Reaching Your Goal Optimally by Playing at Random with no Memory

Julie Parreaux[1]
Benjamin Monmege[2] Pierre-Alain Reynier[2]

[1]Ecole Normale Supérieure de Rennes, France

[2]Aix-Marseille Université, France

CONCUR 2020

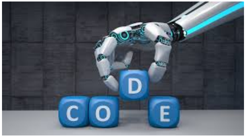
Motivation : game theory for synthesis

Classic approach

Check the correctness of a system

Game theory

Interaction between two antagonistic agents : environment and controller



Code synthesis

Correct by construction : synthesis of controller

Different sorts of games

Qualitative games

Reach or avoid some (sequences of) states

Quantitative games

- Consider quantitative parameters : energy consumption...
- Compare distinct strategies

Different sorts of games

Qualitative games

Reach or avoid some (sequences of) states

Quantitative games

- Consider quantitative parameters : energy consumption...
- Compare distinct strategies

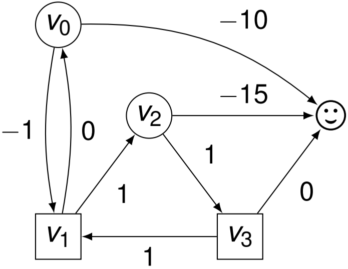
Shortest-Path games

- Combination of a qualitative with a quantitative objective
- Reach a target with a minimum cost

Shortest Path Game

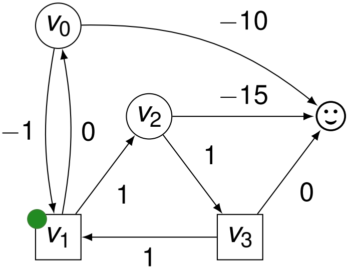
□ Adam ○ Eve

☺ target (T)



Shortest Path Game

□ Adam ○ Eve

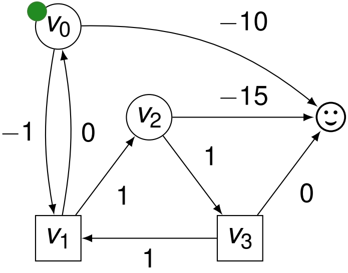


How to play?

Move a token along an edge

$$\pi = v_1$$

Shortest Path Game

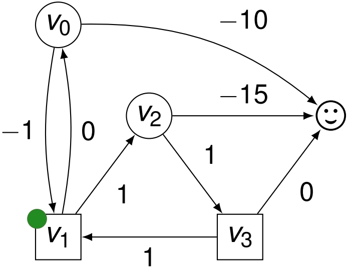


How to play?

Move a token along an edge

$$\pi = v_1 v_0$$

Shortest Path Game

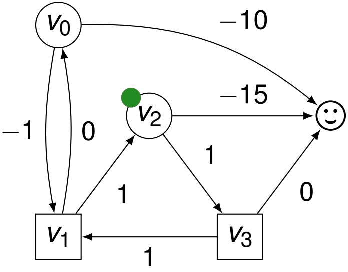


How to play?

Move a token along an edge

$$\pi = v_1 v_0 v_1$$

Shortest Path Game



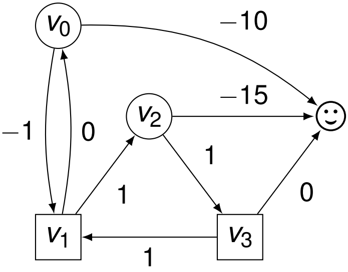
How to play?

Move a token along an edge

$$\pi = v_1 v_0 v_1 v_2$$

Shortest Path Game

□ Adam ○ Eve



Play

Infinite path or reach the target

$$\pi = (v_i)_i \in V^\omega \qquad \pi = (v_i)_i ☺$$

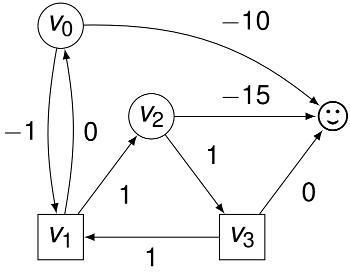
How to play?

Move a token along an edge

$$\pi = v_1 v_0 v_1 v_2 v_3 ☺$$

Shortest Path Game

□ Adam ○ Eve



Play

Infinite path or reach the target

$$\pi = (v_i)_i \in V^\omega \qquad \pi = (v_i)_i ☺$$

How to play?

Move a token along an edge

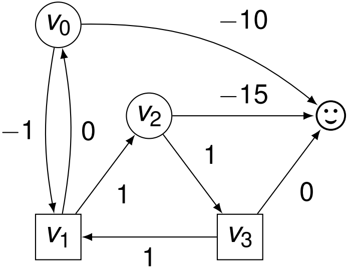
$$\pi = v_1 v_0 v_1 v_2 v_3 ☺$$

Shortest Path payoff of a play π

$$\mathbf{SP}(\pi) = \begin{cases} \sum_{i=0}^{n-1} w((\pi_i, \pi_{i+1})) & \text{if } \exists n \text{ (the smallest) s.t. } \pi_n = ☺ \\ +\infty & \text{if } \pi \text{ does not reach } ☺ \end{cases}$$

Shortest Path Game

Adam Eve



Play

Infinite path or reach the target

$$\pi = (v_i)_i \in V^\omega \qquad \pi = (v_i)_i ☺$$

How to play?

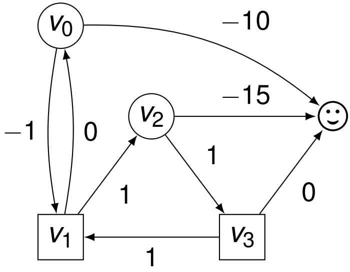
Move a token along an edge

$$\pi = v_1 v_0 v_1 v_2 v_3 ☺$$
$$\textbf{SP}(\pi) = 0 + (-1) + 1 + 1 + 0 = 1$$

Shortest Path payoff of a play π

$$\textbf{SP}(\pi) = \begin{cases} \sum_{i=0}^{n-1} w((\pi_i, \pi_{i+1})) & \text{if } \exists n \text{ (the smallest) s.t. } \pi_n = ☺ \\ +\infty & \text{if } \pi \text{ does not reach } ☺ \end{cases}$$

