Weighted Timed Games: Decidability, Randomisation and Robustness

PhD defense

Julie Parreaux

Aix–Marseille Université

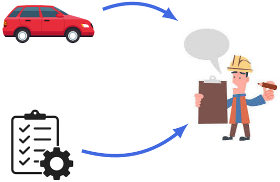
October 24, 2023

From verification to synthesis

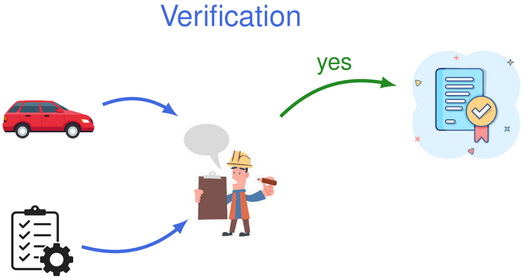
Critical software systems

From verification to synthesis

Verification

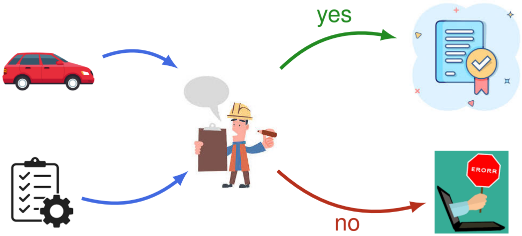


From verification to synthesis



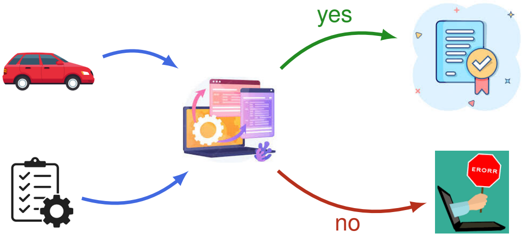
From verification to synthesis

Verification

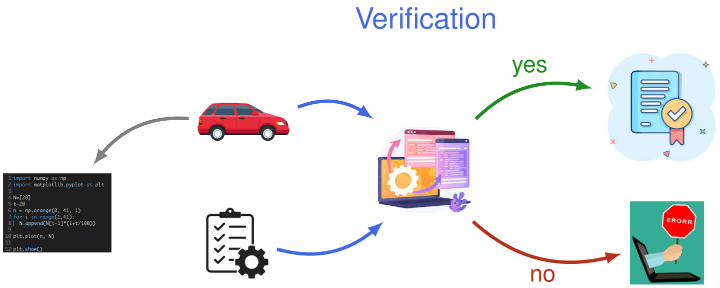


From verification to synthesis

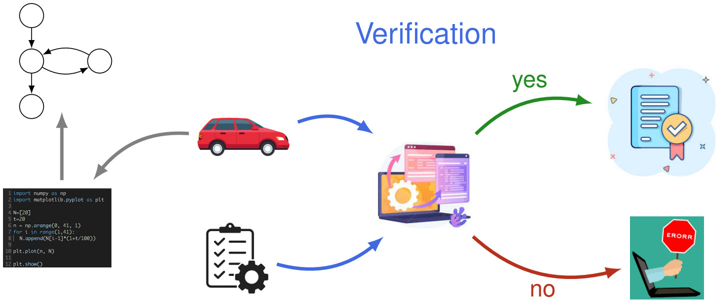
Verification



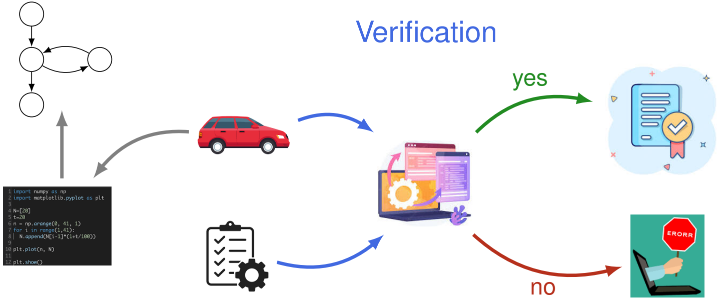
From verification to synthesis



From verification to synthesis

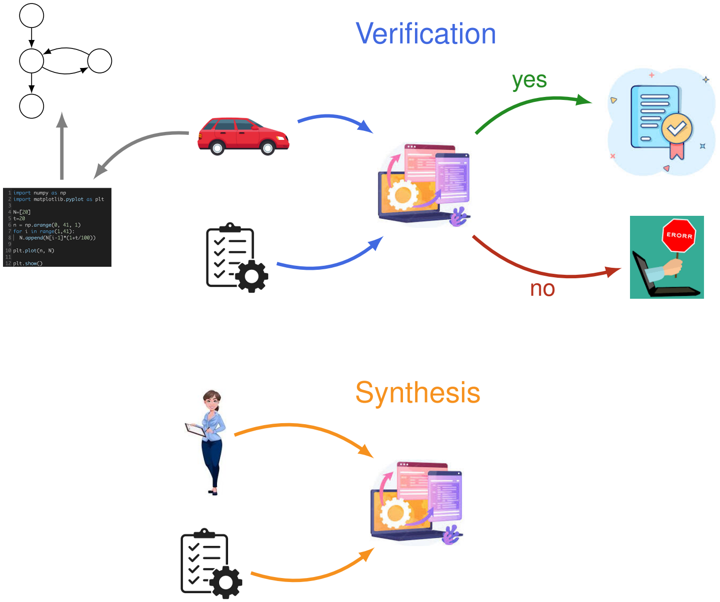


From verification to synthesis

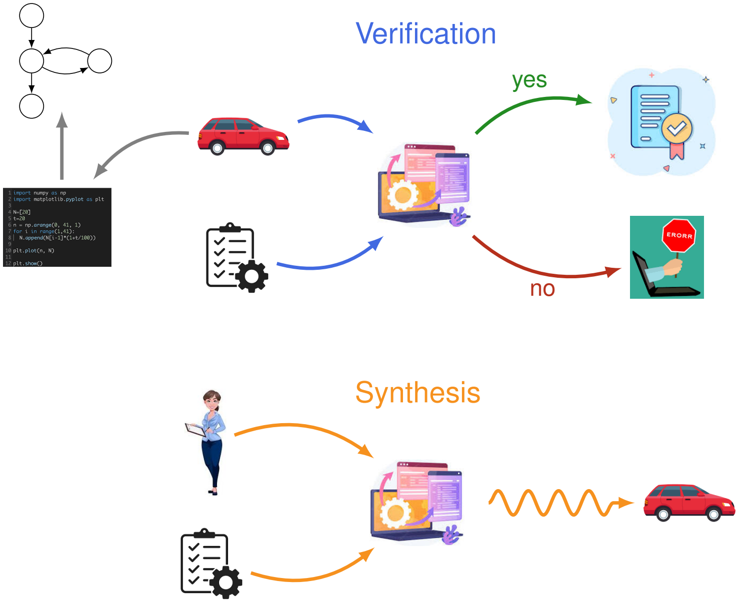


Synthesis

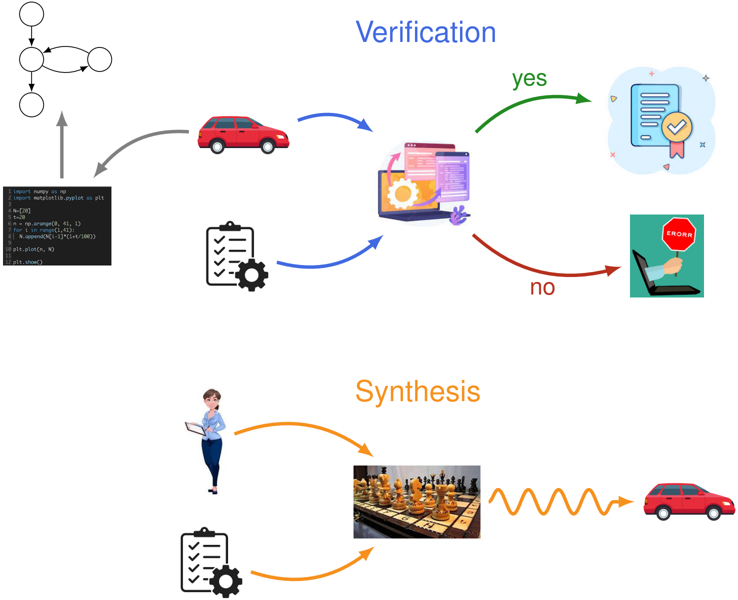
From verification to synthesis



From verification to synthesis



From verification to synthesis



Synthesis based on games on graphs

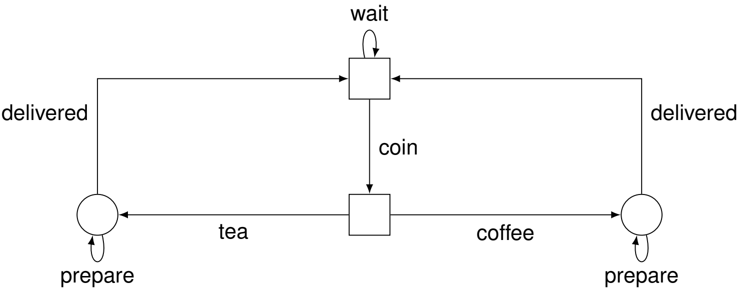
Synthesis based on games on graphs

Synthesis based on games on graphs

If a coin is inserted, then a drink (coffee or tea) will eventually be delivered

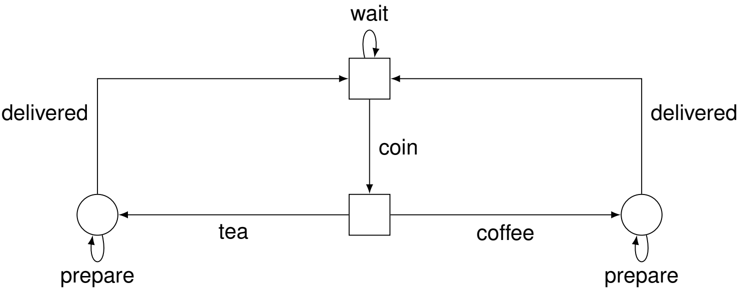
Synthesis based on games on graphs

If a coin is inserted, then a drink (coffee or tea) will eventually be delivered



Synthesis based on games on graphs

If a coin is inserted, then a drink (coffee or tea) will eventually be delivered

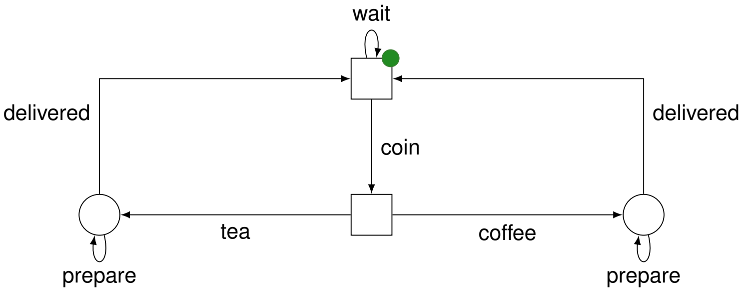


Game synthesis

For the system's point of view: uncontrollable actions from the environment

Synthesis based on games on graphs

If a coin is inserted, then a drink (coffee or tea) will eventually be delivered

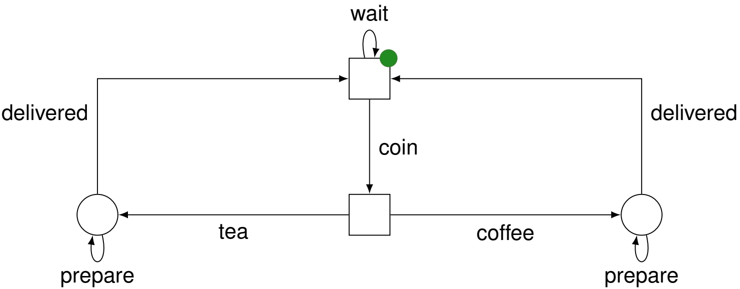


Game synthesis

For the system's point of view: uncontrollable actions from the environment

Synthesis based on games on graphs

If a coin is inserted, then a drink (coffee or tea) will eventually be delivered

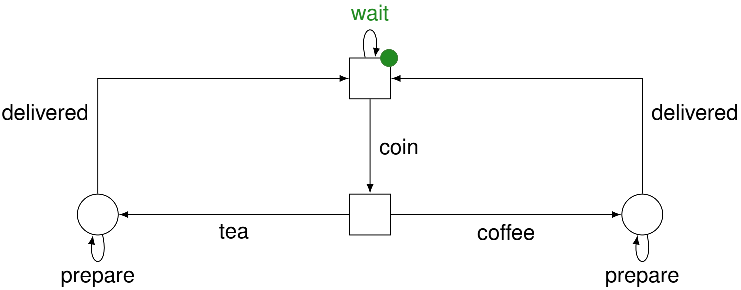


Game synthesis

For the system's point of view: uncontrollable actions from the environment

Synthesis based on games on graphs

If a coin is inserted, then a drink (coffee or tea) will eventually be delivered

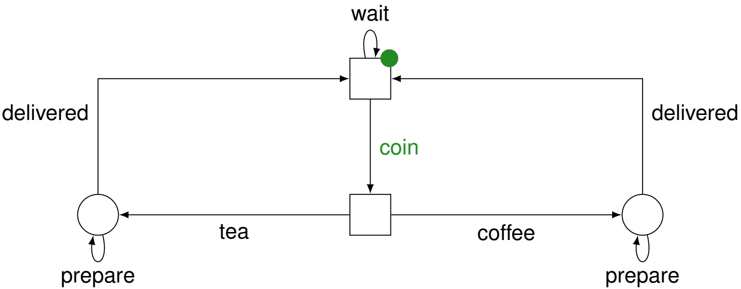


Game synthesis

For the system's point of view: uncontrollable actions from the environment

Synthesis based on games on graphs

If a coin is inserted, then a drink (coffee or tea) will eventually be delivered

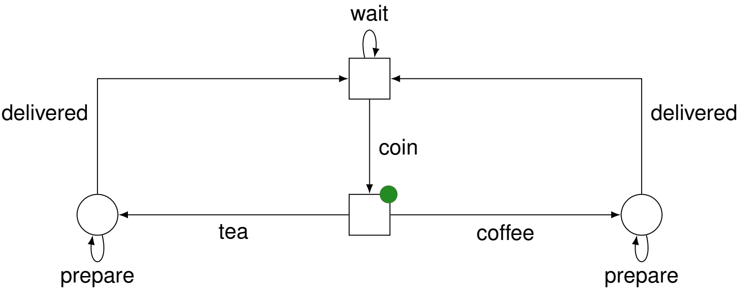


Game synthesis

For the system's point of view: uncontrollable actions from the environment

Synthesis based on games on graphs

If a coin is inserted, then a drink (coffee or tea) will eventually be delivered

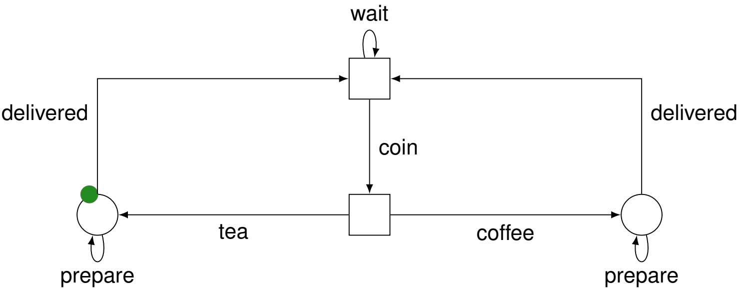


Game synthesis

For the system's point of view: uncontrollable actions from the environment

Synthesis based on games on graphs

If a coin is inserted, then a drink (coffee or tea) will eventually be delivered

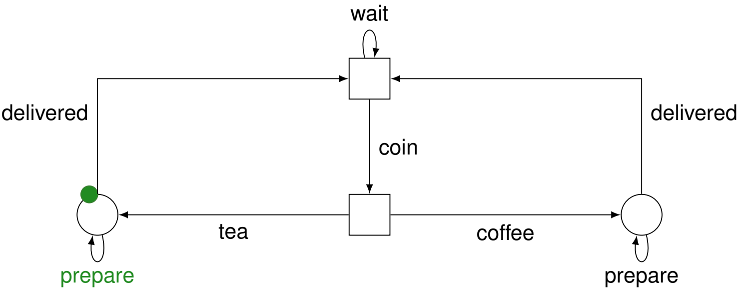


Game synthesis

For the system's point of view: uncontrollable actions from the environment

Synthesis based on games on graphs

If a coin is inserted, then a drink (coffee or tea) will eventually be delivered

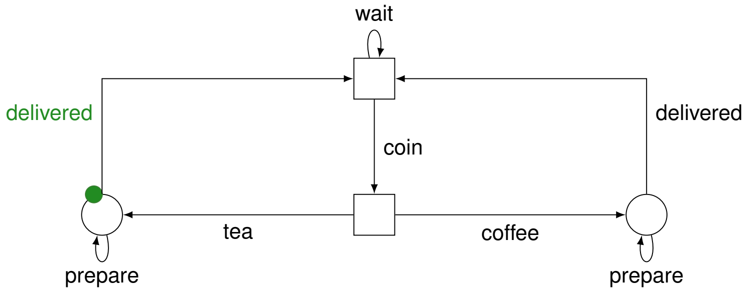


Game synthesis

For the system's point of view: uncontrollable actions from the environment

Synthesis based on games on graphs

If a coin is inserted, then a drink (coffee or tea) will eventually be delivered

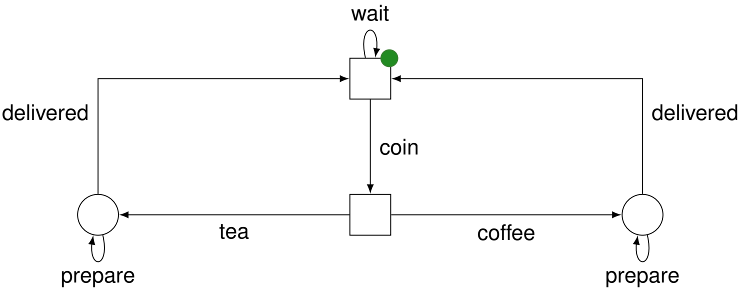


Game synthesis

For the system's point of view: uncontrollable actions from the environment

Synthesis based on games on graphs

If a coin is inserted, then a drink (coffee or tea) will eventually be delivered

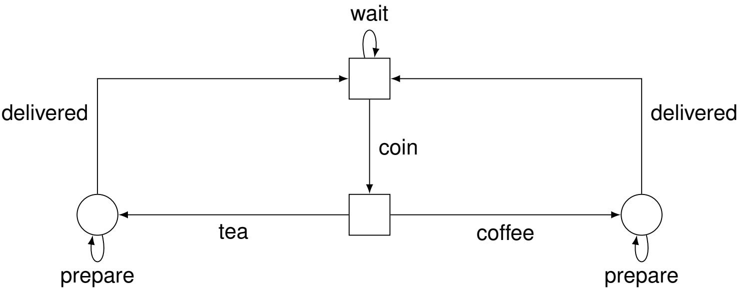


Game synthesis

For the system's point of view: uncontrollable actions from the environment

Synthesis based on games on graphs

If a coin is inserted, then a drink (coffee or tea) will ~~eventually~~ be delivered
within 20 seconds

