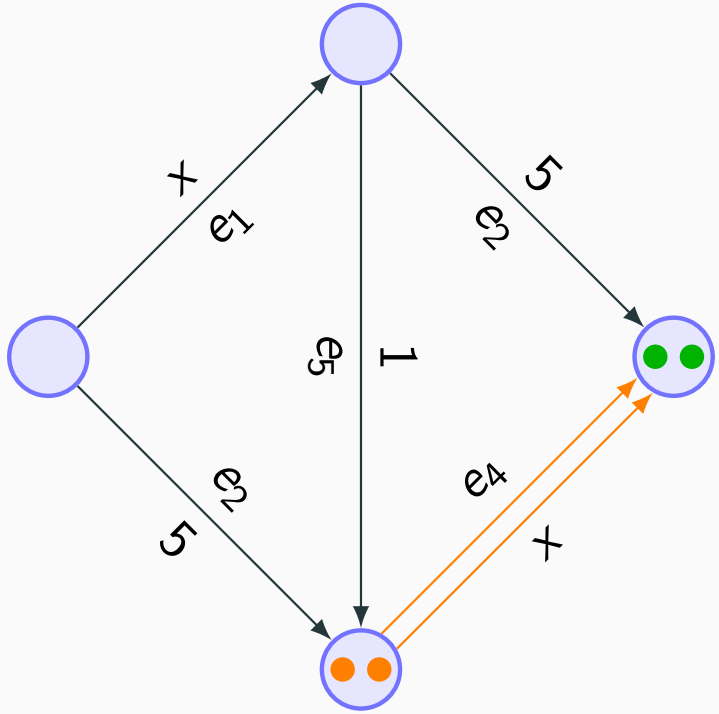


$cost = 2 + 1$

$cost = 5 + 2$

Where we differ?

- Synchronicity: congestion cost only if players take an edge simultaneously
- Dynamic strategies: Strategies are not just paths from source to target

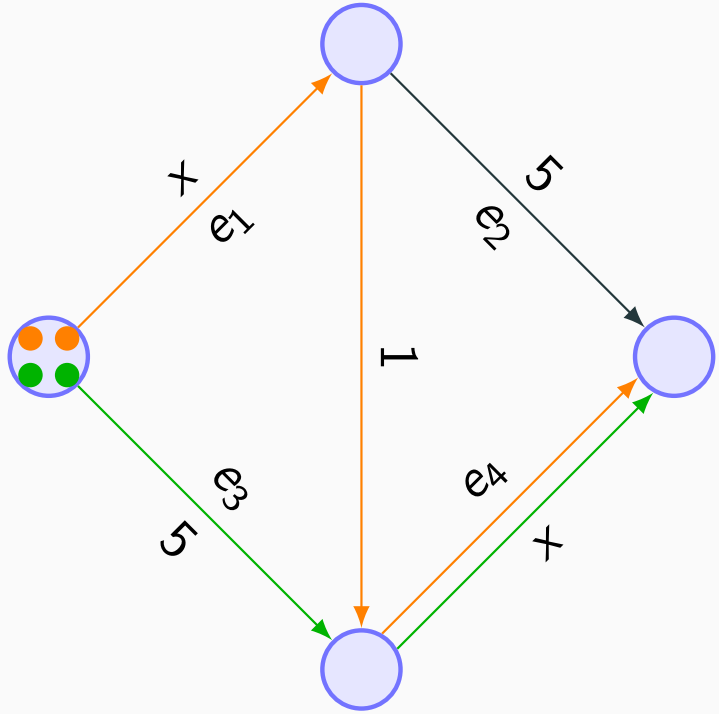


$cost = 2 + 1 + 2 = 5$

$cost = 5 + 2 = 7$

Where we differ?

- Synchronicity: congestion cost only if players take an edge simultaneously
- Dynamic strategies: Strategies are not just paths from source to target



$cost = 2 + 1 + 2 = 5,\ old\ cost = 7$

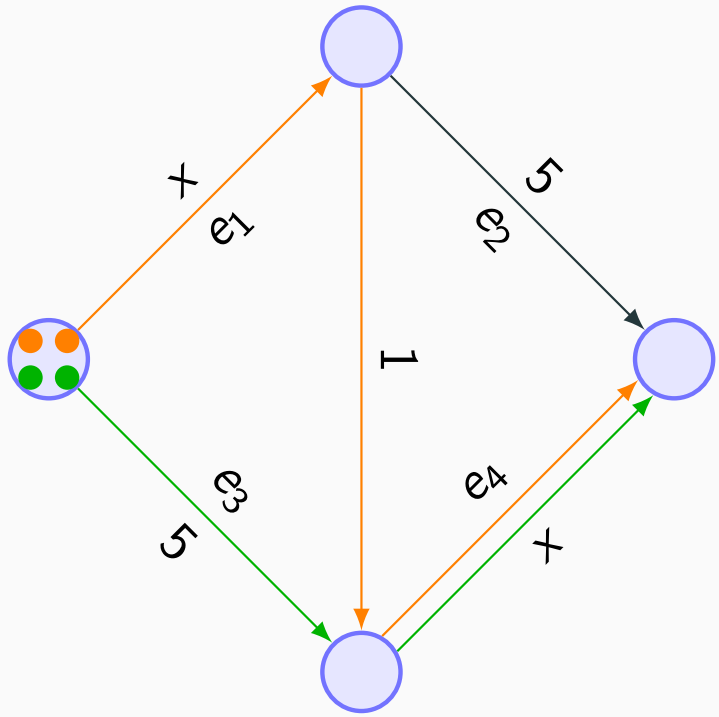
$cost = 5 + 2 = 7,\ old\ cost = 9$

Where we differ?

- Synchronicity: congestion cost only if players take an edge simultaneously
- Dynamic strategies: Strategies are not just paths from source to target

Related model: Dynamic Resource allocation games (DRAG)[2]

- Edges are resources, paths are sets of resources
- Ordered DRAG is acyclic: results do not carry forward to our model

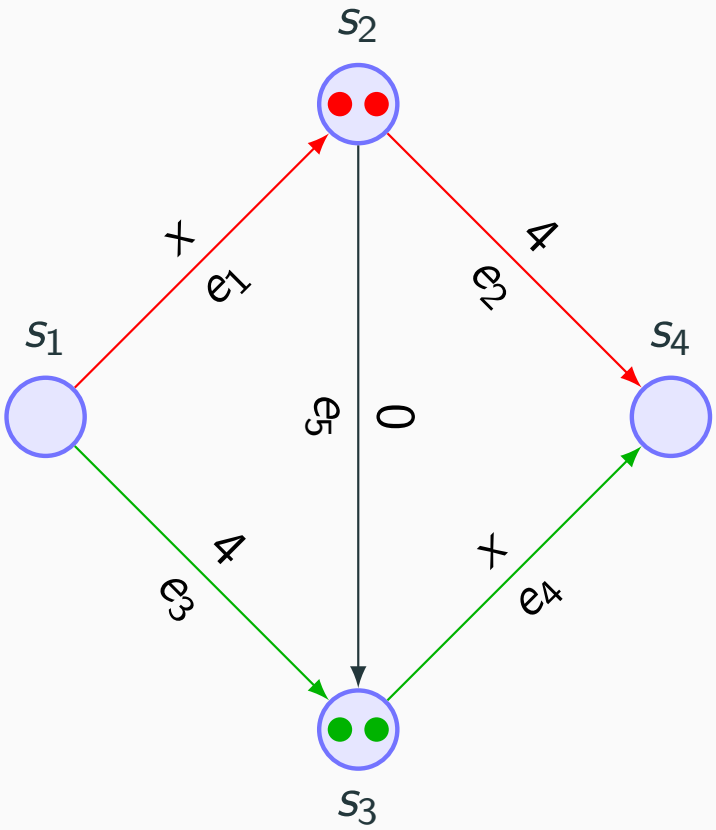


$cost = 2 + 1 + 2 = 5,\ old\ cost = 7$

$cost = 5 + 2 = 7,\ old\ cost = 9$

[2] Avni, Henzinger, and Kupferman 2016.

Our model: Dynamic Network Congestion Games



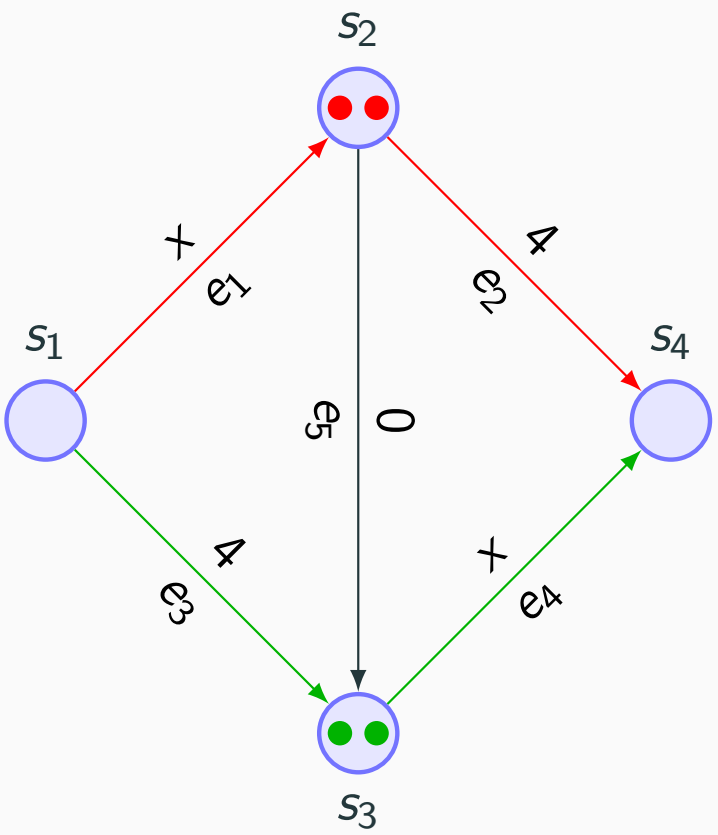
$\mathcal{G} = (V, E, \{cost_e\}_{e \in E}, k, s, t)$

$cost_e : \mathbb{N} \to \mathbb{N}$

k : Number of players

$s, t \in V$: source and target vertices

Our model: Dynamic Network Congestion Games



$\mathcal{G} = (V, E, \{cost_e\}_{e \in E}, k, s, t)$

$cost_e : \mathbb{N} \to \mathbb{N}$

k : Number of players

$s, t \in V$: source and target vertices

Configuration $c = (s_2, s_2, s_3, s_3) \in \mathcal{C}$

$$cost_i(paths) = \sum_{e \in path_i} cost_e(.)$$

- Blind strategies : Players only observe history length, $\sigma_i^{bl} : \mathbb{N} \to E$.
- General strategies: $\sigma_i^{g} : \mathcal{C}^+ \to E$.