Shortest Path Game



Objectives

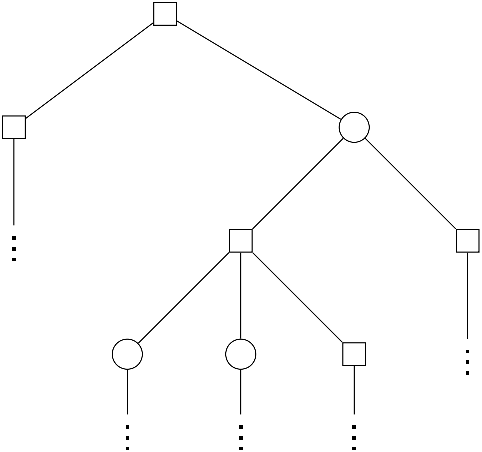
Eve maximise the payoff

Adam minimise the payoff

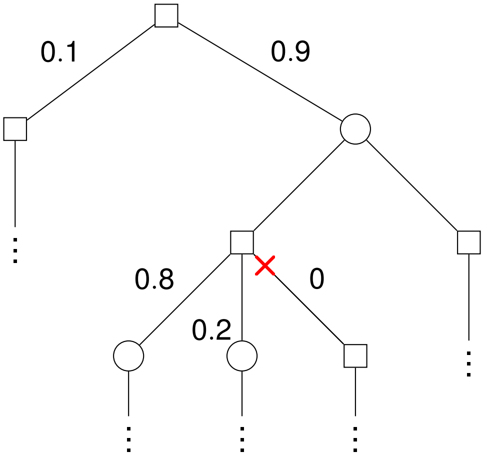
Shortest Path payoff of a play π

$$\mathbf{SP}(\pi) = \begin{cases} \sum_{i=0}^{n-1} w((\pi_i, \pi_{i+1})) & \text{if } \exists n \text{ (the smallest) s.t. } \pi_n = ☺ \\ +\infty & \text{if } \pi \text{ does not reach } ☺ \end{cases}$$

Strategies for Adam



Strategies for Adam



A strategy

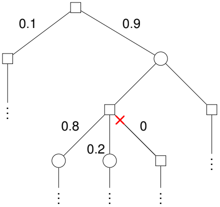
$\sigma : V^* V_{Adam} \to \Delta(V)$

Trading Memory for Randomness, K. Chatterjee, L. Alfaro and T. Henzinger, 2004, QEST

Strategies for Adam

Infinite memory

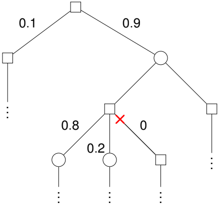
$\sigma : V^* V_{Adam} \to \Delta(V)$



Strategies for Adam

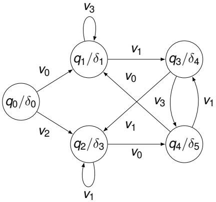
Infinite memory

$\sigma : V^* V_{Adam} \to \Delta(V)$



Finite memory

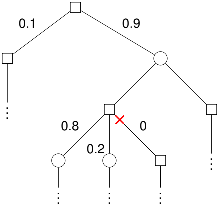
Moore machine



Strategies for Adam

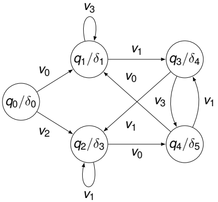
Infinite memory

$\sigma : V^* V_{Adam} \to \Delta(V)$



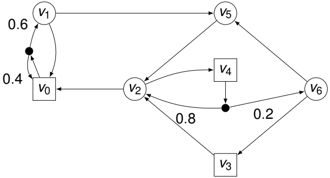
Finite memory

Moore machine



Memoryless

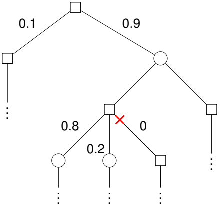
$\sigma : V_{Adam} \to \Delta(V)$



Strategies for Adam

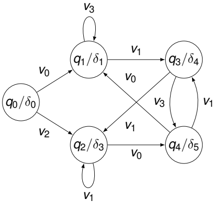
Infinite memory

$\sigma : V^* V_{Adam} \to \Delta(V)$



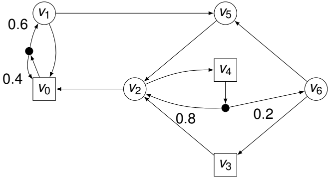
Finite memory

Moore machine



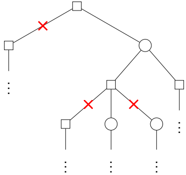
Memoryless

$\sigma : V_{Adam} \to \Delta(V)$



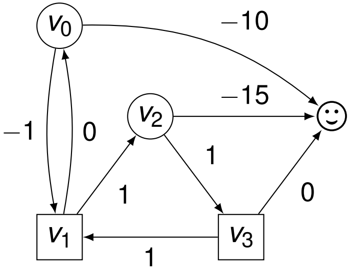
Deterministic

$\sigma : V^* V_{Adam} \to V$



Deterministic Strategies

$\boxed{\sigma}$ Adam $\textcircled{\tau}$ Eve



Value

$$\overline{\mathsf{dVal}}(v) = \inf_{\sigma} \underbrace{\sup_{\tau} \mathbf{SP}(\mathsf{Play}(v, \sigma, \tau))}_{\mathsf{dVal}^{\sigma}(v)}$$

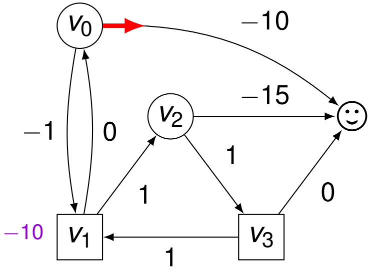
Deterministic strategy

$$\sigma : V^* \times V_{\mathsf{Adam}} \to V$$

Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Deterministic Strategies

$\boxed{\sigma}$ Adam $\bigcirc\!\!\!\!\tau$ Eve



Estimate $\overline{\text{dVal}}(v_1)$

Eve chooses ☺ in v_0 : $\rightsquigarrow -10$

Value

$$\overline{\text{dVal}}(v) = \inf_{\sigma} \underbrace{\sup_{\tau} \mathbf{SP}(\text{Play}(v, \sigma, \tau))}_{\text{dVal}^{\sigma}(v)}$$

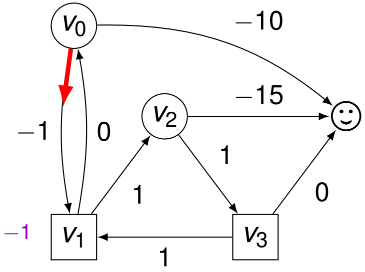
Deterministic strategy

$$\sigma : V^* \times V_{\text{Adam}} \to V$$

Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Deterministic Strategies

σ Adam $\quad$ τ Eve



Estimate $\overline{\mathsf{dVal}}(v_1)$

Eve chooses ☺ in v_0 : $\rightsquigarrow$ -10

Eve chooses v_1 in v_0 : $\dashrightarrow$ -1

Value

$$\overline{\mathsf{dVal}}(v) = \inf_{\sigma} \underbrace{\sup_{\tau} \mathbf{SP}(\mathsf{Play}(v, \sigma, \tau))}_{\mathsf{dVal}^{\sigma}(v)}$$

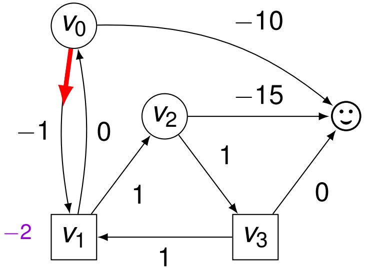
Deterministic strategy

$$\sigma : V^* \times V_{\mathsf{Adam}} \to V$$

Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Deterministic Strategies

σ Adam τ Eve



Estimate $\overline{\text{dVal}}(v_1)$

Eve chooses ☺ in v_0 : $\rightsquigarrow -10$

Eve chooses v_1 in v_0 : $\dashrightarrow -2$

Value

$$\overline{\text{dVal}}(v) = \inf_\sigma \underbrace{\sup_\tau \mathbf{SP}(\text{Play}(v,\sigma,\tau))}_{\text{dVal}^\sigma(v)}$$

