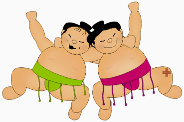


Network Games with Synchronous Cost

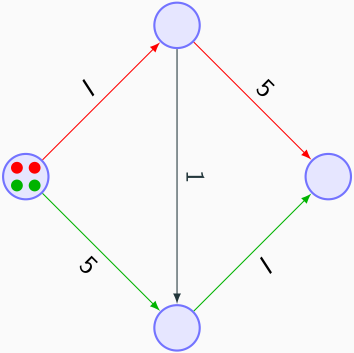
Supervised by Nathalie Bertrand, Nicolas Markey, Ocan Sankur

Suman Sadhukhan

Makushita, September 2019

Introduction

Routing Games

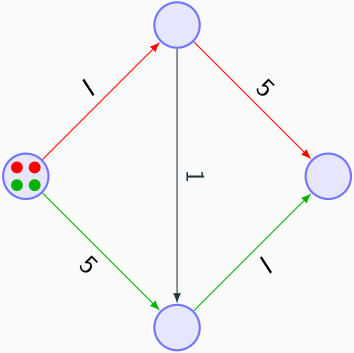


$$cost = 2 + 5 = 7$$

$$cost = 5 + 2 = 7$$

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

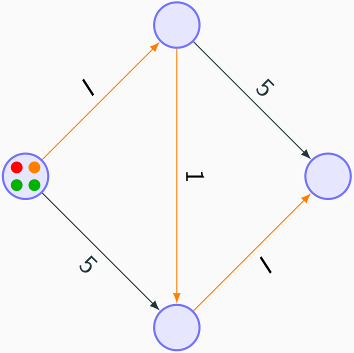
Routing Games



$cost = 7$

$cost = 7$

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



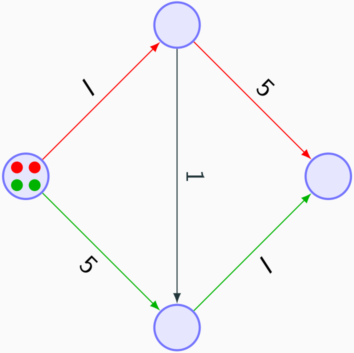
$cost = 7$

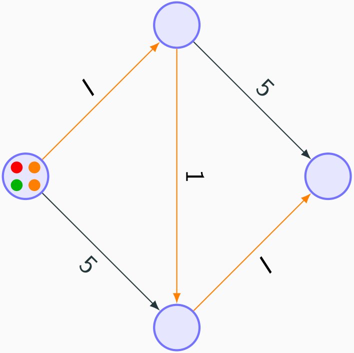
$cost = 5 + 3 = 8$

$cost = 2 + 1 + 3 = 6$

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

Routing Games





$$cost = 7$$

$$cost = 7$$

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

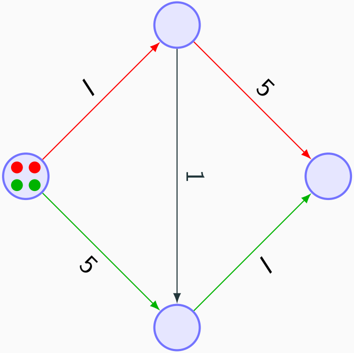
$$cost = 8$$

$$cost = 8$$

$$cost = 3 + 1 + 3 = 7$$

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

Routing Games

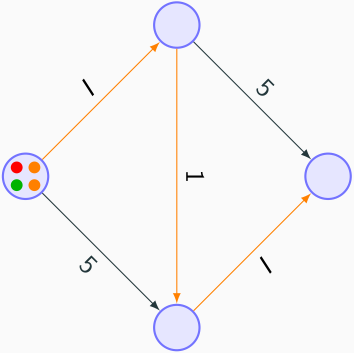


“centralized”

$cost = 7$

$cost = 7$

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



“selfish”

$cost = 8$

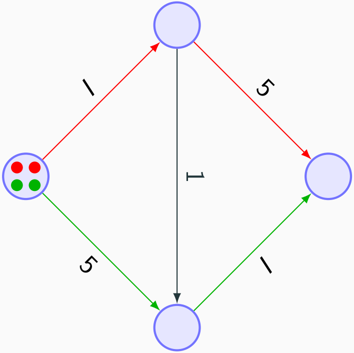
$cost = 8$

$cost = 3 + 1 + 3 = 7$

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

Routing Games

[*Roughgarden* '05]

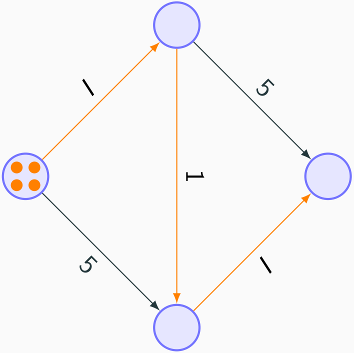


"centralized"

