Reaching Your Goal Optimally by Playing at Random with no Memory

Julie Parreaux
Benjamin Monmege   Pierre-Alain Reynier

September 24, 2020

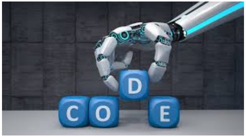
Motivation : game theory for synthesis

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

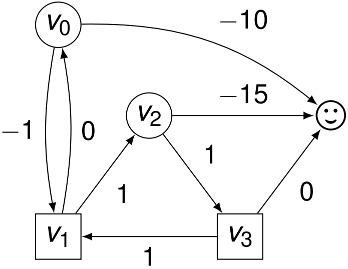
Quantitative games

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

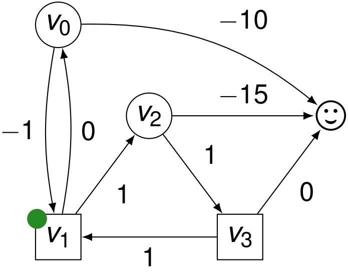
Shortest-Path games

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

Shortest Path Game



Shortest Path Game

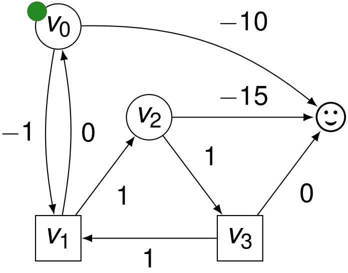


How to play?

Move a token along an edge

$$\pi = v_1$$

Shortest Path Game



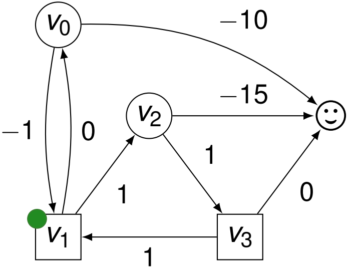
How to play?

Move a token along an edge

$$\pi = v_1 v_0$$

Shortest Path Game

□ Adam ○ Eve

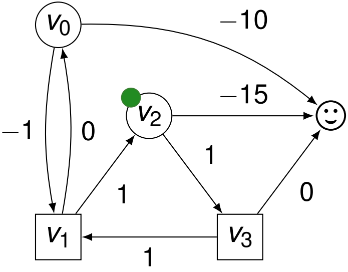


How to play?

Move a token along an edge

$$\pi = v_1 v_0 v_1$$

Shortest Path Game



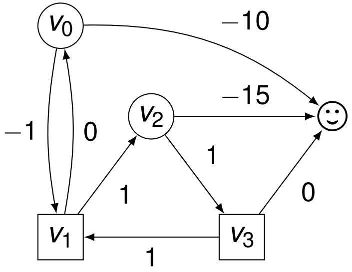
How to play?

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

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

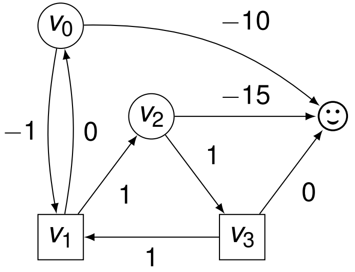
How to play?

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

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

How to play?