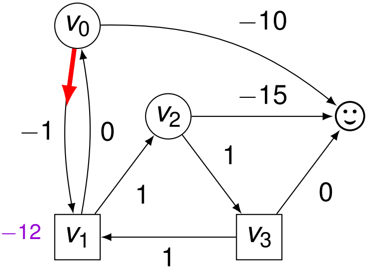
Deterministic strategy

$$\sigma : V^* \times V_{\text{Adam}} \to V$$

Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Deterministic Strategies

$\boxed{\sigma}$ Adam $\quad$ $\textcircled{\tau}$ Eve



Estimate $\overline{\mathrm{dVal}}(v_1)$

Eve chooses ☺ in v_0 : $\rightsquigarrow$ -10

Eve chooses v_1 in v_0 : $\dashrightarrow$ -12

Value

$$\overline{\mathrm{dVal}}(v) = \inf_{\sigma} \underbrace{\sup_{\tau} \mathbf{SP}(\mathrm{Play}(v, \sigma, \tau))}_{\mathrm{dVal}^{\sigma}(v)}$$

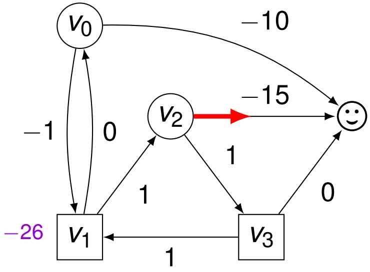
Deterministic strategy

$$\sigma : V^* \times V_{\mathrm{Adam}} \to V$$

Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Deterministic Strategies

$\boxed{\sigma}$ Adam (τ) Eve



Estimate $\overline{\mathsf{dVal}}(v_1)$

Eve chooses ☺ in v_0 : $\rightsquigarrow -10$

Eve chooses v_1 in v_0 : $\dashrightarrow -12$

Eve chooses ☺ in v_2 : $\rightsquigarrow -26$

Value

$$\overline{\mathsf{dVal}}(v) = \inf_{\sigma} \underbrace{\sup_{\tau} \mathbf{SP}(\mathsf{Play}(v, \sigma, \tau))}_{\mathsf{dVal}^{\sigma}(v)}$$

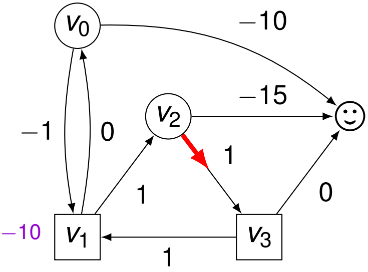
Deterministic strategy

$$\sigma : V^* \times V_{\mathsf{Adam}} \to V$$

Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Deterministic Strategies

$\boxed{\sigma}$ Adam $\quad$ (τ) Eve



Estimate $\overline{\mathrm{dVal}}(v_1)$

Eve chooses ☺ in v_0 : $\rightsquigarrow -10$

Eve chooses v_1 in v_0 : $\dashrightarrow -12$

Eve chooses ☺ in v_2 : $\rightsquigarrow -26$

Eve chooses v_3 in v_2 : $\rightsquigarrow -10$

Value

$$\overline{\mathrm{dVal}}(v) = \inf_{\sigma} \underbrace{\sup_{\tau} \mathbf{SP}(\mathrm{Play}(v, \sigma, \tau))}_{\mathrm{dVal}^{\sigma}(v)}$$

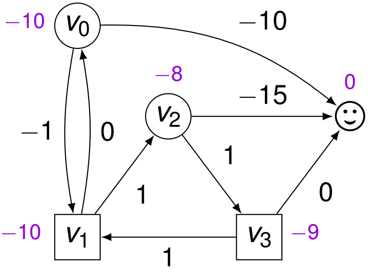
Deterministic strategy

$$\sigma : V^* \times V_{\mathrm{Adam}} \to V$$

Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Deterministic Strategies

$\boxed{\sigma}$ Adam $\textcircled{\tau}$ Eve



Estimate $\overline{\text{dVal}}(v_1)$

Eve chooses ☺ in v_0 : $\rightsquigarrow$ -10

Eve chooses v_1 in v_0 : $\dashrightarrow$ -12

Eve chooses ☺ in v_2 : $\rightsquigarrow$ -26

Eve chooses v_3 in v_2 : $\rightsquigarrow$ -10

Value

$$\overline{\text{dVal}}(v) = \inf_{\sigma} \underbrace{\sup_{\tau} \textbf{SP}(\text{Play}(v, \sigma, \tau))}_{\text{dVal}^{\sigma}(v)}$$

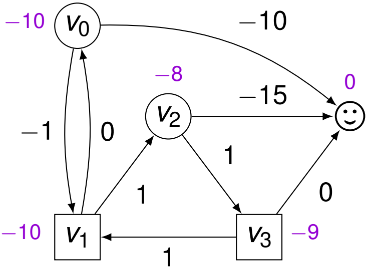
Deterministic strategy

$$\sigma : V^* \times V_{\text{Adam}} \rightarrow V$$

Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Deterministic Strategies

σ Adam τ Eve



Determinacy

$$\mathsf{dVal}(v) = \overline{\mathsf{dVal}}(v) = \underline{\mathsf{dVal}}(v)$$

Value

$$\overline{\mathsf{dVal}}(v) = \inf_{\sigma} \underbrace{\sup_{\tau} \mathbf{SP}(\mathsf{Play}(v, \sigma, \tau))}_{\mathsf{dVal}^{\sigma}(v)}$$

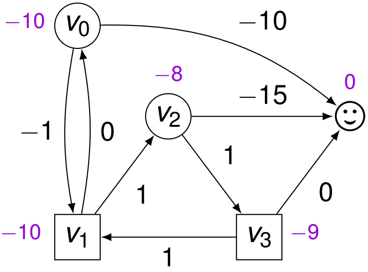
Deterministic strategy

$$\sigma : V^* \times V_{\mathsf{Adam}} \to V$$

Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Deterministic Strategies

$\boxed{\sigma}$ Adam $\quad$ (τ) Eve



Optimal strategy

$$\mathrm{dVal}^{\sigma^*}(v) \leqslant \mathrm{dVal}(v)$$

Value

$$\overline{\mathrm{dVal}}(v) = \inf_{\sigma} \underbrace{\sup_{\tau} \mathbf{SP}(\mathrm{Play}(v, \sigma, \tau))}_{\mathrm{dVal}^{\sigma}(v)}$$

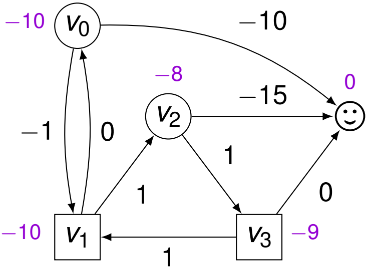
Deterministic strategy

$$\sigma : V^* \times V_{\mathrm{Adam}} \to V$$

Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Deterministic Strategies

$\boxed{\sigma}$ Adam $\quad$ $\textcircled{\tau}$ Eve



Optimal strategy

$$\mathrm{dVal}^{\sigma^*}(v) \leqslant \mathrm{dVal}(v)$$

Optimal strategy for Adam

An optimal strategy for Adam may require finite memory.