$cost = 7$

$cost = 7$

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



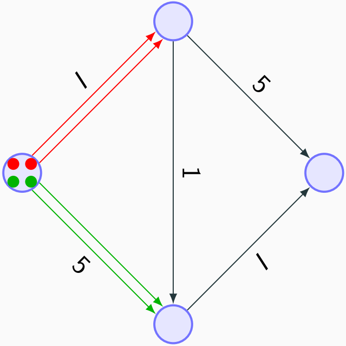
"selfish"

$cost = 4 + 1 + 4 = 9$

$Total\ \ cost = 9 \times 4 = 36$

Where we differ?

- Synchronicity: congestion cost only if players take an edge simultaneously
- Dynamic strategies

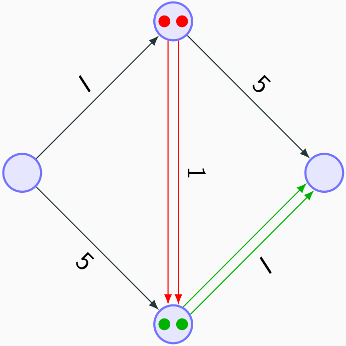


$cost = 2+$

$cost = 5+$

Where we differ?

- Synchronicity: congestion cost only if players take an edge simultaneously
- Dynamic strategies

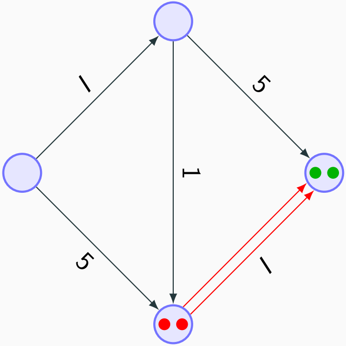


$cost = 2 + 1$

$cost = 5 + 2$

Where we differ?

- Synchronicity: congestion cost only if players take an edge simultaneously
- Dynamic strategies

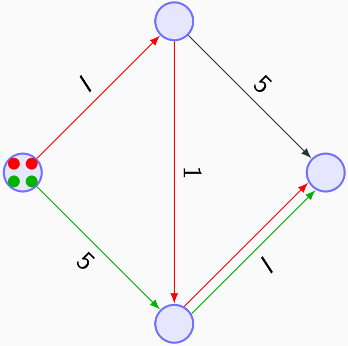


$cost = 2 + 1 + 2 = 5$

$cost = 5 + 2 = 7$

Where we differ?

- Synchronicity: congestion cost only if players take an edge simultaneously
- Dynamic strategies



$\color{red}{cost = 2 + 1 + 2 = 5,\ old\ cost = 7}$

$\color{green}{cost = 5 + 2 = 7,\ old\ cost = 9}$

Our model

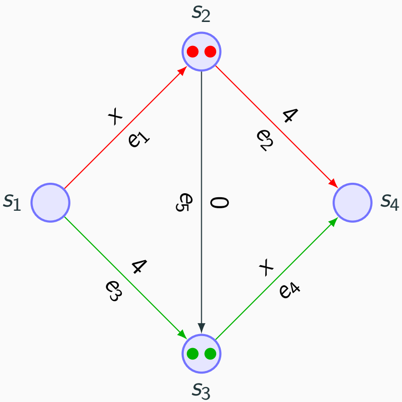
Network Games with Synchronicity (SNG):

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

$cost_e : \mathbb{N} \to \mathbb{N}$, depends on no. of players taking the edge simultaneously

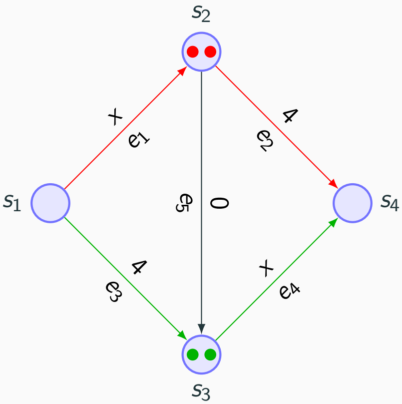
s, t : source and target vertices

Concepts



- Configuration $c = (s_2, s_2, s_3, s_3) \in \mathcal{C}$
- Strategy profile $\mathcal{P} = (e_1 e_2, e_1 e_2, e_3 e_4, e_3 e_4)$
- Outcome: sequence of configurations induced by strategy profile
- Cost of a player: $cost_i = \sum_{e \in path_i} cost_e$

Concepts



- Configuration $c = (s_2, s_2, s_3, s_3) \in \mathcal{C}$
- Strategy profile $\mathcal{P} = (e_1 e_2, e_1 e_2, e_3 e_4, e_3 e_4)$
- Outcome: sequence of configurations induced by strategy profile
- Cost of a player: $cost_i = \sum_{e \in path_i} cost_e$

Strategy classes

- *Blind strategies* : Players only observe history length, $\sigma_i^{bl} : \mathbb{N} \to E$.
- *Local strategies* : Players see vertex sharing players along the history
- *General strategies*: $\sigma_i^g : \mathcal{C}^+ \to E$.

