

Reaching Your Goal Optimally by Playing at Random with no Memory

Julie Parreaux¹

Benjamin Monmege² Pierre-Alain Reynier²

¹Ecole Normale Supérieure de Rennes, France

²Aix-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

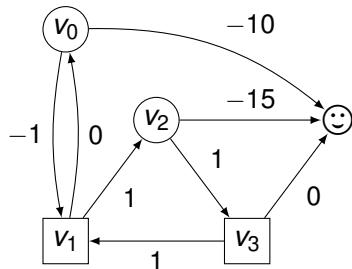
Shortest-Path games

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

Shortest Path Game

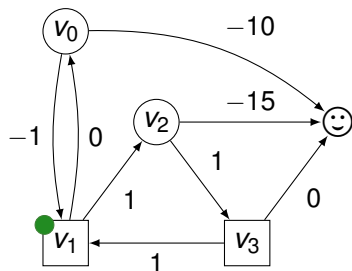
□ Adam ○ Eve

😊 target (T)



Shortest Path Game

□ Adam ○ Eve



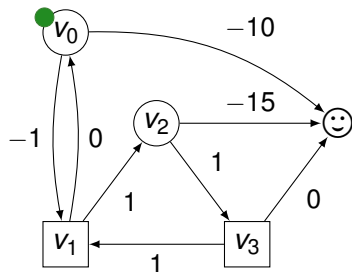
How to play?

Move a token along an edge

$$\pi = v_1$$

Shortest Path Game

□ Adam ○ Eve



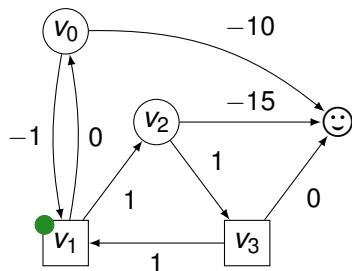
How to play?

Move a token along an edge

$$\pi = v_1 v_0$$

Shortest Path Game

□ Adam ○ Eve



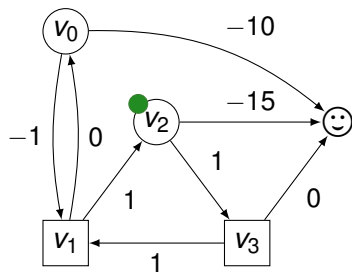
How to play?

Move a token along an edge

$$\pi = v_1 v_0 v_1$$

Shortest Path Game

□ Adam ○ Eve



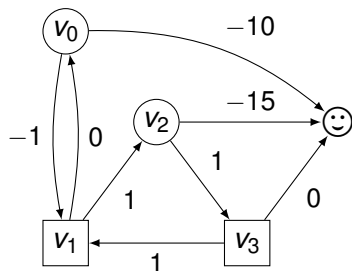
How to play?

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

$$\pi = (v_i)_i \in V^\omega \quad \pi = (v_i)_i \text{😊}$$

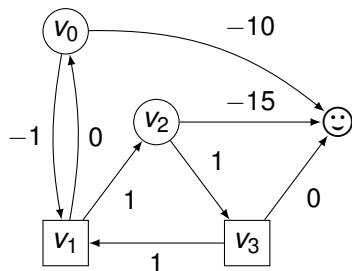
How to play?

Move a token along an edge

$$\pi = v_1 v_0 v_1 v_2 v_3 \text{😊}$$

Shortest Path Game

□ Adam ○ Eve



Play

Infinite path or reach the target

$$\pi = (v_i)_i \in V^\omega \quad \pi = (v_i)_i \text{ ☺}$$

How to play?

Move a token along an edge

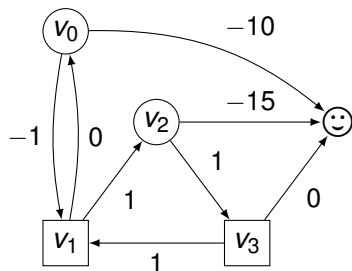
$$\pi = v_1 v_0 v_1 v_2 v_3 \text{ ☺}$$

Shortest Path payoff of a play π

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

Shortest Path Game

□ Adam ○ Eve



Play

Infinite path or reach the target

$$\pi = (v_i)_i \in V^\omega \quad \pi = (v_i)_i \text{😊}$$

How to play?