Value

$$\overline{\mathrm{dVal}}(v) = \inf_{\sigma} \underbrace{\sup_{\tau} \mathbf{SP}(\mathrm{Play}(v,\sigma,\tau))}_{\mathrm{dVal}^{\sigma}(v)}$$

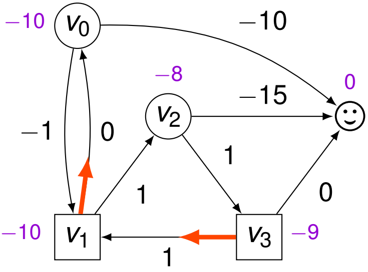
Deterministic strategy

$$\sigma : V^* \times V_{\mathrm{Adam}} \to V$$

Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Deterministic Strategies

$\boxed{\sigma}$ Adam $\quad$ $\textcircled{\tau}$ Eve



Optimal strategy

$\mathrm{dVal}^{\sigma^*}(v) \leqslant \mathrm{dVal}(v)$

Optimal strategy for Adam

The switching strategy:

- σ_1 : reach negative cycle

Value

$$\overline{\mathrm{dVal}}(v) = \inf_{\sigma} \underbrace{\sup_{\tau} \mathbf{SP}(\mathrm{Play}(v, \sigma, \tau))}_{\mathrm{dVal}^{\sigma}(v)}$$

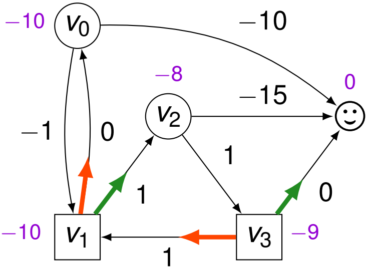
Deterministic strategy

$\sigma : V^* \times V_{\mathrm{Adam}} \to V$

Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Deterministic Strategies

$\boxed{\sigma}$ Adam $\quad (\tau)$ Eve



Optimal strategy

$\mathrm{dVal}^{\sigma^*}(v) \leqslant \mathrm{dVal}(v)$

Optimal strategy for Adam

The switching strategy:

- σ_1 : reach negative cycle
- σ_2 : reach ☺

Value

$$\overline{\mathrm{dVal}}(v) = \inf_{\sigma} \underbrace{\sup_{\tau} \mathbf{SP}(\mathrm{Play}(v, \sigma, \tau))}_{\mathrm{dVal}^{\sigma}(v)}$$

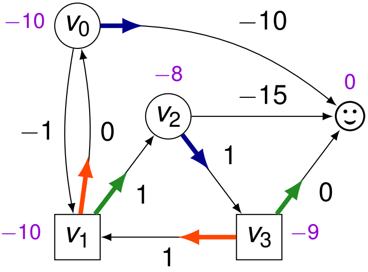
Deterministic strategy

$\sigma : V^* \times V_{\mathrm{Adam}} \to V$

Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Deterministic Strategies

$\boxed{\sigma}$ Adam $\textcircled{\tau}$ Eve



Optimal strategy

$\mathrm{dVal}^{\sigma^*}(v) \leqslant \mathrm{dVal}(v)$

Optimal strategy for Eve

Eve has a memoryless optimal strategy.

Value

$$\overline{\mathrm{dVal}}(v) = \inf_{\sigma} \underbrace{\sup_{\tau} \mathbf{SP}(\mathrm{Play}(v, \sigma, \tau))}_{\mathrm{dVal}^{\sigma}(v)}$$

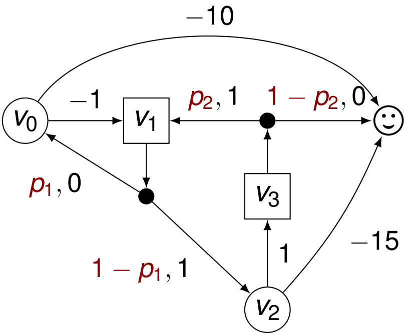
Deterministic strategy

$\sigma : V^* \times V_{\mathrm{Adam}} \to V$

Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Memoryless strategies

σ Adam τ Eve

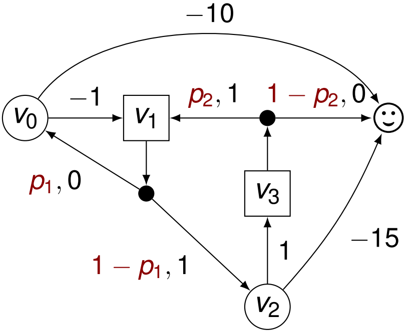


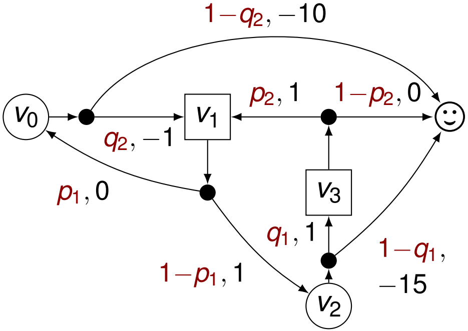
Memoryless strategy

$\sigma : V_{\text{Adam}} \to \Delta(V)$

Memoryless strategies

σ Adam τ Eve





Value

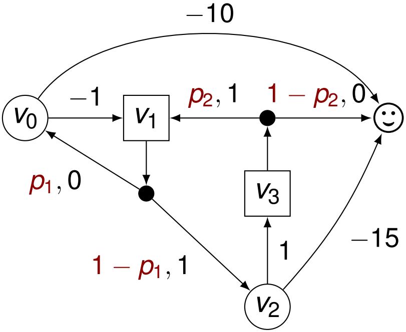
$$\overline{\mathrm{mVal}}(v) = \inf_{\sigma} \underbrace{\sup_{\tau} \mathbb{E}^{\sigma,\tau}(\mathbf{SP})}_{\mathrm{mVal}^{\sigma}(v)}$$

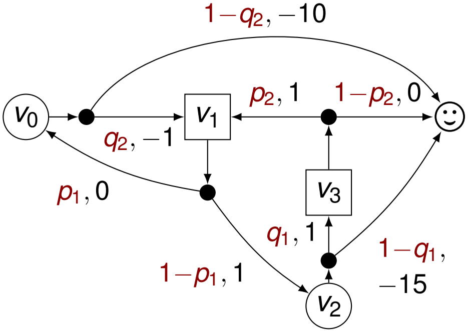
Memoryless strategy

$\sigma : V_{\text{Adam}} \to \Delta(V)$

Memoryless strategies

σ Adam τ Eve





Value

$$\overline{\text{mVal}}(v) = \inf_{\sigma} \underbrace{\sup_{\tau} \mathbb{E}^{\sigma,\tau}(\textbf{SP})}_{\text{mVal}^{\sigma}(v)}$$

Memoryless strategy

$$\sigma : V_{\text{Adam}} \to \Delta(V)$$

ε-optimal strategy

$$\text{mVal}^{\sigma^*}(v) \leqslant \overline{\text{mVal}}(v) + \varepsilon$$

Contributions

Main theorem

For all shortest-path games and vertices v, $\mathrm{dVal}(v) = \overline{\mathrm{mVal}}(v)$.