Price of Anarchy, Price of Stability... [*Roughgarden* '05]

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.

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

Price of Anarchy (PoA) $= \frac{\text{Total cost of "worst" NE}}{\text{Total cost of SO}}$

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

Price of Anarchy, Price of Stability... [*Roughgarden* '05]

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) $= \dfrac{\text{Total cost of "worst" NE}}{\text{Total cost of SO}}$

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

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"

$$\textbf{Price of Anarchy (PoA)} = \frac{\text{Total cost of ``worst'' NE}}{\text{Total cost of SO}}$$

$$\textbf{Price of Stability (PoS)} = \frac{\text{Total cost of ``best'' NE}}{\text{Total cost of SO}}$$

Questions:

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

Results

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*.

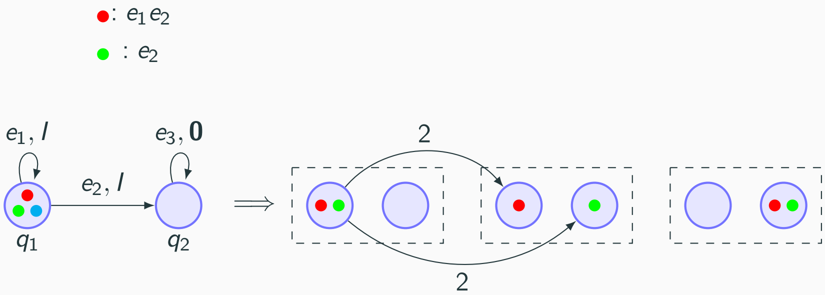
Existence and Computation of NE: Blind Strategies

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?

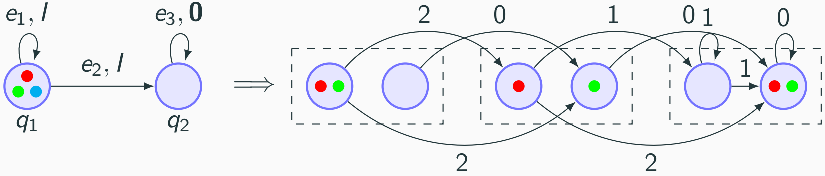
How to solve BR problem?



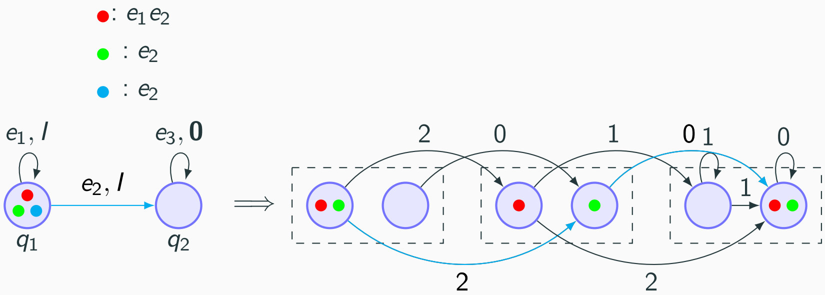
How to solve BR problem?

: $e_1 e_2$

: e_2



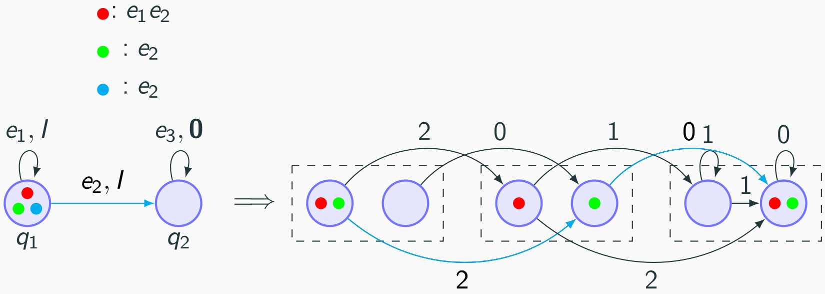
How to solve BR problem?



Reduction is polynomial

Weighted shortest path problem for weighted graphs is in PTIME

How to solve BR problem?