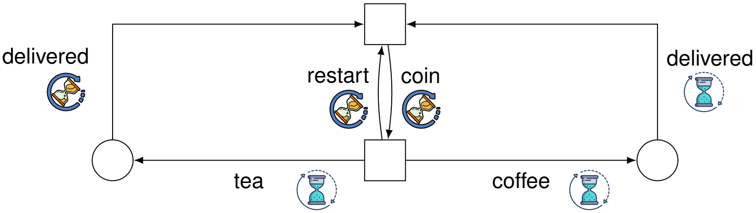


Game synthesis

For the system's point of view: uncontrollable actions from the environment

Synthesis based on games on graphs

If a coin is inserted, then a drink (coffee or tea) will ~~eventually~~ be delivered
within 20 seconds



Game synthesis

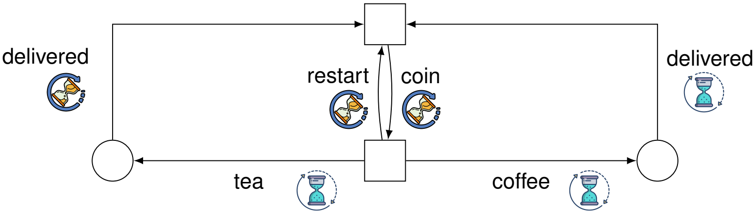
For the system's point of view: uncontrollable actions from the environment

Timed game synthesis

Timed properties requirement over each action

Synthesis based on games on graphs

If ~~a coin~~ is inserted, then a drink (coffee or tea) will ~~eventually~~ be delivered with a cost $\leqslant$ 4€
4€ within 20 seconds



Game synthesis

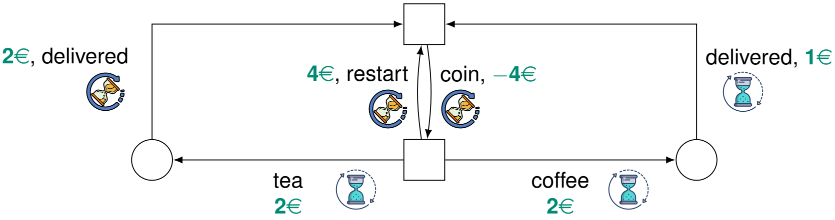
For the system's point of view: uncontrollable actions from the environment

Timed game synthesis

Timed properties requirement over each action

Synthesis based on games on graphs

If ~~a coin~~ 4€ is inserted, then a drink (coffee or tea) will ~~eventually~~ within 20 seconds be delivered with a cost $\leqslant$ 4€



Game synthesis

For the system's point of view: uncontrollable actions from the environment

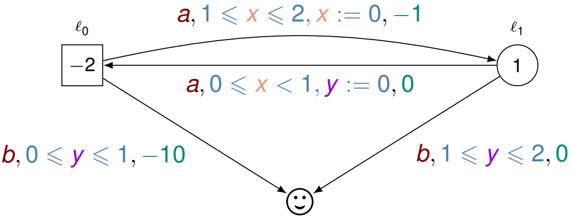
Timed game synthesis

Timed properties requirement over each action

Weighted timed game synthesis

Each action has a cost for the system

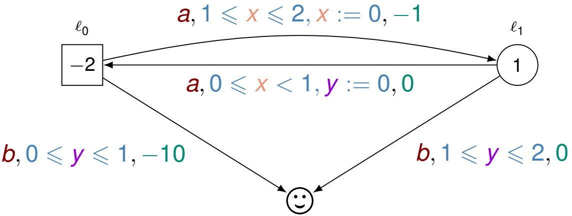
Weighted Timed Games



Weighted Timed Games

○ Min □ Max

☺ target (T)

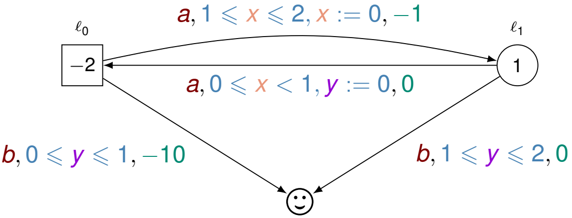


Play ρ $(\ell_1, \begin{bmatrix} x \mapsto 0 \\ y \mapsto 0 \end{bmatrix})$

Weighted Timed Games

◯ Min ☐ Max

☺ target (T)

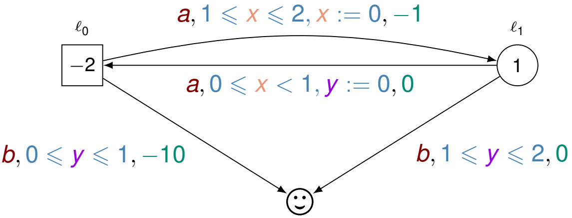


Play ρ $\quad (\ell_1, \begin{bmatrix} 0 \\ 0 \end{bmatrix})$

Weighted Timed Games

Min Max

☺ target (T)

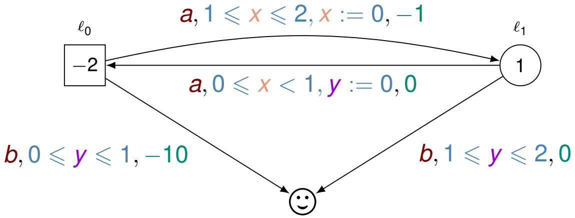


Play ρ $\quad (\ell_1, \begin{bmatrix}0\\0\end{bmatrix}) \xrightarrow{0.5,\ a}$

Weighted Timed Games

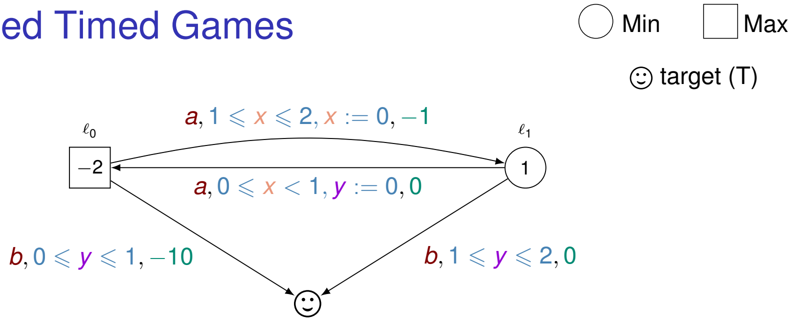
○ Min □ Max

☺ target (T)



Play ρ $\quad (\ell_1, \begin{bmatrix} 0 \\ 0 \end{bmatrix}) \xrightarrow{0.5,\ a} (\ell_0, \begin{bmatrix} 0.5 \\ 0 \end{bmatrix})$

Weighted Timed Games

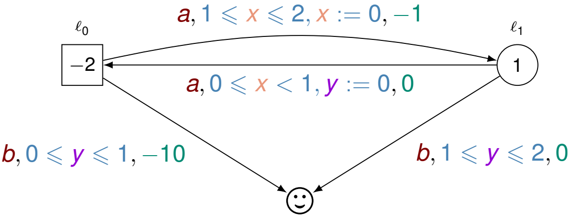


Play ρ $\quad (\ell_1, \begin{bmatrix} 0 \\ 0 \end{bmatrix}) \xrightarrow{0.5,\ a} (\ell_0, \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}) \xrightarrow{1.25,\ a} (\ell_1, \begin{bmatrix} 0 \\ 1.25 \end{bmatrix}) \xrightarrow{1/3,\ b} (☺, \begin{bmatrix} 1/3 \\ 19/12 \end{bmatrix})$

Weighted Timed Games

Min Max

☺ target (T)



Play ρ

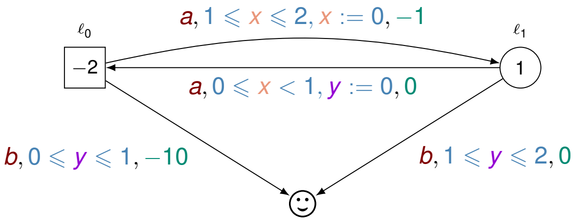
$$(\ell_1, \begin{bmatrix}0\\0\end{bmatrix}) \xrightarrow{0.5,\ a} (\ell_0, \begin{bmatrix}0.5\\0\end{bmatrix}) \xrightarrow{1.25,\ a} (\ell_1, \begin{bmatrix}0\\1.25\end{bmatrix}) \xrightarrow{1/3,\ b} (\smiley, \begin{bmatrix}1/3\\19/12\end{bmatrix})$$

$+ \qquad +$

Weighted Timed Games

◯ Min ☐ Max

☺ target (T)



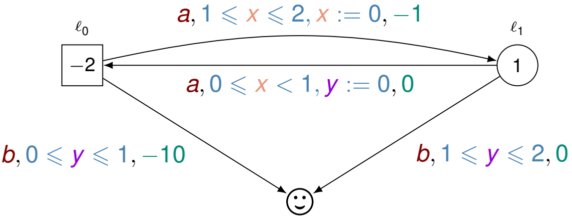
Play ρ $$(\ell_1, \begin{bmatrix} 0 \\ 0 \end{bmatrix}) \xrightarrow{0.5,\ a} (\ell_0, \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}) \xrightarrow{1.25,\ a} (\ell_1, \begin{bmatrix} 0 \\ 1.25 \end{bmatrix}) \xrightarrow{1/3,\ b} (☺, \begin{bmatrix} 1/3 \\ 19/12 \end{bmatrix})$$

$$0 \quad + \quad\quad\quad +$$

Weighted Timed Games

○ Min   □ Max

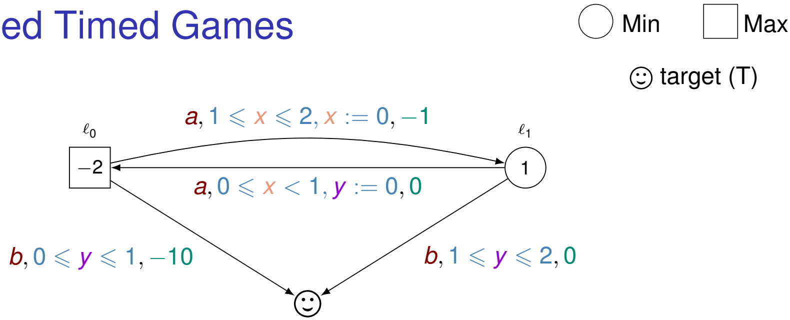
☺ target (T)



Play ρ $\quad (\ell_1, \begin{bmatrix} 0 \\ 0 \end{bmatrix}) \xrightarrow{0.5,\ a} (\ell_0, \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}) \xrightarrow{1.25,\ a} (\ell_1, \begin{bmatrix} 0 \\ 1.25 \end{bmatrix}) \xrightarrow{1/3,\ b} (☺, \begin{bmatrix} 1/3 \\ 19/12 \end{bmatrix})$

$1 \times 0.5 + 0 \qquad + \qquad\qquad +$

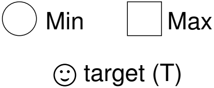
Weighted Timed Games

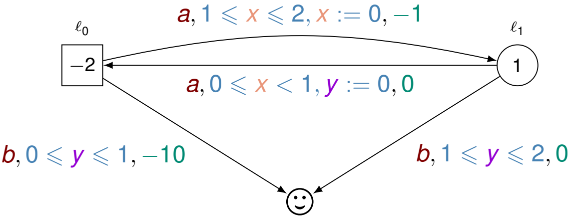


Play ρ $\quad (\ell_1, \begin{bmatrix} 0 \\ 0 \end{bmatrix}) \xrightarrow{0.5,\ a} (\ell_0, \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}) \xrightarrow{1.25,\ a} (\ell_1, \begin{bmatrix} 0 \\ 1.25 \end{bmatrix}) \xrightarrow{1/3,\ b} (☺, \begin{bmatrix} 1/3 \\ 19/12 \end{bmatrix}) \rightsquigarrow -\frac{8}{3}$

$$1 \times 0.5 + 0 \quad + \quad -2 \times 1.25 - 1 \quad + \quad 1 \times \tfrac{1}{3} + 0$$

Weighted Timed Games





Play ρ $\quad (\ell_1, \begin{bmatrix} 0 \\ 0 \end{bmatrix}) \xrightarrow{0.5,\ a} (\ell_0, \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}) \xrightarrow{1.25,\ a} (\ell_1, \begin{bmatrix} 0 \\ 1.25 \end{bmatrix}) \xrightarrow{1/3,\ b} (☺, \begin{bmatrix} 1/3 \\ 19/12 \end{bmatrix})$

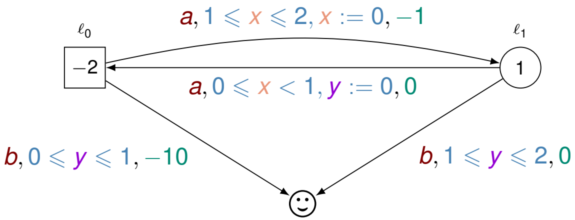
Deterministic strategy

Choose an edge and a delay

Weighted Timed Games

○ Min ☐ Max

☺ target (T)



Play ρ $\quad (\ell_1, \begin{bmatrix} 0 \\ 0 \end{bmatrix}) \xrightarrow{0.5,\ a} (\ell_0, \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}) \xrightarrow{1.25,\ a} (\ell_1, \begin{bmatrix} 0 \\ 1.25 \end{bmatrix}) \xrightarrow{1/3,\ b} (☺, \begin{bmatrix} 1/3 \\ 19/12 \end{bmatrix})$

Deterministic strategy

Choose an edge and a delay

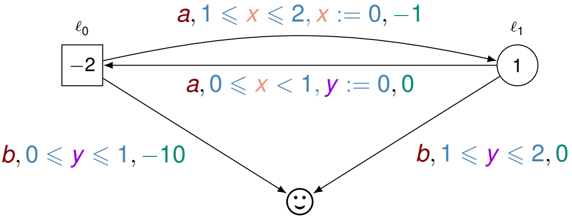
From $(\ell_1, \begin{bmatrix} 0 \\ 0 \end{bmatrix})$

Choose a with $t = \frac{1}{3}$

Weighted Timed Games

○ Min ☐ Max

☺ target (T)



Play ρ $\quad (\ell_1, \begin{bmatrix} 0 \\ 0 \end{bmatrix}) \xrightarrow{0.5,\ a} (\ell_0, \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}) \xrightarrow{1.25,\ a} (\ell_1, \begin{bmatrix} 0 \\ 1.25 \end{bmatrix}) \xrightarrow{1/3,\ b} (☺, \begin{bmatrix} 1/3 \\ 19/12 \end{bmatrix})$

Deterministic strategy

Choose an edge and a delay

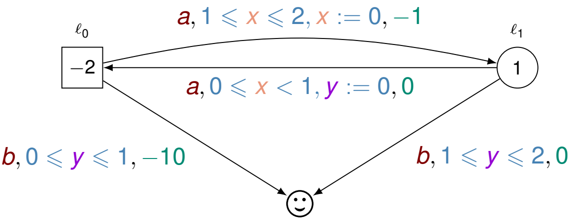
From $(\ell_1, \begin{bmatrix} 0 \\ 0 \end{bmatrix})$

Choose a with $t = \frac{1}{3}$

Weighted Timed Games

○ Min □ Max

☺ target (T)



Play ρ $\quad (\ell_1, \begin{bmatrix} 0 \\ 0 \end{bmatrix}) \xrightarrow{0.5,\ a} (\ell_0, \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}) \xrightarrow{1.25,\ a} (\ell_1, \begin{bmatrix} 0 \\ 1.25 \end{bmatrix}) \xrightarrow{1/3,\ b} (☺, \begin{bmatrix} 1/3 \\ 19/12 \end{bmatrix})$

Deterministic strategy

Choose an edge and a delay

From $(\ell_1, \begin{bmatrix} 0 \\ 0 \end{bmatrix})$

Choose a with $t = \frac{1}{3}$

What features on strategies are needed for Min?