Contributions

Main theorem

For all shortest-path games and vertices v, $\mathrm{dVal}(v) = \overline{\mathrm{mVal}}(v)$.

Optimality proposition

1. We can characterize and test in polynomial time the existence of an optimal memoryless strategy.
2. Adam has an optimal (randomised) memoryless strategy if and only if Adam has an optimal deterministic memoryless strategy.

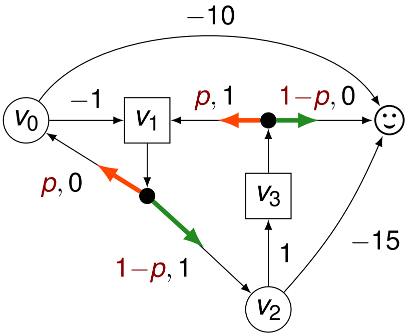
Memoryless simulate deterministic

Claim

For all v, there exists p such that $\mathrm{mVal}^{\sigma_p}(v) \leqslant \mathrm{dVal}(v)$.

Memoryless simulate deterministic

$\boxed{\sigma}$ Adam $\quad$ τ Eve



Claim

For all v, there exists p such that $\mathrm{mVal}^{\sigma_p}(v) \leqslant \mathrm{dVal}(v)$.

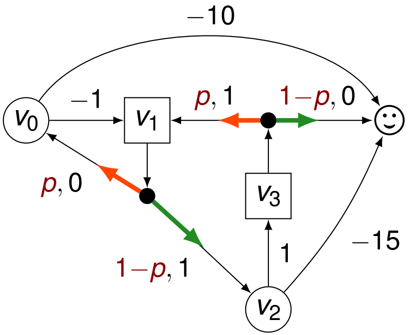
Strategy σ_p

Let $\langle \sigma_1, \sigma_2 \rangle$ be an optimal switching strategy, for all $p \in (0, 1)$,

$$\sigma_p = p \times \sigma_1 + (1 - p) \times \sigma_2$$

Memoryless simulate deterministic

σ Adam τ Eve



Claim

For all v, there exists p such that $\mathrm{mVal}^{\sigma_p}(v) \leqslant \mathrm{dVal}(v)$.

Properties of σ_p

- For all τ, $\mathbb{P}^{\sigma_p,\tau}(\diamond☺) = 1$

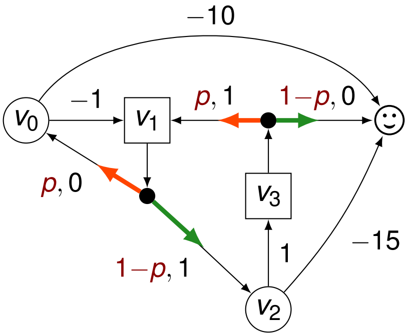
Strategy σ_p

Let $\langle \sigma_1, \sigma_2 \rangle$ be an optimal switching strategy, for all $p \in (0, 1)$,

$$\sigma_p = p \times \sigma_1 + (1-p) \times \sigma_2$$

Memoryless simulate deterministic

$\boxed{\sigma}$ Adam $\quad$ $\textcircled{\tau}$ Eve



Claim

For all v, there exists p such that $\mathrm{mVal}^{\sigma_p}(v) \leqslant \mathrm{dVal}(v)$.

Properties of σ_p

- For all τ, $\mathbb{P}^{\sigma_p,\tau}(\diamond☺) = 1$
- Eve has an optimal memoryless deterministic strategy.

Strategy σ_p

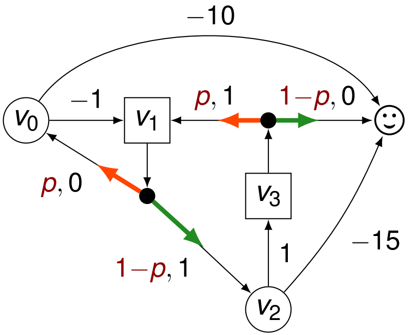
Let $\langle \sigma_1, \sigma_2 \rangle$ be an optimal switching strategy, for all $p \in (0, 1)$,

$$\sigma_p = p \times \sigma_1 + (1-p) \times \sigma_2$$

An analysis of stochastic shortest path problems, D. Bertsekas and J. Tsitsiklis, 1991, Mathematics of Operations Research.

Memoryless simulate deterministic

$\boxed{\sigma}$ Adam $\textcircled{\tau}$ Eve



Claim

For all v, there exists p such that $\text{mVal}^{\sigma_p}(v) \leqslant \text{dVal}(v)$.

Problem

Presence of non-negative cycles

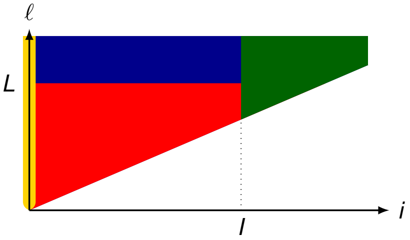
Strategy σ_p

Let $\langle \sigma_1, \sigma_2 \rangle$ be an optimal switching strategy, for all $p \in (0, 1)$,

$$\sigma_p = p \times \sigma_1 + (1-p) \times \sigma_2$$

Memoryless simulate deterministic

$\boxed{\sigma}$ Adam $\quad$ $\textcircled{\tau}$ Eve



Claim

For all v, there exists p such that $\text{mVal}^{\sigma_p}(v) \leqslant \text{dVal}(v)$.