Move a token along an edge

$$\pi = v_1 v_0 v_1 v_2 v_3 \text{😊}$$

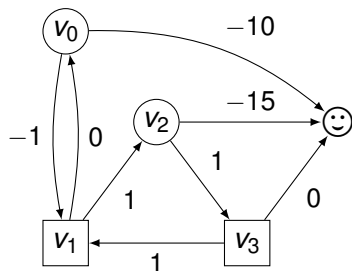
$$\mathbf{SP}(\pi) = 0 + (-1) + 1 + 1 + 0 = 1$$

Shortest Path payoff of a play π

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

Shortest Path Game

□ Adam ○ Eve



Objectives

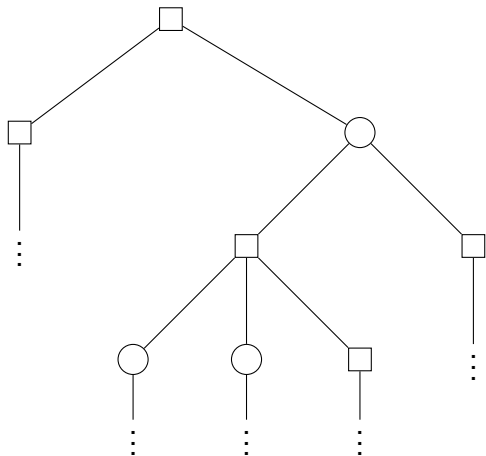
Eve maximise the payoff

Adam minimise the payoff

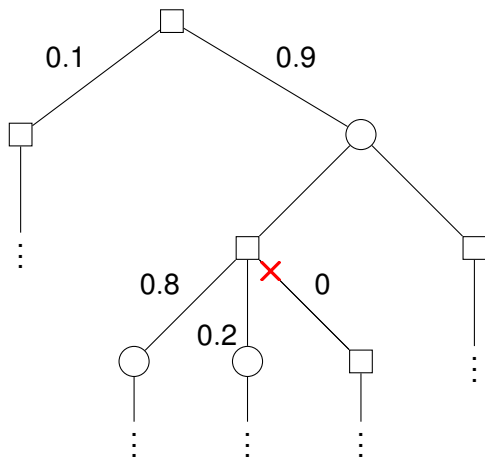
Shortest Path payoff of a play π

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

Strategies for Adam



Strategies for Adam



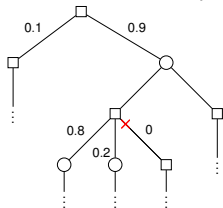
A strategy

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

Strategies for Adam

Infinite memory

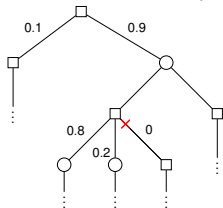
$$\sigma : V^* V_{Adam} \rightarrow \Delta(V)$$



Strategies for Adam

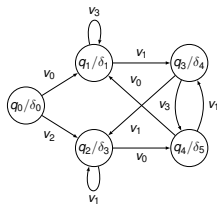
Infinite memory

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



Finite memory

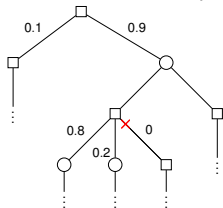
Moore machine



Strategies for Adam

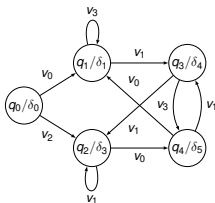
Infinite memory

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



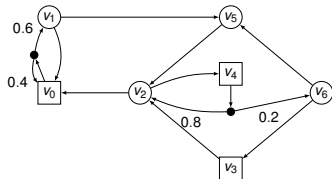
Finite memory

Moore machine



Memoryless

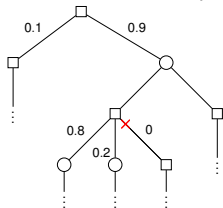
$$\sigma : V_{Adam} \rightarrow \Delta(V)$$