Features on strategies needed for Min

σ Min

τ Max

Features on strategies needed for Min

σ Min

τ Max

Deterministic value

$\mathsf{dVal}(c) = \inf_{\sigma} \sup_{\tau} \mathbf{cost}(\mathsf{Play}(c, \sigma, \tau))$

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

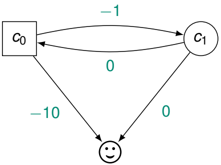
Features on strategies needed for Min

σ Min

τ Max

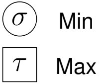
Deterministic value

$\mathsf{dVal}(c) = \inf_{\sigma} \sup_{\tau} \mathbf{cost}(\mathsf{Play}(c, \sigma, \tau))$



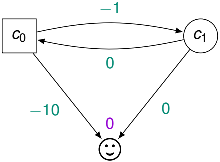
Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Features on strategies needed for Min



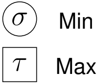
Deterministic value

$$\mathrm{dVal}(c) = \inf_{\sigma} \sup_{\tau} \mathbf{cost}(\mathrm{Play}(c, \sigma, \tau))$$



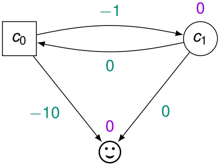
Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Features on strategies needed for Min



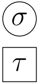
Deterministic value

$$\mathsf{dVal}(c) = \inf_{\sigma} \sup_{\tau} \mathbf{cost}(\mathsf{Play}(c, \sigma, \tau))$$



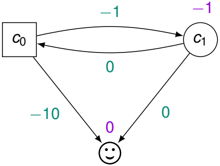
Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Features on strategies needed for Min



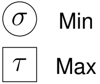
Deterministic value

$\mathsf{dVal}(c) = \inf_\sigma \sup_\tau \mathbf{cost}(\mathsf{Play}(c, \sigma, \tau))$



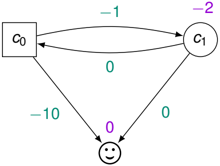
Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Features on strategies needed for Min



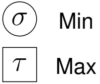
Deterministic value

$\mathrm{dVal}(c) = \inf_{\sigma} \sup_{\tau} \mathbf{cost}(\mathrm{Play}(c, \sigma, \tau))$



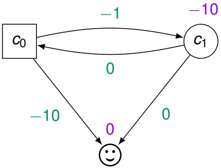
Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Features on strategies needed for Min



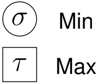
Deterministic value

$$\mathrm{dVal}(c) = \inf_{\sigma} \sup_{\tau} \mathbf{cost}(\mathrm{Play}(c, \sigma, \tau))$$



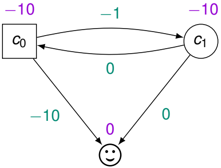
Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Features on strategies needed for Min



Deterministic value

$\mathrm{dVal}(c) = \inf_{\sigma} \sup_{\tau} \mathbf{cost}(\mathrm{Play}(c, \sigma, \tau))$



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

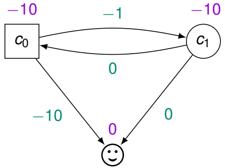
Features on strategies needed for Min

σ Min

τ Max

Deterministic value

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

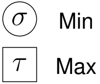


Optimal strategy for Min

$$\mathsf{dVal}^{\sigma}(c) \leqslant \mathsf{dVal}(c)$$

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

Features on strategies needed for Min

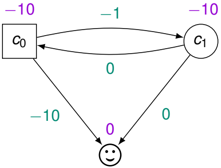


Deterministic value

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

Optimal strategy for Min

$$\mathsf{dVal}^{\sigma}(c) \leqslant \mathsf{dVal}(c)$$



Finite memory

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

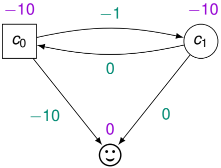
Features on strategies needed for Min

σ Min

τ Max

Deterministic value

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



Optimal strategy for Min

$\mathrm{dVal}^{\sigma}(c) \leqslant \mathrm{dVal}(c)$

Finite memory

Switching 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

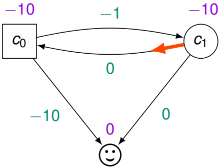
Features on strategies needed for Min

σ Min

τ Max

Deterministic value

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



Optimal strategy for Min

$$\mathsf{dVal}^{\sigma}(c) \leqslant \mathsf{dVal}(c)$$

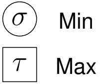
Finite memory

Switching strategy:

- σ_1: reach cycle with a weight $\leqslant -1$

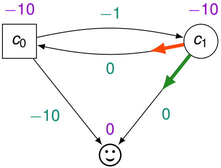
Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Features on strategies needed for Min



Deterministic value

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



Optimal strategy for Min

$\mathrm{dVal}^\sigma(c) \leqslant \mathrm{dVal}(c)$

Finite memory

Switching strategy:

- σ_1: reach cycle with a weight $\leqslant -1$
- σ_2: reach ☺

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

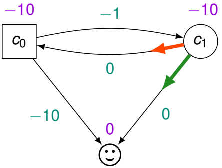
Features on strategies needed for Min

σ Min

τ Max

Deterministic value

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



Optimal strategy for Min

$\mathsf{dVal}^\sigma(c) \leqslant \mathsf{dVal}(c)$

Finite memory

Switching strategy:

- σ_1: reach cycle with a weight $\leqslant -1$
- σ_2: reach ☺
- K: number of turns before switch

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

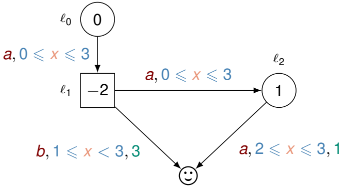
Features on strategies needed for Min

σ Min

τ Max

Deterministic value

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



Optimal strategy for Min

$$\mathsf{dVal}^\sigma(c) \leqslant \mathsf{dVal}(c)$$

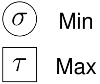
Finite memory

Switching strategy:

- σ_1: reach cycle with a weight $\leqslant -1$
- σ_2: reach ☺
- K: number of turns before switch

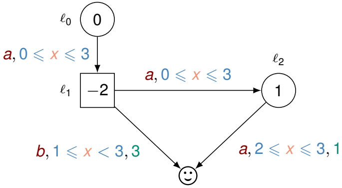
Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Features on strategies needed for Min



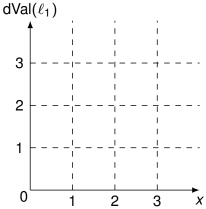
Deterministic value

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



Optimal strategy for Min

$$\mathsf{dVal}^{\sigma}(c) \leqslant \mathsf{dVal}(c)$$



Finite memory

Switching strategy:

- σ_1: reach cycle with a weight $\leqslant -1$
- σ_2: reach ☺
- K: number of turns before switch

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

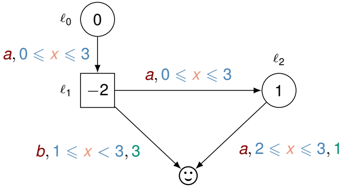
Features on strategies needed for Min

σ Min

τ Max

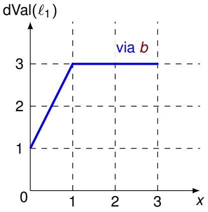
Deterministic value

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



Optimal strategy for Min

$$\mathsf{dVal}^{\sigma}(c) \leqslant \mathsf{dVal}(c)$$



Finite memory

Switching strategy:

- σ_1: reach cycle with a weight $\leqslant -1$
- σ_2: reach ☺
- K: number of turns before switch

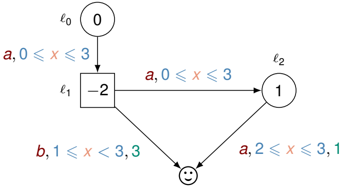
Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

Features on strategies needed for Min

σ Min
τ Max

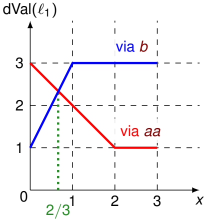
Deterministic value

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



Optimal strategy for Min

$\mathrm{dVal}^\sigma(c) \leqslant \mathrm{dVal}(c)$



Finite memory

Switching strategy:

- σ_1: reach cycle with a weight $\leqslant -1$
- σ_2: reach ☺
- K: number of turns before switch

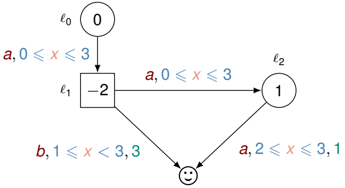
Features on strategies needed for Min

σ Min

τ Max

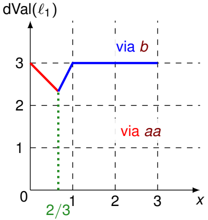
Deterministic value

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



Optimal strategy for Min

$$\mathsf{dVal}^{\sigma}(c) \leqslant \mathsf{dVal}(c)$$



Finite memory

Switching strategy:

- σ_1: reach cycle with a weight $\leqslant -1$
- σ_2: reach ☺
- K: number of turns before switch

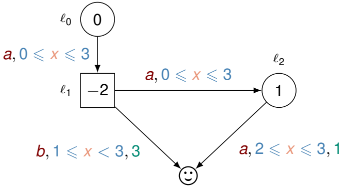
Features on strategies needed for Min

σ Min

τ Max

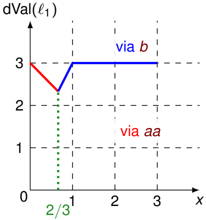
Deterministic value

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



Optimal strategy for Min

$$\mathrm{dVal}^{\sigma}(c) \leqslant \mathrm{dVal}(c)$$



Finite memory

Switching strategy:

- σ_1: reach cycle with a weight $\leqslant -1$
- σ_2: reach ☺
- K: number of turns before switch

Infinite precision

From ℓ_0, Min wants to reach the valuation $2/3$

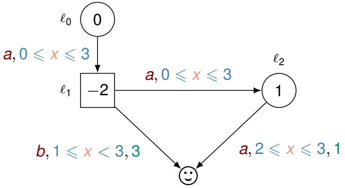
Features on strategies needed for Min

σ Min

τ Max

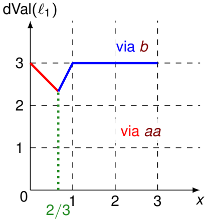
Deterministic value

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



Optimal strategy for Min

$$\mathsf{dVal}^{\sigma}(c) \leqslant \mathsf{dVal}(c)$$



Finite memory

Switching strategy:

- σ_1: reach cycle with a weight $\leqslant -1$
- σ_2: reach ☺
- K: number of turns before switch

Infinite precision

From ℓ_0, Min wants to reach the valuation $2/3$

- if $x \leqslant 2/3$: Min plays $2/3 - x$

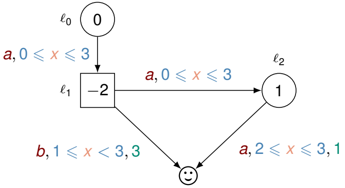
Features on strategies needed for Min

σ Min

τ Max

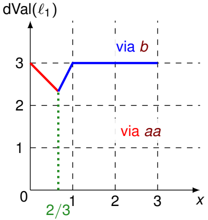
Deterministic value

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



Optimal strategy for Min

$\mathrm{dVal}^{\sigma}(c) \leqslant \mathrm{dVal}(c)$



Finite memory

Switching strategy:

- σ_1: reach cycle with a weight $\leqslant -1$
- σ_2: reach ☺
- K: number of turns before switch

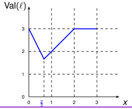
Infinite precision

From ℓ_0, Min wants to reach the valuation $2/3$

- if $x \leqslant 2/3$: Min plays $2/3 - x$
- otherwise, Min plays 0

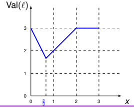
Problems on weighted timed games

Deterministic value problem



Problems on weighted timed games

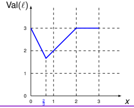
Deterministic value problem



Trading memory with probabilities

Problems on weighted timed games

Deterministic value problem

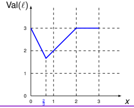


Trading memory with probabilities

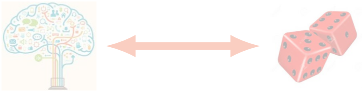
Robust optimal strategies

Problems on weighted timed games

Deterministic value problem



Trading memory with probabilities



Robust optimal strategies

Deterministic value problem

Deciding if $\mathsf{dVal}(c) \leqslant \lambda$?

Deterministic value problem

Deciding if $\mathsf{dVal}(c) \leqslant \lambda$?

	WTG			
$\mathbb{N}$	undecidable			
$\mathbb{Z}$	undecidable			

On Optimal Timed Strategies, T. Brihaye, V. Bruyère and J.-F. Raskin, 2005, FORMATS

Adding Negative Prices to Priced Timed Games, T. Brihaye, G. Geeraerts, S. Krishna, L. Manasa, B. Monmege, and A. Trivedi, 2014, CONCUR

Deterministic value problem

Deciding if dVal(c) $\leqslant \lambda$?

	WTG	0-clock		
$\mathbb{N}$	undecidable			
$\mathbb{Z}$	undecidable			

Deterministic value problem

Deciding if dVal(c) $\leqslant \lambda$?

	WTG	0-clock		
$\mathbb{N}$	undecidable	PTIME		
$\mathbb{Z}$	undecidable	pseudo-polynomial		

On Short Paths Interdiction Problems: Total and Node-Wise Limited Interdiction, L. Khachiyan, E. Boros, K. Borys, K. Elbassioni, V. Gurvich, G. Rudolf, and J. Zhao, 2008, Theory of Computing Systems

Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games., T. Brihaye, G. Geeraerts, A. Haddad, and B. Monmege, 2017, Acta Informatica

Deterministic value problem

Deciding if dVal(c) $\leqslant \lambda$?

	WTG	0-clock	divergent	
$\mathbb{N}$	undecidable	PTIME		
$\mathbb{Z}$	undecidable	pseudo-polynomial		

Deterministic value problem

Deciding if dVal(c) $\leqslant \lambda$?

	WTG	0-clock	divergent	
$\mathbb{N}$	undecidable	PTIME		
$\mathbb{Z}$	undecidable	pseudo-polynomial		

Property of divergence

All SCCs of the WTG contain only cycles with a weight $\leqslant -1$ or $\geqslant 1$

Optimal Reachability for Weighted Timed Game., R. Alur, M. Bernadsky, and P. Madhusudan, 2004, ICALP

Optimal Strategies in Priced Timed Game Automata, P. Bouyer, F. Cassez, E.l Fleury, and K. Larsen, 2004, FSTTCS

Optimal Reachability in Divergent Weighted Timed Games., D. Busatto-Gaston, B. Monmege, and P.-A. Reynier, 2017, FOSSACS

Deterministic value problem

Deciding if dVal(c) $\leqslant \lambda$?

	WTG	0-clock	divergent	
$\mathbb{N}$	undecidable	PTIME	EXPTIME	
$\mathbb{Z}$	undecidable	pseudo-polynomial	EXPTIME	

Property of divergence

All SCCs of the WTG contain only cycles with a weight $\leqslant -1$ or $\geqslant 1$

Optimal Reachability for Weighted Timed Game., R. Alur, M. Bernadsky, and P. Madhusudan, 2004, ICALP

Optimal Strategies in Priced Timed Game Automata, P. Bouyer, F. Cassez, E.l Fleury, and K. Larsen, 2004, FSTTCS

Optimal Reachability in Divergent Weighted Timed Games., D. Busatto-Gaston, B. Monmege, and P.-A. Reynier, 2017, FOSSACS

Deterministic value problem

Deciding if $\mathsf{dVal}(c) \leqslant \lambda$?

	WTG	0-clock	divergent	1-clock
$\mathbb{N}$	undecidable	PTIME	EXPTIME	
$\mathbb{Z}$	undecidable	pseudo-polynomial	EXPTIME	

Property of divergence

All SCCs of the WTG contain only cycles with a weight $\leqslant -1$ or $\geqslant 1$

Deterministic value problem

Deciding if dVal(c) $\leqslant \lambda$?