Move a token along an edge

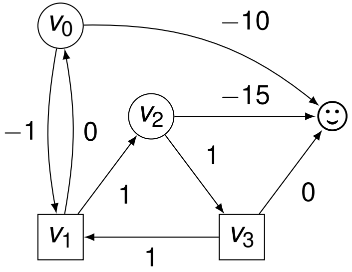
$\pi = v_1 v_0 v_1 v_2 v_3$☺

Shortest Path payoff of a play π

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

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

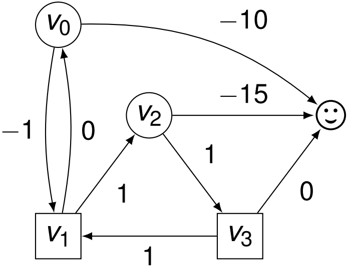
$$\pi = v_1 v_0 v_1 v_2 v_3 ☺$$

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

Shortest Path payoff of a play π

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

Shortest Path Game



Objectives

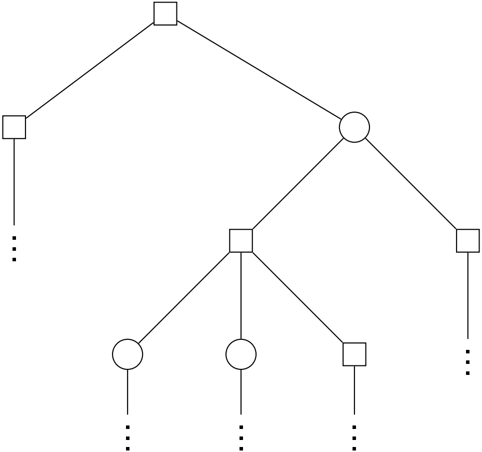
Eve maximise the payoff

Adam minimise the payoff

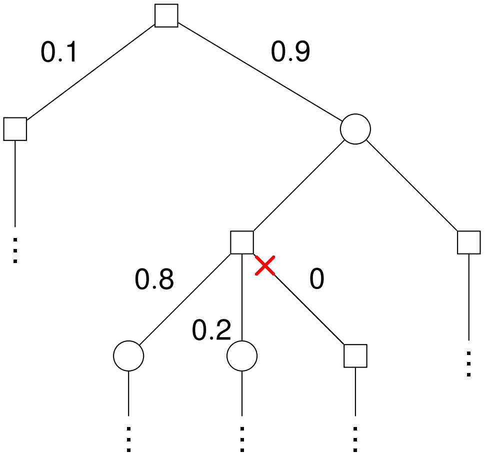
Shortest Path payoff of a play π

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

Strategies for Adam



Strategies for Adam



A strategy

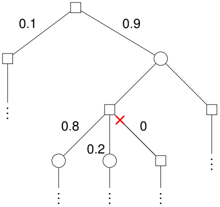
$\sigma : V^* V_{Adam} \to \Delta(V)$

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

Strategies for Adam

Infinite memory

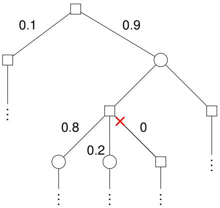
$\sigma : V^* V_{Adam} \to \Delta(V)$



Strategies for Adam

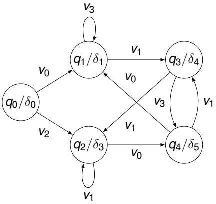
Infinite memory

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



Finite memory

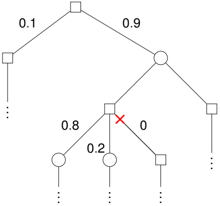
Moore machine



Strategies for Adam

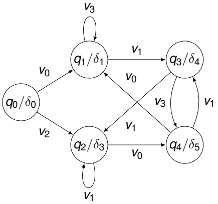
Infinite memory

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



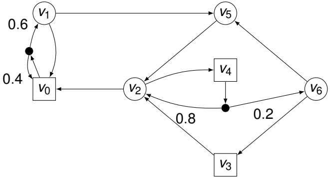
Finite memory

Moore machine



Memoryless

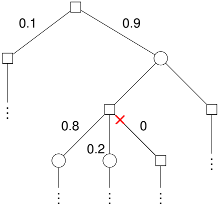
$\sigma : V_{Adam} \to \Delta(V)$



Strategies for Adam

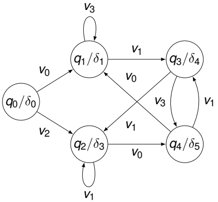
Infinite memory

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



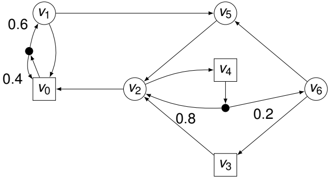
Finite memory

Moore machine



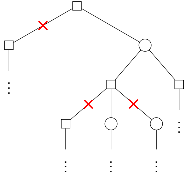
Memoryless

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



Deterministic

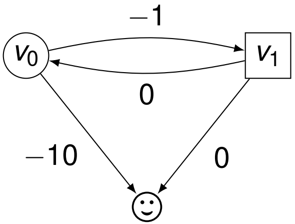
$\sigma : V^* V_{Adam} \to V$



Deterministic Strategies

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

σ Adam $\quad$ τ Eve



Value

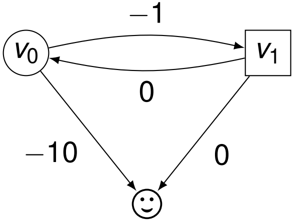
$$\overline{\text{dVal}}(v) = \inf_{\sigma} \underbrace{\sup_{\tau} \textbf{SP}(\text{Play}(v, \sigma, \tau))}_{\text{dVal}^{\sigma}(v)}$$

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

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

σ Adam $\quad$ τ Eve



Value

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

Determinacy

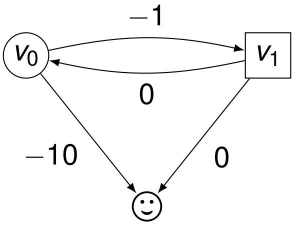
$$\mathrm{dVal}(v) = \overline{\mathrm{dVal}}(v) = \underline{\mathrm{dVal}}(v)$$

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

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

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



Value

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

Bellman equation

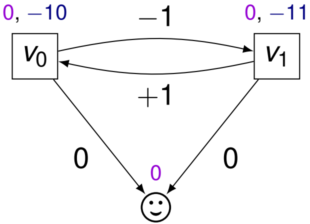
$$\mathrm{dVal}(v) = \begin{cases} 0 & \text{if } v = ☺ \\ \max_{v'} \left(w(v, v') + \mathrm{dVal}(v')\right) & \text{if } v \in V_{\text{Eve}} \\ \min_{v'} \left(w(v, v') + \mathrm{dVal}(v')\right) & \text{if } v \in V_{\text{Adam}} \end{cases}$$

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

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

$\boxed{\sigma}$ Adam $\quad$ τ Eve



Value

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

Unicity

Bellman equation may have many solutions.

Bellman equation

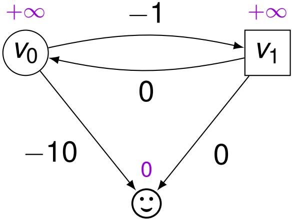
$$\mathrm{dVal}(v) = \begin{cases} 0 & \text{if } v = ☺ \\ \max_{v'} (w(v, v') + \mathrm{dVal}(v')) & \text{if } v \in V_{\text{Eve}} \\ \min_{v'} (w(v, v') + \mathrm{dVal}(v')) & \text{if } v \in V_{\text{Adam}} \end{cases}$$

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

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

$\boxed{\sigma}$ Adam $\quad$ τ Eve



Value

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

Value iteration

- Compute dVal as a greatest fixed point

Bellman equation

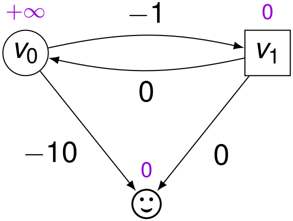
$$\text{dVal}(v) = \begin{cases} 0 & \text{if } v = ☺ \\ \max_{v'} (w(v, v') + \text{dVal}(v')) & \text{if } v \in V_{\text{Eve}} \\ \min_{v'} (w(v, v') + \text{dVal}(v')) & \text{if } v \in V_{\text{Adam}} \end{cases}$$

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

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

$\boxed{\sigma}$ Adam $\quad$ τ Eve



Value

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

Value iteration

- Compute dVal as a greatest fixed point

Bellman equation

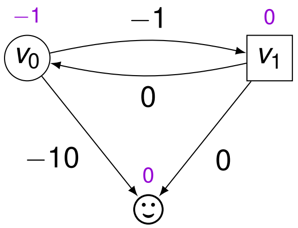
$$\mathrm{dVal}(v) = \begin{cases} 0 & \text{if } v = ☺ \\ \max_{v'} (w(v, v') + \mathrm{dVal}(v')) & \text{if } v \in V_{\text{Eve}} \\ \min_{v'} (w(v, v') + \mathrm{dVal}(v')) & \text{if } v \in V_{\text{Adam}} \end{cases}$$

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

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

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



Value

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

Value iteration

- Compute dVal as a greatest fixed point

Bellman equation

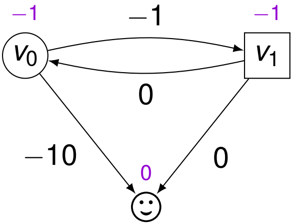
$$\text{dVal}(v) = \begin{cases} 0 & \text{if } v = \smile \\ \max_{v'} (w(v, v') + \text{dVal}(v')) & \text{if } v \in V_{\text{Eve}} \\ \min_{v'} (w(v, v') + \text{dVal}(v')) & \text{if } v \in V_{\text{Adam}} \end{cases}$$

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

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

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



Value

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

Value iteration

- Compute dVal as a greatest fixed point

Bellman equation

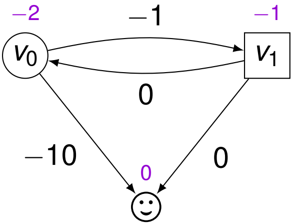
$$\text{dVal}(v) = \begin{cases} 0 & \text{if } v = ☺ \\ \max_{v'} \left(w(v, v') + \text{dVal}(v')\right) & \text{if } v \in V_{\text{Eve}} \\ \min_{v'} \left(w(v, v') + \text{dVal}(v')\right) & \text{if } v \in V_{\text{Adam}} \end{cases}$$

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

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

$\boxed{\sigma}$ Adam τ Eve



Value

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

Value iteration

- Compute dVal as a greatest fixed point

Bellman equation

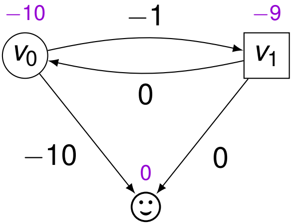
$$\text{dVal}(v) = \begin{cases} 0 & \text{if } v = ☺ \\ \max_{v'} (w(v, v') + \text{dVal}(v')) & \text{if } v \in V_{\text{Eve}} \\ \min_{v'} (w(v, v') + \text{dVal}(v')) & \text{if } v \in V_{\text{Adam}} \end{cases}$$

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

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

$\boxed{\sigma}$ Adam τ Eve



Value

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

Value iteration

- Compute dVal as a greatest fixed point

Bellman equation

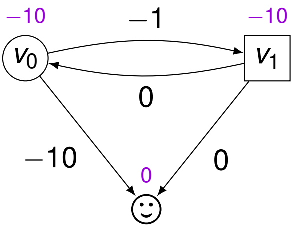
$$\text{dVal}(v) = \begin{cases} 0 & \text{if } v = ☺ \\ \max_{v'} (w(v, v') + \text{dVal}(v')) & \text{if } v \in V_{\text{Eve}} \\ \min_{v'} (w(v, v') + \text{dVal}(v')) & \text{if } v \in V_{\text{Adam}} \end{cases}$$