Strategies for Adam

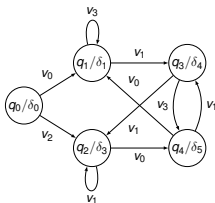
Infinite memory

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



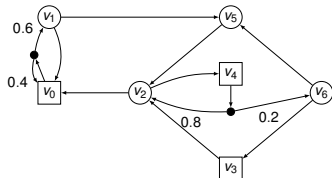
Finite memory

Moore machine



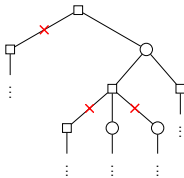
Memoryless

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



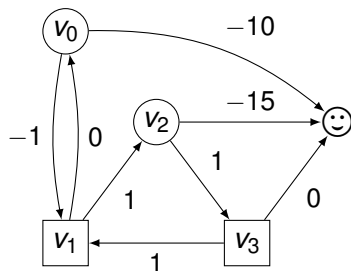
Deterministic

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



Deterministic Strategies

σ Adam τ Eve



Value

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

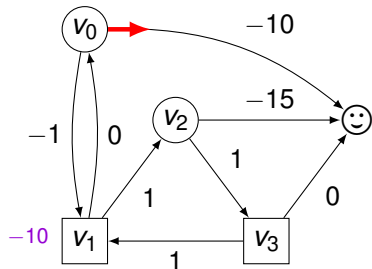
Deterministic strategy

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

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



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

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

Value

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

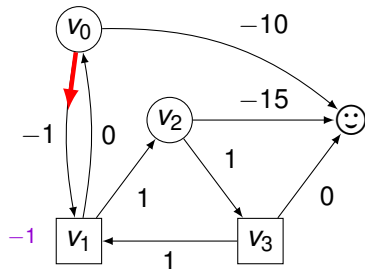
Deterministic strategy

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

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



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

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

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

Value

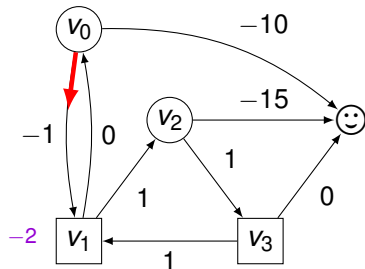
$$\overline{\text{dVal}}(v) = \inf_{\sigma} \sup_{\tau} \underbrace{\text{SP}(\text{Play}(v, \sigma, \tau))}_{\text{dVal}^{\sigma}(v)}$$

Deterministic strategy

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

Deterministic Strategies

σ Adam τ Eve



Estimate $\overline{dVal}(v_1)$

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

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

Value

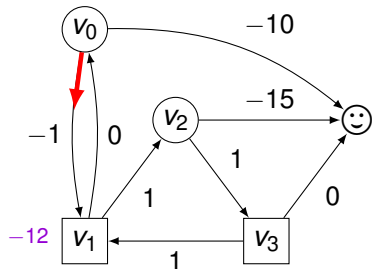
$$\overline{dVal}(v) = \inf_{\sigma} \sup_{\tau} \underbrace{\mathbf{SP}(\text{Play}(v, \sigma, \tau))}_{dVal^{\sigma}(v)}$$

Deterministic strategy

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

Deterministic Strategies

σ Adam τ Eve



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

Eve chooses smiley face in v_0 : $\rightsquigarrow -10$

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

Value

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

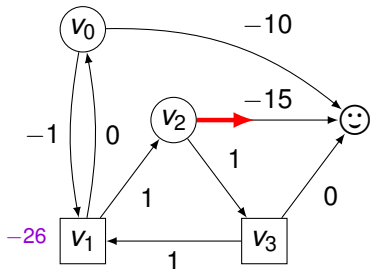
Deterministic strategy

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

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



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 : $\rightsquigarrow -26$

Value

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

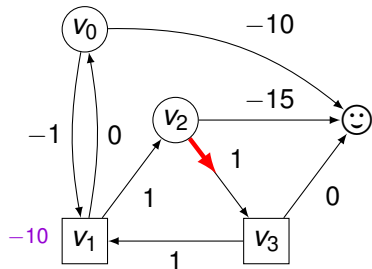
Deterministic strategy

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

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



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

- Eve chooses smiley face in v_0 : $\rightsquigarrow -10$
- Eve chooses v_1 in v_0 : $\dashrightarrow -12$
- Eve chooses smiley face in v_2 : $\rightsquigarrow -26$
- Eve chooses v_3 in v_2 : $\rightsquigarrow -10$

Value

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

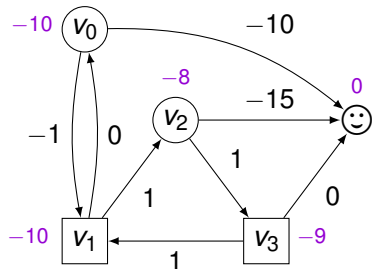
Deterministic strategy

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

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



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

- Eve chooses smiley face in v_0 : $\rightsquigarrow -10$
- Eve chooses v_1 in v_0 : $\dashrightarrow -12$
- Eve chooses smiley face in v_2 : $\rightsquigarrow -26$
- Eve chooses v_3 in v_2 : $\rightsquigarrow -10$

Value

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

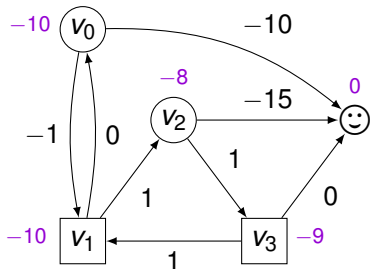
Deterministic strategy

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

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

$$d\text{Val}(v) = \overline{d\text{Val}}(v) = \underline{d\text{Val}}(v)$$

Value

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

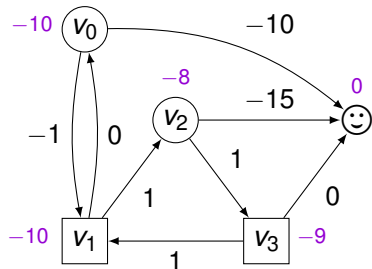
Deterministic strategy

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

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



Optimal strategy

$$d\text{Val}^{\sigma^*}(v) \leq d\text{Val}(v)$$

Value

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

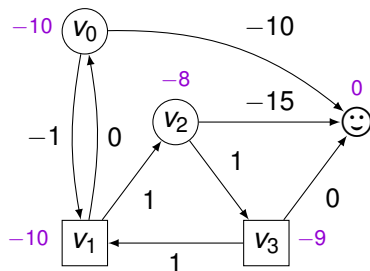
Deterministic strategy

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

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



Optimal strategy

$$dVal^{\sigma^*}(v) \leq dVal(v)$$

Optimal strategy for Adam

An optimal strategy for Adam may require finite memory.