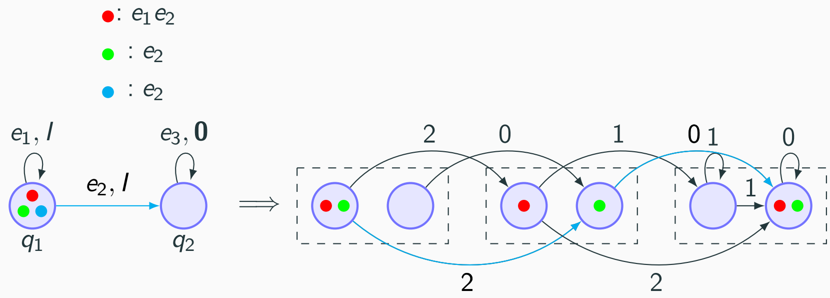


Reduction is polynomial

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

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

Convergence of iterative procedure

Termination is guaranteed by potential function

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.

Convergence of iterative procedure

Termination is guaranteed by potential function

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

Existence and computation of NE: General Strategies

Claim

Blind strategy NE are general strategy NE.

Existence and computation of NE: General Strategies

Claim
Blind strategy NE are general strategy NE.

That shows existence of general strategy NE.

Existence and computation of NE: General Strategies

Claim

Blind strategy NE are general strategy NE.

That shows existence of general strategy NE.

Are there more?

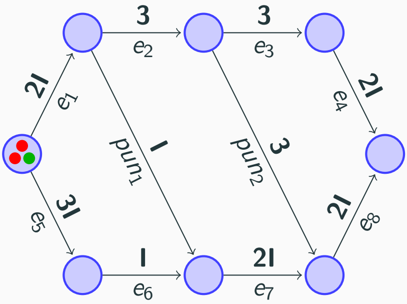
Existence and computation of NE: General Strategies

Claim

Blind strategy NE are general strategy NE.

That shows existence of general strategy NE.

Are there more?

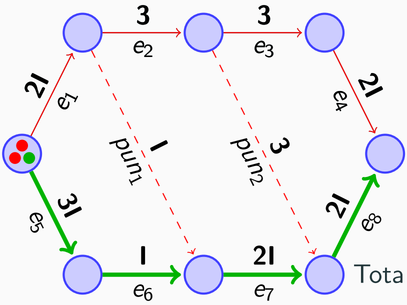


Claim

Blind strategy NE are general strategy NE.

That shows existence of general strategy NE.

Are there more?



- (red) cost $= 14$
- (green) cost $= 8$

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

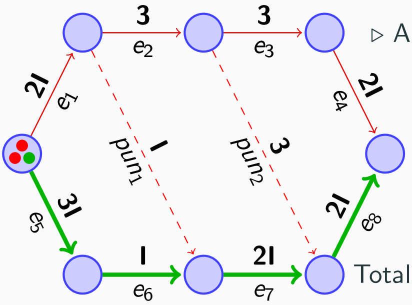
Existence and computation of NE: General Strategies

Claim

Blind strategy NE are general strategy NE.

That shows existence of general strategy NE.

Are there more?



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

● cost = 14
● cost = 8

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

Characterization of NE : General Strategy

A play ρ is called *well-fit* if

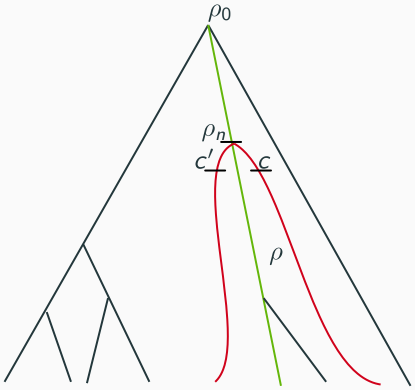
$$\forall i \in [k], \forall n > 0, \forall c \in Succ_\rho(i,n)$$
$$cost_i(\rho_{\geq n}) \leq v_{i,c} + cost_i(\rho_n, c)$$

$v_{i,c}$ = cost that player-i can not avoid if the game starts from confiuration $c = \min_{\sigma_i} \max_{\sigma_{-i}} cost_i^c((\sigma_i, \sigma_{-i}))$
$cost_i(\rho_n, c)$ = cost that player-i pays for transitioning from configuration ρ_n to c in one step

Characterization of NE : General Strategy

A play ρ is called *well-fit* if

$$\forall i \in [k], \forall n > 0, \forall c \in Succ_{\rho}(i, n)$$
$$cost_i(\rho_{\geq n}) \leq v_{i,c} + cost_i(\rho_n, c)$$



Characterization of NE : General Strategy

A play ρ is called *well-fit* if

$$\forall i \in [k], \forall n > 0, \forall c \in Succ_\rho(i, n)$$
$$cost_i(\rho_{\geq n}) \leq v_{i,c} + cost_i(\rho_n, c)$$

A play $\rho \in \mathcal{C}^+$ of a SNG $\mathcal{S}$ is the outcome of a Nash Equilibrium if and only if it is a well-fit play.