	WTG	0-clock	divergent	1-clock
$\mathbb{N}$	undecidable	PTIME	EXPTIME	EXPTIME
$\mathbb{Z}$	undecidable	pseudo-polynomial	EXPTIME	

Property of divergence

All SCCs of the WTG contain only cycles with a weight $\leqslant -1$ or $\geqslant 1$

Almost optimal strategies in one clock priced timed games, P. Bouyer, K. Larsen, N. Markey, and J. Rasmussen, 2006, FSTTCS

Two-Player Reachability-Price Games on Single Clock Timed Automata., M. Rutkowski, 2011, QAPL

A Faster Algorithm for Solving One-Clock Priced Timed Games, T. Dueholm Hansen, R. Ibsen-Jensen, and P. Bro Miltersen, 2013, CONCUR

Deterministic value problem

Deciding if $\mathrm{dVal}(c) \leqslant \lambda$?

	WTG	0-clock	divergent	1-clock
$\mathbb{N}$	undecidable	PTIME	EXPTIME	EXPTIME
$\mathbb{Z}$	undecidable	pseudo-polynomial	EXPTIME	

Property of divergence

All SCCs of the WTG contain only cycles with a weight $\leqslant -1$ or $\geqslant 1$

Deterministic value problem

Deciding if dVal(c) $\leqslant \lambda$?

	WTG	0-clock	divergent	1-clock
$\mathbb{N}$	undecidable	PTIME	EXPTIME	EXPTIME
$\mathbb{Z}$	undecidable	pseudo-polynomial	EXPTIME	

Property of divergence

All SCCs of the WTG contain only cycles with a weight $\leqslant -1$ or $\geqslant 1$

PSPACE lower bound

The deterministic value problem is PSPACE-hard for 1-clock WTG

One-Clock Priced Timed Games are PSPACE-hard., J. Fearnley, R. Ibsen-Jensen, and R. Savani, 2020, LICS

Deterministic value problem

Deciding if dVal(c) $\leqslant \lambda$?

	WTG	0-clock	divergent	1-clock
$\mathbb{N}$	undecidable	PTIME	EXPTIME	EXPTIME
$\mathbb{Z}$	undecidable	pseudo-polynomial	EXPTIME	

Property of divergence

All SCCs of the WTG contain only cycles with a weight $\leqslant -1$ or $\geqslant 1$

PSPACE lower bound

The deterministic value problem is PSPACE-hard for 1-clock WTG

Theorem (CONCUR'22): the problem is decidable for 1-clock WTG

Deterministic value problem

Deciding if dVal(c) $\leqslant \lambda$?

	WTG	0-clock	divergent	1-clock
$\mathbb{N}$	undecidable	PTIME	EXPTIME	EXPTIME
$\mathbb{Z}$	undecidable	pseudo-polynomial	EXPTIME	

Property of divergence

All SCCs of the WTG contain only cycles with a weight $\leqslant -1$ or $\geqslant 1$

PSPACE lower bound

The deterministic value problem is PSPACE-hard for 1-clock WTG

Theorem (CONCUR'22): the problem is decidable for 1-clock WTG

$c \mapsto$ Val(c) is computable in exponential time

Deterministic value problem

Deciding if $\mathrm{dVal}(c) \leqslant \lambda$?

	WTG	0-clock	divergent	1-clock
$\mathbb{N}$	undecidable	PTIME	EXPTIME	EXPTIME
$\mathbb{Z}$	undecidable	pseudo-polynomial	EXPTIME	

Property of divergence

All SCCs of the WTG contain only cycles with a weight $\leqslant -1$ or $\geqslant 1$

PSPACE lower bound

The deterministic value problem is PSPACE-hard for 1-clock WTG

Theorem (CONCUR'22): the problem is decidable for 1-clock WTG

$c \mapsto \mathrm{Val}(c)$ is computable in exponential time

- Back-time algorithm: compute $c \mapsto \mathrm{Val}(c)$ from $x = 1$ to 0

Deterministic value problem

Deciding if $\mathrm{dVal}(c) \leqslant \lambda$?

	WTG	0-clock	divergent	1-clock
$\mathbb{N}$	undecidable	PTIME	EXPTIME	EXPTIME
$\mathbb{Z}$	undecidable	pseudo-polynomial	EXPTIME	

Property of divergence

All SCCs of the WTG contain only cycles with a weight $\leqslant -1$ or $\geqslant 1$

PSPACE lower bound

The deterministic value problem is PSPACE-hard for 1-clock WTG

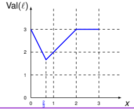
Theorem (CONCUR'22): the problem is decidable for 1-clock WTG

$c \mapsto \mathrm{Val}(c)$ is computable in exponential time

- Back-time algorithm: compute $c \mapsto \mathrm{Val}(c)$ from $x = 1$ to 0
- Value iteration algorithm: deterministic value is a fixed point of a given operator

Problems on weighted timed games

Deterministic value problem



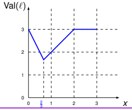
Trading memory with probabilities

Robust optimal strategies

Problems on weighted timed games

Decidability for 1-clock WTG

Deterministic value problem

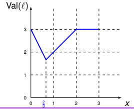


Trading memory with probabilities

Robust optimal strategies

Problems on weighted timed games

Deterministic value problem



Decidability for 1-clock WTG

Software prototype for 1-clock WTG

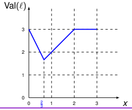
Trading memory with probabilities

Robust optimal strategies

Problems on weighted timed games

Fixpoint characterisation

Deterministic value problem



Decidability for 1-clock WTG

Software prototype for 1-clock WTG

Trading memory with probabilities

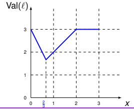
Robust optimal strategies

Problems on weighted timed games

Fixpoint characterisation

Switching strategies in divergent WTG

Deterministic value problem



Decidability for 1-clock WTG

Software prototype for 1-clock WTG

Trading memory with probabilities

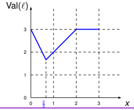
Robust optimal strategies

Problems on weighted timed games

Fixpoint characterisation

Switching strategies in divergent WTG

Deterministic value problem



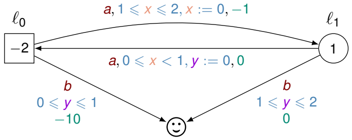
Decidability for 1-clock WTG

Software prototype for 1-clock WTG

Trading memory with probabilities

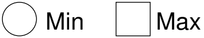
Robust optimal strategies

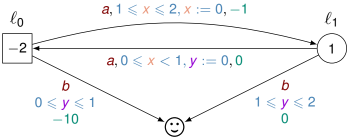
Stochastic strategies



Stochastic Timed Automata, N. Bertrand, P. Bouyer, T. Brihaye, Q. Menet, C. Baier, M. Grosser, and M. Jurdzinzki, 2014, Logical Methods in Computer Science

Stochastic strategies





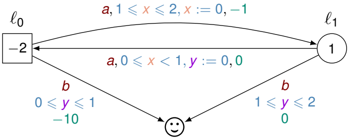
Stochastic strategy

Distribution over possible choices

Stochastic Timed Automata, N. Bertrand, P. Bouyer, T. Brihaye, Q. Menet, C. Baier, M. Grosser, and M. Jurdzinzki, 2014, Logical Methods in Computer Science

Stochastic strategies

Min Max



Stochastic strategy

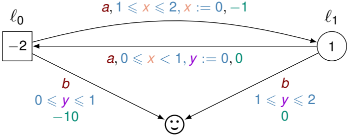
Distribution over possible choices

1. Edge *a*: finite distribution

Stochastic Timed Automata, N. Bertrand, P. Bouyer, T. Brihaye, Q. Menet, C. Baier, M. Grosser, and M. Jurdzinzki, 2014, Logical Methods in Computer Science

Stochastic strategies

○ Min □ Max



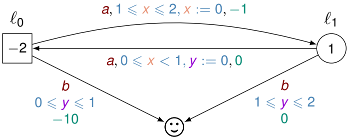
Stochastic strategy

Distribution over possible choices

1. Edge *a*: finite distribution
2. Delay for *a*: infinite distribution

Stochastic Timed Automata, N. Bertrand, P. Bouyer, T. Brihaye, Q. Menet, C. Baier, M. Grosser, and M. Jurdzinzki, 2014, Logical Methods in Computer Science

Stochastic strategies



From $(\ell_1, \begin{bmatrix} 0 \\ 0 \end{bmatrix})$

Choose between *a* or *b* with $\mathcal{B}(\frac{1}{2})$

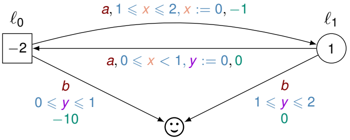
Stochastic strategy

Distribution over possible choices

1. Edge *a*: finite distribution
2. Delay for *a*: infinite distribution

Stochastic Timed Automata, N. Bertrand, P. Bouyer, T. Brihaye, Q. Menet, C. Baier, M. Grosser, and M. Jurdzinzki, 2014, Logical Methods in Computer Science

Stochastic strategies



From $(\ell_1, \begin{bmatrix} 0 \\ 0 \end{bmatrix})$

Choose between a or b with $\mathcal{B}(\frac{1}{2})$

- a: choose t with $\mathcal{U}([0, 1))$

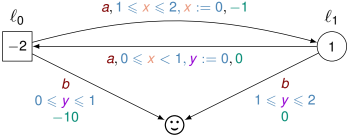
Stochastic strategy

Distribution over possible choices

1. Edge a: finite distribution
2. Delay for a: infinite distribution

Stochastic Timed Automata, N. Bertrand, P. Bouyer, T. Brihaye, Q. Menet, C. Baier, M. Grosser, and M. Jurdzinzki, 2014, Logical Methods in Computer Science

Stochastic strategies



From $(\ell_1, \begin{bmatrix} 0 \\ 0 \end{bmatrix})$

Choose between *a* or *b* with $\mathcal{B}(\frac{1}{2})$

- *a*: choose t with $\mathcal{U}([0, 1))$
- *b*: choose t with $\delta_{1.5}$

Stochastic strategy

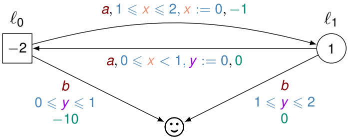
Distribution over possible choices

1. Edge *a*: finite distribution
2. Delay for *a*: infinite distribution

Stochastic Timed Automata, N. Bertrand, P. Bouyer, T. Brihaye, Q. Menet, C. Baier, M. Grosser, and M. Jurdzinzki, 2014, Logical Methods in Computer Science

Stochastic strategies

η Min θ Max



From $(\ell_1, \begin{bmatrix} 0 \\ 0 \end{bmatrix})$

Choose between *a* or *b* with $\mathcal{B}(\frac{1}{2})$

- *a*: choose *t* with $\mathcal{U}([0, 1))$
- *b*: choose *t* with $\delta_{1.5}$

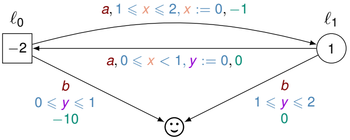
Stochastic strategy

Distribution over possible choices

1. Edge *a*: finite distribution
2. Delay for *a*: infinite distribution

When we fix two strategies

Stochastic strategies



From $(\ell_1, \begin{bmatrix} 0 \\ 0 \end{bmatrix})$

Choose between a or b with $\mathcal{B}(\frac{1}{2})$

- a: choose t with $\mathcal{U}([0,1))$
- b: choose t with $\delta_{1.5}$

Stochastic strategy

Distribution over possible choices

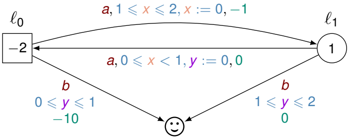
1. Edge a: finite distribution
2. Delay for a: infinite distribution

When we fix two strategies

- Infinite Markov Chain

Stochastic strategies

η Min θ Max



From $(\ell_1, \begin{bmatrix} 0 \\ 0 \end{bmatrix})$

Choose between *a* or *b* with $\mathcal{B}(\frac{1}{2})$

- *a*: choose t with $\mathcal{U}([0, 1))$
- *b*: choose t with $\delta_{1.5}$

Stochastic strategy

Distribution over possible choices

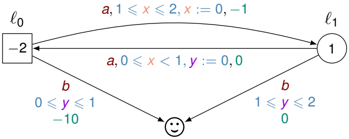
1. Edge *a*: finite distribution
2. Delay for *a*: infinite distribution

When we fix two strategies

- Infinite Markov Chain
- Replace **cost**(Play(c, η, θ)) by $\mathbb{E}_c^{\eta,\theta}$(**cost**)

Stochastic strategies

η Min θ Max



From $(\ell_1, \begin{bmatrix} 0 \\ 0 \end{bmatrix})$

Choose between a or b with $\mathcal{B}(\frac{1}{2})$

- a: choose t with $\mathcal{U}([0,1))$
- b: choose t with $\delta_{1.5}$

Stochastic strategy

Distribution over possible choices

1. Edge a: finite distribution
2. Delay for a: infinite distribution

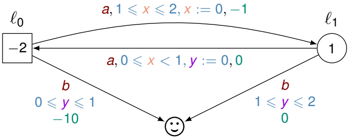
When we fix two strategies

- Infinite Markov Chain
- Replace $\textbf{cost}(\text{Play}(c, \eta, \theta))$ by $\mathbb{E}_c^{\eta,\theta}(\textbf{cost})$

⚠ Measurability conditions on η and θ

Stochastic strategies

η Min θ Max



From $(\ell_1, \begin{bmatrix} 0 \\ 0 \end{bmatrix})$

Choose between a or b with $\mathcal{B}(\frac{1}{2})$

- a: choose t with $\mathcal{U}([0, 1))$
- b: choose t with $\delta_{1.5}$

Stochastic strategy

Distribution over possible choices

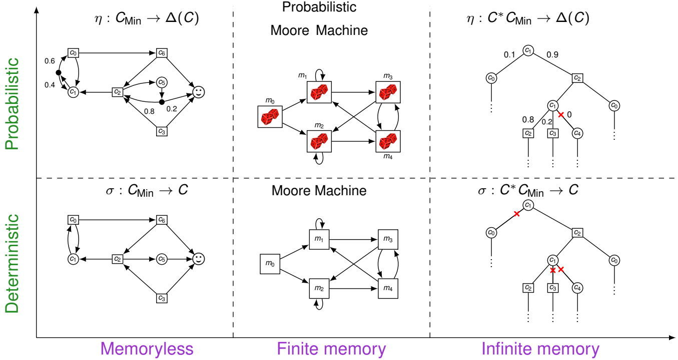
1. Edge a: finite distribution
2. Delay for a: infinite distribution

When we fix two strategies

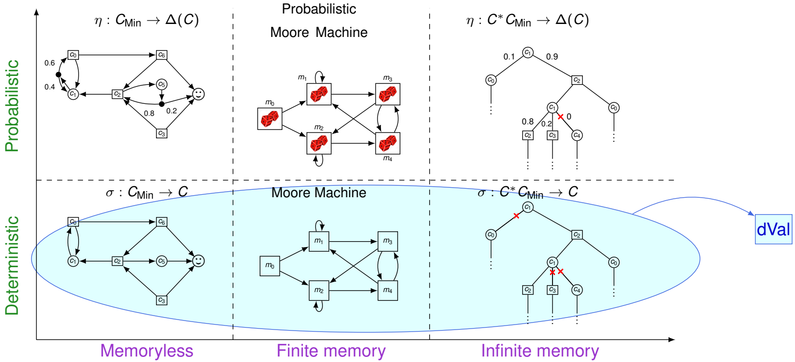
- Infinite Markov Chain
- Replace $\mathbf{cost}(\mathrm{Play}(c, \eta, \theta))$ by $\mathbb{E}_c^{\eta,\theta}(\mathbf{cost})$