Value

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

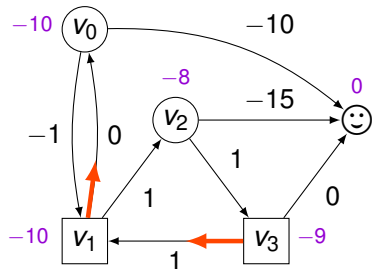
Deterministic strategy

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

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



Optimal strategy

$$dVal^{\sigma^*}(v) \leq dVal(v)$$

Optimal strategy for Adam

The switching strategy:

► σ_1 : reach negative cycle

Value

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

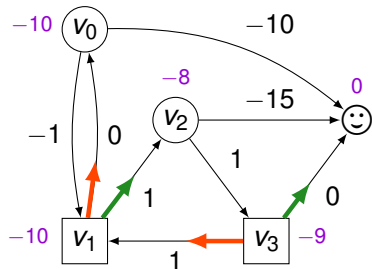
Deterministic strategy

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

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



Optimal strategy

$$dVal^{\sigma^*}(v) \leq dVal(v)$$

Optimal strategy for Adam

The switching strategy:

- ▶ σ_1 : reach negative cycle
- ▶ σ_2 : reach 😊

Value

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

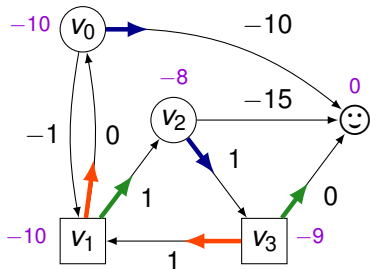
Deterministic strategy

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

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



Optimal strategy

$$dVal^{\sigma^*}(v) \leq dVal(v)$$

Optimal strategy for Eve

Eve has a memoryless optimal strategy.

Value

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

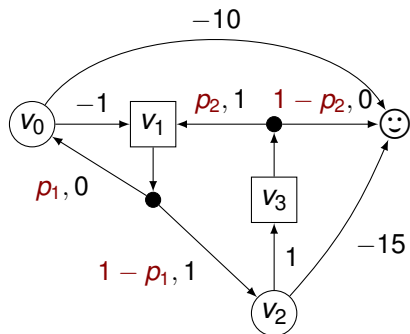
Deterministic strategy

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

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

Memoryless strategies

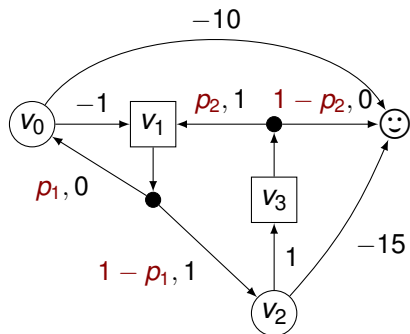
σ Adam τ Eve



Memoryless strategy

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

Memoryless strategies

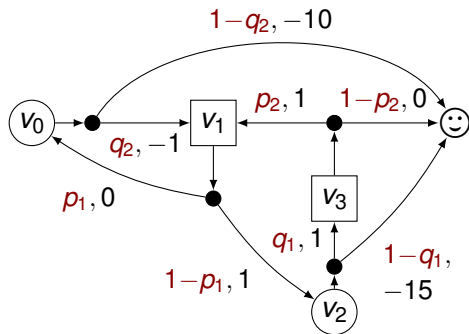


Value

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

$\underbrace{\hspace{10em}}_{\text{mVal}^{\sigma}(v)}$

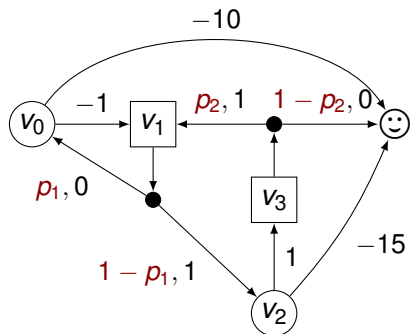
\square Adam \circ Eve



Memoryless strategy

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

Memoryless strategies

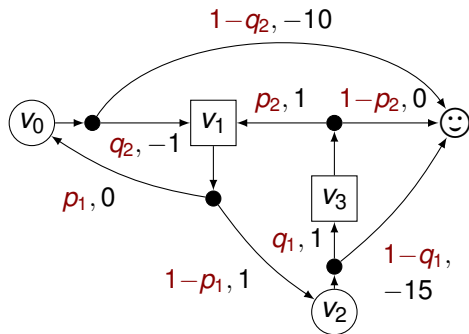


Value

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

$\underbrace{\hspace{10em}}_{\text{mVal}^{\sigma}(v)}$

σ Adam τ Eve



Memoryless strategy

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

ϵ -optimal strategy

$$\text{mVal}^{\sigma^*}(v) \leq \overline{\text{mVal}}(v) + \epsilon$$

Contributions

Main theorem

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

Contributions

Main theorem

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

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.