Standard Concepts

Nash Equilibrium (NE). A strategy profile $(\sigma_1, \sigma_2, \ldots \sigma_k)$ is a NE if no player has an incentive to deviate from its current strategy. $\rightsquigarrow$ "selfish"

Social Optimal (SO). A strategy profile $(\sigma_1, \sigma_2, \ldots \sigma_k)$ is a SO if $cost = \sum_i cost_i$ is minimum. $\rightsquigarrow$ "centralized"

Price of Anarchy (PoA)[a] $= \dfrac{\text{Total cost of "worst" NE}}{\text{Total cost of SO}}$

Price of Stability (PoS) $= \dfrac{\text{Total cost of "best" NE}}{\text{Total cost of SO}}$

[a] Koutsopias and Papadimitrou 2009.

Standard Concepts

Nash Equilibrium (NE). A strategy profile $(\sigma_1, \sigma_2, \ldots \sigma_k)$ is a NE if no player has an incentive to deviate from its current strategy. $\rightsquigarrow$ "selfish"

Social Optimal (SO). A strategy profile $(\sigma_1, \sigma_2, \ldots \sigma_k)$ is a SO if $cost = \sum_i cost_i$ is minimum. $\rightsquigarrow$ "centralized"

Price of Anarchy (PoA)[a] $= \dfrac{\text{Total cost of "worst" NE}}{\text{Total cost of SO}}$

Price of Stability (PoS) $= \dfrac{\text{Total cost of "best" NE}}{\text{Total cost of SO}}$

[a] Koutsopias and Papadimitrou 2009.

Standard Concepts

Nash Equilibrium (NE). A strategy profile $(\sigma_1, \sigma_2, \ldots \sigma_k)$ is a NE if no player has an incentive to deviate from its current strategy. $\rightsquigarrow$ "selfish"

Social Optimal (SO). A strategy profile $(\sigma_1, \sigma_2, \ldots \sigma_k)$ is a SO if $cost = \sum_i cost_i$ is minimum. $\rightsquigarrow$ "centralized"

Price of Anarchy (PoA)[a] $= \dfrac{\text{Total cost of "worst" NE}}{\text{Total cost of SO}}$

Price of Stability (PoS) $= \dfrac{\text{Total cost of "best" NE}}{\text{Total cost of SO}}$

[a] Koutsopias and Papadimitrou 2009.

Questions:

- **Is PoA/PoS always well-defined?**
- **How to compute these measures?**

Is PoA/PoS always well-defined for our model?

Existence and Computation of NE: Blind Strategies

Best-Response Problem (BR). Given a strategy profile $\mathcal{P} = (\pi_1, \pi_2, \ldots \pi_k)$, can player i improve its cost by deviating? If so, what is its optimal strategy?

Existence and Computation of NE: Blind Strategies

Best-Response Problem (BR). Given a strategy profile $\mathcal{P} = (\pi_1, \pi_2, \ldots \pi_k)$, can player i improve its cost by deviating? If so, what is its optimal strategy?

Applying algorithm for BR iteratively yields a NE *in the limit*.

Existence and Computation of NE: Blind Strategies

Best-Response Problem (BR). Given a strategy profile $\mathcal{P} = (\pi_1, \pi_2, \ldots \pi_k)$, can player i improve its cost by deviating? If so, what is its optimal strategy?

Applying algorithm for BR iteratively yields a NE *in the limit*.

- ▷ How to solve BR problem?
- ▷ Does this iterative procedure converge?

Existence and Computation of NE: Blind Strategies

Best-Response Problem (BR). Given a strategy profile $\mathcal{P} = (\pi_1, \pi_2, \ldots \pi_k)$, can player i improve its cost by deviating? If so, what is its optimal strategy?

Applying algorithm for BR iteratively yields a NE *in the limit*.

- ▷ How to solve BR problem? Shortest weighted path problem
- ▷ Does this iterative procedure converge? Existence of Potential function[3]

[3] D.Monderer and Shapley 1996.

Existence and Computation of NE: Blind Strategies

Best-Response Problem (BR). Given a strategy profile $\mathcal{P} = (\pi_1, \pi_2, \ldots \pi_k)$, can player i improve its cost by deviating? If so, what is its optimal strategy?

Applying algorithm for BR iteratively yields a NE *in the limit*.

- ▷ How to solve BR problem? Shortest weighted path problem: PTIME
- ▷ Does this iterative procedure converge? Existence of Potential function[3] Only for blind strategies

[3] D.Monderer and Shapley 1996.

Existence and computation of NE: General Strategies

Theorem
Blind strategy NE are general strategy NE.

Existence and computation of NE: General Strategies

Theorem
Blind strategy NE are general strategy NE.

That shows PoS/PoA are always well-defined for our model.

Existence and computation of NE: General Strategies

Theorem
Blind strategy NE are general strategy NE.

That shows PoS/PoA are always well-defined for our model.

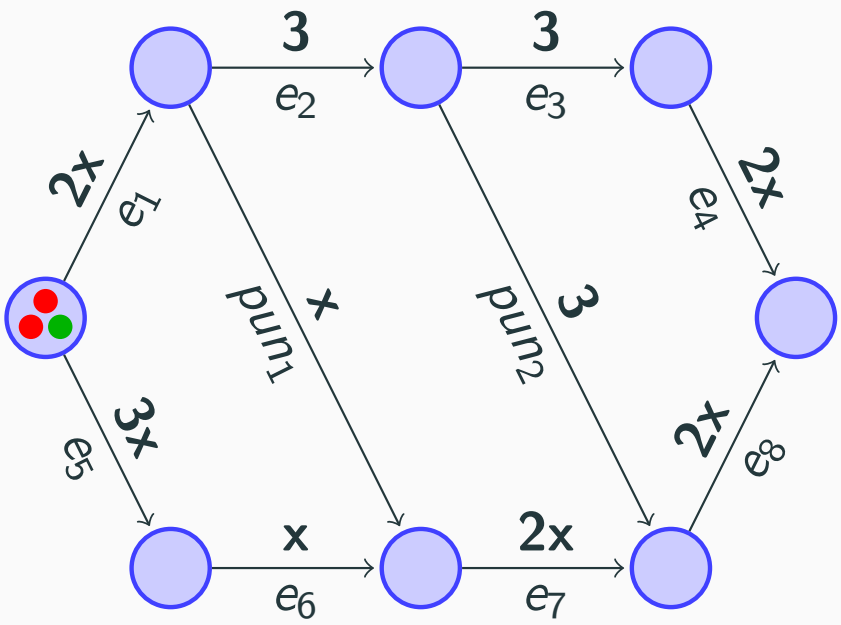
Are there more?

Existence and computation of NE: General Strategies

Theorem
Blind strategy NE are general strategy NE.

That shows PoS/PoA are always well-defined for our model.

Are there more?

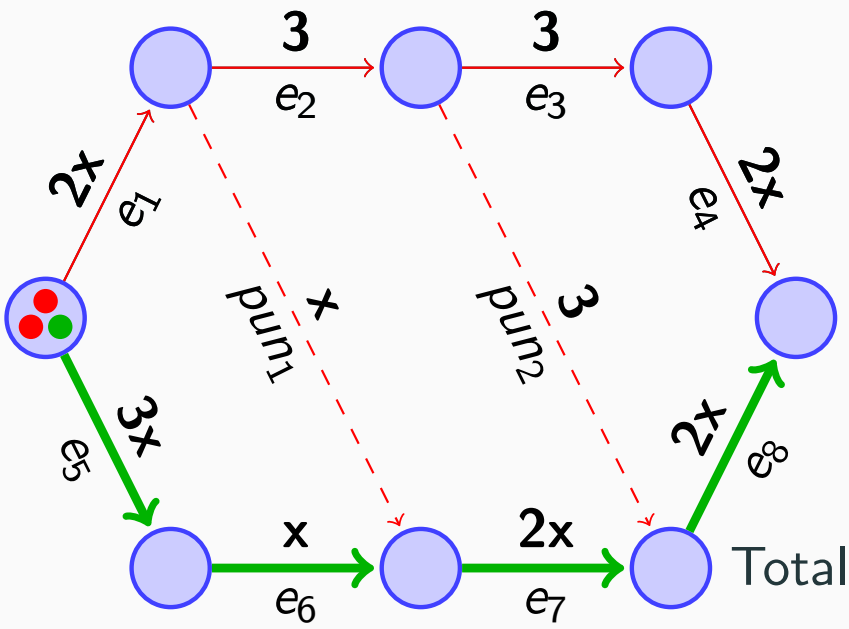


Existence and computation of NE: General Strategies

Theorem
Blind strategy NE are general strategy NE.

That shows PoS/PoA are always well-defined for our model.

Are there more?



- cost = 14
- cost = 8

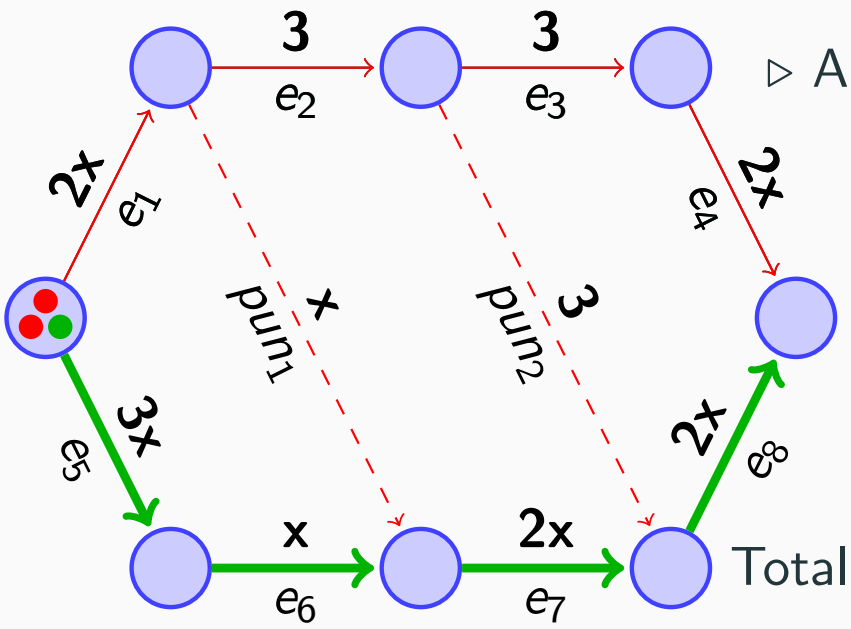
Total cost $= 14 \times 2 + 8 = 36$

Existence and computation of NE: General Strategies

Theorem
Blind strategy NE are general strategy NE.

That shows PoS/PoA are always well-defined for our model.

Are there more?