⚠ Measurability conditions on η and θ

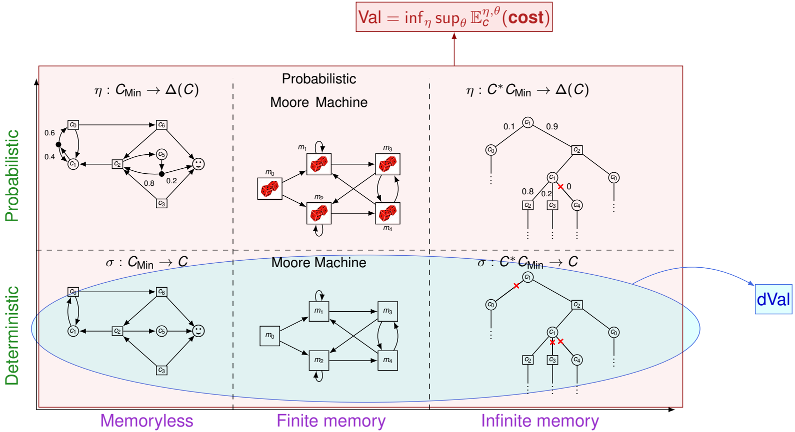
Stochastic values



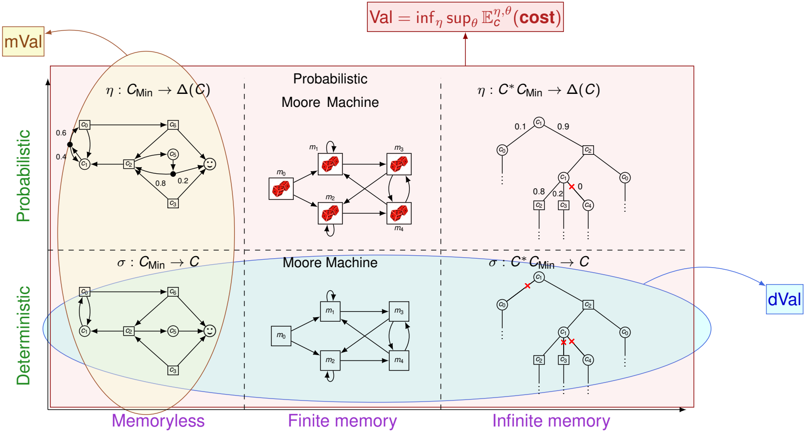
Stochastic values



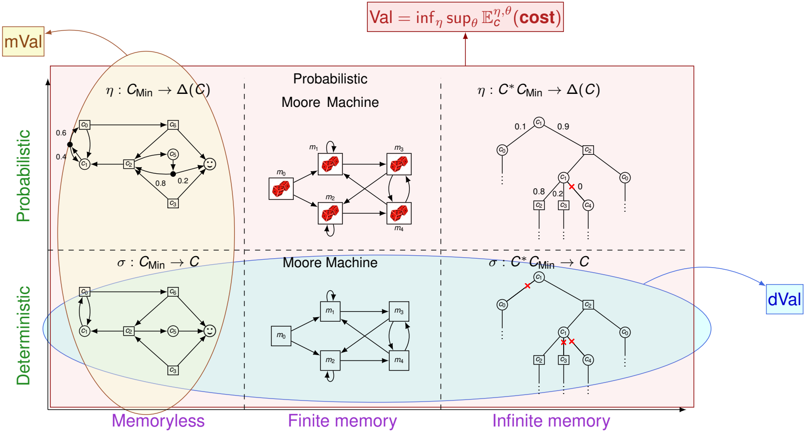
Stochastic values



Stochastic values

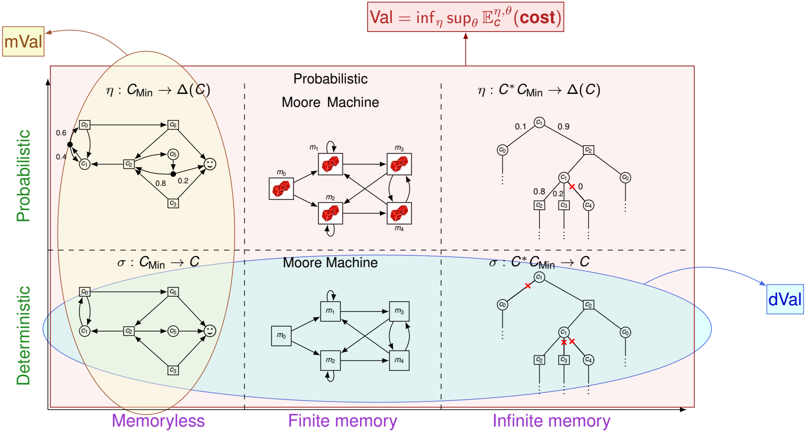


Stochastic values



Theorem (CONCUR'20, ICALP'21): Trading memory with probabilities

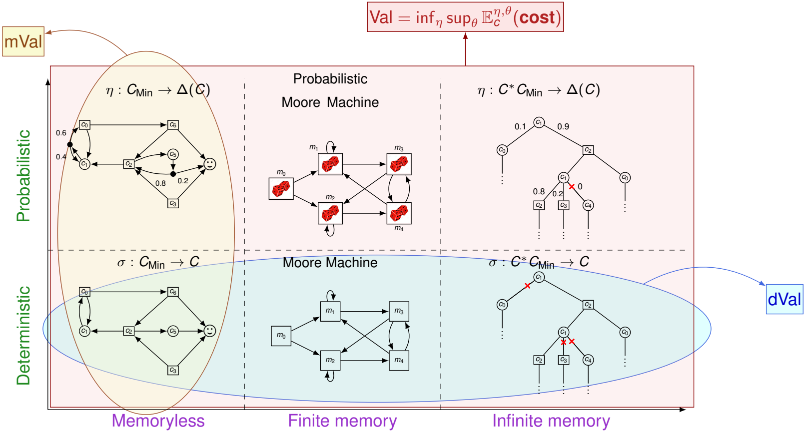
Stochastic values



Theorem (CONCUR'20, ICALP'21): Trading memory with probabilities

$$\text{dVal} \quad = \quad \text{Val} \quad = \quad \text{mVal}$$

Stochastic values

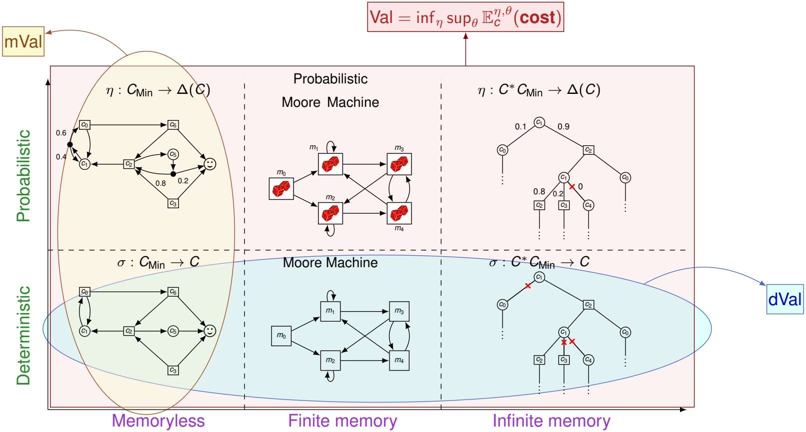


Theorem (CONCUR'20, ICALP'21): Trading memory with probabilities

$$\text{dVal} \quad = \quad \text{Val} \quad = \quad \text{mVal}$$

- 0-clock weighted timed games

Stochastic values

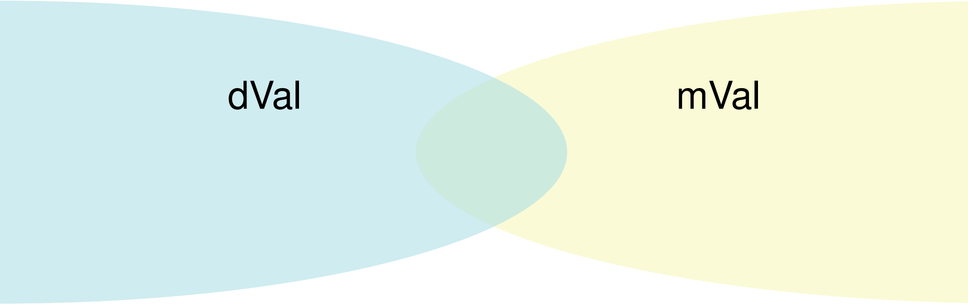


Theorem (CONCUR'20, ICALP'21): Trading memory with probabilities

$$\mathsf{dVal} \quad = \quad \mathsf{Val} \quad = \quad \mathsf{mVal}$$

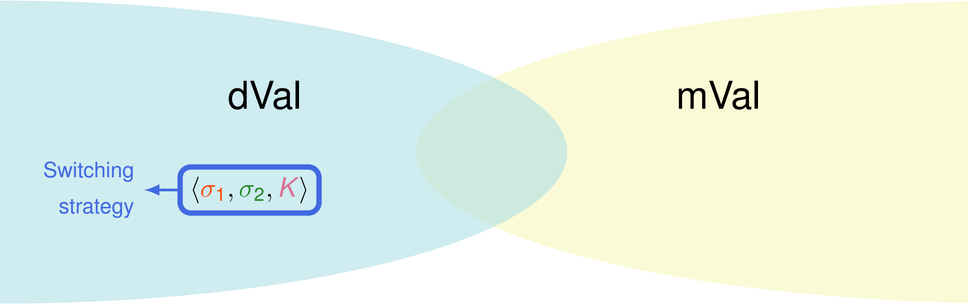
- 0-clock weighted timed games
- divergent weighted timed games

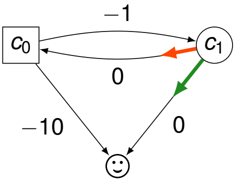
Trading memory with probabilities



Trading memory with probabilities

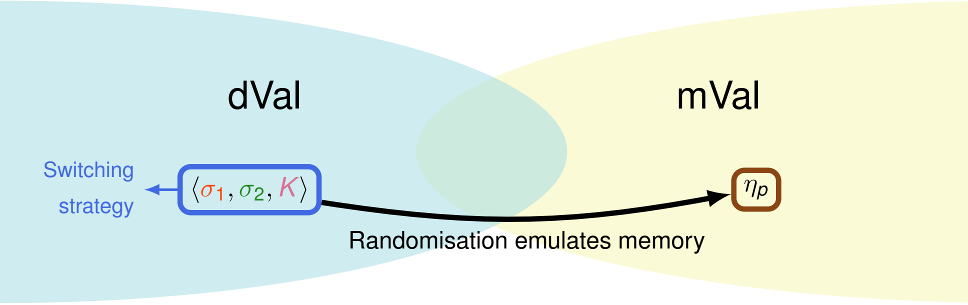
○ Min □ Max

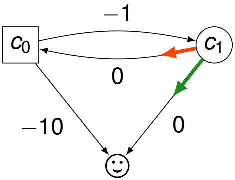




Trading memory with probabilities

○ Min □ Max





Trading memory with probabilities

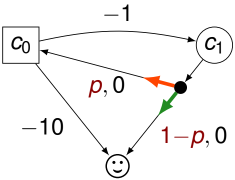
○ Min □ Max

dVal

mVal

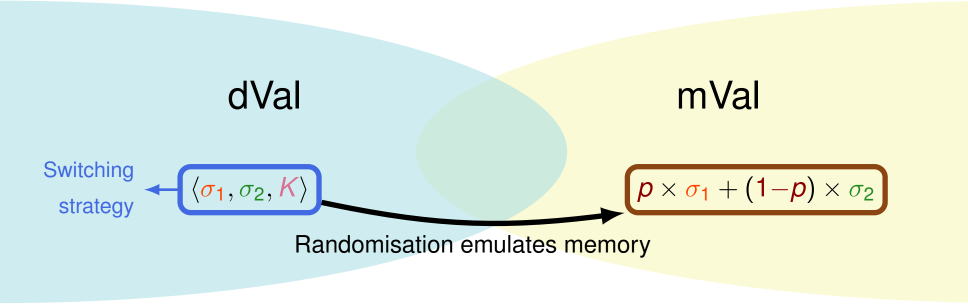
Switching strategy ← $\langle \sigma_1, \sigma_2, K \rangle$ → $p \times \sigma_1 + (1-p) \times \sigma_2$

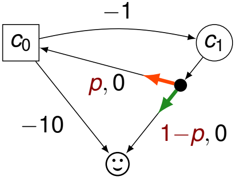
Randomisation emulates memory



Trading memory with probabilities

○ Min □ Max

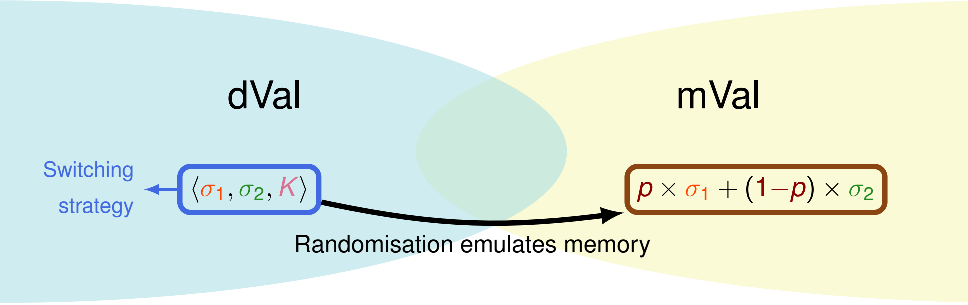


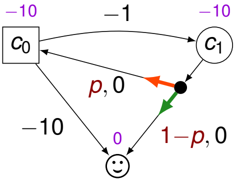


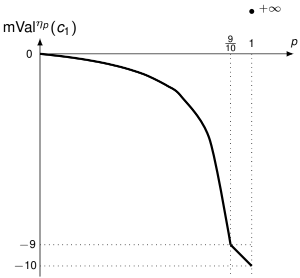
- Max has a best response deterministic memoryless strategy: τ

Trading memory with probabilities

◯ Min ☐ Max



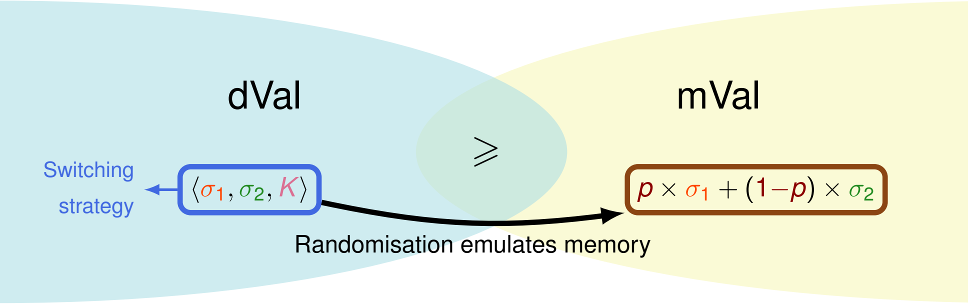


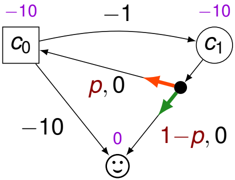


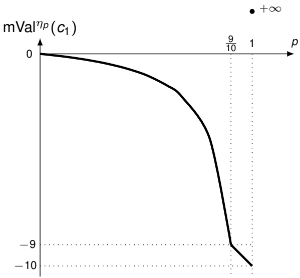
- Max has a best response deterministic memoryless strategy: τ

Trading memory with probabilities

○ Min □ Max







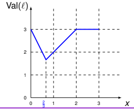
- Max has a best response deterministic memoryless strategy: τ

Problems on weighted timed games

Fixpoint characterisation

Switching strategies in divergent WTG

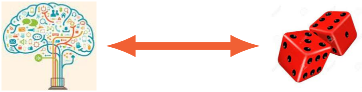
Deterministic value problem



Decidability for 1-clock WTG

Software prototype for 1-clock WTG

Trading memory with probabilities



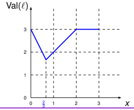
Robust optimal strategies

Problems on weighted timed games

Fixpoint characterisation

Switching strategies in divergent WTG

Deterministic value problem



Decidability for 1-clock WTG

Software prototype for 1-clock WTG

Definition of stochastic values

Trading memory with probabilities

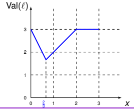
Robust optimal strategies

Problems on weighted timed games

Fixpoint characterisation

Switching strategies in divergent WTG

Deterministic value problem



Decidability for 1-clock WTG

Software prototype for 1-clock WTG

Definition of stochastic values

Memory is useless in divergent WTG and 0-clock WTG

Trading memory with probabilities

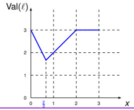
Robust optimal strategies

Problems on weighted timed games

Fixpoint characterisation

Switching strategies in divergent WTG

Deterministic value problem



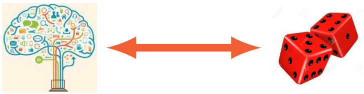
Decidability for 1-clock WTG

Software prototype for 1-clock WTG

Definition of stochastic values

Memory is useless in divergent WTG and 0-clock WTG

Trading memory with probabilities



Probabilities are useless in 1-clock WTG, divergent WTG, and 0-clock WTG

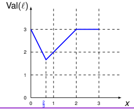
Robust optimal strategies

Problems on weighted timed games

Fixpoint characterisation

Switching strategies in divergent WTG

Deterministic value problem



Decidability for 1-clock WTG

Software prototype for 1-clock WTG

Definition of stochastic values

Memory is useless in divergent WTG and 0-clock WTG

Trading memory with probabilities

Probabilities are useless in 1-clock WTG, divergent WTG, and 0-clock WTG

Robust optimal strategies

Robustness in weighted timed games

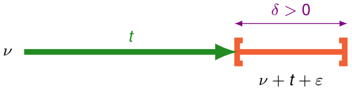
$$\nu \xrightarrow{t} \nu + t$$

Robustness in weighted timed games

Give to Max the power to perturb the delay chosen by Min

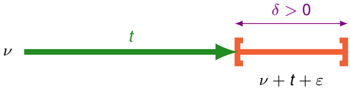
Robustness in weighted timed games

Give to Max the power to perturb the delay chosen by Min



Robustness in weighted timed games

Give to Max the power to perturb the delay chosen by Min

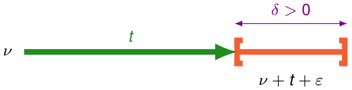


Fixed-δ semantics

Check the guard after the perturbation

Robustness in weighted timed games

Give to Max the power to perturb the delay chosen by Min

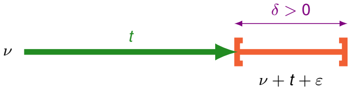


Fixed-δ semantics

Check the guard after the perturbation
$\forall \varepsilon \in [0, \delta], \nu + t + \varepsilon$ satisfies the guard