Characterization of NE : General Strategy

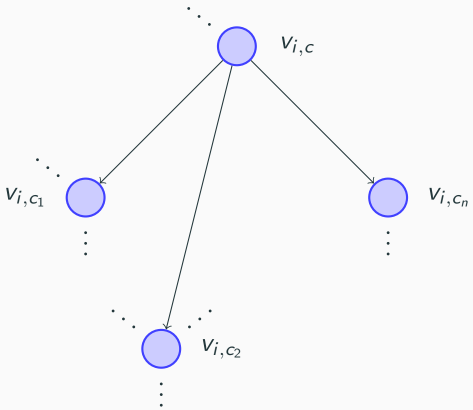
A play ρ is called *well-fit* if

$$\forall i \in [k], \forall n > 0, \forall c \in Succ_\rho(i, n)$$
$$cost_i(\rho_{\geq n}) \leq v_{i,c} + cost_i(\rho_n, c)$$

A play $\rho \in \mathcal{C}^+$ of a SNG $\mathcal{S}$ is the outcome of a Nash Equilibrium if and only if it is a well-fit play.

Searching for NE might not be necessary: maybe we can directly deal with outcomes to compute PoA, PoS?

Computing $v_{i,c}$



$$v_{i,c} = \min_{c_j[i]} \max_{c_j[-i]} (v_{i,c_j} + cost_i(c, c_j))$$

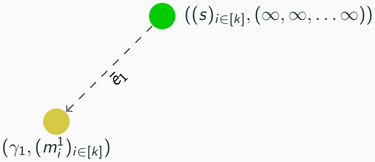
We can compute $v_{i,c}$ by a fixed-point algorithm by initializing suitably over the configuration graph.

Identifying outcomes of best NE

$((s)_{i \in [k]}, (\infty, \infty, \ldots \infty))$

- $(\min_c v_{i,c} + cost_i(\rho_n, c)) - cost_i(\rho_{\geq n}) \geq 0 \;\forall n \forall c \in succ_\rho(i, n) \forall i \in [k]$

Identifying outcomes of best NE

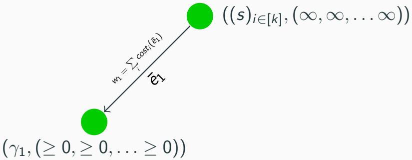


- $(\min_c v_{i,c} + cost_i(\rho_n, c)) - cost_i(\rho_{\geq n}) \geq 0 \;\forall n \forall c \in succ_\rho(i, n) \forall i \in [k]$
- There is an edge from $(\gamma, (m_i)_{i\in[k]})$ to $(\gamma', (m_i')_{i\in[k]})$ if $\forall i \in [k]$

  $$m_i' = \min(m_i - cost_i(\gamma, \gamma'), ctr^i_{\gamma,\gamma'})$$

  where $ctr^i_{\gamma,\gamma'} = (\min_{\gamma''} v_{i,\gamma''} + cost_i(\gamma, \gamma'')) - cost_i(\gamma, \gamma')$

Identifying outcomes of best NE

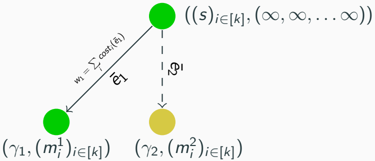


- $(\min_c v_{i,c} + cost_i(\rho_n, c)) - cost_i(\rho_{\geq n}) \geq 0 \ \forall n \forall c \in succ_\rho(i, n) \forall i \in [k]$
- There is an edge from $(\gamma, (m_i)_{i\in[k]})$ to $(\gamma', (m'_i)_{i\in[k]})$ if $\forall i \in [k]$

  $$m'_i = \min(m_i - cost_i(\gamma, \gamma'), ctr^i_{\gamma,\gamma'})$$

  where $ctr^i_{\gamma,\gamma'} = (\min_{\gamma''} v_{i,\gamma''} + cost_i(\gamma, \gamma'')) - cost_i(\gamma, \gamma')$

Identifying outcomes of best NE

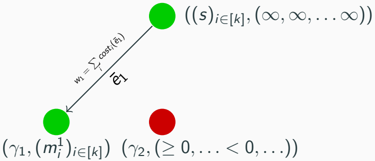


- $(\min_c v_{i,c} + cost_i(\rho_n, c)) - cost_i(\rho_{\geq n}) \geq 0 \;\forall n \forall c \in succ_\rho(i, n) \forall i \in [k]$
- There is an edge from $(\gamma, (m_i)_{i\in[k]})$ to $(\gamma', (m_i')_{i\in[k]})$ if $\forall i \in [k]$

  $$m_i' = \min(m_i - cost_i(\gamma, \gamma'), ctr^i_{\gamma,\gamma'})$$

  where
  $ctr^i_{\gamma,\gamma'} = (\min_{\gamma''} v_{i,\gamma''} + cost_i(\gamma, \gamma'')) - cost_i(\gamma, \gamma')$