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

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

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



Value

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

Value iteration

- Compute dVal as a greatest fixed point

Bellman equation

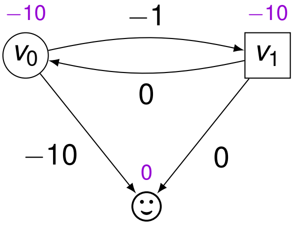
$$\text{dVal}(v) = \begin{cases} 0 & \text{if } v = ☺ \\ \max_{v'} (w(v, v') + \text{dVal}(v')) & \text{if } v \in V_{\text{Eve}} \\ \min_{v'} (w(v, v') + \text{dVal}(v')) & \text{if } v \in V_{\text{Adam}} \end{cases}$$

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

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

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



Value

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

Value iteration

- Compute dVal as a greatest fixed point
- Complexity: pseudo-polynomial

Bellman equation

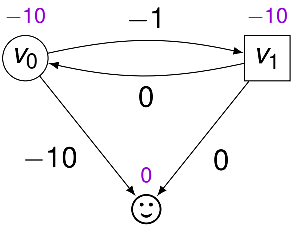
$$\text{dVal}(v) = \begin{cases} 0 & \text{if } v = ☺ \\ \max_{v'} (w(v, v') + \text{dVal}(v')) & \text{if } v \in V_{\text{Eve}} \\ \min_{v'} (w(v, v') + \text{dVal}(v')) & \text{if } v \in V_{\text{Adam}} \end{cases}$$

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

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

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



Value

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

Optimal strategy

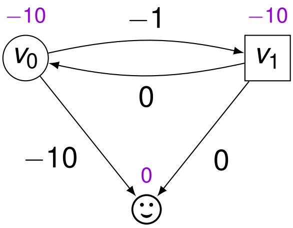
$\mathsf{dVal}^{\sigma^*}(v) \leqslant \mathsf{dVal}(v)$

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

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

$\boxed{\sigma}$ Adam τ Eve



Value

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

Optimal strategy for Adam

An optimal strategy for Adam may require finite memory.

Optimal strategy

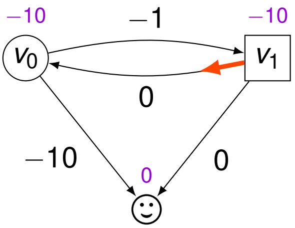
$\mathrm{dVal}^{\sigma^*}(v) \leqslant \mathrm{dVal}(v)$

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

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

$\boxed{\sigma}$ Adam $\quad$ τ Eve



Value

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

Optimal strategy for Adam

Switching strategy:

- σ_1 : reach negative cycle

Optimal strategy

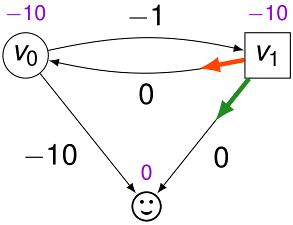
$\mathrm{dVal}^{\sigma^*}(v) \leqslant \mathrm{dVal}(v)$

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

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



Value

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

Optimal strategy for Adam

Switching strategy:

- σ_1 : reach negative cycle
- σ_2 : reach ☺

Optimal strategy

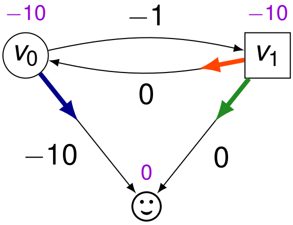
$$\mathsf{dVal}^{\sigma^*}(v) \leqslant \mathsf{dVal}(v)$$

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

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

$\boxed{\sigma}$ Adam $\quad$ (τ) Eve



Value

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

Optimal strategy for Adam

Switching strategy:

- σ_1 : reach negative cycle
- σ_2 : reach ☺

Optimal strategy

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

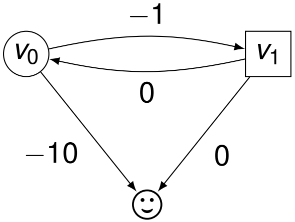
Optimal strategy for Eve

Eve has a memoryless optimal strategy.

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

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



σ Adam τ Eve

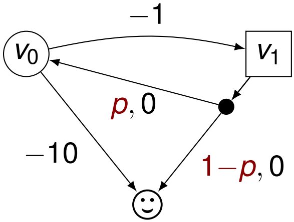
Value

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

Memoryless strategies

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

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



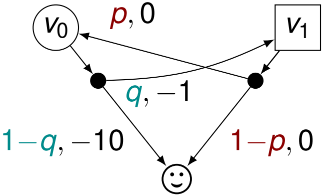
Value

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

Memoryless strategies

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

$\boxed{\sigma}$ Adam τ Eve

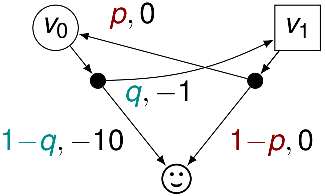


Value

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

Memoryless strategies

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



σ Adam τ Eve

Value

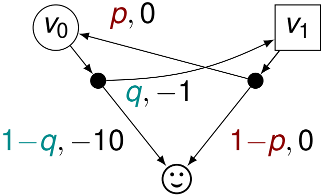
$$\overline{\text{mVal}}(v) = \inf_\sigma \underbrace{\sup_\tau \mathbb{E}^{\sigma,\tau}(\mathbf{SP})}_{\text{mVal}^\sigma(v)}$$

Bellman equation in a Markov Chain

$$\text{mVal}^{\sigma,\tau}(v) = \sum_{v'} \mathbb{P}^{\sigma,\tau}(v, v')(w(v, v') + \text{mVal}^{\sigma,\tau}(v'))$$

Memoryless strategies

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



σ Adam τ Eve

Value

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

Unicity

A unique fix point : $\text{mVal}^{\sigma,\tau}$

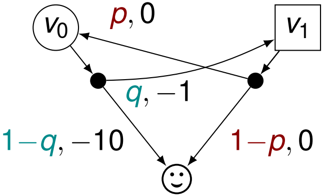
Bellman equation in a Markov Chain

$$\text{mVal}^{\sigma,\tau}(v) = \sum_{v'} \mathbb{P}^{\sigma,\tau}(v, v')(w(v, v') + \text{mVal}^{\sigma,\tau}(v'))$$

Memoryless strategies

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

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



Value

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

Compute $\text{mVal}^{\sigma,\tau}$

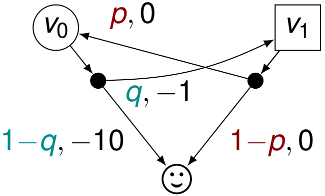
$$\text{mVal}^{\sigma,\tau}(v_1) = p \times \text{mVal}^{\sigma,\tau}(v_0)$$
$$\text{mVal}^{\sigma,\tau}(v_0) = q(\text{mVal}^{\sigma,\tau}(v_1) - 1) - 10(1-q)$$

Bellman equation in a Markov Chain

$$\text{mVal}^{\sigma,\tau}(v) = \sum_{v'} \mathbb{P}^{\sigma,\tau}(v,v')(w(v,v') + \text{mVal}^{\sigma,\tau}(v'))$$

Memoryless strategies

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



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

Value

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

Compute $\text{mVal}^{\sigma,\tau}$

$$\text{mVal}^{\sigma,\tau}(v_1) = p\frac{-q-10(1-q)}{1-pq}$$

$$\text{mVal}^{\sigma,\tau}(v_0) = \frac{-q-10(1-q)}{1-pq}$$

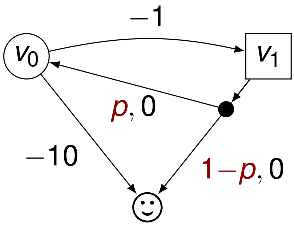
Bellman equation in a Markov Chain

$$\text{mVal}^{\sigma,\tau}(v) = \sum_{v'} \mathbb{P}^{\sigma,\tau}(v, v')(w(v, v') + \text{mVal}^{\sigma,\tau}(v'))$$