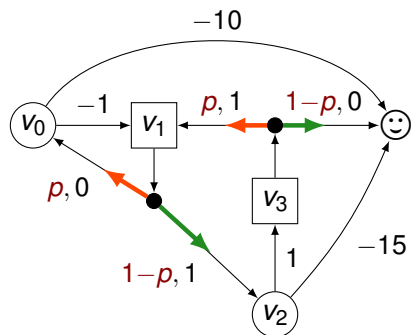
Memoryless simulate deterministic

Claim

For all v , there exists ρ such that $mVal^{\sigma\rho}(v) \leq dVal(v)$.

Memoryless simulate deterministic

\square Adam \circ Eve



Claim

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

Strategy σ_p

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

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

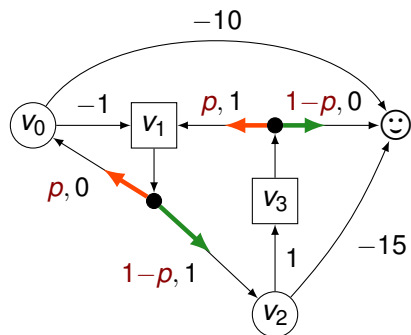
Memoryless simulate deterministic



Adam



Eve



Claim

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

Properties of σ_p

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

Strategy σ_p

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

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

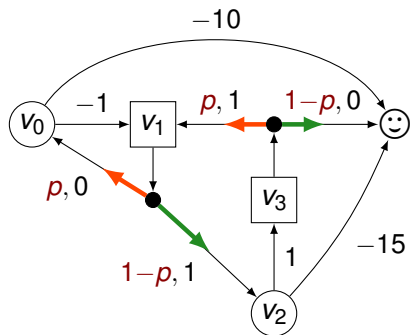
Memoryless simulate deterministic



Adam



Eve



Claim

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

Properties of σ_p

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

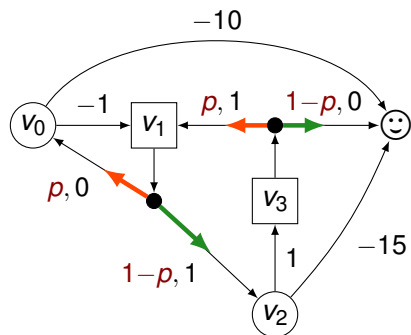
Strategy σ_p

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

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

Memoryless simulate deterministic

\square Adam \circ Eve



Claim

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

Problem

Presence of non-negative cycles

Strategy σ_p

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

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

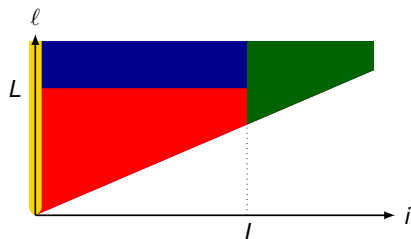
Memoryless simulate deterministic



Adam



Eve



Claim

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

Problem

Presence of non-negative cycles

Tool for the proof

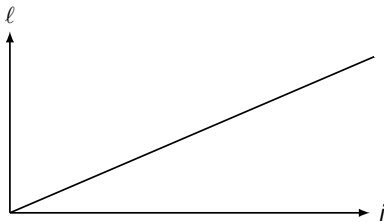
Control the non-negative cycles with a partition of plays

Strategy σ_p

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

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

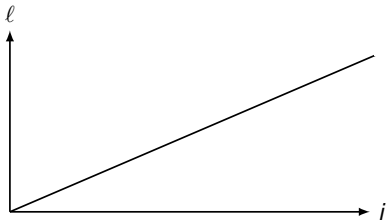
Focus on the partition of plays ℓ size of play reaching the target
 i number of non-negative cycles