- ▷ All blind NEs have total cost at least 37.
- ▷ PoS depends on strategy space.

- cost $= 14$
- cost $= 8$

Total cost $= 14 \times 2 + 8 = 36$

How to compute PoS/PoA for our model?

Decision problem(s)

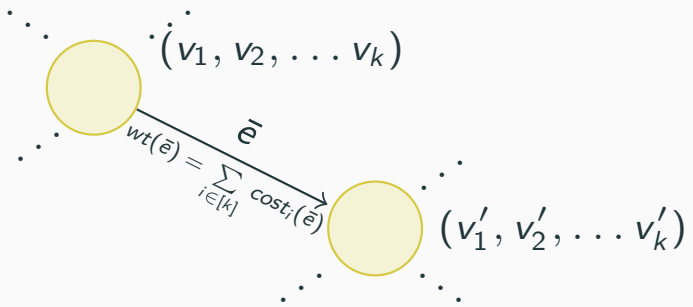
Social optimal: $\exists$ strategy profile.$\rightsquigarrow$ Computes SOpt
Input. A dynamic NCG $\mathcal{G}$, and $M_{opt} \in \mathbb{N} \cup \{0\}$.
Question. Does there exist a strategy profile with social cost $\leq M_{opt}$.

Best NE problem: $\exists$ NE profile.$\rightsquigarrow$ Computes PoS
Input. A dynamic NCG $\mathcal{G}$, and $M_{\exists} \in \mathbb{N} \cup \{0\}$.
Question. Does there exist a NE profile with social cost $\leq M_{\exists}$.

Worst NE problem: $\forall$ NE profile.$\rightsquigarrow$ Computes PoA
Input. A dynamic NCG $\mathcal{G}$, and $M_{\forall} \in \mathbb{N} \cup \{0\}$.
Output. Do all NE profiles in $\mathcal{G}$ have social cost $\leq M_{\forall}$.

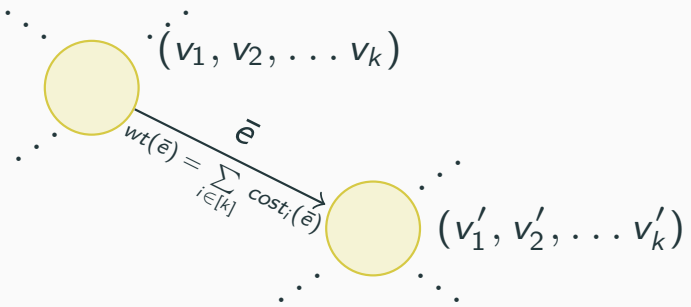
Social Optimal Problem

Configuration Graph ($\mathcal{C}_\mathcal{G}$): $(V^k, \mathcal{E})$
Make it a weighted graph with weight function, $wt(\bar{e}) = \sum_{i \in [k]} cost_i(\bar{e})$



Social Optimal Problem

Configuration Graph ($\mathcal{C}_{\mathcal{G}}$): $(V^k, \mathcal{E})$
Make it a weighted graph with weight function, $wt(\bar{e}) = \sum_{i \in [k]} cost_i(\bar{e})$

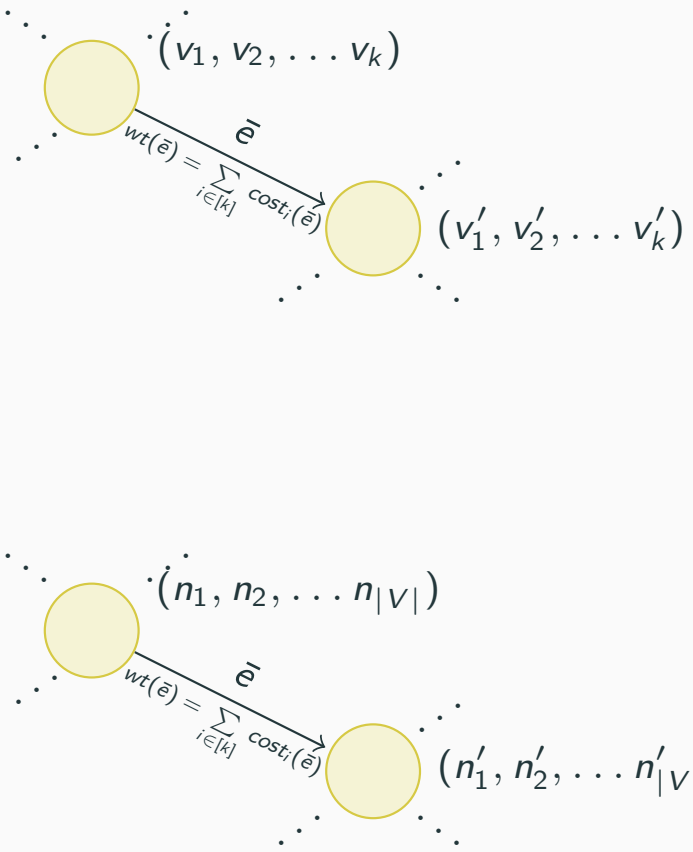


Shortest weighted path in this graph = the outcome of an SO
$\rightsquigarrow$ Can compute cost of SO from this! ☺

Social Optimal Problem

Configuration Graph ($\mathcal{C}_{\mathcal{G}}$): $(V^k, \mathcal{E})$
Make it a weighted graph with weight function, $wt(\bar{e}) = \sum_{i \in [k]} cost_i(\bar{e})$

Shortest weighted path in this graph = the outcome of an SO
$\rightsquigarrow$ Can compute cost of SO from this! ☺
$\rightsquigarrow$ Configuration graph is double-exponential in k ☹

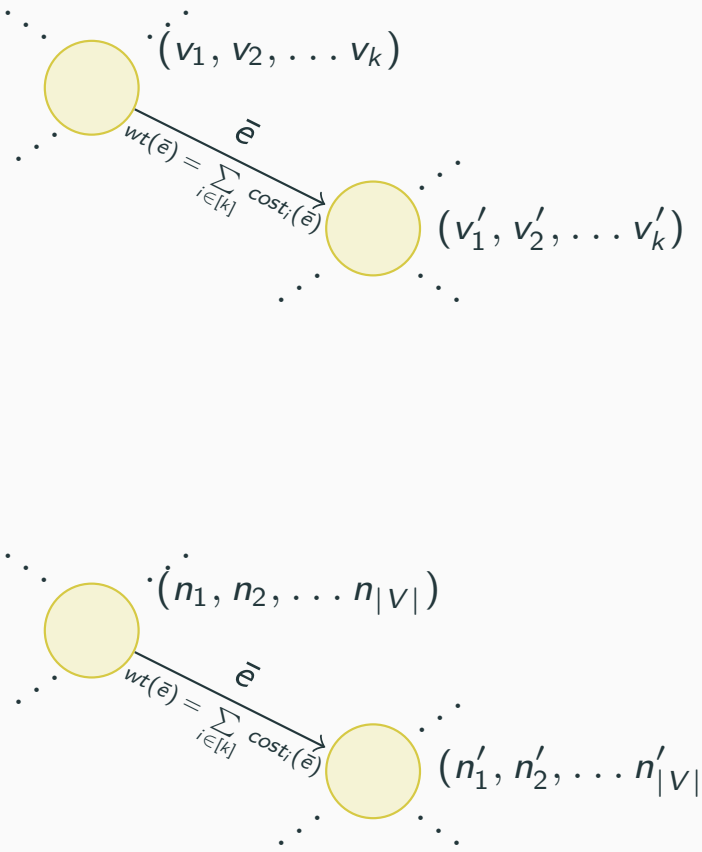


Social Optimal Problem

Configuration Graph ($\mathcal{C}_\mathcal{G}$): $(V^k, \mathcal{E})$
Make it a weighted graph with weight function, $wt(\bar{e}) = \sum_{i \in [k]} cost_i(\bar{e})$

Shortest weighted path in this graph = the outcome of an SO
$\rightsquigarrow$ Can compute cost of SO from this! ☺
$\rightsquigarrow$ Configuration graph is double-exponential in k ☹

Configuration graph does not store any information about NE property
$\rightsquigarrow$ Can't compute cost of NEs from $\mathcal{C}_\mathcal{G}$ as it is!! ☹



PSPACE Complexity for $\exists$ strategy profile

Figure 1: An arena $\mathcal{A}$

PSPACE Complexity for ∃ strategy profile

Figure 1: An arena $\mathcal{A}$

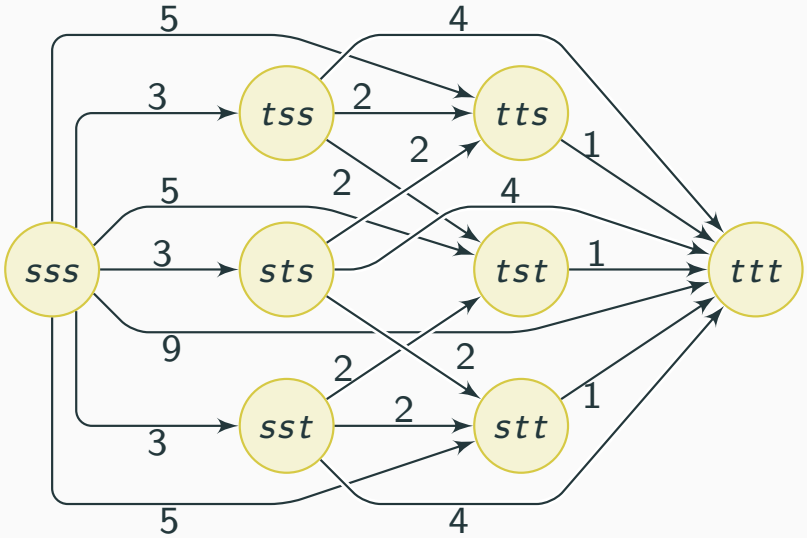


Figure 2: The configuration graph associated with $(\mathcal{A}, 3)$ (loops omitted)

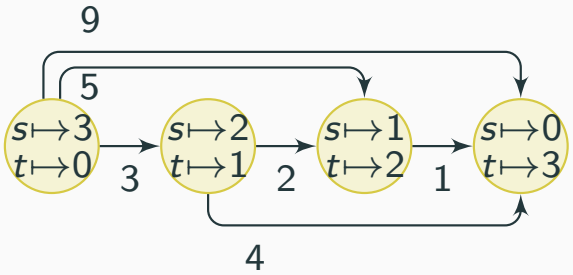


Figure 3: The abstract weighted graph associated with $(\mathcal{A}, 3)$ (loops omitted)