Memoryless strategies

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

$\boxed{\sigma}$ Adam $\quad$ (τ) Eve



Value

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

Compute mVal$^{\sigma,\tau}$

$$\text{mVal}^{\sigma,\tau}(v_1) = p\frac{-q-10(1-q)}{1-pq}$$

$$\text{mVal}^{\sigma,\tau}(v_0) = \frac{-q-10(1-q)}{1-pq}$$

Compute mVal$^\sigma$

- If $p < \frac{9}{10}$
- If $p \geqslant \frac{9}{10}$

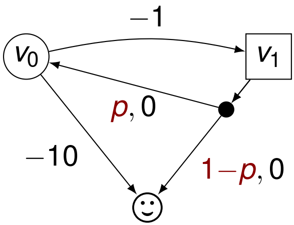
Bellman equation in a Markov Chain

$$\text{mVal}^{\sigma,\tau}(v) = \sum_{v'} \mathbb{P}^{\sigma,\tau}(v, v')(w(v, v') + \text{mVal}^{\sigma,\tau}(v'))$$

Memoryless strategies

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

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



Value

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

Compute $\text{mVal}^{\sigma,\tau}$

$$\text{mVal}^{\sigma,\tau}(v_1) = p\frac{-q-10(1-q)}{1-pq}$$

$$\text{mVal}^{\sigma,\tau}(v_0) = \frac{-q-10(1-q)}{1-pq}$$

Compute mVal^{σ}

- If $p < \frac{9}{10}$, then $q = 1$:
- If $p \geqslant \frac{9}{10}$

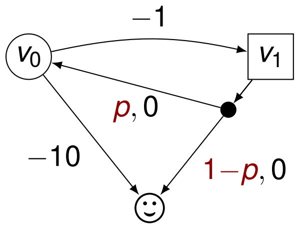
Bellman equation in a Markov Chain

$$\text{mVal}^{\sigma,\tau}(v) = \sum_{v'} \mathbb{P}^{\sigma,\tau}(v, v')(w(v, v') + \text{mVal}^{\sigma,\tau}(v'))$$

Memoryless strategies

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

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



Value

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

Compute $\text{mVal}^{\sigma,\tau}$

$$\text{mVal}^{\sigma,\tau}(v_1) = p\frac{-q-10(1-q)}{1-pq}$$

$$\text{mVal}^{\sigma,\tau}(v_0) = \frac{-q-10(1-q)}{1-pq}$$

Compute mVal^{σ}

- If $p < \frac{9}{10}$, then $q = 1$:

  $$\text{mVal}^{\sigma}(v_1) = \frac{-p}{1-p}$$

  $$\text{mVal}^{\sigma}(v_0) = \frac{-1}{1-p}$$

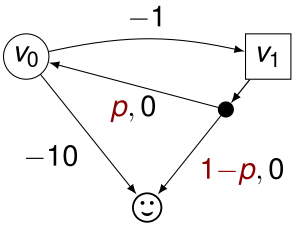
- If $p \geqslant \frac{9}{10}$

Bellman equation in a Markov Chain

$$\text{mVal}^{\sigma,\tau}(v) = \sum_{v'} \mathbb{P}^{\sigma,\tau}(v,v')(w(v,v') + \text{mVal}^{\sigma,\tau}(v'))$$

Memoryless strategies

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



σ Adam τ Eve

Value

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

Compute $\text{mVal}^{\sigma,\tau}$

$$\text{mVal}^{\sigma,\tau}(v_1) = p\frac{-q-10(1-q)}{1-pq}$$

$$\text{mVal}^{\sigma,\tau}(v_0) = \frac{-q-10(1-q)}{1-pq}$$

Compute mVal^{σ}

- If $p < \frac{9}{10}$, then $q = 1$:

  $\text{mVal}^{\sigma}(v_1) = \frac{-p}{1-p}$

  $\text{mVal}^{\sigma}(v_0) = \frac{-1}{1-p}$

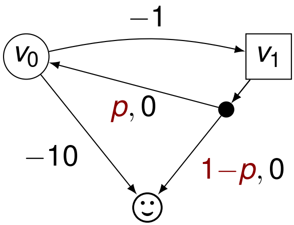
- If $p \geqslant \frac{9}{10}$, then $q = 0$:

Bellman equation in a Markov Chain

$$\text{mVal}^{\sigma,\tau}(v) = \sum_{v'} \mathbb{P}^{\sigma,\tau}(v, v')(w(v, v') + \text{mVal}^{\sigma,\tau}(v'))$$

Memoryless strategies

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



Compute $\text{mVal}^{\sigma,\tau}$

$$\text{mVal}^{\sigma,\tau}(v_1) = p\frac{-q-10(1-q)}{1-pq}$$
$$\text{mVal}^{\sigma,\tau}(v_0) = \frac{-q-10(1-q)}{1-pq}$$

σ Adam τ Eve

Value

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

Compute mVal^σ

- If $p < \frac{9}{10}$, then $q = 1$:
  $$\text{mVal}^\sigma(v_1) = \frac{-p}{1-p}$$
  $$\text{mVal}^\sigma(v_0) = \frac{-1}{1-p}$$
- If $p \geqslant \frac{9}{10}$, then $q = 0$:
  $$\text{mVal}^\sigma(v_1) = -10p$$
  $$\text{mVal}^\sigma(v_0) = -10$$

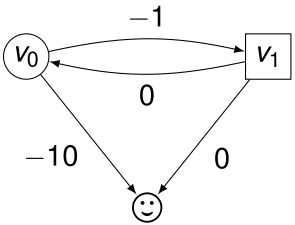
Bellman equation in a Markov Chain

$$\text{mVal}^{\sigma,\tau}(v) = \sum_{v'} \mathbb{P}^{\sigma,\tau}(v,v')(w(v,v') + \text{mVal}^{\sigma,\tau}(v'))$$

Memoryless strategies

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

$\boxed{\sigma}$ Adam τ Eve



Value

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

Compute $\text{mVal}^{\sigma,\tau}$

$$\text{mVal}^{\sigma,\tau}(v_1) = p\frac{-q-10(1-q)}{1-pq}$$

$$\text{mVal}^{\sigma,\tau}(v_0) = \frac{-q-10(1-q)}{1-pq}$$

Compute mVal^{σ}

- If $p < \frac{9}{10}$, then $q = 1$:

  $$\text{mVal}^{\sigma}(v_1) = \frac{-p}{1-p}$$

  $$\text{mVal}^{\sigma}(v_0) = \frac{-1}{1-p}$$

- If $p \geqslant \frac{9}{10}$, then $q = 0$:

  $$\text{mVal}^{\sigma}(v_1) = -10p$$

  $$\text{mVal}^{\sigma}(v_0) = -10$$

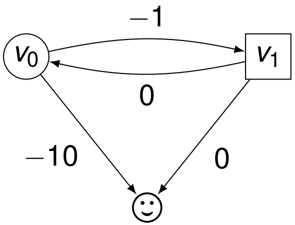
Bellman equation in a Markov Chain

$$\text{mVal}^{\sigma,\tau}(v) = \sum_{v'} \mathbb{P}^{\sigma,\tau}(v, v')(w(v, v') + \text{mVal}^{\sigma,\tau}(v'))$$