Problem

Presence of non-negative cycles

Tool for the proof

Control the non-negative cycles with a partition of plays

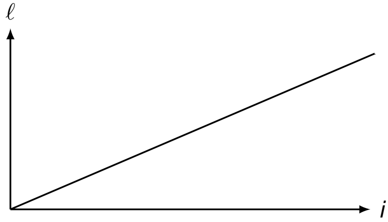
Strategy σ_p

Let $\langle \sigma_1, \sigma_2 \rangle$ be an optimal switching strategy, for all $p \in (0, 1)$,

$$\sigma_p = p \times \sigma_1 + (1-p) \times \sigma_2$$

Focus on the partition of plays

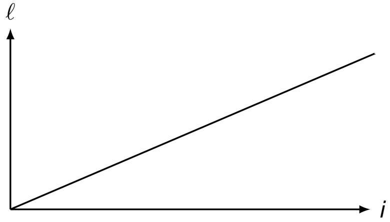
ℓ size of play reaching the target
i number of non-negative cycles



Focus on the partition of plays

ℓ size of play reaching the target
i number of non-negative cycles

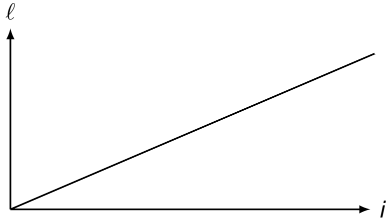
Fix a memoryless strategy for Eve



Focus on the partition of plays

ℓ size of play reaching the target
i number of non-negative cycles

Fix a memoryless strategy for Eve



Good zones

SP $\leqslant$ dVal
$\Rightarrow \mathbb{E}($**SP**$) \leqslant$ dVal

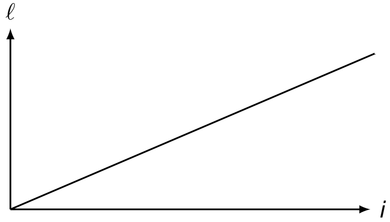
Focus on the partition of plays

Fix a memoryless strategy for Eve

ℓ size of play reaching the target
i number of non-negative cycles

Good zones

SP $\leqslant$ dVal
$\Rightarrow \mathbb{E}($**SP**$) \leqslant$ dVal



Zones to control

$\mathbb{E}($**SP**$) \leqslant \varepsilon$

Focus on the partition of plays

ℓ size of play reaching the target
i number of non-negative cycles

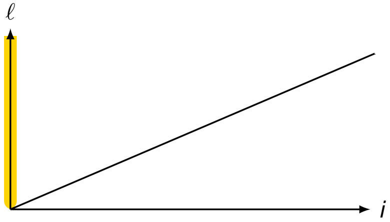
Fix a memoryless strategy for Eve

Yellow zone

All plays conforming to σ_1

Good zones

$\mathbf{SP} \leqslant \mathrm{dVal}$
$\Rightarrow \mathbb{E}(\mathbf{SP}) \leqslant \mathrm{dVal}$



Zones to control

$\mathbb{E}(\mathbf{SP}) \leqslant \varepsilon$

Strategy σ_p

Let $\langle \sigma_1, \sigma_2 \rangle$ be an optimal switching strategy, for all $p \in (0, 1)$,

$$\sigma_p = p \times \sigma_1 + (1 - p) \times \sigma_2$$

Focus on the partition of plays

ℓ size of play reaching the target
i number of non-negative cycles

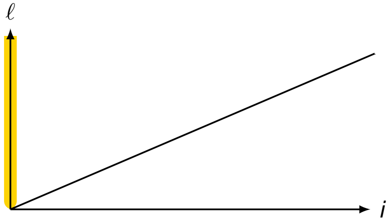
Fix a memoryless strategy for Eve

Yellow zone

All plays conforming to σ_1
Weight of each play is $\leqslant$ dVal

Good zones

SP $\leqslant$ dVal
$\Rightarrow \mathbb{E}(\mathbf{SP}) \leqslant$ dVal



Zones to control

$\mathbb{E}(\mathbf{SP}) \leqslant \varepsilon$

Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Focus on the partition of plays

ℓ size of play reaching the target
i number of non-negative cycles

Fix a memoryless strategy for Eve

Yellow zone

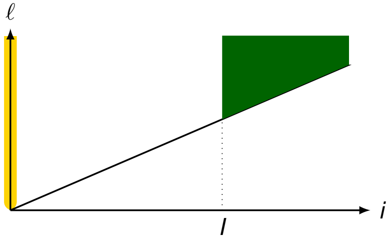
All plays conforming to σ_1
Weight of each play is $\leqslant$ dVal

Green zone

Plays contain many non-negative cycles

Good zones

SP $\leqslant$ dVal
$\Rightarrow \mathbb{E}(\mathbf{SP}) \leqslant$ dVal



Zones to control

$\mathbb{E}(\mathbf{SP}) \leqslant \varepsilon$

Focus on the partition of plays

ℓ size of play reaching the target
i number of non-negative cycles

Fix a memoryless strategy for Eve

Yellow zone

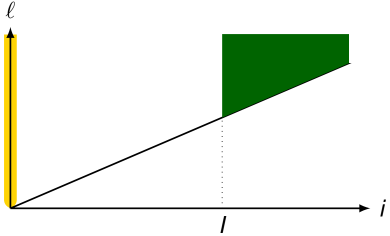
All plays conforming to σ_1
Weight of each play is $\leqslant$ dVal

Green zone

Plays contain many non-negative cycles
$\forall p, \exists I$ s.t. expectation $\leqslant \frac{\varepsilon}{2}$

Good zones

SP $\leqslant$ dVal
$\Rightarrow \mathbb{E}(\mathbf{SP}) \leqslant$ dVal



Zones to control

$\mathbb{E}(\mathbf{SP}) \leqslant \varepsilon$

Focus on the partition of plays

ℓ size of play reaching the target
i number of non-negative cycles

Fix a memoryless strategy for Eve

Yellow zone

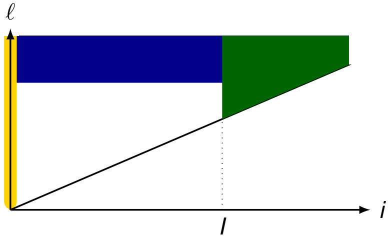
All plays conforming to σ_1
Weight of each play is $\leqslant$ dVal

Green zone

Plays contain many non-negative cycles
$\forall p, \exists I$ s.t. expectation $\leqslant \frac{\varepsilon}{2}$

Good zones

SP $\leqslant$ dVal
$\Rightarrow \mathbb{E}(\mathbf{SP}) \leqslant$ dVal



Zones to control

$\mathbb{E}(\mathbf{SP}) \leqslant \varepsilon$

Blue zone

Plays with many negative cycles and few non-negative cycles

Focus on the partition of plays

ℓ size of play reaching the target
i number of non-negative cycles

Fix a memoryless strategy for Eve

Yellow zone

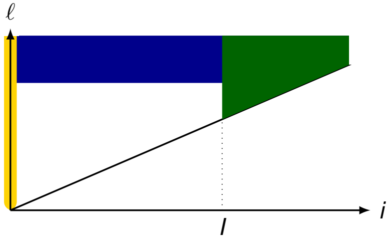
All plays conforming to σ_1
Weight of each play is $\leqslant$ dVal

Green zone

Plays contain many non-negative cycles
$\forall p, \exists I$ s.t. expectation $\leqslant \frac{\varepsilon}{2}$

Good zones

$\mathbf{SP} \leqslant$ dVal
$\Rightarrow \mathbb{E}(\mathbf{SP}) \leqslant$ dVal



Zones to control

$\mathbb{E}(\mathbf{SP}) \leqslant \varepsilon$

Blue zone

Plays with many negative cycles and few non-negative cycles
$\forall p, I, \ \exists L$ s.t. weights $\leqslant$ dVal