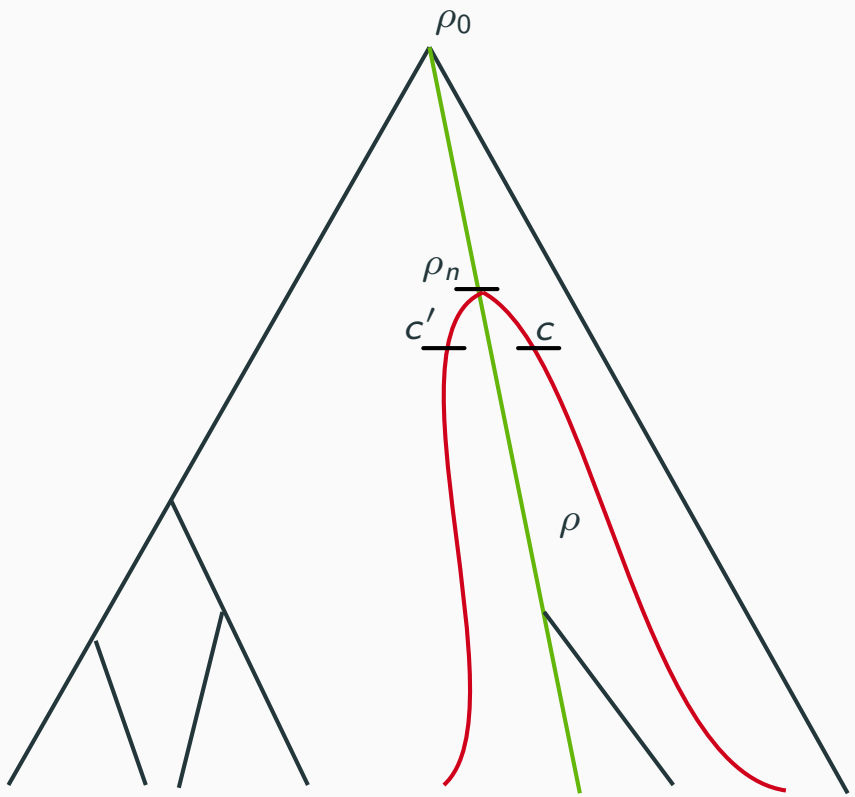
Characterization of outcomes of NE

Theorem
A play ρ is the outcome of an NE iff

$$\forall i \in [k], \forall n < |\rho|, \forall c \in Succ_\rho(i,n)$$
$$cost_i(\rho_{\geq n}) \leq val_{i,c} + cost_i(\rho_n, c)$$

where
$Succ_\rho(i,n)$ = siblings of ρ_{n+1} that can be reached from ρ by varying only player-i's choice
$=\{c | (\rho_n, c) \in \mathcal{E} \& c[j] = \rho_{n+1}[j] \forall j \neq i\}$

$val_{i,c}$ = cost that player-i can not avoid paying from configuration c if others form a coalition against that player
$= \min_{\sigma_i} \max_{\sigma_{-i}} cost_i^c((\sigma_i, \sigma_{-i}))$



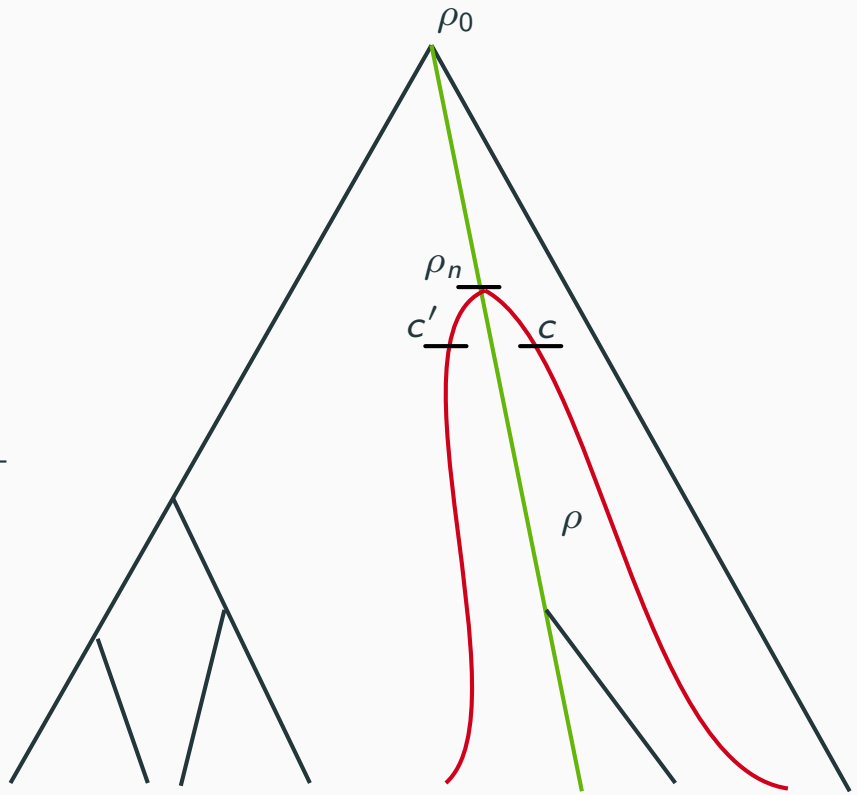
Characterization of outcomes of NE

Theorem
A play ρ is the outcome of an NE iff

$$\forall i \in [k], \forall n < |\rho|, \forall c \in Succ_\rho(i, n)$$
$$cost_i(\rho_{\geq n}) \leq val_{i,c} + cost_i(\rho_n, c)$$

Computation of $val_{i,c}$ is reduced to the value computation of a 2-player turn-based game[a]

[a] Khachiyan et al. 2008.



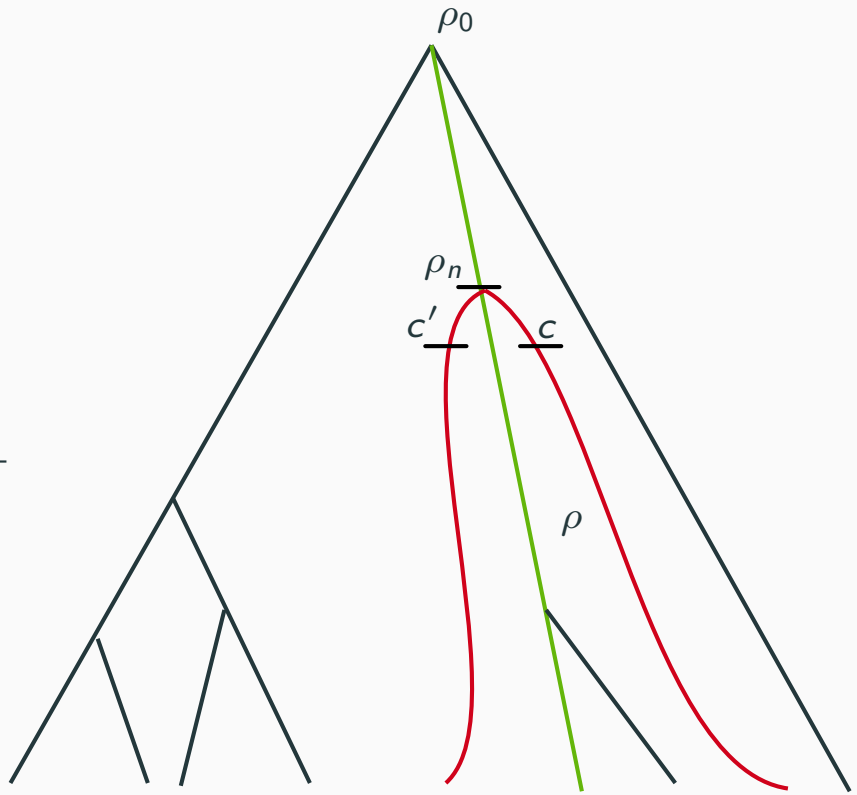
Characterization of outcomes of NE

Theorem
A play ρ is the outcome of an NE iff

$$\forall i \in [k], \forall n < |\rho|, \forall c \in Succ_\rho(i, n)$$
$$cost_i(\rho_{\geq n}) \leq val_{i,c} + cost_i(\rho_n, c)$$

Computation of $val_{i,c}$ is reduced to the value computation of a 2-player turn-based game[a]

[a] Khachiyan et al. 2008.



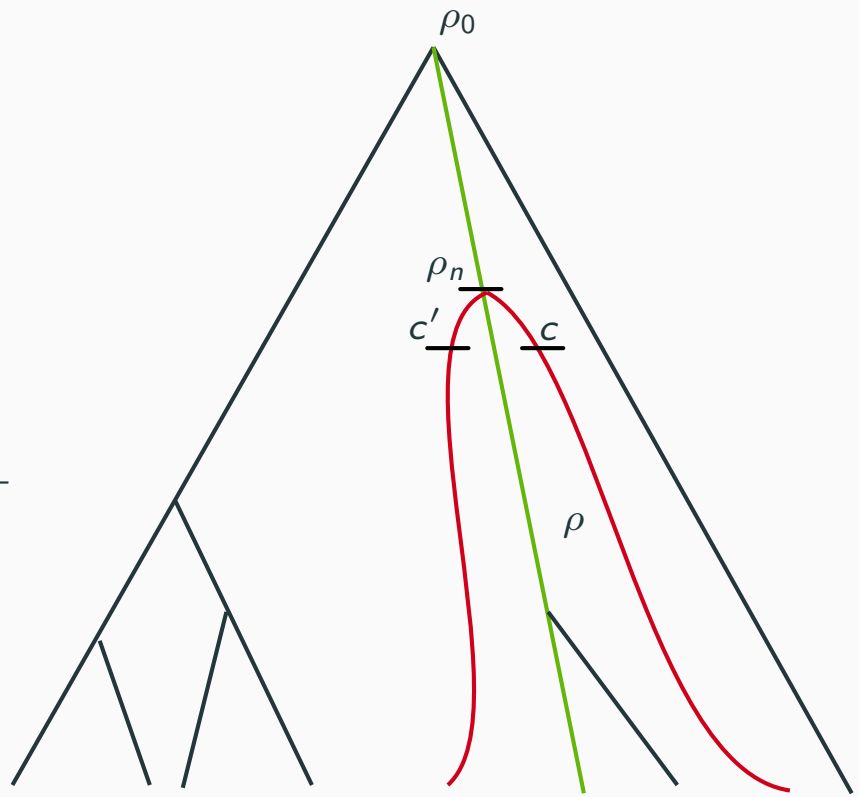
Characterization of outcomes of NE

Theorem
A play ρ is the outcome of an NE iff

$$\forall i \in [k], \forall n < |\rho|, \forall c \in Succ_\rho(i, n)$$
$$cost_i(\rho_{\geq n}) \leq val_{i,c} + cost_i(\rho_n, c)$$

Computation of $val_{i,c}$ is reduced to the value computation of a 2-player turn-based game[a] $\rightsquigarrow$ in EXPTIME

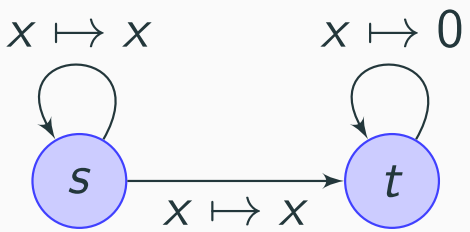
[a] Khachiyan et al. 2008.