Focus on the partition of plays

Fix a memoryless strategy for Eve

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



Focus on the partition of plays

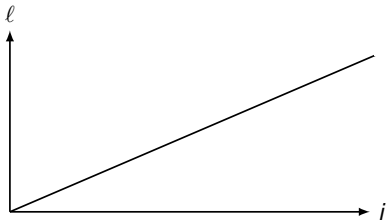
Fix a memoryless strategy for Eve

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

Good zones

$$\mathbf{SP} \leq d\text{Val}$$

$$\Rightarrow \mathbb{E}(\mathbf{SP}) \leq d\text{Val}$$



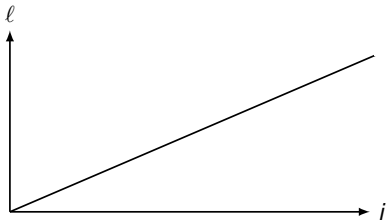
Focus on the partition of plays

Fix a memoryless strategy for Eve

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

Good zones

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



Zones to control

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

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

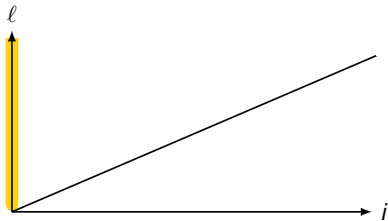
All plays conforming to σ_1



Good zones

$$\mathbf{SP} \leq d\text{Val}$$

$$\Rightarrow \mathbb{E}(\mathbf{SP}) \leq d\text{Val}$$



Zones to control

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

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

Fix a memoryless strategy for Eve

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

Yellow zone

All plays conforming to σ_1

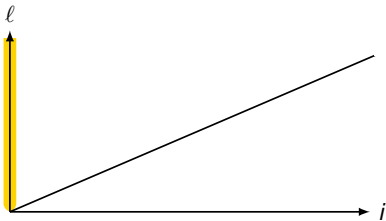
Weight of each play is $\leq dVal$



Good zones

$SP \leq dVal$

$\Rightarrow \mathbb{E}(SP) \leq dVal$



Zones to control

$\mathbb{E}(SP) \leq \epsilon$

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

Yellow zone

All plays conforming to σ_1

Weight of each play is $\leq dVal$

Green zone

Plays contain many non-negative cycles

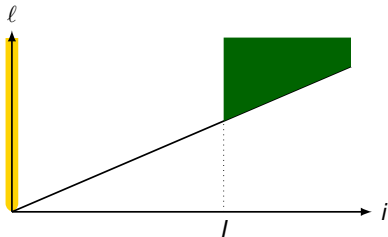
Good zones

$$SP \leq dVal$$

$$\Rightarrow \mathbb{E}(SP) \leq dVal$$

Zones to control

$$\mathbb{E}(SP) \leq \epsilon$$



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

Yellow zone

All plays conforming to σ_1

Weight of each play is $\leq dVal$

Green zone

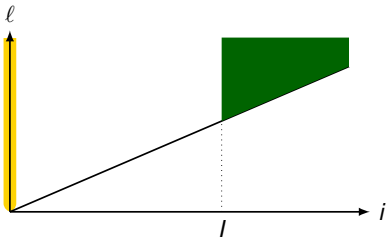
Plays contain many non-negative cycles

$\forall p, \exists l$ s.t. expectation $\leq \frac{\epsilon}{2}$

Good zones

$SP \leq dVal$

$\Rightarrow \mathbb{E}(SP) \leq dVal$



Zones to control

$\mathbb{E}(SP) \leq \epsilon$

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 $\leq dVal$

Green zone

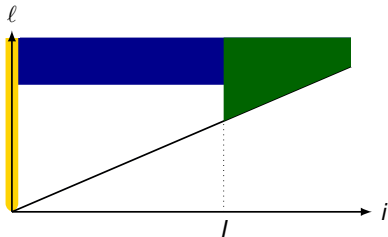
Plays contain many non-negative cycles

$\forall p, \exists l$ s.t. expectation $\leq \frac{\epsilon}{2}$

Good zones

$SP \leq dVal$

$\Rightarrow \mathbb{E}(SP) \leq dVal$



Zones to control

$\mathbb{E}(SP) \leq \epsilon$

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 $\leq dVal$

Green zone

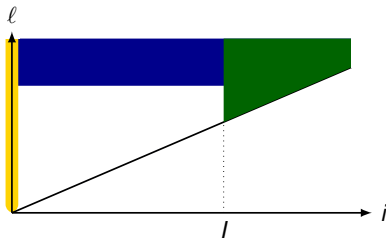
Plays contain many non-negative cycles

$\forall p, \exists l$ s.t. expectation $\leq \frac{\epsilon}{2}$

Good zones

$SP \leq dVal$

$\Rightarrow \mathbb{E}(SP) \leq dVal$



Zones to control

$\mathbb{E}(SP) \leq \epsilon$

Blue zone

Plays with many negative cycles and few non-negative cycles