Memoryless strategies

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

$\boxed{\sigma}$ Adam τ Eve



Value

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

Compute $\text{mVal}^{\sigma,\tau}$

$$\text{mVal}^{\sigma,\tau}(v_1) = p\frac{-q-10(1-q)}{1-pq}$$

$$\text{mVal}^{\sigma,\tau}(v_0) = \frac{-q-10(1-q)}{1-pq}$$

Compute mVal^{σ}

- If $p < \frac{9}{10}$, then $q = 1$:

  $\text{mVal}^{\sigma}(v_1) = \frac{-p}{1-p} < -9$

  $\text{mVal}^{\sigma}(v_0) = \frac{-1}{1-p}$

- If $p \geqslant \frac{9}{10}$, then $q = 0$:

  $\text{mVal}^{\sigma}(v_1) = -10p$

  $\text{mVal}^{\sigma}(v_0) = -10$

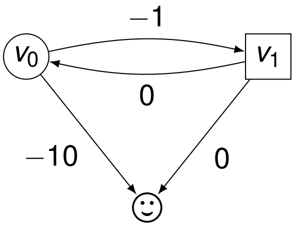
Bellman equation in a Markov Chain

$$\text{mVal}^{\sigma,\tau}(v) = \sum_{v'} \mathbb{P}^{\sigma,\tau}(v,v')(w(v,v') + \text{mVal}^{\sigma,\tau}(v'))$$

Memoryless strategies

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

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



Value

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

Compute mVal^{σ}

- If $p < \frac{9}{10}$, then $q = 1$:

  $\text{mVal}^{\sigma}(v_1) = \frac{-p}{1-p} < -9$

  $\text{mVal}^{\sigma}(v_0) = \frac{-1}{1-p} < -10$

- If $p \geqslant \frac{9}{10}$, then $q = 0$:

  $\text{mVal}^{\sigma}(v_1) = -10p$

  $\text{mVal}^{\sigma}(v_0) = -10$

Compute $\text{mVal}^{\sigma,\tau}$

$\text{mVal}^{\sigma,\tau}(v_1) = p\frac{-q-10(1-q)}{1-pq}$

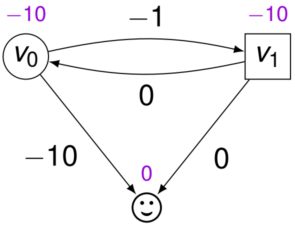
$\text{mVal}^{\sigma,\tau}(v_0) = \frac{-q-10(1-q)}{1-pq}$

Bellman equation in a Markov Chain

$$\text{mVal}^{\sigma,\tau}(v) = \sum_{v'} \mathbb{P}^{\sigma,\tau}(v,v')(w(v,v') + \text{mVal}^{\sigma,\tau}(v'))$$

Memoryless strategies

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



σ Adam τ Eve

Value

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

Compute $\text{mVal}^{\sigma,\tau}$

$$\text{mVal}^{\sigma,\tau}(v_1) = p\frac{-q-10(1-q)}{1-pq}$$
$$\text{mVal}^{\sigma,\tau}(v_0) = \frac{-q-10(1-q)}{1-pq}$$

Compute mVal^{σ}

- If $p < \frac{9}{10}$, then $q = 1$:
  $$\text{mVal}^{\sigma}(v_1) = \frac{-p}{1-p} < -9$$
  $$\text{mVal}^{\sigma}(v_0) = \frac{-1}{1-p} < -10$$
- If $p \geqslant \frac{9}{10}$, then $q = 0$:
  $$\text{mVal}^{\sigma}(v_1) = -10p$$
  $$\text{mVal}^{\sigma}(v_0) = -10$$

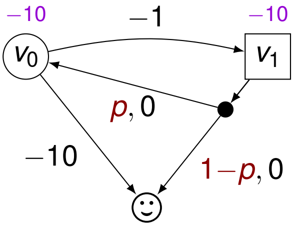
Bellman equation in a Markov Chain

$$\text{mVal}^{\sigma,\tau}(v) = \sum_{v'} \mathbb{P}^{\sigma,\tau}(v,v')(w(v,v') + \text{mVal}^{\sigma,\tau}(v'))$$

Memoryless strategies

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

$\boxed{\sigma}$ Adam $\quad$ τ Eve



Value

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

Value in a MDP

Computable in polynomial time

Bellman equation in a Markov Chain

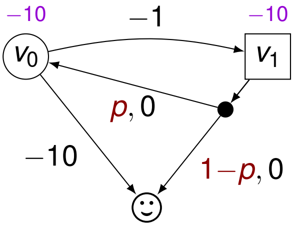
$$\text{mVal}^{\sigma,\tau}(v) = \sum_{v'} \mathbb{P}^{\sigma,\tau}(v, v')(w(v, v') + \text{mVal}^{\sigma,\tau}(v'))$$

Stochastic Shortest Paths and Weight-Bounded Properties in Markov Decision Processes, C. Baier, N. Bertrand, C. Dubslaff, D. Gburek and O. Sankur, 2018, LICS.

Memoryless strategies

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

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



Value

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

Value in a MDP

Computable in polynomial time

ε-optimal strategy

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

Bellman equation in a Markov Chain

$$\mathsf{mVal}^{\sigma,\tau}(v) = \sum_{v'} \mathbb{P}^{\sigma,\tau}(v, v')(w(v, v') + \mathsf{mVal}^{\sigma,\tau}(v'))$$

Stochastic Shortest Paths and Weight-Bounded Properties in Markov Decision Processes, C. Baier, N. Bertrand, C. Dubslaff, D. Gburek and O. Sankur, 2018, LICS.

Contribution

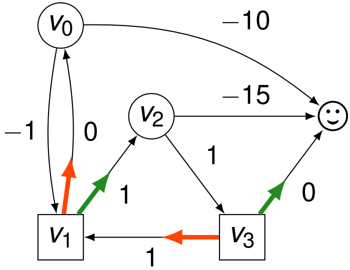
$$\text{dVal} = \overline{\text{mVal}}$$

Memoryless simulate deterministic

Claim

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

Memoryless simulate deterministic



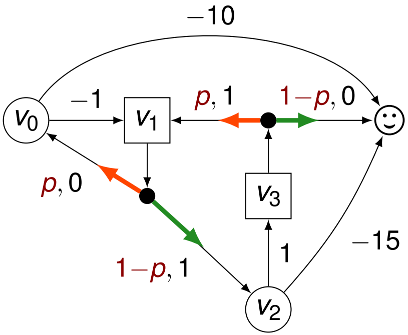
Claim

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

Strategy ρ_p

Let $\langle \sigma_1, \sigma_2 \rangle$ be an optimal switching strategy,

Memoryless simulate deterministic



Claim

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

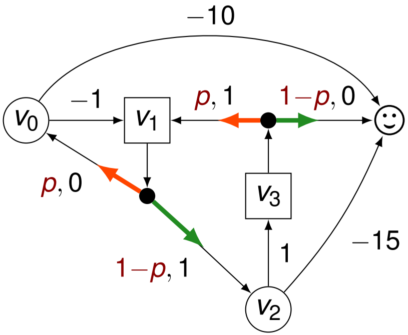
Strategy ρ_p

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

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

Memoryless simulate deterministic

□ Adam ○ Eve



Claim

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

Properties of ρ_p

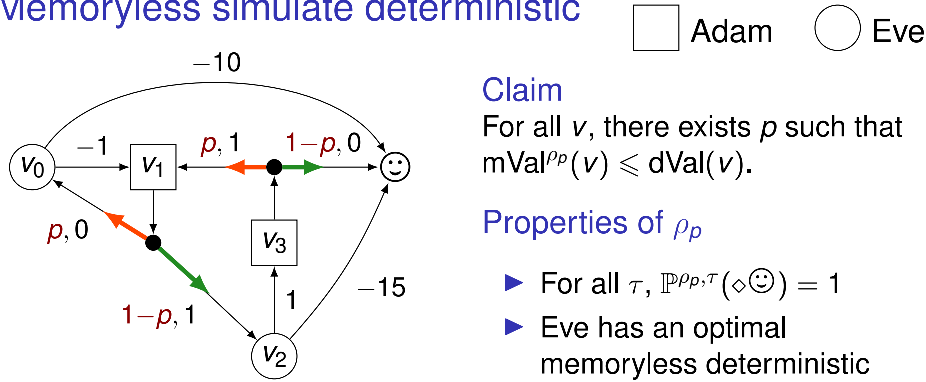
- For all τ, $\mathbb{P}^{\rho_p,\tau}(\diamond ☺) = 1$