Identifying outcomes of best NE

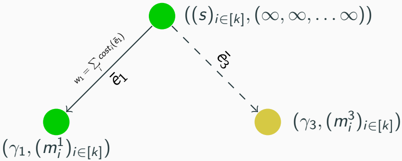


- $(\min_c v_{i,c} + cost_i(\rho_n, c)) - cost_i(\rho_{\geq n}) \geq 0 \ \forall n \forall c \in succ_\rho(i, n) \forall i \in [k]$
- There is an edge from $(\gamma, (m_i)_{i\in[k]})$ to $(\gamma', (m_i')_{i\in[k]})$ if $\forall i \in [k]$

  $$m_i' = \min(m_i - cost_i(\gamma, \gamma'), ctr^i_{\gamma,\gamma'})$$

  where
  $ctr^i_{\gamma,\gamma'} = (\min_{\gamma''} v_{i,\gamma''} + cost_i(\gamma, \gamma'')) - cost_i(\gamma, \gamma')$

Identifying outcomes of best NE

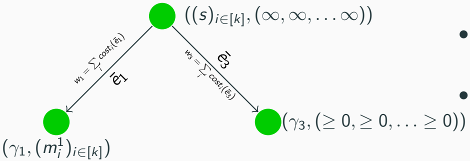


- $(\min_c v_{i,c} + cost_i(\rho_n, c)) - cost_i(\rho_{\geq n}) \geq 0 \; \forall n \forall c \in succ_\rho(i, n) \forall i \in [k]$
- There is an edge from $(\gamma, (m_i)_{i\in[k]})$ to $(\gamma', (m_i')_{i\in[k]})$ if $\forall i \in [k]$

  $$m_i' = \min(m_i - cost_i(\gamma, \gamma'), ctr^i_{\gamma,\gamma'})$$

  where
  $ctr^i_{\gamma,\gamma'} = (\min_{\gamma''} v_{i,\gamma''} + cost_i(\gamma, \gamma'')) - cost_i(\gamma, \gamma')$

Identifying outcomes of best NE

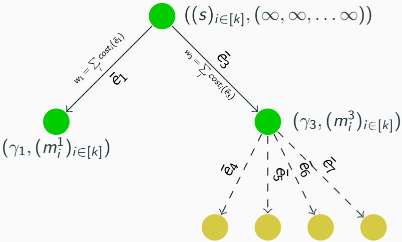


- $(\min_c v_{i,c} + cost_i(\rho_n, c)) - cost_i(\rho_{\geq n}) \geq 0 \; \forall n \forall c \in succ_\rho(i, n) \forall i \in [k]$
- There is an edge from $(\gamma, (m_i)_{i\in[k]})$ to $(\gamma', (m'_i)_{i\in[k]})$ if $\forall i \in [k]$

  $$m'_i = \min(m_i - cost_i(\gamma, \gamma'), ctr^i_{\gamma,\gamma'})$$

  where
  $ctr^i_{\gamma,\gamma'} = (\min_{\gamma''} v_{i,\gamma''} + cost_i(\gamma, \gamma'')) - cost_i(\gamma, \gamma')$

Identifying outcomes of best NE

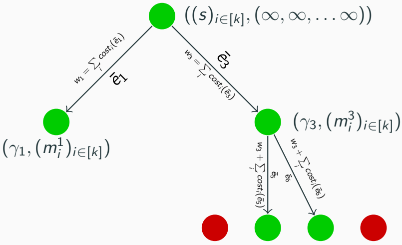


- $(\min_c v_{i,c} + cost_i(\rho_n, c)) - cost_i(\rho_{\geq n}) \geq 0 \; \forall n \forall c \in succ_\rho(i,n) \forall i \in [k]$
- There is an edge from $(\gamma, (m_i)_{i\in[k]})$ to $(\gamma', (m_i')_{i\in[k]})$ if $\forall i \in [k]$

  $$m_i' = \min(m_i - cost_i(\gamma,\gamma'), ctr^i_{\gamma,\gamma'})$$

  where $ctr^i_{\gamma,\gamma'} = (\min_{\gamma''} v_{i,\gamma''} + cost_i(\gamma,\gamma'')) - cost_i(\gamma,\gamma')$
- Total cost is the priority