Robustness in weighted timed games

Give to Max the power to perturb the delay chosen by Min



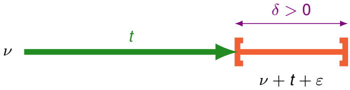
Fixed-δ semantics

Check the guard after the perturbation
$\forall \varepsilon \in [0, \delta], \nu + t + \varepsilon$ satisfies the guard

Two problems induced by our knowledge on δ

Robustness in weighted timed games

Give to Max the power to perturb the delay chosen by Min



Fixed-δ semantics

Check the guard after the perturbation
$\forall \varepsilon \in [0, \delta], \nu + t + \varepsilon$ satisfies the guard

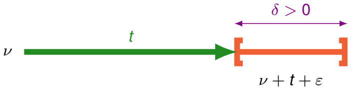
Two problems induced by our knowledge on δ

- δ is fixed and known

Robustness in weighted timed games

χ Min
ζ Max

Give to Max the power to perturb the delay chosen by Min



Fixed-δ semantics

Check the guard after the perturbation
$\forall \varepsilon \in [0, \delta], \nu + t + \varepsilon$ satisfies the guard

Two problems induced by our knowledge on δ

- δ is fixed and known

Fixed-δ robust value

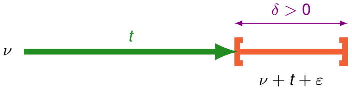
$$\mathsf{rVal}^{\delta}(c) = \inf_{\substack{\chi \\ \delta\text{-robust}}} \sup_{\substack{\zeta \\ \delta\text{-robust}}} \mathbf{cost}(\mathsf{Play}(c, \chi, \zeta))$$

Robustness in weighted timed games

χ Min

ζ Max

Give to Max the power to perturb the delay chosen by Min



Fixed-δ semantics

Check the guard after the perturbation
$\forall \varepsilon \in [0, \delta], \nu + t + \varepsilon$ satisfies the guard

Two problems induced by our knowledge on δ

- δ is fixed and known

Fixed-δ robust value

$$\mathsf{rVal}^{\delta}(c) = \inf_{\substack{\chi \\ \delta\text{-robust}}} \sup_{\substack{\zeta \\ \delta\text{-robust}}} \mathbf{cost}(\mathsf{Play}(c, \chi, \zeta))$$

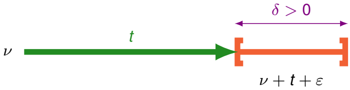
✓ Encoding fixed-δ semantics into exact one

Robustness in weighted timed games

χ Min

ζ Max

Give to Max the power to perturb the delay chosen by Min



Fixed-δ semantics

Check the guard after the perturbation
$\forall \varepsilon \in [0, \delta], \nu + t + \varepsilon$ satisfies the guard

Two problems induced by our knowledge on δ

- δ is fixed and known

Fixed-δ robust value

$$\mathrm{rVal}^{\delta}(c) = \inf_{\substack{\chi \\ \delta\text{-robust}}} \sup_{\substack{\zeta \\ \delta\text{-robust}}} \mathbf{cost}(\mathrm{Play}(c, \chi, \zeta))$$

Encoding fixed-δ semantics into exact one

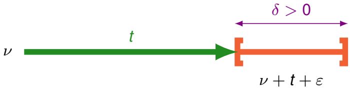
Need a new clock

Robustness in weighted timed games

χ Min

ζ Max

Give to Max the power to perturb the delay chosen by Min



Fixed-δ semantics

Check the guard after the perturbation
$\forall \varepsilon \in [0, \delta], \nu + t + \varepsilon$ satisfies the guard

Two problems induced by our knowledge on δ

- δ is fixed and known
- δ tends to 0

Fixed-δ robust value

$$\mathsf{rVal}^{\delta}(c) = \inf_{\substack{\chi \\ \delta\text{-robust}}} \sup_{\substack{\zeta \\ \delta\text{-robust}}} \mathbf{cost}(\mathsf{Play}(c, \chi, \zeta))$$

Encoding fixed-δ semantics into exact one

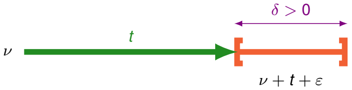
Need a new clock

Robustness in weighted timed games

χ Min

ζ Max

Give to Max the power to perturb the delay chosen by Min



Fixed-δ semantics

Check the guard after the perturbation
$\forall \varepsilon \in [0, \delta], \nu + t + \varepsilon$ satisfies the guard

Two problems induced by our knowledge on δ

- δ is fixed and known
- δ tends to 0

Fixed-δ robust value

$$\mathrm{rVal}^\delta(c) = \inf_{\substack{\chi \\ \delta\text{-robust}}} \sup_{\substack{\zeta \\ \delta\text{-robust}}} \mathbf{cost}(\mathrm{Play}(c, \chi, \zeta))$$

Limit robust value

$$\mathrm{rVal}(c) = \lim_{\substack{\delta \to 0 \\ \delta > 0}} \mathrm{rVal}^\delta(c)$$

Encoding fixed-δ semantics into exact one

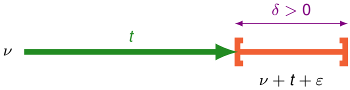
Need a new clock

Robustness in weighted timed games

χ Min

ζ Max

Give to Max the power to perturb the delay chosen by Min



Fixed-δ semantics

Check the guard after the perturbation
$\forall \varepsilon \in [0, \delta], \nu + t + \varepsilon$ satisfies the guard

Two problems induced by our knowledge on δ

- δ is fixed and known
- δ tends to 0

Fixed-δ robust value

$$\mathrm{rVal}^{\delta}(c) = \inf_{\substack{\chi \\ \delta\text{-robust}}} \sup_{\substack{\zeta \\ \delta\text{-robust}}} \mathbf{cost}(\mathrm{Play}(c, \chi, \zeta))$$

Encoding fixed-δ semantics into exact one

Need a new clock

Limit robust value

$$\mathrm{rVal}(c) = \lim_{\substack{\delta \to 0 \\ \delta > 0}} \mathrm{rVal}^{\delta}(c)$$

rVal^{δ} is monotonic in δ

Robust value problems

Deciding if $\mathsf{rVal}^{\delta}(c)$ (resp. $\mathsf{rVal}(c)$) is at most equal to λ?

Robust value problems

Deciding if $\mathrm{rVal}^{\delta}(c)$ (resp. $\mathrm{rVal}(c)$) is at most equal to λ?

	WTG			
rVal^{δ}	undecidable			
rVal	undecidable			

Robust Weighted Timed Automata and Games, P. Bouyer, N. Markey, and O. Sankur, 2013, FORMATS

Robust value problems

Deciding if $\mathrm{rVal}^{\delta}(c)$ (resp. $\mathrm{rVal}(c)$) is at most equal to λ?

	WTG	acyclic	divergent	1-clock
rVal^{δ}	undecidable			
rVal	undecidable			

Robust value problems

Deciding if $\mathsf{rVal}^{\delta}(c)$ (resp. $\mathsf{rVal}(c)$) is at most equal to λ?

	WTG	acyclic	divergent	1-clock
rVal^{δ}	undecidable			decidable (in $\mathbb{N}$)
rVal	undecidable			

Revisiting Robustness in Priced Timed Game, S. Guha, S. Krishna, L. Manasa, and A. Trivedi, 2015, FSTTCS

Robust value problems

Deciding if $\mathrm{rVal}^{\delta}(c)$ (resp. $\mathrm{rVal}(c)$) is at most equal to λ?

	WTG	acyclic	divergent	1-clock
rVal^{δ}	undecidable	decidable	decidable	decidable (in $\mathbb{N}$)
rVal	undecidable			

Robust value problems

Deciding if $\mathsf{rVal}^{\delta}(c)$ (resp. $\mathsf{rVal}(c)$) is at most equal to λ?

	WTG	acyclic	divergent	1-clock
rVal^{δ}	undecidable	decidable	decidable	decidable (in $\mathbb{N}$)
rVal	undecidable	decidable		

Theorem (SUBMITTED): Decidability of the robust value problem in acyclic WTG

Robust value problems

Deciding if $\mathrm{rVal}^\delta(c)$ (resp. $\mathrm{rVal}(c)$) is at most equal to λ?

	WTG	acyclic	divergent	1-clock
rVal^δ	undecidable	decidable	decidable	decidable (in $\mathbb{N}$)
rVal	undecidable	decidable		

Theorem (SUBMITTED): Decidability of the robust value problem in acyclic WTG

A combination of two existing methods

Robust value problems

Deciding if $\mathrm{rVal}^\delta(c)$ (resp. $\mathrm{rVal}(c)$) is at most equal to λ?

	WTG	acyclic	divergent	1-clock
rVal^δ	undecidable	decidable	decidable	decidable (in $\mathbb{N}$)
rVal	undecidable	decidable		

Theorem (SUBMITTED): Decidability of the robust value problem in acyclic WTG

A combination of two existing methods

Cells

Optimal reachability for weighted timed games, R. Alur, M. Bernadsky and P. Madhusudan, 2004, ICALP

Robust value problems

Deciding if $\mathrm{rVal}^{\delta}(c)$ (resp. $\mathrm{rVal}(c)$) is at most equal to λ?

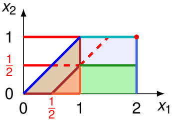
	WTG	acyclic	divergent	1-clock
rVal^{δ}	undecidable	decidable	decidable	decidable (in $\mathbb{N}$)
rVal	undecidable	decidable		

Theorem (SUBMITTED): Decidability of the robust value problem in acyclic WTG

A combination of two existing methods

Cells

$y = \sum_i a_i\, x_i + b$



Optimal reachability for weighted timed games, R. Alur, M. Bernadsky and P. Madhusudan, 2004, ICALP

Robust value problems

Deciding if $\mathrm{rVal}^{\delta}(c)$ (resp. $\mathrm{rVal}(c)$) is at most equal to λ?

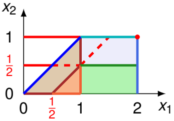
	WTG	acyclic	divergent	1-clock
rVal^{δ}	undecidable	decidable	decidable	decidable (in $\mathbb{N}$)
rVal	undecidable	decidable		

Theorem (SUBMITTED): Decidability of the robust value problem in acyclic WTG

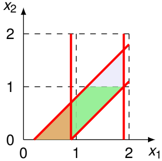
A combination of two existing methods

Cells

$y = \sum_i a_i\, x_i + b$



Shrunk DBM



Shrinking timed automata, O. Sankur, P. Bouyer, and N. Markey, 2011, FSTTCS

Robust value problems

Deciding if $\mathrm{rVal}^\delta(c)$ (resp. $\mathrm{rVal}(c)$) is at most equal to λ?

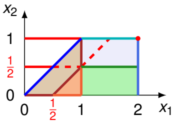
	WTG	acyclic	divergent	1-clock
rVal^δ	undecidable	decidable	decidable	decidable (in $\mathbb{N}$)
rVal	undecidable	decidable		

Theorem (SUBMITTED): Decidability of the robust value problem in acyclic WTG

A combination of two existing methods

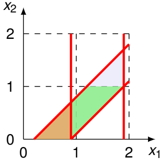
Cells

$y = \sum_i a_i\, x_i + b$



Shrunk cells

Shrunk DBM



Robust value problems

Deciding if $\mathrm{rVal}^\delta(c)$ (resp. $\mathrm{rVal}(c)$) is at most equal to λ?

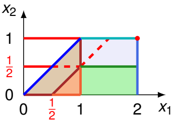
	WTG	acyclic	divergent	1-clock
rVal^δ	undecidable	decidable	decidable	decidable (in $\mathbb{N}$)
rVal	undecidable	decidable		

Theorem (SUBMITTED): Decidability of the robust value problem in acyclic WTG

A combination of two existing methods

Cells

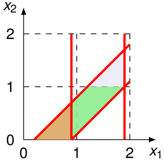
$y = \sum_i a_i\, x_i + b$



Shrunk cells

$y = \sum_i a_i\, x_i + b$

Shrunk DBM



Robust value problems

Deciding if $\mathrm{rVal}^{\delta}(c)$ (resp. $\mathrm{rVal}(c)$) is at most equal to λ?

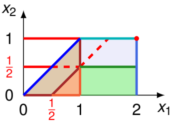
	WTG	acyclic	divergent	1-clock
rVal^{δ}	undecidable	decidable	decidable	decidable (in $\mathbb{N}$)
rVal	undecidable	decidable		

Theorem (SUBMITTED): Decidability of the robust value problem in acyclic WTG

A combination of two existing methods

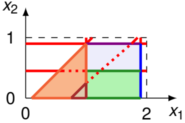
Cells

$y = \sum_i a_i\, x_i + b$

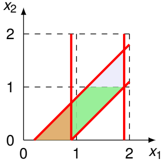


Shrunk cells

$y = \sum_i a_i\, x_i + b + c\,\delta$



Shrunk DBM



Robust value problems

Deciding if $\mathrm{rVal}^{\delta}(c)$ (resp. $\mathrm{rVal}(c)$) is at most equal to λ?

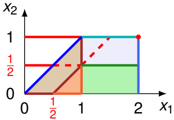
	WTG	acyclic	divergent	1-clock
rVal^{δ}	undecidable	decidable	decidable	decidable (in $\mathbb{N}$)
rVal	undecidable	decidable	decidable	

Theorem (SUBMITTED): Decidability of the robust value problem in acyclic WTG

A combination of two existing methods

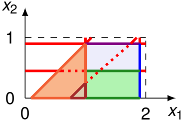
Cells

$y = \sum_i a_i \, x_i + b$

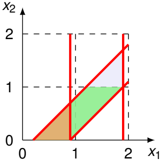


Shrunk cells

$y = \sum_i a_i \, x_i + b + c \, \delta$



Shrunk DBM

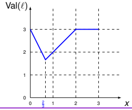


Problems on weighted timed games

Fixpoint characterisation

Switching strategies in divergent WTG

Deterministic value problem



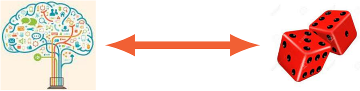
Decidability for 1-clock WTG

Software prototype for 1-clock WTG

Definition of stochastic values

Memory is useless in divergent WTG and 0-clock WTG

Trading memory with probabilities



Probabilities are useless in 1-clock WTG, divergent WTG, and 0-clock WTG

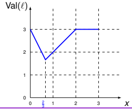
Robust optimal strategies

Problems on weighted timed games

Fixpoint characterisation

Switching strategies in divergent WTG

Deterministic value problem



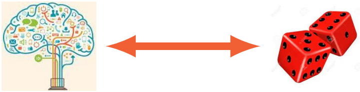
Decidability for 1-clock WTG

Software prototype for 1-clock WTG

Definition of stochastic values

Memory is useless in divergent WTG and 0-clock WTG

Trading memory with probabilities



Probabilities are useless in 1-clock WTG, divergent WTG, and 0-clock WTG

Definition of robust values

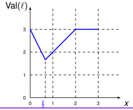
Robust optimal strategies

Problems on weighted timed games

Fixpoint characterisation

Switching strategies in divergent WTG

Deterministic value problem



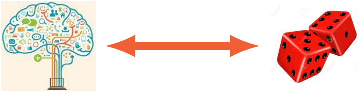
Decidability for 1-clock WTG

Software prototype for 1-clock WTG

Definition of stochastic values

Memory is useless in divergent WTG and 0-clock WTG

Trading memory with probabilities



Probabilities are useless in 1-clock WTG, divergent WTG, and 0-clock WTG

Definition of robust values

Robust optimal strategies

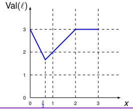
Decidability of $\mathrm{rVal}(c) < +\infty$ in all WTGs

Problems on weighted timed games

Fixpoint characterisation

Switching strategies in divergent WTG

Deterministic value problem



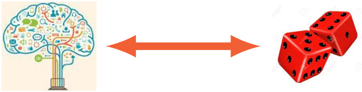
Decidability for 1-clock WTG

Software prototype for 1-clock WTG

Definition of stochastic values

Memory is useless in divergent WTG and 0-clock WTG

Trading memory with probabilities



Probabilities are useless in 1-clock WTG, divergent WTG, and 0-clock WTG

Definition of robust values

Computing robust values in divergent (and acyclic) WTG

Robust optimal strategies

Decidability of $\mathrm{rVal}(c) < +\infty$ in all WTGs

Perspectives

Perspectives

Computation of (new) values

Perspectives

Computation of (new) values

Trading memory with probabilities in 1-clock WTG

Perspectives

Computation of (new) values

Trading memory with probabilities in 1-clock WTG

Robust values in 1-clock WTG

Perspectives

Computation of (new) values

Trading memory with probabilities in 1-clock WTG

Stochastic values in stochastic timed games

Robust values in 1-clock WTG

Perspectives

Computation of (new) values

Trading memory with probabilities in 1-clock WTG

Stochastic values in stochastic timed games

Robust values in 1-clock WTG

Robust stochastic value from strategies with continuous distribution on delays

Perspectives

Computation of (new) values

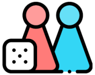
Trading memory with probabilities in 1-clock WTG