Strategy ρ_p

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

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

Memoryless simulate deterministic



Claim

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

Properties of ρ_p

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

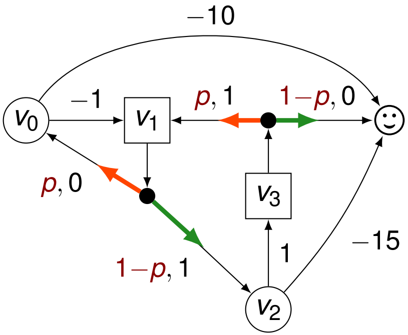
Strategy ρ_p

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

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

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

Memoryless simulate deterministic



Claim

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

Problem

Presence of non-negative cycles

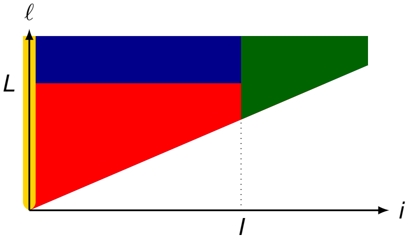
Strategy ρ_p

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

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

Memoryless simulate deterministic

□ Adam ○ Eve



Claim

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

Problem

Presence of non-negative cycles

Tool for the proof

Control the non-negative cycles with a partition of plays

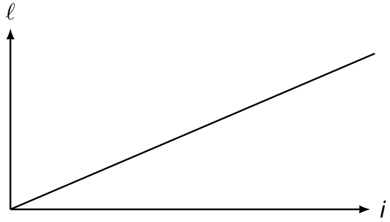
Strategy ρ_p

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

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

Focus on the partition of plays

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

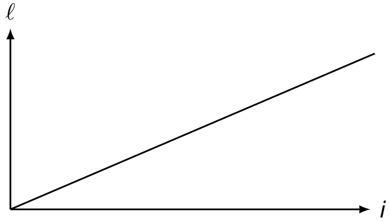


Focus on the partition of plays

ℓ size of play reaching the target

i number of non-negative cycles

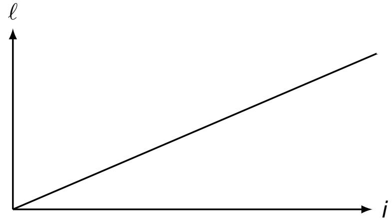
Fix a memoryless strategy for Eve



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

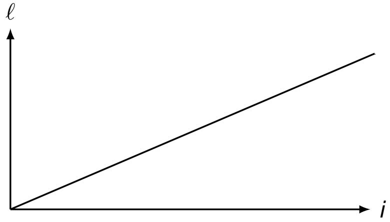
Focus on the partition of plays

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

Fix a memoryless strategy for Eve

Good zones

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



Zones to control

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

Focus on the partition of plays

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

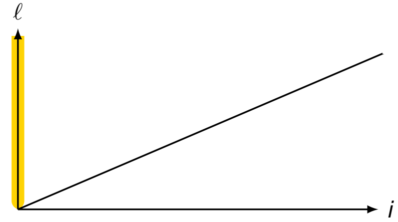
Fix a memoryless strategy for Eve

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

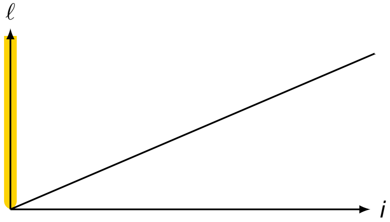
Fix a memoryless strategy for Eve

Yellow zone

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

Good zones

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



Zones to control

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

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

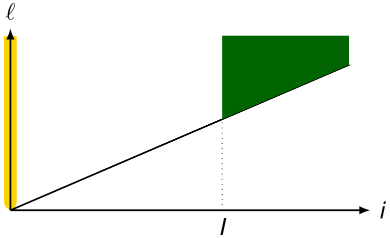
All plays conforming to σ_1
Weight of each play is $\leqslant$ dVal

Green zone

Plays contain many non-negative cycles

Good zones

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



Zones to control

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

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

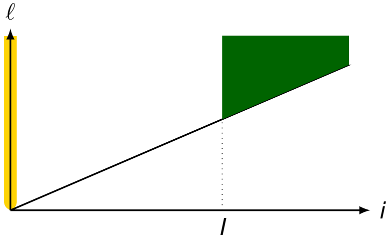
All plays conforming to σ_1
Weight of each play is $\leqslant$ dVal

Green zone

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

Good zones

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



Zones to control

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

Focus on the partition of plays

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

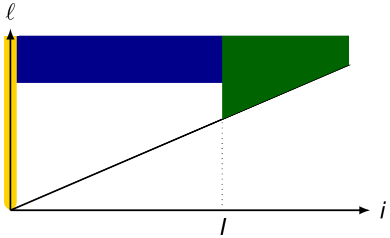
Fix a memoryless strategy for Eve

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

Zones to control

$\mathbb{E}($**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

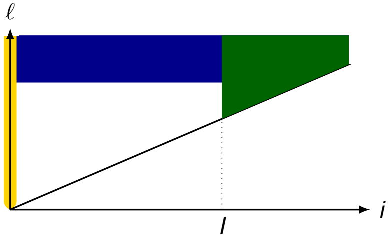
All plays conforming to σ_1
Weight of each play is $\leqslant$ dVal

Green zone

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

Good zones

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

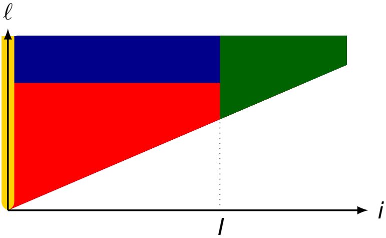
All plays conforming to σ_1
Weight of each play is $\leqslant$ dVal

Green zone

Plays contain many non-negative cycles
$\forall p, \exists l$ 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
$\forall p, l, \ \exists L$ s.t. weights $\leqslant$ dVal

Red zone

Rest of plays

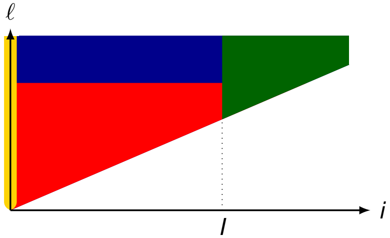
Focus on the partition of plays

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

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

Zones to control

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

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

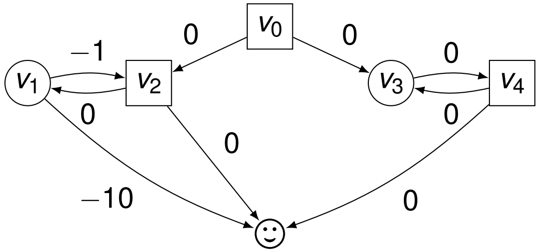
For all v and for all memoryless strategies ρ, there exists a deterministic strategy σ such that