EXPSPACE algorithm for best & worst NE

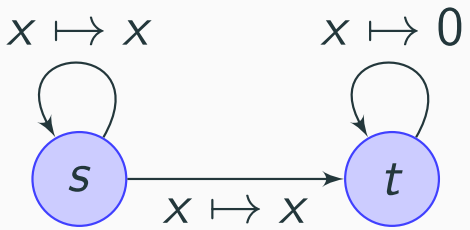
$$m_i = \min\{\underbrace{m_i - cost_i(\bar{e})}_{\text{From history}}, \underbrace{\min_c \langle val_{i,c} + cost_i(\_, c) - cost_i(\bar{e})\rangle}_{\text{For current state}}\}$$

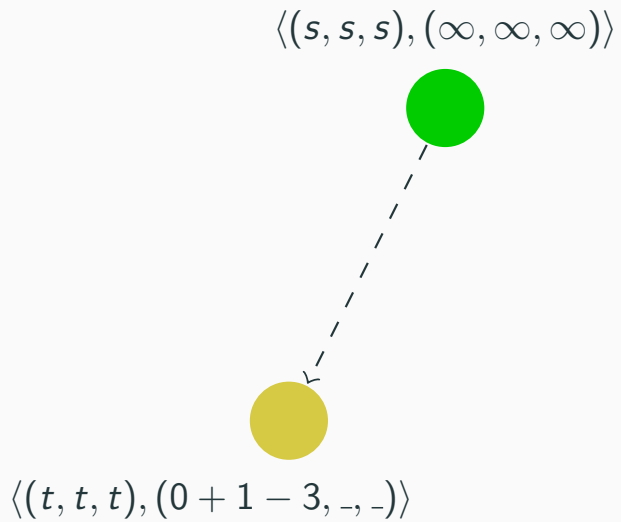
$\langle (s,s,s), (\infty,\infty,\infty)\rangle$



EXPSPACE algorithm for best & worst NE

$$m_i = \min\{\underbrace{m_i - cost_i(\bar{e})}_{\text{From history}}, \underbrace{\min_c \langle val_{i,c} + cost_i(\_, c) - cost_i(\bar{e})\rangle}_{\text{For current state}}\}$$

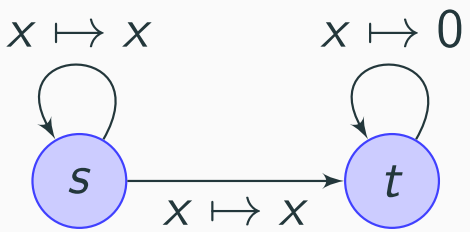




EXPSPACE algorithm for best & worst NE

$$m_i = \min\{\underbrace{m_i - cost_i(\bar{e})}_{\text{From history}}, \underbrace{\min_c \langle val_{i,c} + cost_i(\_, c) - cost_i(\bar{e})\rangle}_{\text{For current state}}\}$$

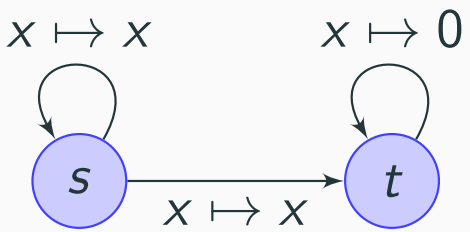
$\langle (s, s, s), (\infty, \infty, \infty)\rangle$

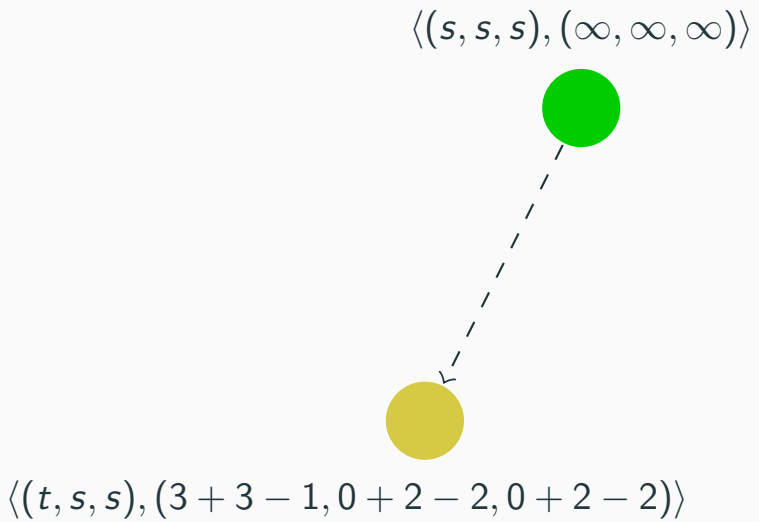


$\langle (t, t, t), (< 0, \_, \_)\rangle$

EXPSPACE algorithm for best & worst NE

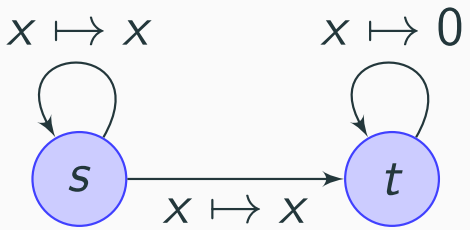
$$m_i = \min\{\underbrace{m_i - cost_i(\bar{e})}_{\text{From history}}, \underbrace{\min_c \langle val_{i,c} + cost_i(\_, c) - cost_i(\bar{e})\rangle}_{\text{For current state}}\}$$

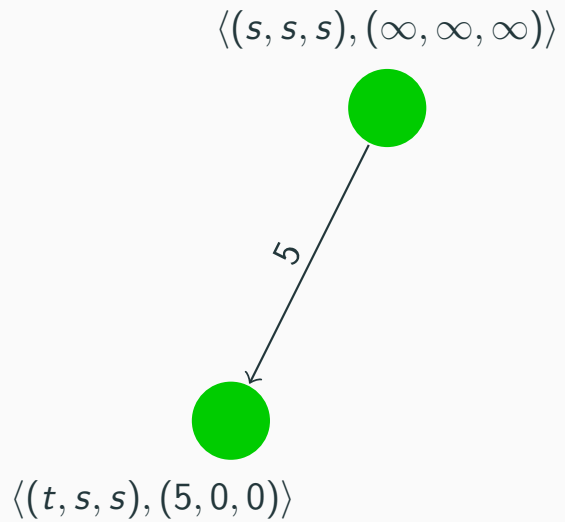




EXPSPACE algorithm for best & worst NE

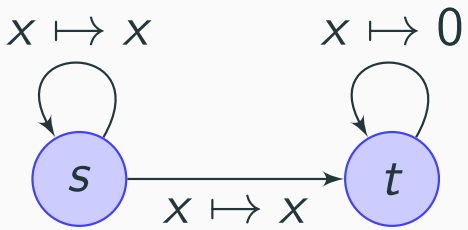
$$m_i = \min\{\underbrace{m_i - cost_i(\bar{e})}_{\text{From history}}, \underbrace{\min_c \langle val_{i,c} + cost_i(\_, c) - cost_i(\bar{e})\rangle}_{\text{For current state}}\}$$

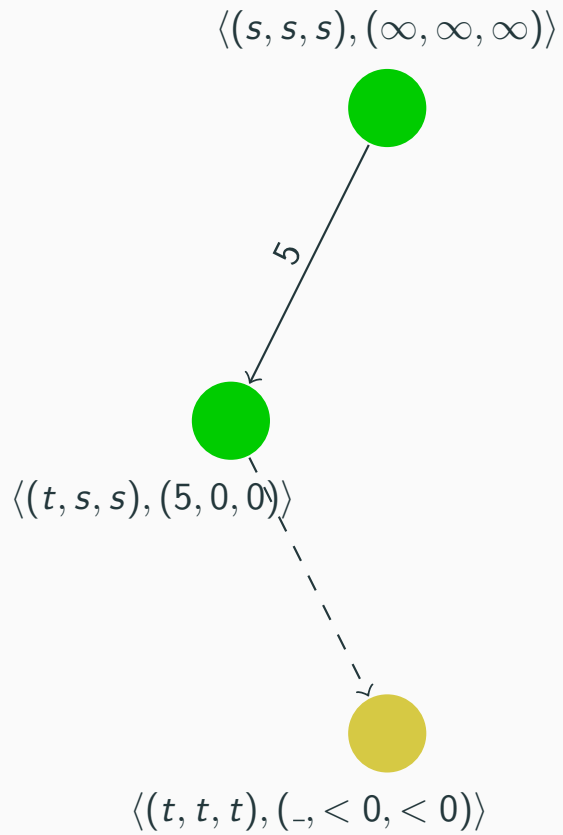




EXPSPACE algorithm for best & worst NE

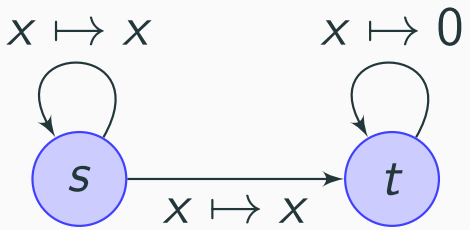
$$m_i = \min\{\underbrace{m_i - cost_i(\bar{e})}_{\text{From history}}, \underbrace{\min_c \langle val_{i,c} + cost_i(\_, c) - cost_i(\bar{e})\rangle}_{\text{For current state}}\}$$

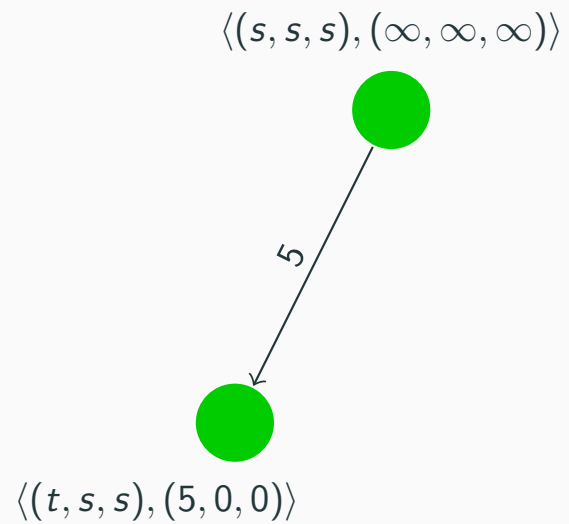


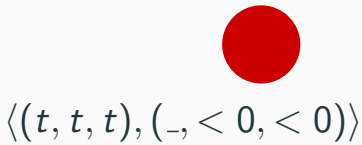


EXPSPACE algorithm for best & worst NE

$$m_i = \min\{\underbrace{m_i - cost_i(\bar{e})}_{\text{From history}}, \underbrace{\min_c \langle val_{i,c} + cost_i(\_, c) - cost_i(\bar{e})\rangle}_{\text{For current state}}\}$$