Focus on the partition of plays

ℓ size of play reaching the target
i number of non-negative cycles

Fix a memoryless strategy for Eve

Yellow zone

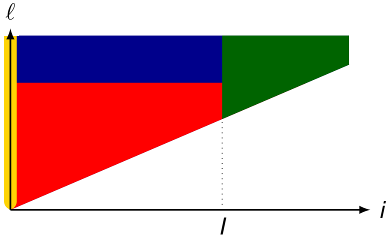
All plays conforming to σ_1
Weight of each play is $\leqslant$ dVal

Green zone

Plays contain many non-negative cycles
$\forall p, \exists I$ s.t. expectation $\leqslant \frac{\varepsilon}{2}$

Good zones

SP $\leqslant$ dVal
$\Rightarrow \mathbb{E}(\mathbf{SP}) \leqslant$ dVal



Zones to control

$\mathbb{E}(\mathbf{SP}) \leqslant \varepsilon$

Blue zone

Plays with many negative cycles and few non-negative cycles
$\forall p, I, \ \exists L$ s.t. weights $\leqslant$ dVal

Red zone

Rest of plays

Focus on the partition of plays

ℓ size of play reaching the target
i number of non-negative cycles

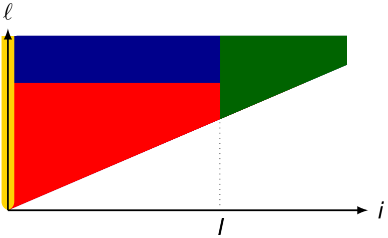
Fix a memoryless strategy for Eve

Yellow zone
All plays conforming to σ_1
Weight of each play is $\leqslant$ dVal

Green zone
Plays contain many non-negative cycles
$\forall p, \exists I$ s.t. expectation $\leqslant \frac{\varepsilon}{2}$

Good zones

$\mathbf{SP} \leqslant \mathrm{dVal}$
$\Rightarrow \mathbb{E}(\mathbf{SP}) \leqslant \mathrm{dVal}$



Zones to control

$\mathbb{E}(\mathbf{SP}) \leqslant \varepsilon$

Blue zone
Plays with many negative cycles and few non-negative cycles
$\forall p, I, \ \exists L$ s.t. weights $\leqslant$ dVal

Red zone
Rest of plays
$\exists p$ s.t. expectation $\leqslant \frac{\varepsilon}{2}$

Deterministic simulate memoryless

□ Adam ○ Eve

Claim

For all v and for all memoryless strategies ρ, there exists a deterministic strategy σ such that

$$\mathrm{dVal}^{\sigma}(v) \leqslant \mathrm{mVal}^{\rho}(v)$$

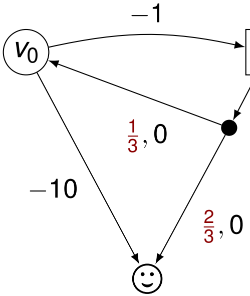
Deterministic simulate memoryless

□ Adam ○ Eve

Claim

For all v and for all memoryless strategies ρ, there exists a deterministic strategy σ such that

$$\mathrm{dVal}^{\sigma}(v) \leqslant \mathrm{mVal}^{\rho}(v)$$



Counter-example of intuition

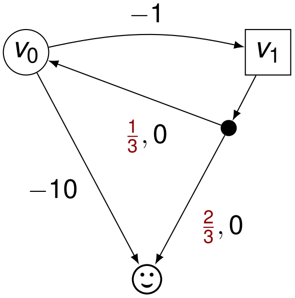
$$-\frac{1}{2} = \mathrm{mVal}^{\rho}(v_1)$$

Deterministic simulate memoryless ☐ Adam ◯ Eve

Claim

For all v and for all memoryless strategies ρ, there exists a deterministic strategy σ such that

$$\mathrm{dVal}^{\sigma}(v) \leqslant \mathrm{mVal}^{\rho}(v)$$



Intuition

In v_1, Adam chooses two times ☺ and one time v_0.

Counter-example of intuition

$$-\frac{1}{2} = \mathrm{mVal}^{\rho}(v_1)$$

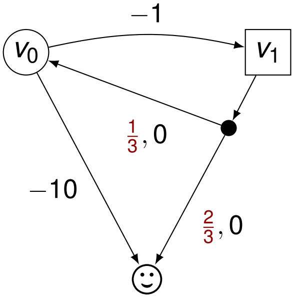
Deterministic simulate memoryless

□ Adam ○ Eve

Claim

For all v and for all memoryless strategies ρ, there exists a deterministic strategy σ such that

$$\mathrm{dVal}^{\sigma}(v) \leqslant \mathrm{mVal}^{\rho}(v)$$



Intuition

In v_1, Adam chooses two times ☺ and one time v_0.

Counter-example of intuition

$$\mathrm{dVal}^{\sigma}(v_1) = 0 > -\frac{1}{2} = \mathrm{mVal}^{\rho}(v_1)$$

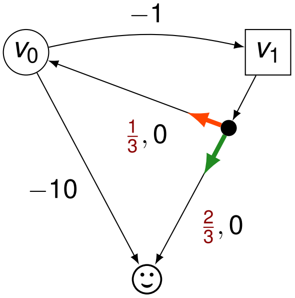
Deterministic simulate memoryless

□ Adam ○ Eve

Claim

For all v and for all memoryless strategies ρ, there exists a deterministic strategy σ such that

$$\mathrm{dVal}^{\sigma}(v) \leqslant \mathrm{mVal}^{\rho}(v)$$

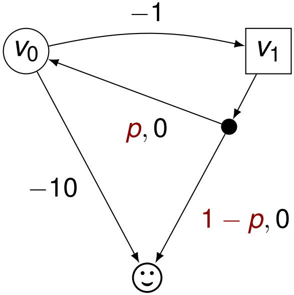


Tools for the proof

- Build a switching strategy $\sigma = \langle \sigma_1, \sigma_2 \rangle$ for ρ
- Value iteration for the fixpoint that gives the value

Focus on the switching strategy

Let ρ be a memoryless strategy

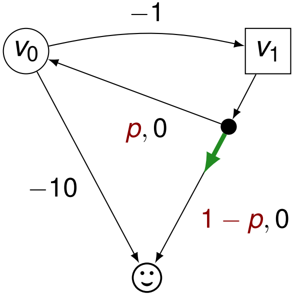


Focus on the switching strategy

□ Adam ○ Eve

Let ρ be a memoryless strategy

Fix σ_2 an attractor strategy



Focus on the switching strategy

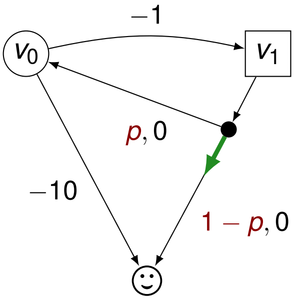
□ Adam ○ Eve

Let ρ be a memoryless strategy

Fix σ_2 an attractor strategy

Compute mVal^{ρ}

Based on an equation given by a fixed-point



Focus on the switching strategy

□ Adam ○ Eve

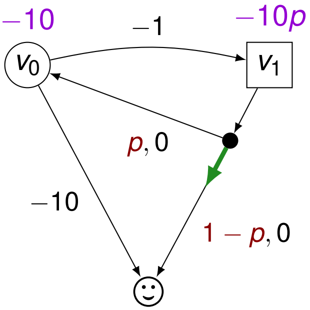
Let ρ be a memoryless strategy

Fix σ_2 an attractor strategy

Compute mVal^{ρ}

Based on an equation given by a fixed-point

$\mathrm{mVal}^{\rho}(v) = \sum_{v' \in E(v)} \mathbb{P}(v, v')(w(v, v') + \mathrm{mVal}^{\rho}(v'))$



Focus on the switching strategy

□ Adam ○ Eve

Let ρ be a memoryless strategy

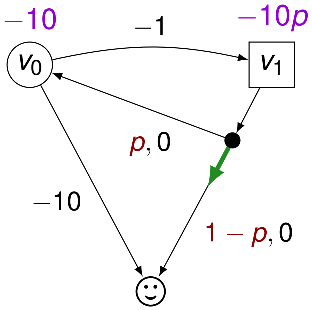
Fix σ_2 an attractor strategy

We know mVal^{ρ} for all vertices.