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

Deterministic simulate memoryless

Claim

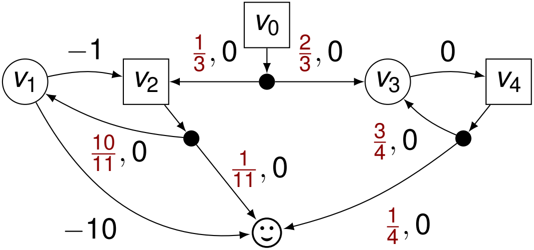
For all v and for all memoryless strategies ρ, there exists a deterministic strategy σ such that $\mathrm{dVal}^{\sigma}(v) \leqslant \mathrm{mVal}^{\rho}(v)$



Deterministic simulate memoryless

Claim

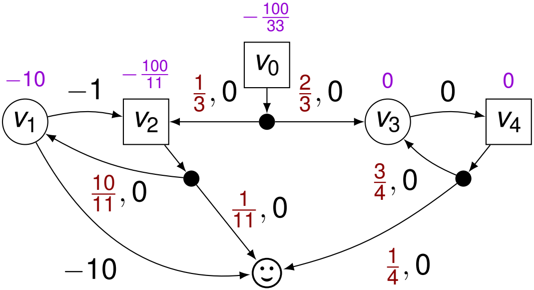
For all v and for all memoryless strategies ρ, there exists a deterministic strategy σ such that $\mathrm{dVal}^{\sigma}(v) \leqslant \mathrm{mVal}^{\rho}(v)$



Deterministic simulate memoryless

Claim

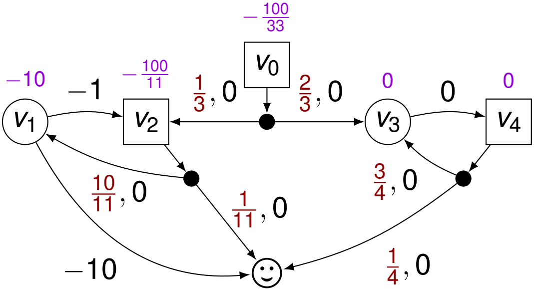
For all v and for all memoryless strategies ρ, there exists a deterministic strategy σ such that $\mathrm{dVal}^{\sigma}(v) \leqslant \mathrm{mVal}^{\rho}(v)$



Deterministic simulate memoryless □ Adam ○ Eve

Claim

For all v and for all memoryless strategies ρ, there exists a deterministic strategy σ such that $\mathrm{dVal}^{\sigma}(v) \leqslant \mathrm{mVal}^{\rho}(v)$



First Idea: Adam's strategy

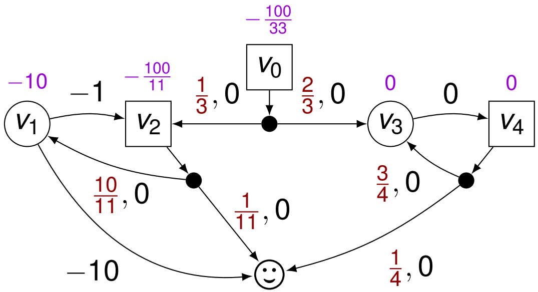
- v_2: 10 times v_1 and 1 time ☺

Deterministic simulate memoryless

□ Adam ○ Eve

Claim

For all v and for all memoryless strategies ρ, there exists a deterministic strategy σ such that $\mathrm{dVal}^{\sigma}(v) \leqslant \mathrm{mVal}^{\rho}(v)$



First Idea: Adam's strategy

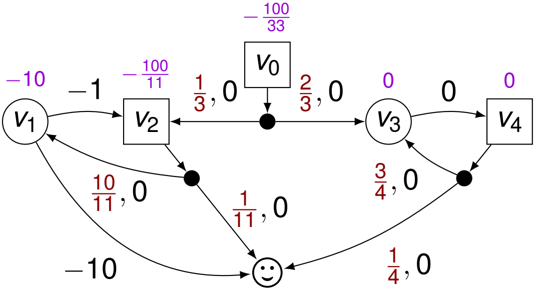
- v_2: 10 times v_1 and 1 time ☺
- v_4: 3 times v_3 and 1 time ☺

Deterministic simulate memoryless

□ Adam ○ Eve

Claim

For all v and for all memoryless strategies ρ, there exists a deterministic strategy σ such that $\mathrm{dVal}^{\sigma}(v) \leqslant \mathrm{mVal}^{\rho}(v)$



First Idea: Adam's strategy

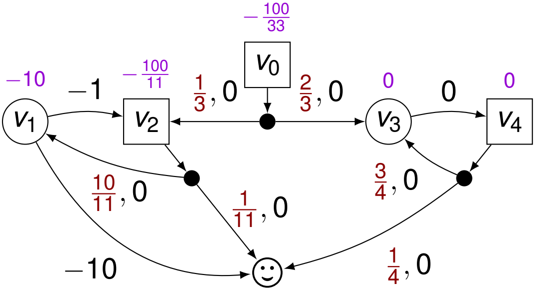
- v_2: 10 times v_1 and 1 time ☺
- v_4: 3 times v_3 and 1 time ☺
- v_0: 2 times v_3 and 1 time v_2

Deterministic simulate memoryless

□ Adam ○ Eve

Claim

For all v and for all memoryless strategies ρ, there exists a deterministic strategy σ such that $\mathrm{dVal}^{\sigma}(v) \leqslant \mathrm{mVal}^{\rho}(v)$



First Idea: Adam's strategy

- v_2: 10 times v_1 and 1 time ☺
- v_4: 3 times v_3 and 1 time ☺
- v_0: 2 times v_3 and 1 time v_2

Counter-example

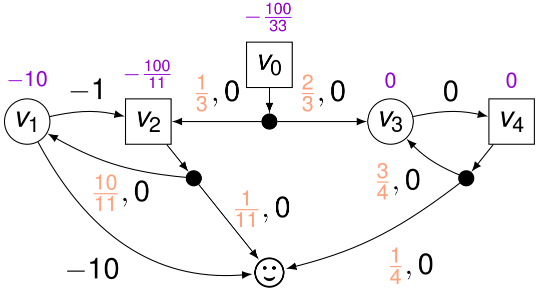
$$\mathrm{dVal}^{\sigma}(v_1) = 0 > -\frac{100}{33} = \mathrm{mVal}^{\rho}(v_1)$$

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

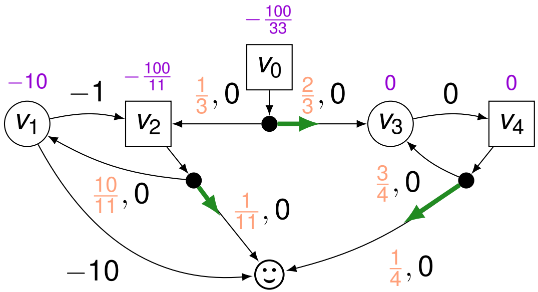
- Build a switching strategy $\sigma = \langle \sigma_1, \sigma_2 \rangle$ for ρ
- Value iteration for the fixpoint that gives the value

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

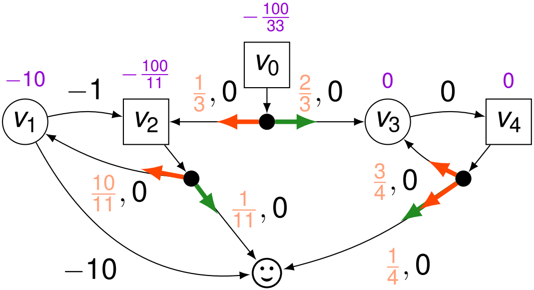
- Build a switching strategy $\sigma = \langle \sigma_1, \sigma_2 \rangle$ for ρ
- Value iteration for the fixpoint that gives the value