Stochastic values in stochastic timed games

Robust values in 1-clock WTG

Robust stochastic value from strategies with continuous distribution on delays

Using probabilities in (others) games



Perspectives

Computation of (new) values

Trading memory with probabilities in 1-clock WTG

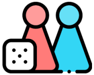
Stochastic values in stochastic timed games

Robust values in 1-clock WTG

Robust stochastic value from strategies with continuous distribution on delays

Using probabilities in (others) games

Polynomial algorithm to solve 0-clock WTG by strategy iteration



Perspectives

Computation of (new) values

Trading memory with probabilities in 1-clock WTG

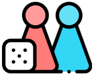
Stochastic values in stochastic timed games

Robust values in 1-clock WTG

Robust stochastic value from strategies with continuous distribution on delays

Using probabilities in (others) games

Polynomial algorithm to solve 0-clock WTG by strategy iteration



Characterisation of memory needed when probabilities are allowed

Perspectives

Computation of (new) values

Trading memory with probabilities in 1-clock WTG

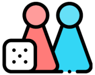
Stochastic values in stochastic timed games

Robust values in 1-clock WTG

Robust stochastic value from strategies with continuous distribution on delays

Using probabilities in (others) games

Polynomial algorithm to solve 0-clock WTG by strategy iteration



Characterisation of memory needed when probabilities are allowed

Implementation

Perspectives

Computation of (new) values

Trading memory with probabilities in 1-clock WTG

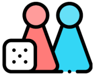
Stochastic values in stochastic timed games

Robust values in 1-clock WTG

Robust stochastic value from strategies with continuous distribution on delays

Using probabilities in (others) games

Polynomial algorithm to solve 0-clock WTG by strategy iteration



Characterisation of memory needed when probabilities are allowed

Implementation

Solving 1-clock WTG

Perspectives

Computation of (new) values

Trading memory with probabilities in 1-clock WTG

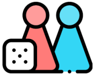
Stochastic values in stochastic timed games

Robust values in 1-clock WTG

Robust stochastic value from strategies with continuous distribution on delays

Using probabilities in (others) games

Polynomial algorithm to solve 0-clock WTG by strategy iteration



Characterisation of memory needed when probabilities are allowed

Implementation

Solving 1-clock WTG

Solving robust acyclic (1-clock) WTG

Appendix

Value Iteration does not converge in finite time in 1-clock WTG

Computation of deterministic value for 1-clock WTG

Existence of the expectation

Partition to compute stochastic values

Robust reachability

Value Iteration for 1-clock WTG

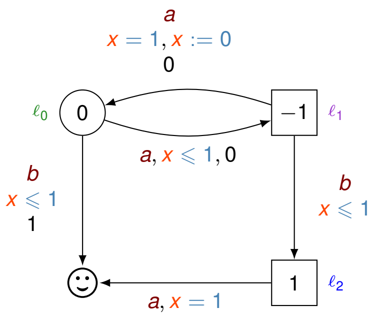
○ Min ☐ Max

$$\mathcal{F}(X)(\ell,\nu) = \begin{cases} 0 & \text{if } \ell = ☺; \\ \inf_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathsf{wt}(a) + t\,\mathsf{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Min;} \\ \sup_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathsf{wt}(a) + t\,\mathsf{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Max.} \end{cases}$$

Value Iteration for 1-clock WTG

Min Max

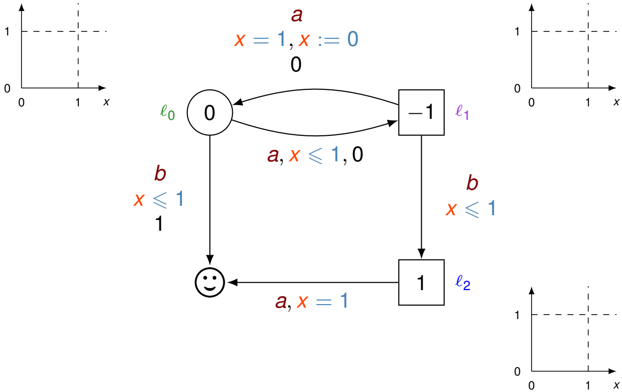
$$
\mathcal{F}(X)(\ell,\nu) = \begin{cases} 0 & \text{if } \ell = ☺; \\ \inf_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathsf{wt}(a) + t\,\mathsf{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Min;} \\ \sup_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathsf{wt}(a) + t\,\mathsf{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Max.} \end{cases}
$$



Value Iteration for 1-clock WTG

○ Min □ Max

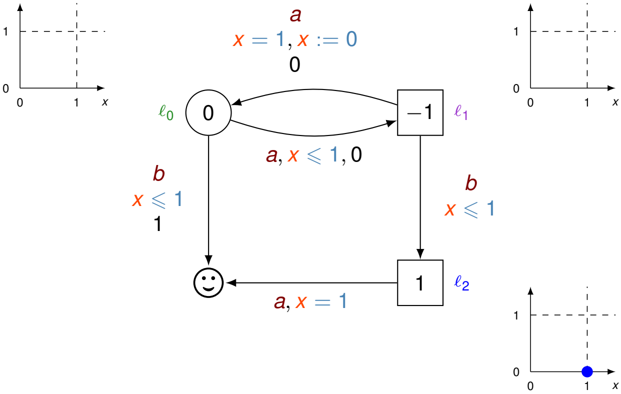
$$\mathcal{F}(X)(\ell,\nu) = \begin{cases} 0 & \text{if } \ell = ☺; \\ \inf_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathrm{wt}(a) + t\,\mathrm{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Min;} \\ \sup_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathrm{wt}(a) + t\,\mathrm{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Max.} \end{cases}$$



Value Iteration for 1-clock WTG

○ Min □ Max

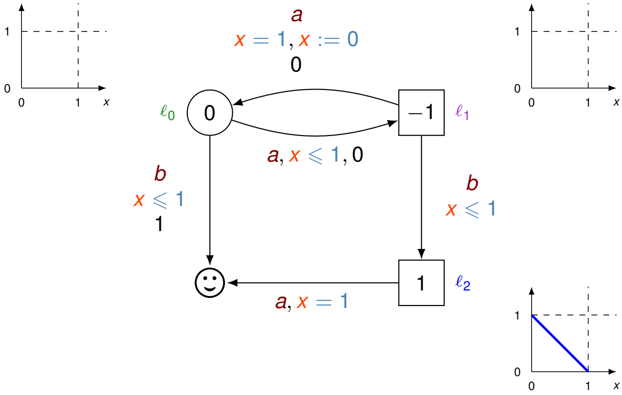
$$\mathcal{F}(X)(\ell,\nu) = \begin{cases} 0 & \text{if } \ell = ☺; \\ \inf_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \left(\mathrm{wt}(a) + t\,\mathrm{wt}(\ell) + X(\ell',\nu')\right) & \text{if } \ell \text{ belongs to Min;} \\ \sup_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \left(\mathrm{wt}(a) + t\,\mathrm{wt}(\ell) + X(\ell',\nu')\right) & \text{if } \ell \text{ belongs to Max.} \end{cases}$$



Value Iteration for 1-clock WTG

Min Max

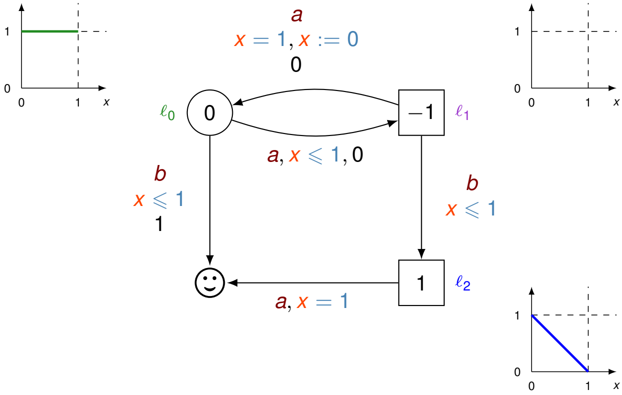
$$\mathcal{F}(X)(\ell,\nu) = \begin{cases} 0 & \text{if } \ell = \smiley; \\ \inf_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathrm{wt}(a) + t\,\mathrm{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Min;} \\ \sup_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathrm{wt}(a) + t\,\mathrm{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Max.} \end{cases}$$



Value Iteration for 1-clock WTG

○ Min □ Max

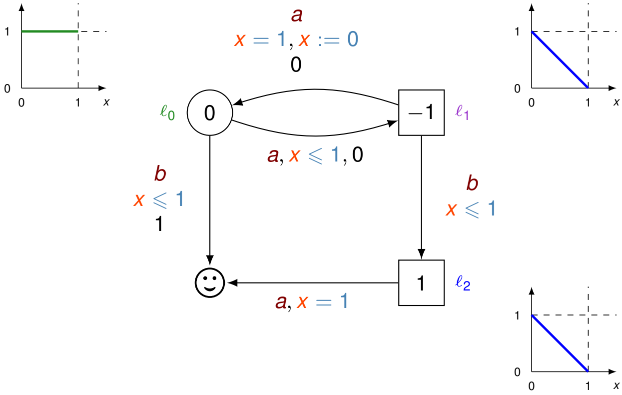
$$\mathcal{F}(X)(\ell,\nu) = \begin{cases} 0 & \text{if } \ell = \smile; \\ \inf_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathrm{wt}(a) + t\,\mathrm{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Min;} \\ \sup_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathrm{wt}(a) + t\,\mathrm{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Max.} \end{cases}$$



Value Iteration for 1-clock WTG

○ Min □ Max

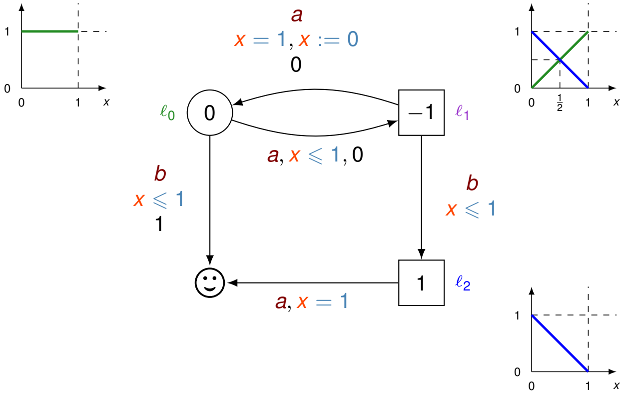
$$\mathcal{F}(X)(\ell,\nu) = \begin{cases} 0 & \text{if } \ell = ☺; \\ \inf_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathrm{wt}(a) + t\,\mathrm{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Min;} \\ \sup_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathrm{wt}(a) + t\,\mathrm{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Max.} \end{cases}$$



Value Iteration for 1-clock WTG

○ Min □ Max

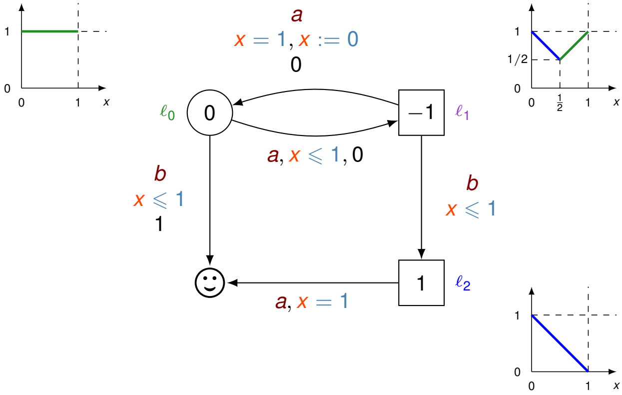
$$\mathcal{F}(X)(\ell,\nu) = \begin{cases} 0 & \text{if } \ell = ☺; \\ \inf_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathrm{wt}(a) + t\,\mathrm{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Min;} \\ \sup_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathrm{wt}(a) + t\,\mathrm{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Max.} \end{cases}$$



Value Iteration for 1-clock WTG

Min Max

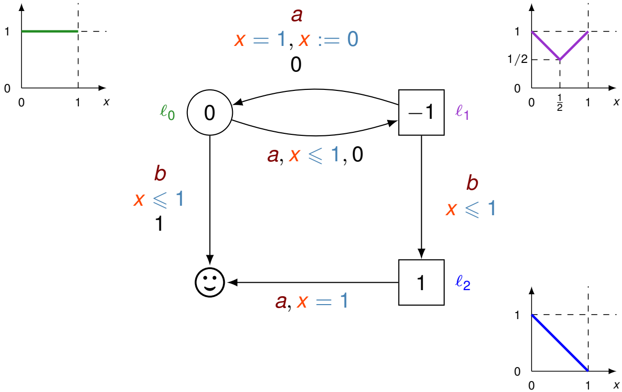
$$\mathcal{F}(X)(\ell,\nu) = \begin{cases} 0 & \text{if } \ell = ☺; \\ \inf_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathrm{wt}(a) + t\,\mathrm{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Min;} \\ \sup_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathrm{wt}(a) + t\,\mathrm{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Max.} \end{cases}$$



Value Iteration for 1-clock WTG

○ Min □ Max

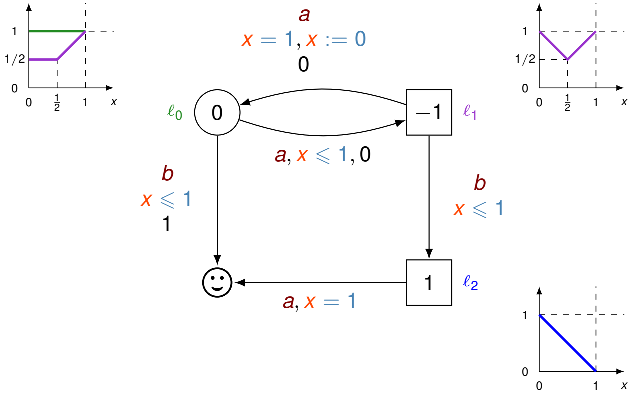
$$\mathcal{F}(X)(\ell,\nu) = \begin{cases} 0 & \text{if } \ell = ☺; \\ \inf_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathrm{wt}(a) + t\,\mathrm{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Min;} \\ \sup_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathrm{wt}(a) + t\,\mathrm{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Max.} \end{cases}$$



Value Iteration for 1-clock WTG

◯ Min ☐ Max

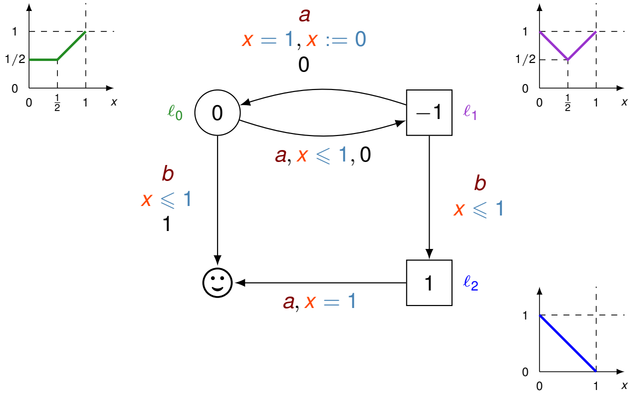
$$\mathcal{F}(X)(\ell,\nu) = \begin{cases} 0 & \text{if } \ell = ☺; \\ \inf_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathrm{wt}(a) + t\,\mathrm{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Min;} \\ \sup_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathrm{wt}(a) + t\,\mathrm{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Max.} \end{cases}$$



Value Iteration for 1-clock WTG

Min Max

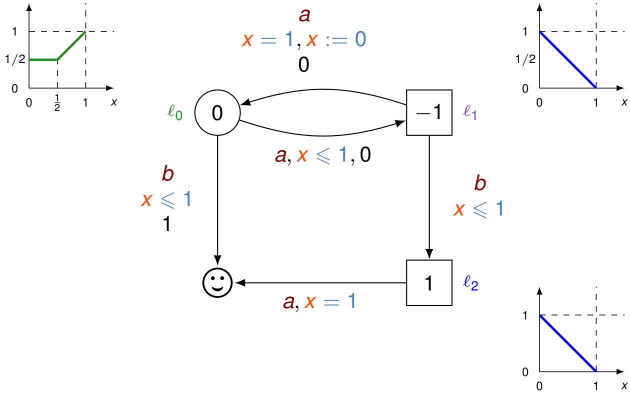
$$\mathcal{F}(X)(\ell,\nu) = \begin{cases} 0 & \text{if } \ell = ☺; \\ \inf_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathrm{wt}(a) + t\,\mathrm{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Min;} \\ \sup_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathrm{wt}(a) + t\,\mathrm{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Max.} \end{cases}$$



Value Iteration for 1-clock WTG

○ Min ☐ Max

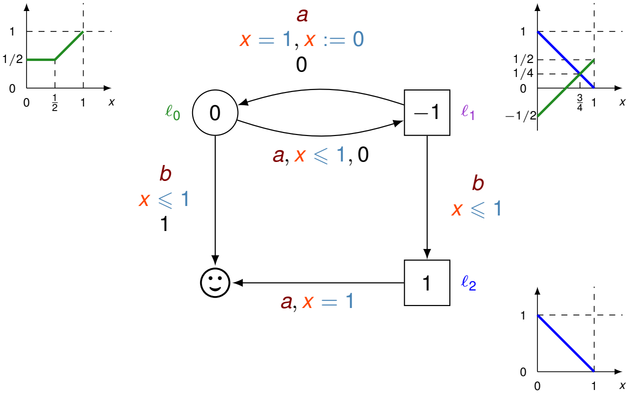
$$\mathcal{F}(X)(\ell,\nu) = \begin{cases} 0 & \text{if } \ell = \smiley; \\ \inf_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathrm{wt}(a) + t\,\mathrm{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Min;} \\ \sup_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathrm{wt}(a) + t\,\mathrm{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Max.} \end{cases}$$