$\forall p, l, \exists L$ s.t. weights $\leq dVal$

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 $\leq dVal$

Green zone

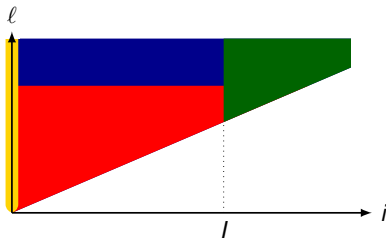
Plays contain many non-negative cycles

$\forall p, \exists l$ s.t. expectation $\leq \frac{\epsilon}{2}$

Good zones

$SP \leq dVal$

$\Rightarrow \mathbb{E}(SP) \leq dVal$



Zones to control

$\mathbb{E}(SP) \leq \epsilon$

Blue zone

Plays with many negative cycles and few non-negative cycles

$\forall p, l, \exists L$ s.t. weights $\leq dVal$

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 $\leq dVal$

Green zone

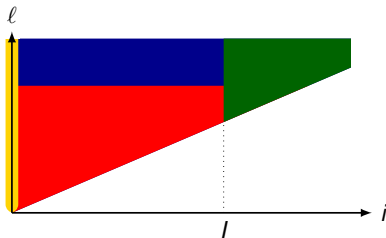
Plays contain many non-negative cycles

$\forall p, \exists l$ s.t. expectation $\leq \frac{\epsilon}{2}$

Good zones

$SP \leq dVal$

$\Rightarrow \mathbb{E}(SP) \leq dVal$



Zones to control

$\mathbb{E}(SP) \leq \epsilon$

Blue zone

Plays with many negative cycles and few non-negative cycles

$\forall p, l, \exists L$ s.t. weights $\leq dVal$

Red zone

Rest of plays

$\exists p$ s.t. expectation $\leq \frac{\epsilon}{2}$

Deterministic simulate memoryless



Adam



Eve

Claim

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

$$\text{dVal}^\sigma(v) \leq \text{mVal}^\rho(v)$$

Deterministic simulate memoryless



Adam

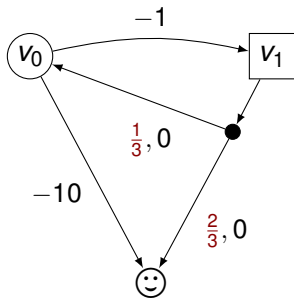


Eve

Claim

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

$$dVal^\sigma(v) \leq mVal^\rho(v)$$



Counter-example of intuition

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

Deterministic simulate memoryless



Adam

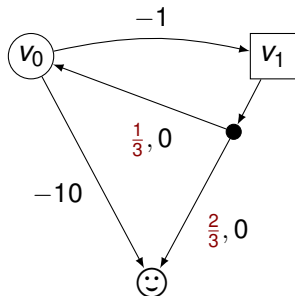


Eve

Claim

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

$$dVal^\sigma(v) \leq mVal^\rho(v)$$



Intuition

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

Counter-example of intuition

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

Deterministic simulate memoryless



Adam

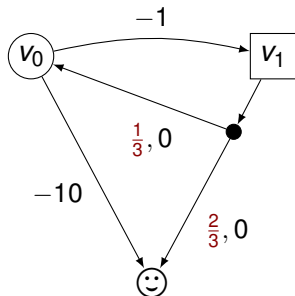


Eve

Claim

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

$$dVal^\sigma(v) \leq mVal^\rho(v)$$



Intuition

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

Counter-example of intuition

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

Deterministic simulate memoryless



Adam

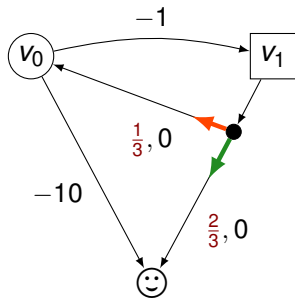


Eve

Claim

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

$$dVal^\sigma(v) \leq mVal^\rho(v)$$



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

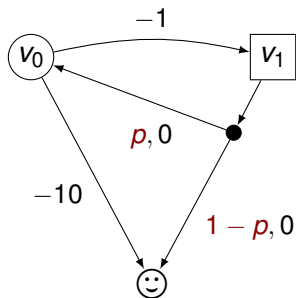
Let ρ be a memoryless strategy



Adam



Eve



Focus on the switching strategy

Let ρ be a memoryless strategy

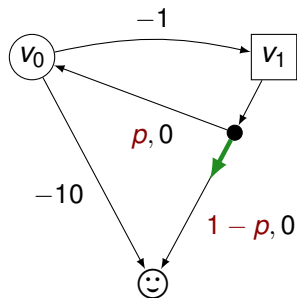
Fix σ_2 an attractor strategy



Adam



Eve



Focus on the switching strategy

Let ρ be a memoryless strategy

Fix σ_2 an attractor strategy

Compute $mVal^\rho$

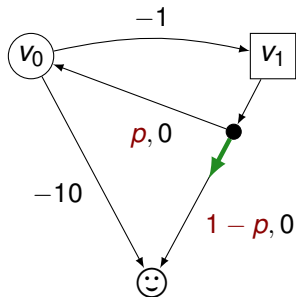
Based on an equation given by a fixed-point