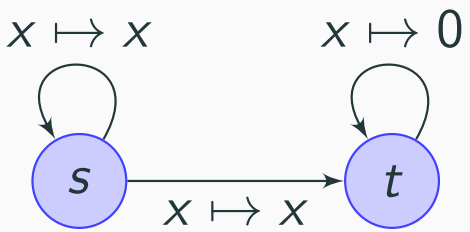




EXPSPACE algorithm for best & worst NE

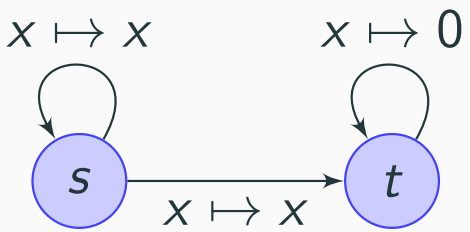
$$m_i = \min\{\underbrace{m_i - cost_i(\bar{e})}_{\text{From history}}, \underbrace{\min_c \langle val_{i,c} + cost_i(\_, c) - cost_i(\bar{e})\rangle}_{\text{For current state}}\}$$

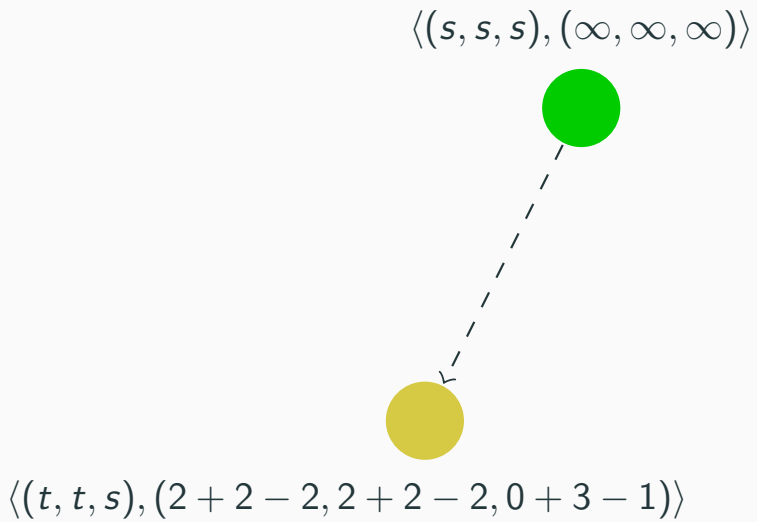
$\langle (s,s,s), (\infty,\infty,\infty)\rangle$



EXPSPACE algorithm for best & worst NE

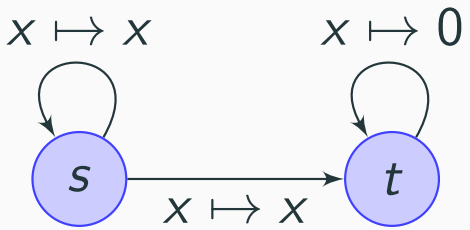
$$m_i = \min\{\underbrace{m_i - cost_i(\bar{e})}_{\text{From history}}, \underbrace{\min_c \langle val_{i,c} + cost_i(\_, c) - cost_i(\bar{e})\rangle}_{\text{For current state}}\}$$

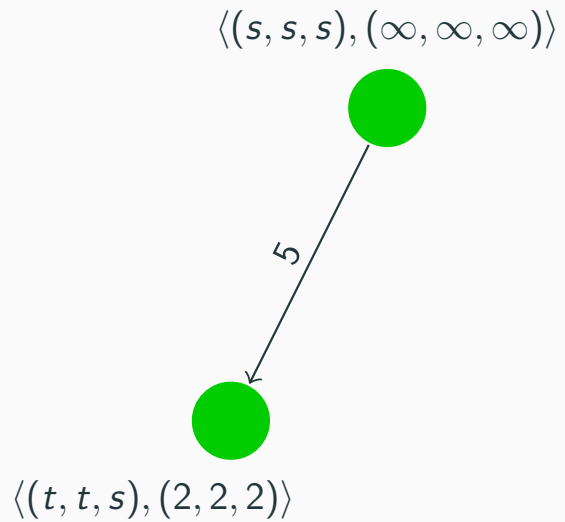




EXPSPACE algorithm for best & worst NE

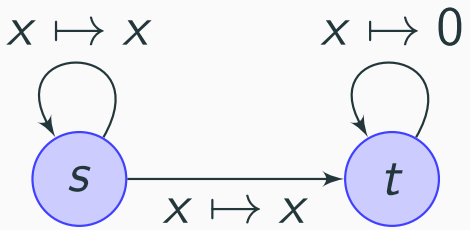
$$m_i = \min\{\underbrace{m_i - cost_i(\bar{e})}_{\text{From history}}, \underbrace{\min_c \langle val_{i,c} + cost_i(\_, c) - cost_i(\bar{e})\rangle}_{\text{For current state}}\}$$

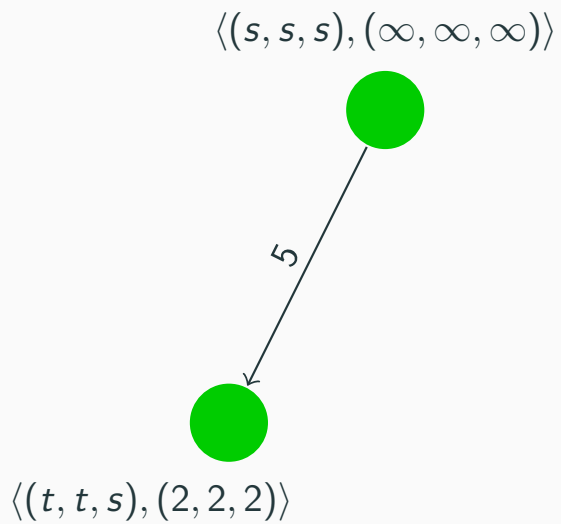


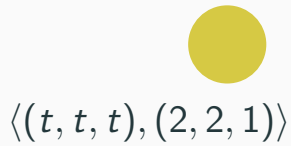


EXPSPACE algorithm for best & worst NE

$$m_i = \min\{\underbrace{m_i - cost_i(\bar{e})}_{\text{From history}}, \underbrace{\min_c \langle val_{i,c} + cost_i(\_, c) - cost_i(\bar{e})\rangle}_{\text{For current state}}\}$$

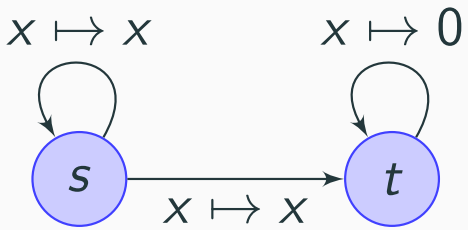


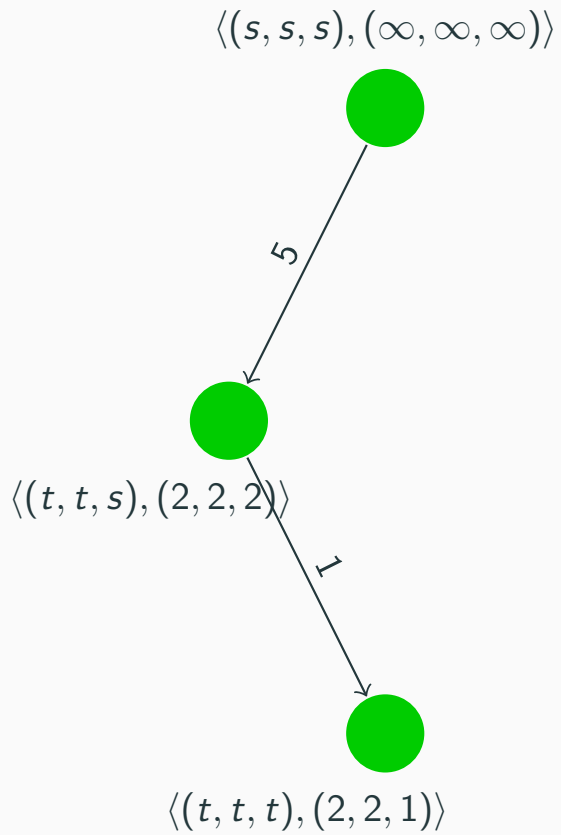




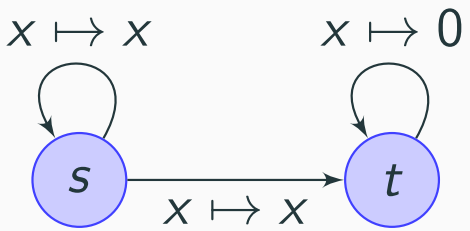
EXPSPACE algorithm for best & worst NE

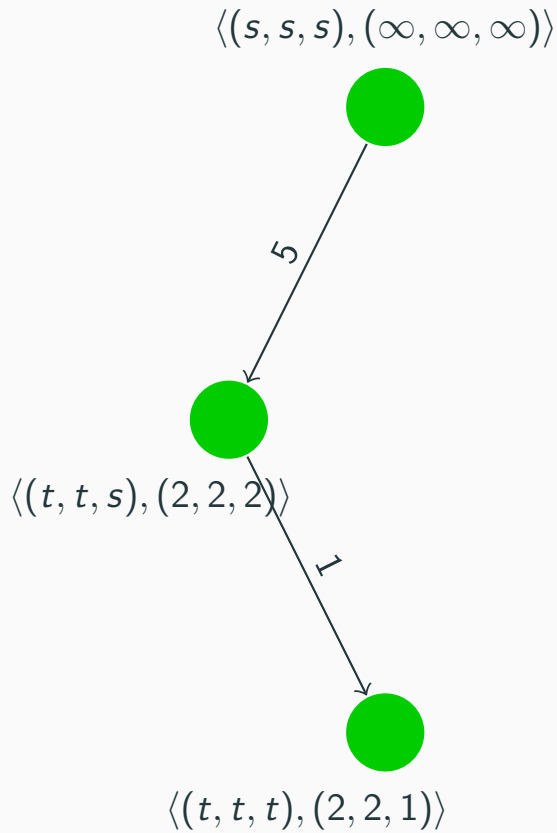
$$m_i = \min\{\underbrace{m_i - cost_i(\bar{e})}_{\text{From history}}, \underbrace{\min_c \langle val_{i,c} + cost_i(\_, c) - cost_i(\bar{e})\rangle}_{\text{For current state}}\}$$





EXPSPACE algorithm for best & worst NE





- Along any path, values are non-increasing, and lower bounded by 0 $\implies$ The graph is acyclic.
- For $\exists$ NE, guess such a path, and verify sum of the weights on the edges $\leq M_{\exists}$.
- For $\forall$ NE, change the sign of the weights, guess such a path with social cost $\leq -M_{\forall}$, and verify it. If we get such a path, the answer of $\forall$ NE is "No".
- The values are upper bounded by $|E| \max_{e \in E} cost_e(k)$, at each step we need polynomial space for storing but need EXPTIME to compute $val_{i,c}$, hence decision problem is in EXPSPACE. This implies we can compute PoA, PoS in 2EXPTIME.