Identifying outcomes of best NE

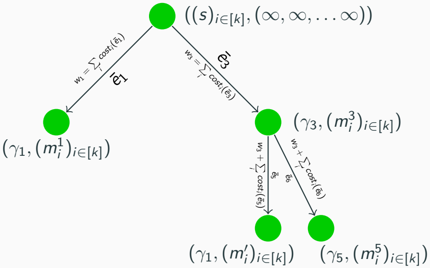


- $(\min_c v_{i,c} + cost_i(\rho_n, c)) - cost_i(\rho_{\geq n}) \geq 0 \; \forall n \forall c \in succ_\rho(i, n) \forall i \in [k]$
- There is an edge from $(\gamma, (m_i)_{i\in[k]})$ to $(\gamma', (m'_i)_{i\in[k]})$ if $\forall i \in [k]$

  $$m'_i = \min(m_i - cost_i(\gamma, \gamma'), ctr^i_{\gamma,\gamma'})$$

  where $ctr^i_{\gamma,\gamma'} = (\min_{\gamma''} v_{i,\gamma''} + cost_i(\gamma, \gamma'')) - cost_i(\gamma, \gamma')$
- Total cost is the priority

Identifying outcomes of best NE

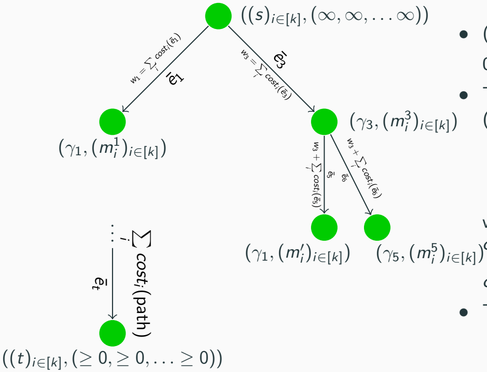


- $(\min_c v_{i,c} + cost_i(\rho_n, c)) - cost_i(\rho_{\geq n}) \geq 0 \; \forall n \forall c \in succ_\rho(i, n) \forall i \in [k]$
- There is an edge from $(\gamma, (m_i)_{i\in[k]})$ to $(\gamma', (m_i')_{i\in[k]})$ if $\forall i \in [k]$

  $$m_i' = \min(m_i - cost_i(\gamma, \gamma'), ctr^i_{\gamma,\gamma'})$$

  where $ctr^i_{\gamma,\gamma'} = (\min_{\gamma''} v_{i,\gamma''} + cost_i(\gamma, \gamma'')) - cost_i(\gamma, \gamma')$
- Total cost is the priority

Identifying outcomes of best NE

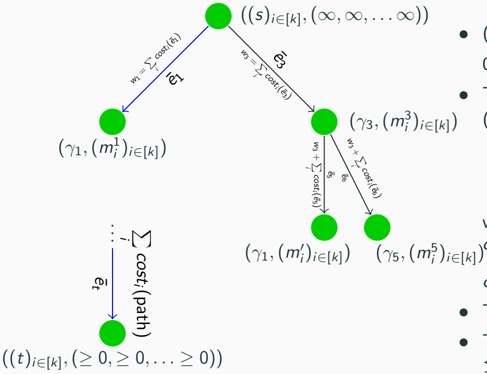


- $(\min_c v_{i,c} + cost_i(\rho_n, c)) - cost_i(\rho_{\geq n}) \geq 0 \ \forall n \forall c \in succ_\rho(i, n) \forall i \in [k]$
- There is an edge from $(\gamma, (m_i)_{i\in[k]})$ to $(\gamma', (m_i')_{i\in[k]})$ if $\forall i \in [k]$

$$m_i' = \min(m_i - cost_i(\gamma, \gamma'), ctr^i_{\gamma,\gamma'})$$

  where
  $ctr^i_{\gamma,\gamma'} = (\min_{\gamma''} v_{i,\gamma''} + cost_i(\gamma, \gamma'')) - cost_i(\gamma, \gamma')$
- Total cost is the priority

Identifying outcomes of best NE

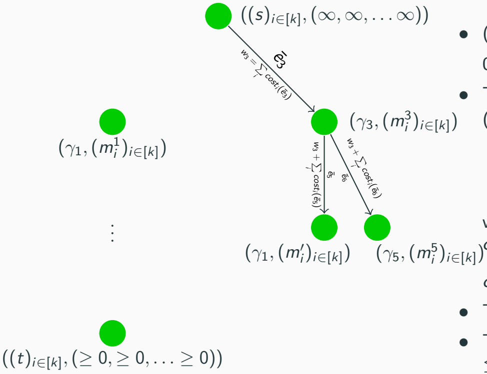


- $(\min_c v_{i,c} + cost_i(\rho_n, c)) - cost_i(\rho_{\geq n}) \geq 0 \; \forall n \forall c \in succ_\rho(i, n) \forall i \in [k]$
- There is an edge from $(\gamma, (m_i)_{i\in[k]})$ to $(\gamma', (m_i')_{i\in[k]})$ if $\forall i \in [k]$

$$m_i' = \min(m_i - cost_i(\gamma, \gamma'), ctr^i_{\gamma,\gamma'})$$

  where $ctr^i_{\gamma,\gamma'} = (\min_{\gamma''} v_{i,\gamma''} + cost_i(\gamma, \gamma'')) - cost_i(\gamma, \gamma')$
- Total cost is the priority
- Tree size $(\mathcal{T})$ $\leq |\mathcal{C}| \times (\text{max. value } m_i \text{ can attend})^k$
- For Dijkstra-like algorithm: $\mathcal{O}(|E_\mathcal{T}| + |V_\mathcal{T}| \log |V_\mathcal{T}|)$