Adam



Eve



Focus on the switching strategy



Adam



Eve

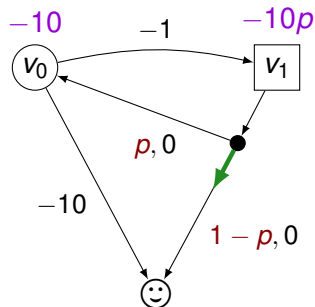
Let ρ be a memoryless strategy

Fix σ_2 an attractor strategy

Compute $mVal^\rho$

Based on an equation given by a fixed-point

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



Focus on the switching strategy



Adam



Eve

Let ρ be a memoryless strategy

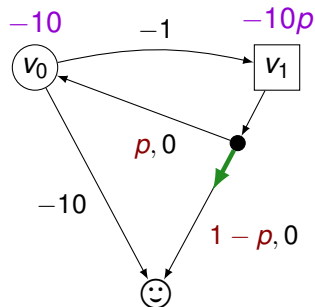
Fix σ_2 an attractor strategy

We know $mVal^\rho$ for all vertices.

Compute $mVal^\rho$

Based on an equation given by a fixed-point

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



Focus on the switching strategy



Adam



Eve

Let ρ be a memoryless strategy

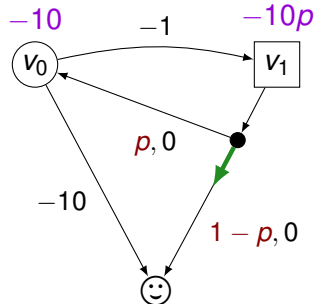
Fix σ_2 an attractor strategy

We know $mVal^\rho$ for all vertices.

Compute $mVal^\rho$

Based on an equation given by a fixed-point

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



Properties of σ_1

Plays conforming to σ_1 reach

- ▶ the target with weight $\leq dVal$
- ▶ a negative cycle

Focus on the switching strategy



Adam



Eve

Let ρ be a memoryless strategy

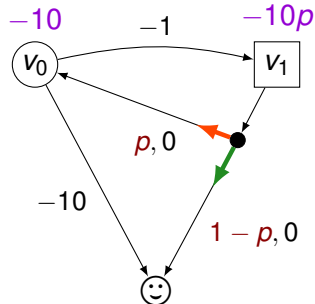
Fix σ_2 an attractor strategy

We know $mVal^\rho$ for all vertices.

Compute $mVal^\rho$

Based on an equation given by a fixed-point

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



Properties of σ_1

Plays conforming to σ_1 reach

- ▶ the target with weight $\leq dVal$ 😊
- ▶ a negative cycle 😊

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

Focus on the switching strategy



Adam



Eve

Let ρ be a memoryless strategy

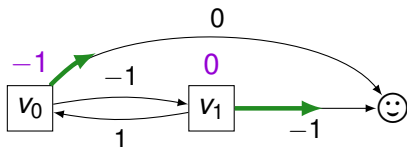
Fix σ_2 an attractor strategy

We know $mVal^\rho$ for all vertices.

Compute $mVal^\rho$

Based on an equation given by a fixed-point

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



Properties of σ_1

Plays conforming to σ_1 reach

- ▶ the target with weight $\leq dVal$ 😊
- ▶ a negative cycle 😊

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

Focus on the switching strategy



Adam



Eve

Let ρ be a memoryless strategy

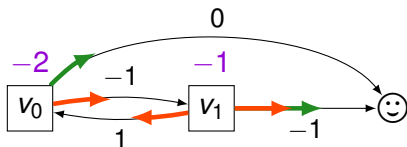
Fix σ_2 an attractor strategy

We know $mVal^\rho$ for all vertices.

Compute $mVal^\rho$

Based on an equation given by a fixed-point

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



Properties of σ_1

Plays conforming to σ_1 reach

- ▶ the target with weight $\leq dVal$ 😊
- ▶ a negative cycle

Null cycle

How do we choose the good vertex?

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

Focus on the switching strategy



Adam



Eve

Let ρ be a memoryless strategy

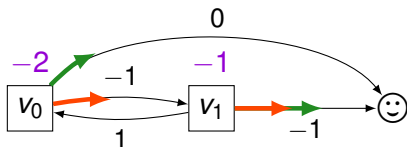
Fix σ_2 an attractor strategy

We know $mVal^\rho$ for all vertices.

Compute $mVal^\rho$

Based on an equation given by a fixed-point

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



Properties of σ_1

Plays conforming to σ_1 reach

- ▶ the target with weight $\leq dVal$ 😊
- ▶ a negative cycle 😊

Null cycle

How do we choose the good vertex?

Attractor distance

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

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

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?