Subgame Perfect Equilibria(SPE)

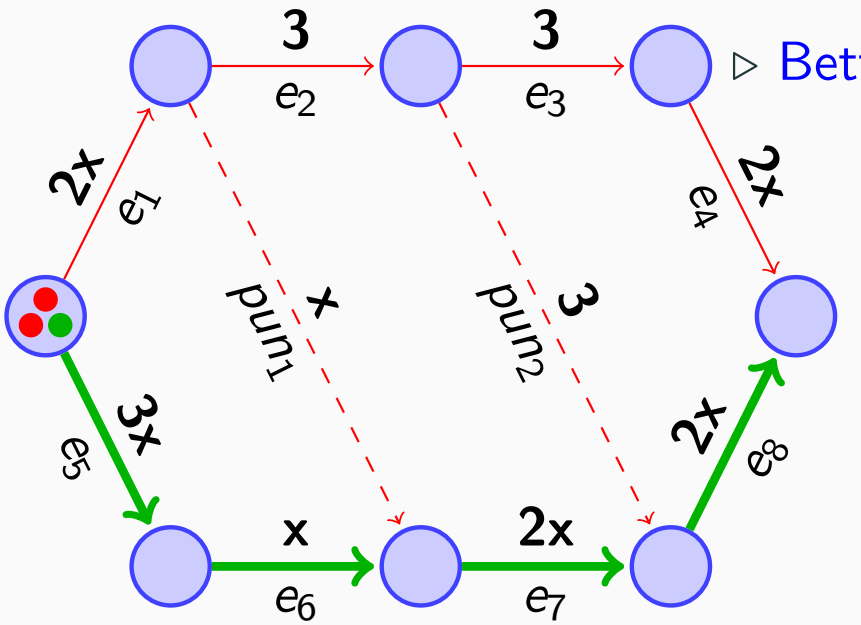
An NE profile $\mathcal{P}$ is an SPE[ab] if for any initialized subgame of the original game, $\mathcal{P}$ works as an NE profile.

[a] Avni, Henzinger, and Kupferman 2016.
[b] Brihaye et al. 2019.

Subgame Perfect Equilibria(SPE)

An NE profile $\mathcal{P}$ is an SPE[ab] if for any initialized subgame of the original game, $\mathcal{P}$ works as an NE profile.

[a] Avni, Henzinger, and Kupferman 2016.
[b] Brihaye et al. 2019.

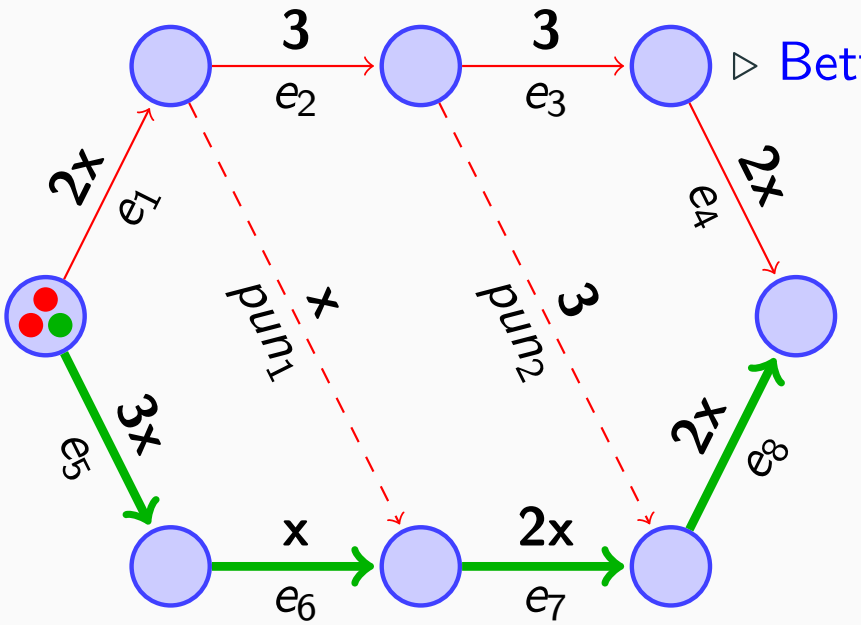


▷ Better Equilibria: Avoids non-credible threats!

Subgame Perfect Equilibria(SPE)

An NE profile $\mathcal{P}$ is an SPE[ab] if for any initialized subgame of the original game, $\mathcal{P}$ works as an NE profile.

[a] Avni, Henzinger, and Kupferman 2016.
[b] Brihaye et al. 2019.



▷ Better Equilibria: Avoids non-credible threats!

▷ Ongoing work: ∃ SPE problem is in EXPSPACE, no decisive conclusion for existence of SPE in the model yet

Conclusion

Recap

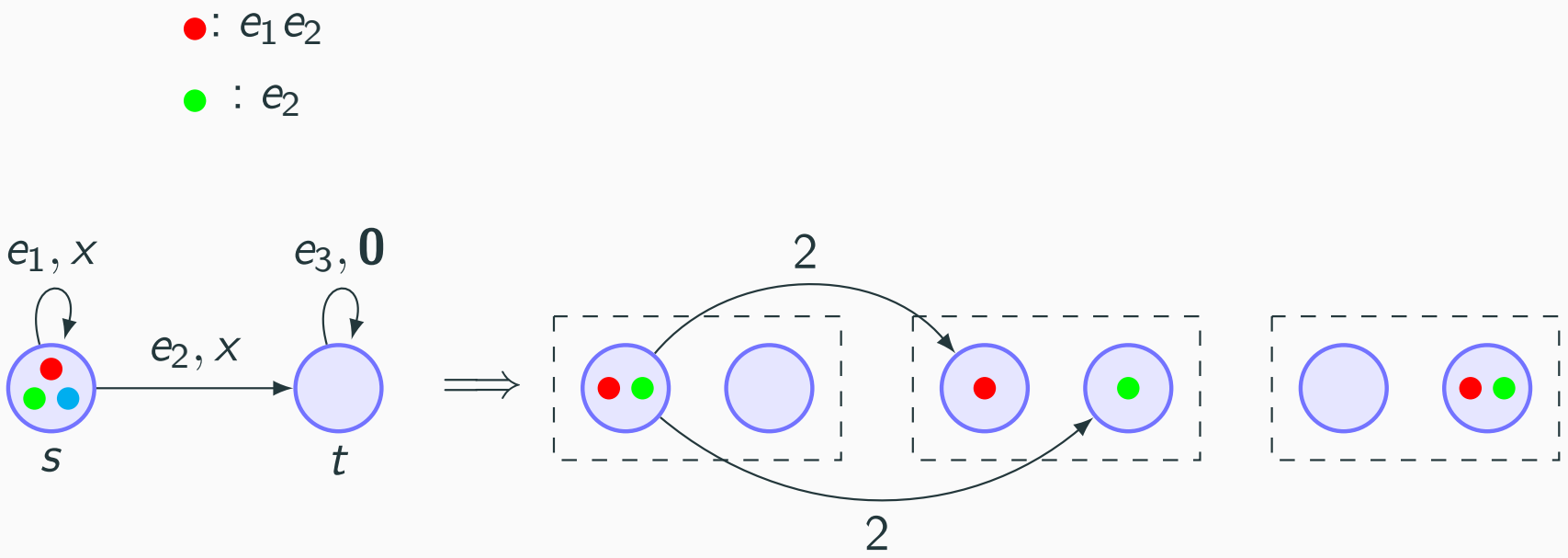
- Defined dynamic network games: with Synchronous cost computation.
- Existence of Nash Equilibrium.
- General strategy NE are better than blind NE.
- Computed cost of SO, best and worst NE. $\rightsquigarrow$ 2-EXPTIME upper bound for the decision problem(s) we proposed.
- EXPSPACE algorithm for $\exists$ SPE problem.

Future Objectives

- Finding *better* lower bound complexity result for the decision problem(s).
- We plan to consider number of players as a parameter of the instance, to study whether we can draw any relation between PoS (resp. PoA), NE with this parameter.
- Investigating **Subgame Perfect Equilibria** (SPE) in depth: specifically we are interested whether we can achieve any decisive results on the existence of SPE in our model
- We have plans to consider objectives other than reachability, like regularity.

Thanks!

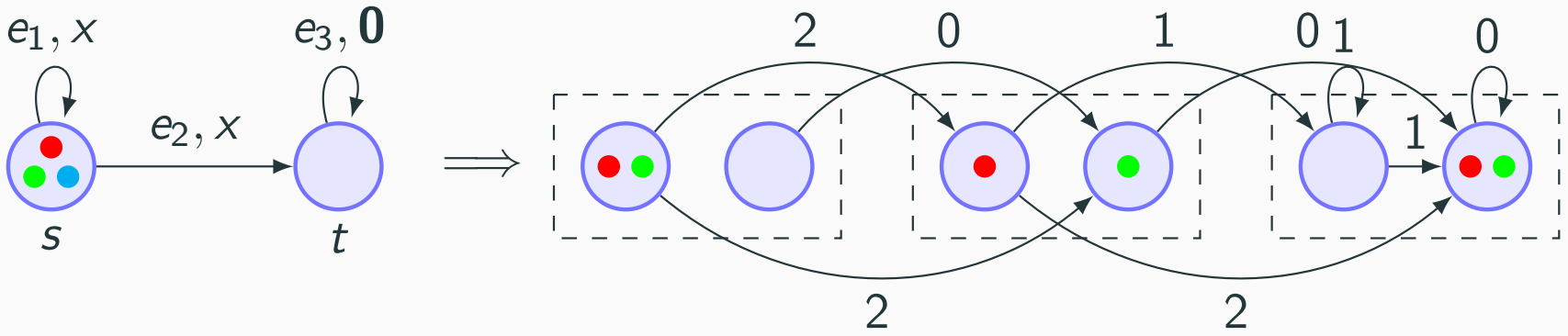
How to solve BR problem?