Deterministic simulate memoryless

□ Adam ○ Eve

Claim

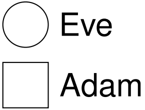
For all v and for all memoryless strategies ρ, there exists a deterministic strategy σ such that $\mathrm{dVal}^{\sigma}(v) \leqslant \mathrm{mVal}^{\rho}(v)$

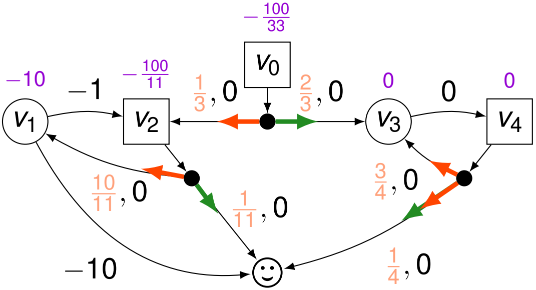


Tools for the proof

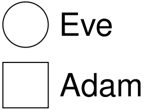
- Build a switching strategy $\sigma = \langle \sigma_1, \sigma_2 \rangle$ for ρ
- Value iteration for the fixpoint that gives the value

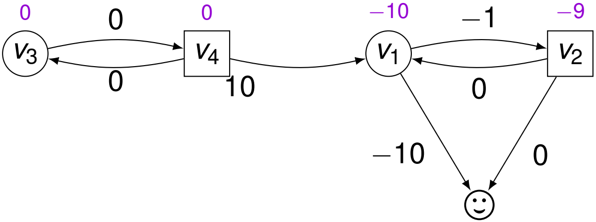
How do we choose the good vertex?





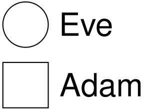
How do we choose the good vertex?

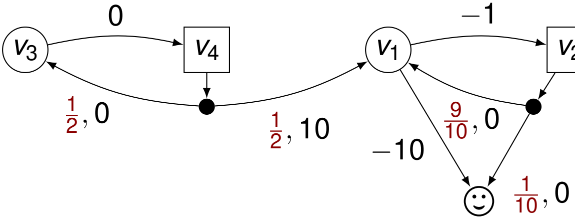




How do we choose the good vertex?

Let ρ be a memoryless strategy

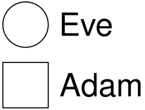


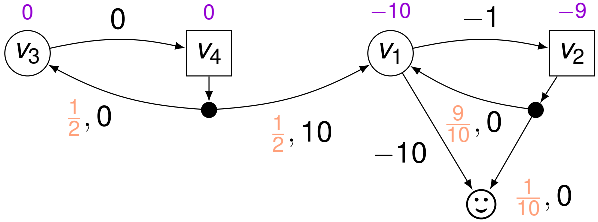


How do we choose the good vertex?

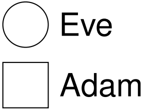
Let ρ be a memoryless strategy

We know mVal^{ρ} for all vertices.





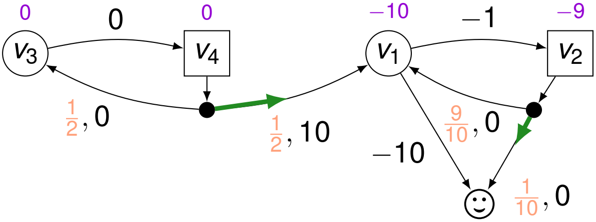
How do we choose the good vertex?



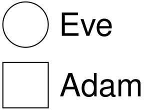
Let ρ be a memoryless strategy

Fix σ_2 an attractor strategy

We know mVal^{ρ} for all vertices.



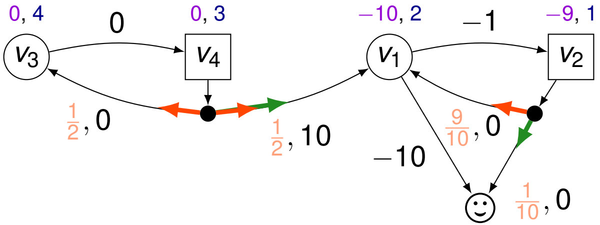
How do we choose the good vertex?



Let ρ be a memoryless strategy

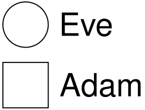
Fix σ_2 an attractor strategy

We know mVal^{ρ} for all vertices.



Attractor distance

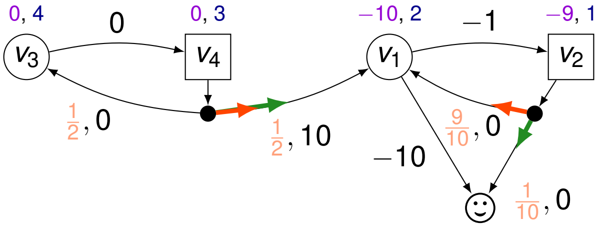
How do we choose the good vertex?



Let ρ be a memoryless strategy

Fix σ_2 an attractor strategy

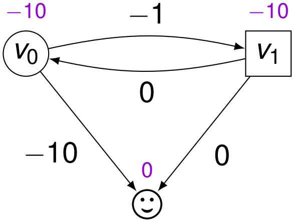
We know mVal$^{\rho}$ for all vertices.



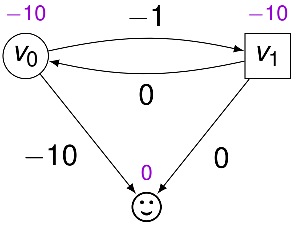
Attractor distance

$\sigma_1(v)$ chooses the minimal distance

Optimality

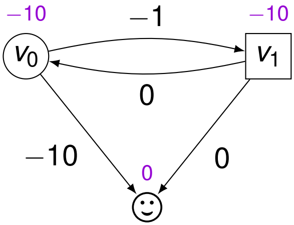


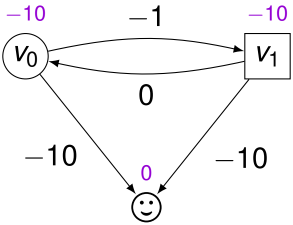
Optimality



No optimal memoryless strategy

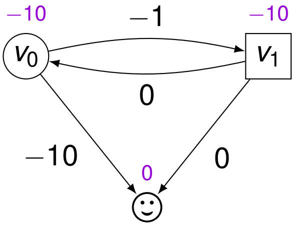
Optimality



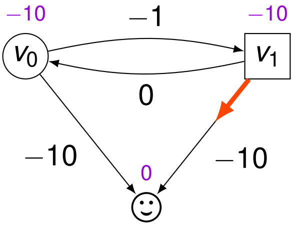


No optimal memoryless strategy

Optimality

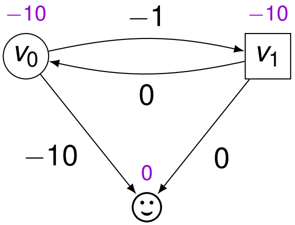


No optimal memoryless strategy

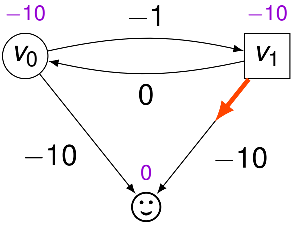


Optimal memoryless strategy

Optimality



No optimal memoryless strategy



Optimal memoryless strategy

Optimality proposition

1. We can characterize and test in polynomial time the existence of an optimal memoryless strategy.
2. A memoryless strategy optimal is deterministic.

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?