Compute mVal^{ρ}

Based on an equation given by a fixed-point

$\mathrm{mVal}^{\rho}(v) = \sum_{v' \in E(v)} \mathbb{P}(v, v')(w(v, v') + \mathrm{mVal}^{\rho}(v'))$



Focus on the switching strategy

□ Adam ○ Eve

Let ρ be a memoryless strategy

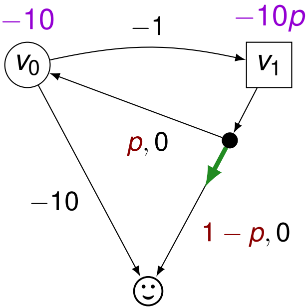
Fix σ_2 an attractor strategy

We know mVal^{ρ} for all vertices.

Compute mVal^{ρ}

Based on an equation given by a fixed-point

$\mathrm{mVal}^{\rho}(v) = \sum_{v' \in E(v)} \mathbb{P}(v, v')(w(v, v') + \mathrm{mVal}^{\rho}(v'))$



Properties of σ_1

Plays conforming to σ_1 reach

- the target with weight $\leqslant$ dVal
- a negative cycle

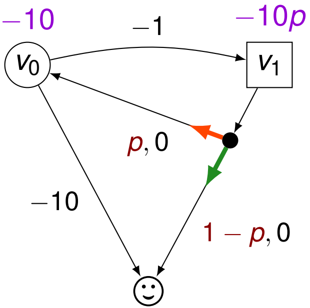
Focus on the switching strategy

Let ρ be a memoryless strategy
Fix σ_2 an attractor strategy

We know mVal^ρ for all vertices.

Compute mVal^ρ

Based on an equation given by a fixed-point
$\mathrm{mVal}^\rho(v) = \sum_{v' \in E(v)} \mathbb{P}(v, v')(w(v, v') + \mathrm{mVal}^\rho(v'))$



Properties of σ_1

Plays conforming to σ_1 reach

- the target with weight $\leqslant$ dVal ☺
- a negative cycle ☺

$\sigma_1(v) \subseteq \mathrm{argmin}_{v' \in E(v)}\{w(v, v') + \mathrm{mVal}^\rho(v')\} = G(v)$

Focus on the switching strategy

□ Adam ○ Eve

Let ρ be a memoryless strategy

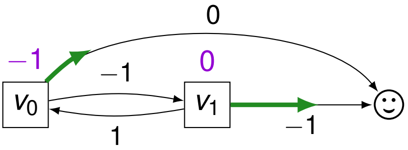
Fix σ_2 an attractor strategy

We know mVal^{ρ} for all vertices.

Compute mVal^{ρ}

Based on an equation given by a fixed-point

$\mathrm{mVal}^{\rho}(v) = \sum_{v' \in E(v)} \mathbb{P}(v, v')(w(v, v') + \mathrm{mVal}^{\rho}(v'))$



Properties of σ_1

Plays conforming to σ_1 reach

- the target with weight $\leqslant$ dVal ☺
- a negative cycle ☺

$\sigma_1(v) \subseteq \mathrm{argmin}_{v' \in E(v)}\{w(v, v') + \mathrm{mVal}^{\rho}(v')\} = G(v)$

Focus on the switching strategy

Adam Eve

Let ρ be a memoryless strategy

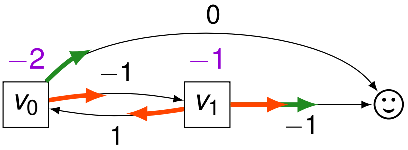
Fix σ_2 an attractor strategy

We know mVal^{ρ} for all vertices.

Compute mVal^{ρ}

Based on an equation given by a fixed-point

$\mathrm{mVal}^{\rho}(v) = \sum_{v' \in E(v)} \mathbb{P}(v, v')(w(v, v') + \mathrm{mVal}^{\rho}(v'))$



Properties of σ_1

Plays conforming to σ_1 reach

- the target with weight $\leqslant$ dVal ☺
- a negative cycle

Null cycle

How do we choose the good vertex?

$\sigma_1(v) \subseteq \mathrm{argmin}_{v' \in E(v)}\{w(v, v') + \mathrm{mVal}^{\rho}(v')\} = G(v)$

Focus on the switching strategy

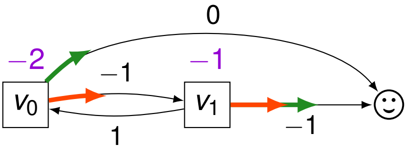
□ Adam ○ Eve

Let ρ be a memoryless strategy
Fix σ_2 an attractor strategy

We know mVal^ρ for all vertices.

Compute mVal^ρ

Based on an equation given by a fixed-point
$\mathrm{mVal}^\rho(v) = \sum_{v' \in E(v)} \mathbb{P}(v, v')(w(v, v') + \mathrm{mVal}^\rho(v'))$



Properties of σ_1

Plays conforming to σ_1 reach

- the target with weight $\leqslant$ dVal ☺
- a negative cycle ☺

Null cycle

How do we choose the good vertex?

Attractor distance

$\sigma_1(v) = \min_{v' \in G(v)} d(v')$

$\sigma_1(v) \subseteq \mathrm{argmin}_{v' \in E(v)} \{w(v, v') + \mathrm{mVal}^\rho(v')\} = G(v)$

Conclusion

Contributions

1. Adam has the same hope using memory or randomness.
2. Existence of an optimal memoryless strategy for Adam is testable in polynomial time.

Conclusion

Contributions

1. Adam has the same hope using memory or randomness.
2. Existence of an optimal memoryless strategy for Adam is testable in polynomial time.

Perspectives

- A polynomial-time algorithm to compute the value
- Extension to probabilistic value (memory and randomisation)

Conclusion

Contributions

1. Adam has the same hope using memory or randomness.
2. Existence of an optimal memoryless strategy for Adam is testable in polynomial time.

Perspectives

- A polynomial-time algorithm to compute the value
- Extension to probabilistic value (memory and randomisation)

Thank you! Questions?