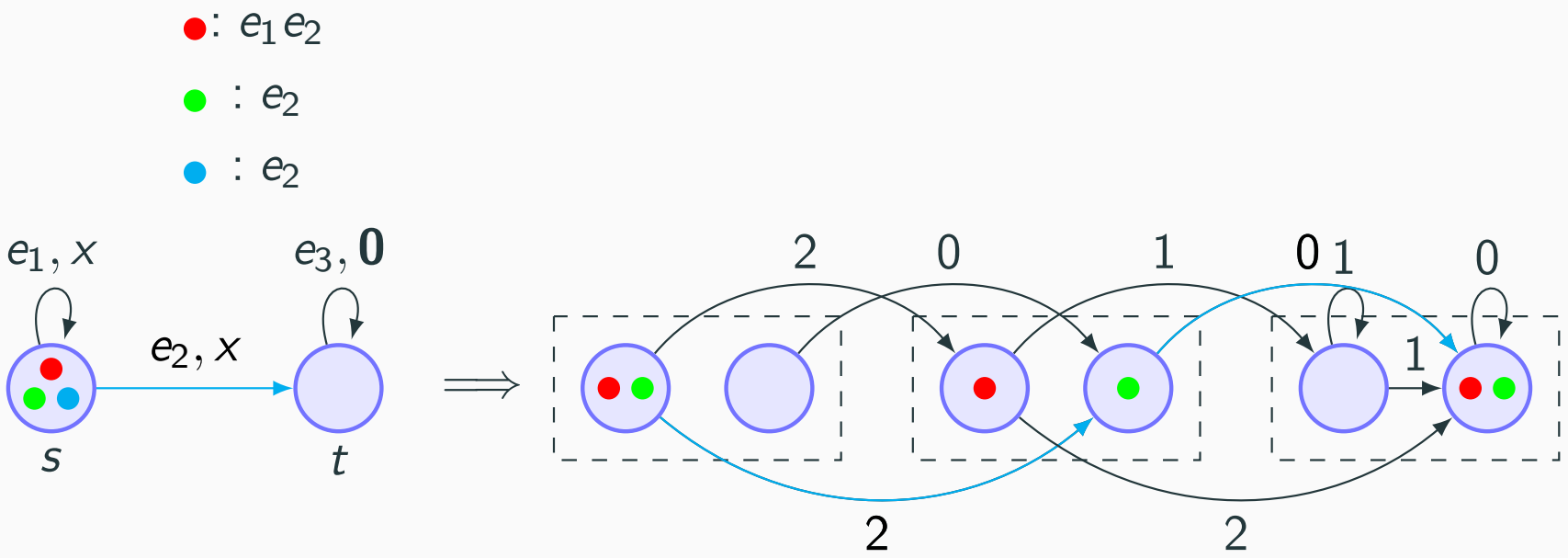
How to solve BR problem?

●: $e_1 e_2$

● : e_2



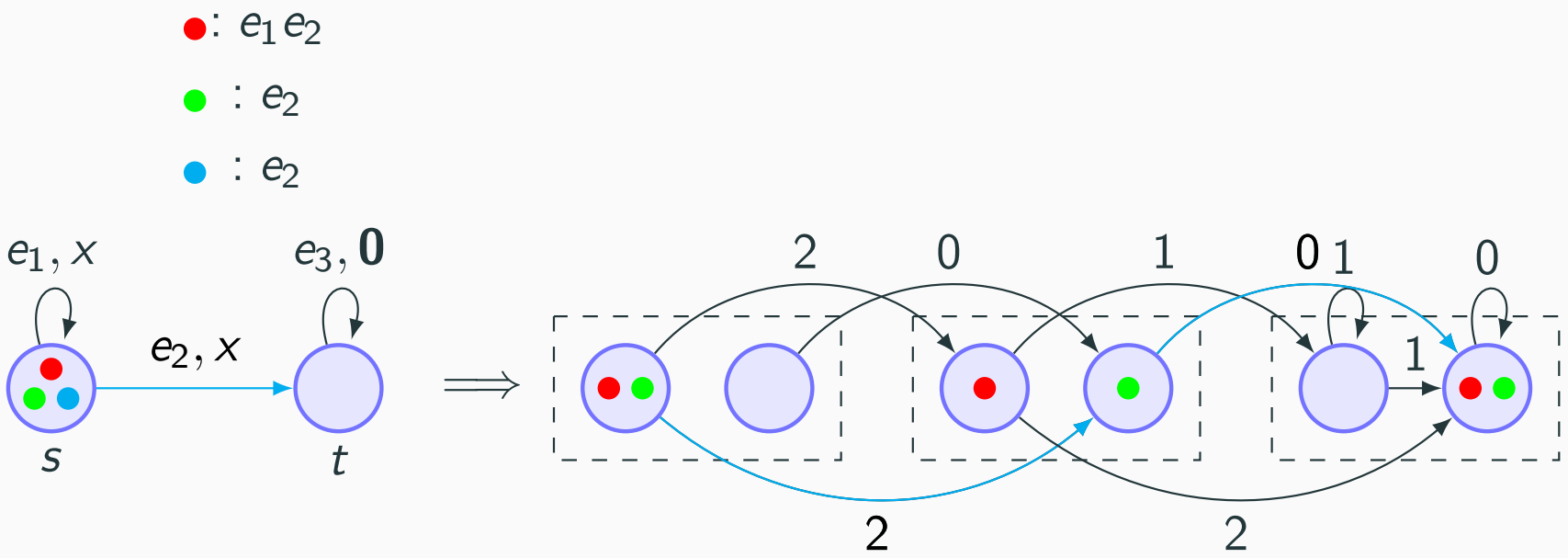
How to solve BR problem?



Reduction is polynomial

Weighted shortest path problem for weighted graphs is in PTIME

How to solve BR problem?

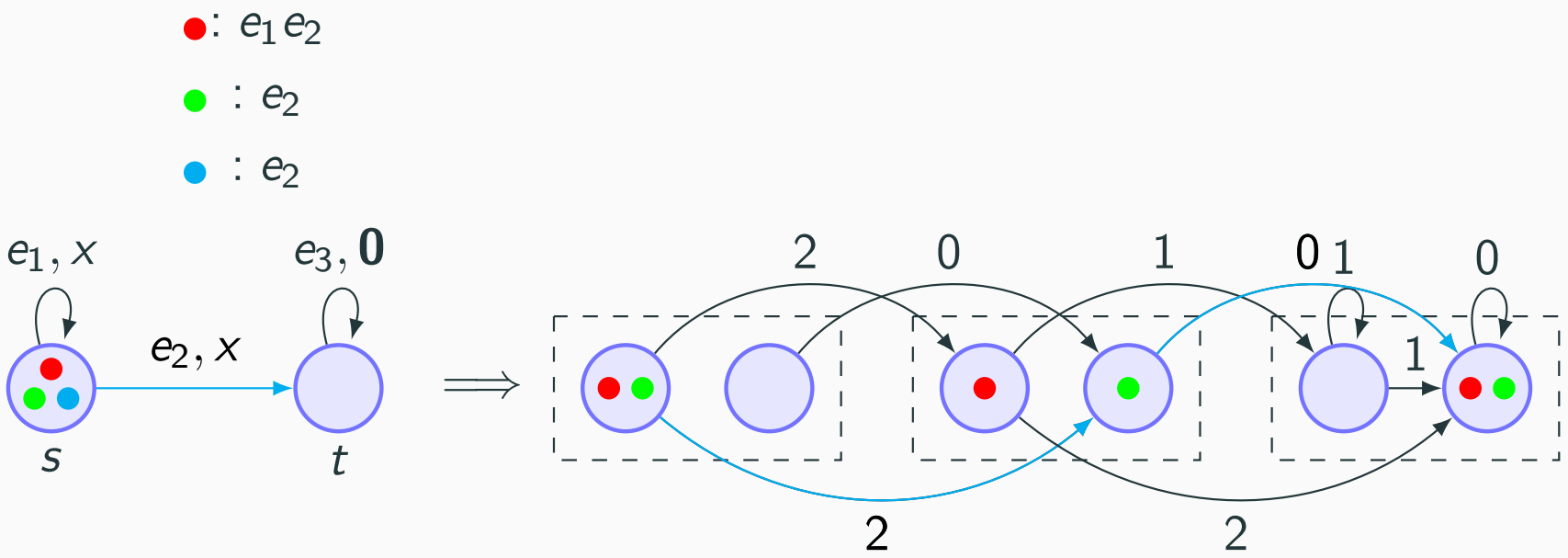


Reduction is polynomial

Weighted shortest path problem for weighted graphs is in PTIME

BR problem for blind strategies is in PTIME

How to solve BR problem?



Reduction is polynomial

Weighted shortest path problem for weighted graphs is in PTIME

BR problem for blind strategies is in PTIME

Remark. BR problem for general strategies is also in PTIME!!

Convergence of iterative procedure

Termination is guaranteed by potential function[4]

Potential function $\Pi : \{\text{Set of blind strategy profiles}\} \to \mathbb{N}$

- decreases at each application of BR algorithm
- lower bounded by 0.

For blind strategies

$$\Pi(\mathcal{P}) = \sum_{j=1}^{N_{\mathcal{P}}} \sum_{e \in E} \sum_{i=1}^{load^{j}_{\mathcal{P},e}} cost_e(i)$$

where $N_{\mathcal{P}}$ is the maximum length of strategy "path"
and *load* is the number of players simultaneously using an edge

[4] D.Monderer and Shapley 1996.

Convergence of iterative procedure

Termination is guaranteed by potential function[4]

Potential function $\Pi : \{\text{Set of blind strategy profiles}\} \to \mathbb{N}$

- decreases at each application of BR algorithm
- lower bounded by 0.

For blind strategies

$$\Pi(\mathcal{P}) = \sum_{j=1}^{N_{\mathcal{P}}} \sum_{e \in E} \sum_{i=1}^{load^{j}_{\mathcal{P},e}} cost_e(i)$$

where $N_{\mathcal{P}}$ is the maximum length of strategy "path"
and *load* is the number of players simultaneously using an edge

This shows existence of NE for blind strategies.

[4] D.Monderer and Shapley 1996.

Convergence of iterative procedure

Termination is guaranteed by potential function[4]

Potential function $\Pi : \{\text{Set of blind strategy profiles}\} \to \mathbb{N}$

- decreases at each application of BR algorithm
- lower bounded by 0.

For blind strategies

$$\Pi(\mathcal{P}) = \sum_{j=1}^{N_{\mathcal{P}}} \sum_{e \in E} \sum_{i=1}^{load^{j}_{\mathcal{P},e}} cost_e(i)$$

where $N_{\mathcal{P}}$ is the maximum length of strategy "path"
and *load* is the number of players simultaneously using an edge

This shows existence of NE for blind strategies.

Remark. Termination for general strategies is unknown

[4] D.Monderer and Shapley 1996.