Identifying outcomes of best NE



- $(\min_c v_{i,c} + cost_i(\rho_n, c)) - cost_i(\rho_{\geq n}) \geq 0 \; \forall n \forall c \in succ_\rho(i, n) \forall i \in [k]$
- There is an edge from $(\gamma, (m_i)_{i\in[k]})$ to $(\gamma', (m_i')_{i\in[k]})$ if $\forall i \in [k]$

  $$m_i' = \min(m_i - cost_i(\gamma, \gamma'), ctr^i_{\gamma,\gamma'})$$

  where $ctr^i_{\gamma,\gamma'} = (\min_{\gamma''} v_{i,\gamma''} + cost_i(\gamma, \gamma'')) - cost_i(\gamma, \gamma')$
- Total cost is the priority
- Tree size $(\mathcal{T})$ $\leq |\mathcal{C}| \times (\text{max. value } m_i \text{ can attend})^k$
- For Dijkstra-like algorithm: $\mathcal{O}(|E_\mathcal{T}| + |V_\mathcal{T}| \log |V_\mathcal{T}|)$
- EXPTIME in number of players to compute total cost of a best NE

Outcomes for worst NE and SO

- Outcome of an SO can be identified by similar algorithm just by omitting the checking for NE constraints at each node.
- Outcome of an worst NE can be identified by similar algorithm by putting negative cost at each edge, then finding the shortest weighted path.

Outcomes for worst NE and SO

- Outcome of an SO can be identified by similar algorithm just by omitting the checking for NE constraints at each node.
- Outcome of an worst NE can be identified by similar algorithm by putting negative cost at each edge, then finding the shortest weighted path. This works because there is no negative cycle!

Outcomes for worst NE and SO

- Outcome of an SO can be identified by similar algorithm just by omitting the checking for NE constraints at each node.
- Outcome of an worst NE can be identified by similar algorithm by putting negative cost at each edge, then finding the shortest weighted path. This works because there is no negative cycle!

Can compute PoS and PoA now!

Other Equilibria: Subgame Perfect Equilibria(SPE)

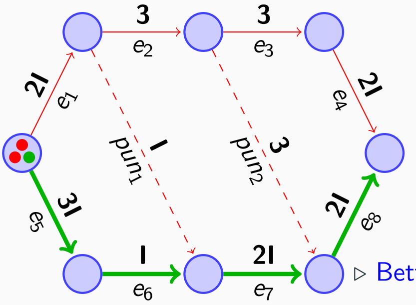
An NE profile $\mathcal{P}$ is an SPE[a][b] if for any initialized subgame of the original game, $\mathcal{P}$ works as an NE profile.

[a] Avni, Henzinger, and Kupferman, "Dynamic Resource Allocation Games".
[b] Brihaye et al., "The Complexity of Subgame Perfect Equilibria in Quantitative Reachability Games".

Other Equilibria: Subgame Perfect Equilibria(SPE)

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

[a] Avni, Henzinger, and Kupferman, “Dynamic Resource Allocation Games”.
[b] Brihaye et al., “The Complexity of Subgame Perfect Equilibria in Quantitative Reachability Games”.

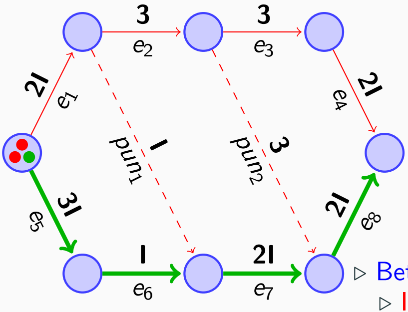


▷ Better Equilibria: Avoids non-credible threats!

Other Equilibria: Subgame Perfect Equilibria(SPE)

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

[a] Avni, Henzinger, and Kupferman, "Dynamic Resource Allocation Games".
[b] Brihaye et al., "The Complexity of Subgame Perfect Equilibria in Quantitative Reachability Games".



▷ Better Equilibria: Avoids non-credible threats!
▷ In our settings, we are looking for results.

Conclusion

Future Objectives

- Finding lower bound complexity result for finding PoS (resp. PoA).
- 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.
- We can consider objectives other than reachability, like regularity.

Thanks!