Value Iteration for 1-clock WTG

Min Max

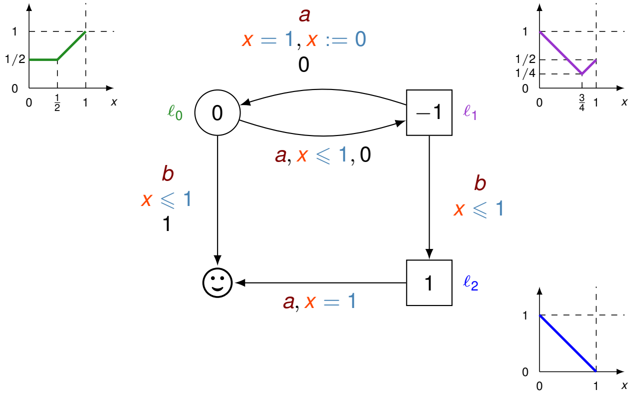
$$\mathcal{F}(X)(\ell,\nu) = \begin{cases} 0 & \text{if } \ell = ☺; \\ \inf_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathsf{wt}(a) + t\,\mathsf{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Min;} \\ \sup_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathsf{wt}(a) + t\,\mathsf{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Max.} \end{cases}$$



Value Iteration for 1-clock WTG

Min Max

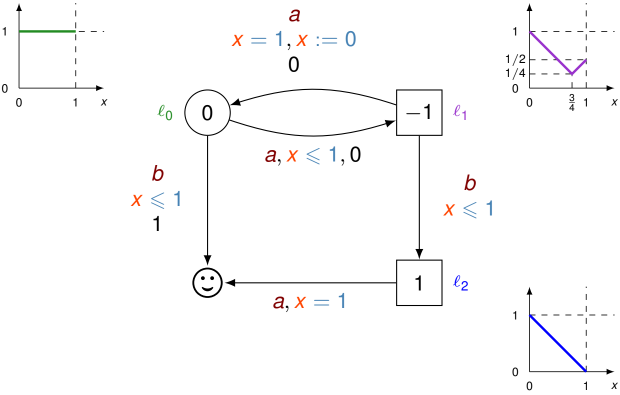
$$\mathcal{F}(X)(\ell,\nu) = \begin{cases} 0 & \text{if } \ell = ☺; \\ \inf_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathrm{wt}(a) + t\,\mathrm{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Min;} \\ \sup_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathrm{wt}(a) + t\,\mathrm{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Max.} \end{cases}$$



Value Iteration for 1-clock WTG

◯ Min ☐ Max

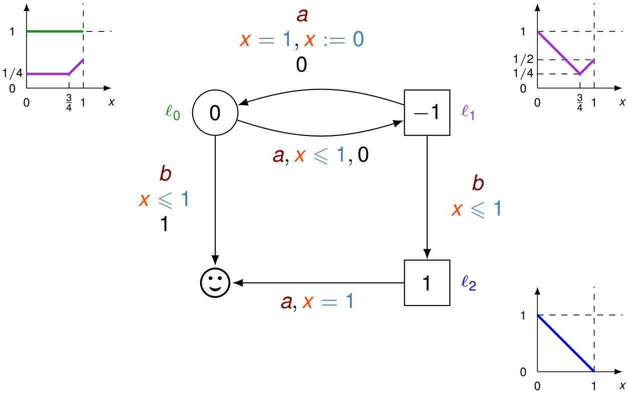
$$\mathcal{F}(X)(\ell,\nu) = \begin{cases} 0 & \text{if } \ell = ☺; \\ \inf_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathrm{wt}(a) + t\,\mathrm{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Min;} \\ \sup_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathrm{wt}(a) + t\,\mathrm{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Max.} \end{cases}$$



Value Iteration for 1-clock WTG

◯ Min ☐ Max

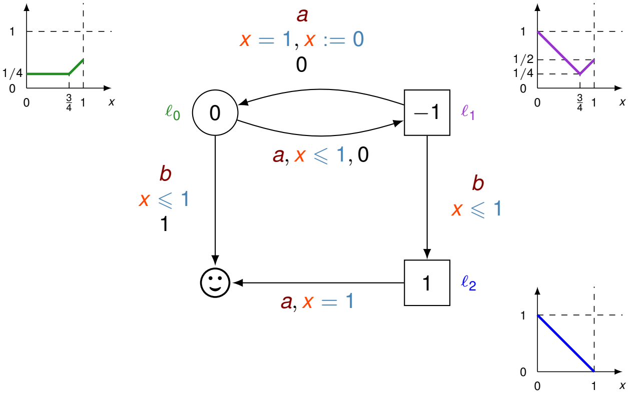
$$\mathcal{F}(X)(\ell,\nu) = \begin{cases} 0 & \text{if } \ell = ☺; \\ \inf_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathrm{wt}(a) + t\,\mathrm{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Min;} \\ \sup_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathrm{wt}(a) + t\,\mathrm{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Max.} \end{cases}$$



Value Iteration for 1-clock WTG

Min Max

$$\mathcal{F}(X)(\ell,\nu) = \begin{cases} 0 & \text{if } \ell = ☺; \\ \inf_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathrm{wt}(a) + t\,\mathrm{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Min;} \\ \sup_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathrm{wt}(a) + t\,\mathrm{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Max.} \end{cases}$$

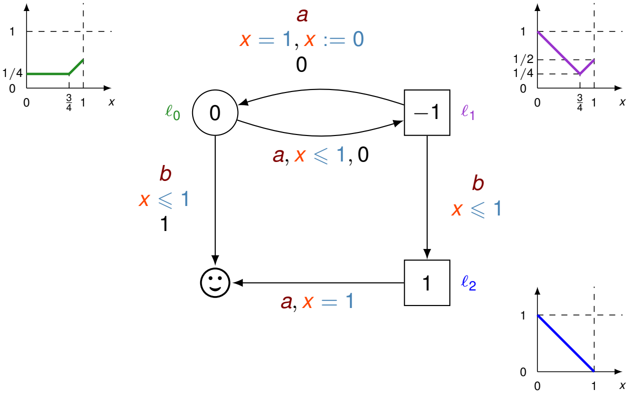


Value Iteration for 1-clock WTG

○ Min □ Max

Does not converge in finite time

$$\mathcal{F}(X)(\ell,\nu) = \begin{cases} 0 & \text{if } \ell = ☺; \\ \inf_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathrm{wt}(a) + t\,\mathrm{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Min;} \\ \sup_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathrm{wt}(a) + t\,\mathrm{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Max.} \end{cases}$$

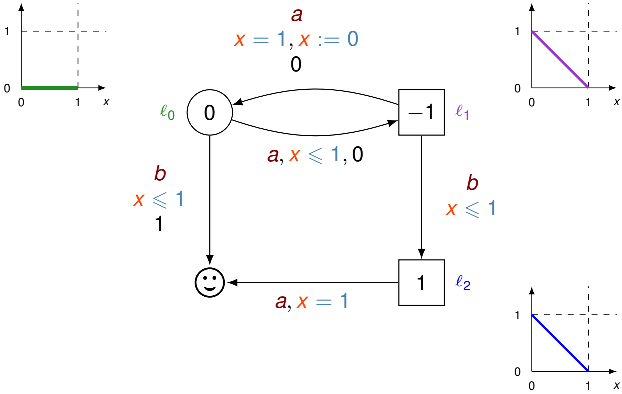


Value Iteration for 1-clock WTG

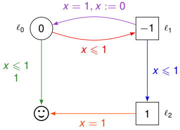
○ Min □ Max

Does not converge in finite time

$$\mathcal{F}(X)(\ell,\nu) = \begin{cases} 0 & \text{if } \ell = ☺; \\ \inf_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathrm{wt}(a) + t\,\mathrm{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Min;} \\ \sup_{(\ell,\nu)\xrightarrow{a,t}(\ell',\nu')} \big(\mathrm{wt}(a) + t\,\mathrm{wt}(\ell) + X(\ell',\nu')\big) & \text{if } \ell \text{ belongs to Max.} \end{cases}$$



Decidability in 1-clock WTGs

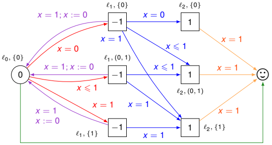


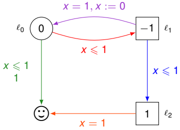
$c \mapsto \mathrm{Val}(c)$ is computable

Decidability in 1-clock WTGs

Encoding

Regions





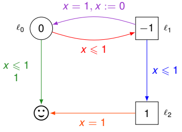
$c \mapsto \mathrm{Val}(c)$ is computable

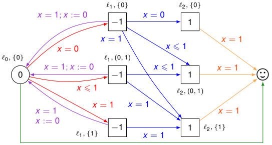
Decidability in 1-clock WTGs

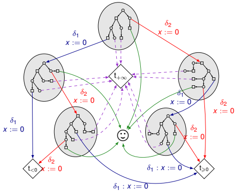
Finite unfolding

Encoding

Regions





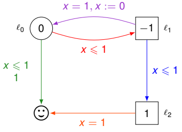


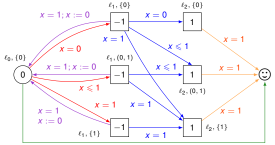
$c \mapsto \mathrm{Val}(c)$ is computable

Decidability in 1-clock WTGs

Encoding

Regions

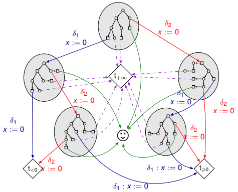




$c \mapsto \mathrm{Val}(c)$ is computable

Finite unfolding

bound the number of reset



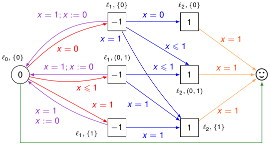
Decidability in 1-clock WTGs

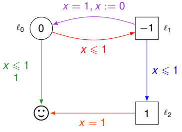
Finite unfolding

bound the number of reset

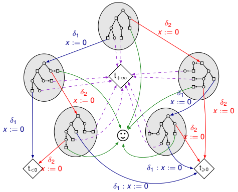
Encoding

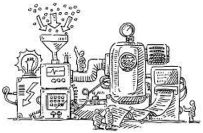
Regions

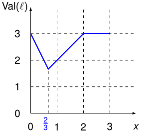




$c \mapsto \text{Val}(c)$ is computable







Back-time algorithm

Solving

One-Clock Priced Timed Games with Negative Weights, T. Brihaye, G. Geeraerts, A. Haddad, E. Lefaucheux, B. Monmege, Log. Methods Comput. Sci., 2022

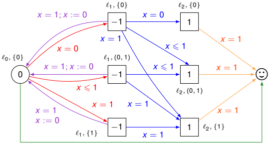
Decidability in 1-clock WTGs

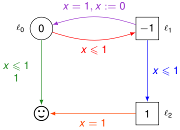
Finite unfolding

bound the number of reset

Encoding

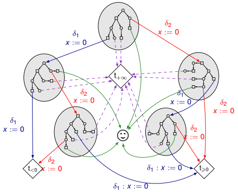
Regions

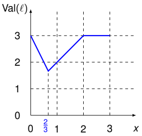


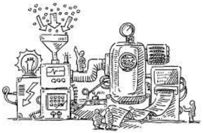


$c \mapsto \mathrm{Val}(c)$ is computable

in exponential time







Back-time algorithm

Solving

Decidability in 1-clock WTGs

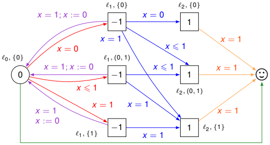
Finite unfolding

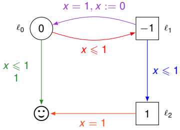
bound the number of reset

Encoding

Regions

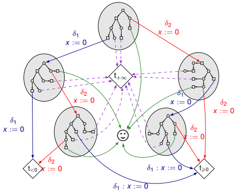
polynomial

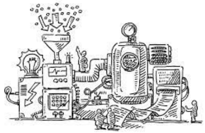


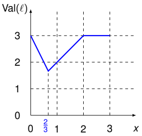


$c \mapsto \mathrm{Val}(c)$ is computable

in exponential time







Back-time algorithm

Solving

Decidability in 1-clock WTGs

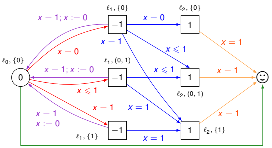
Finite unfolding

bound the number of reset

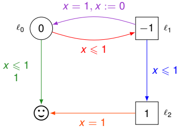
Encoding

Regions

polynomial

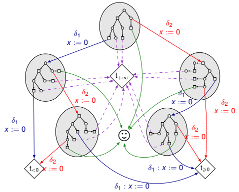


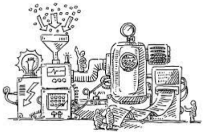
exponential

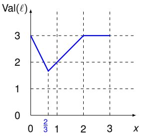


$c \mapsto \mathrm{Val}(c)$ is computable

in exponential time







Back-time algorithm

Solving

Decidability in 1-clock WTGs

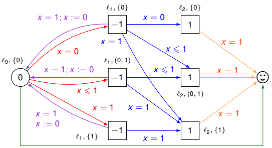
Finite unfolding

bound the number of reset

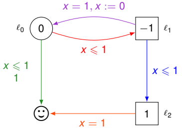
Encoding

Regions

polynomial

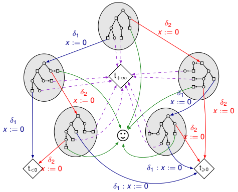


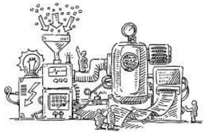
exponential

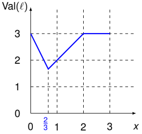


$c \mapsto \text{Val}(c)$ is computable

in exponential time







Back-time algorithm

pseudo-

polynomial

Solving

One-Clock Priced Timed Games with Negative Weights, T. Brihaye, G. Geeraerts, A. Haddad, E. Lefaucheux, B. Monmege, Log. Methods Comput. Sci., 2022

Existence of the expectation: $\mathbb{E}_c^{\eta,\theta}(\mathbf{cost})$ η Min θ Max

Existence of the expectation: $\mathbb{E}_c^{\eta,\theta}(\textbf{cost})$ (η) Min [θ] Max

$\eta, \theta : C^*C \to \Delta(C)$

Distribution over possible choices

1. Edge *a*: finite distribution $\eta_E(c)$
2. Delay for *a*: infinite distribution: $\eta_{\mathbb{R}^+}(c, a)$

Existence of the expectation: $\mathbb{E}_c^{\eta,\theta}(\mathbf{cost})$ $\bigcirc\!\eta$ Min $\boxed{\theta}$ Max

$\eta, \theta : C^* C \to \Delta(C)$

Distribution over possible choices

1. Edge a: finite distribution $\eta_E(c)$
2. Delay for a: infinite distribution: $\eta_{\mathbb{R}^+}(c, a)$

Path $\quad \pi = (c, a_1 \dots a_n) = \{t_1, \dots, t_n \mid c \xrightarrow{t_1, a_1} \dots \xrightarrow{t_n, e_n}\}$

Existence of the expectation: $\mathbb{E}_c^{\eta,\theta}(\textbf{cost})$ η Min θ Max

$\eta, \theta : C^* C \to \Delta(C)$

Distribution over possible choices

1. Edge a: finite distribution $\eta_E(c)$
2. Delay for a: infinite distribution: $\eta_{\mathbb{R}^+}(c, a)$

Path $\pi = (c, a_1 \dots a_n) = \{t_1, \dots, t_n \mid c \xrightarrow{t_1, a_1} \dots \xrightarrow{t_n, e_n}\}$

Probability of a path

$$\mathbb{P}_c^{\eta,\theta}(a\,\pi) = \int_{t \in I(c,a)} \eta_E(c)(a)\, \mathbb{P}_{c_1}^{\eta,\theta}(\pi)\, \mathrm{d}\eta_{\mathbb{R}^+}(c, a)(t)$$

Stochastic Timed Automata, N. Bertrand, P. Bouyer, T. Brihaye, Q. Menet, C. Baier, M. Grosser, and M. Jurdzinzki, 2014, Logical Methods in Computer Science

Existence of the expectation: $\mathbb{E}_c^{\eta,\theta}$(**cost**) $\textcircled{\eta}$ Min $\boxed{\theta}$ Max

$\eta, \theta : C^*C \to \Delta(C)$

Distribution over possible choices

1. Edge a: finite distribution $\eta_E(c)$
2. Delay for a: infinite distribution: $\eta_{\mathbb{R}^+}(c, a)$

Path $\quad \pi = (c, a_1 \dots a_n) = \{t_1, \dots, t_n \mid c \xrightarrow{t_1, a_1} \dots \xrightarrow{t_n, e_n}\}$

Probability of a path

$$\mathbb{P}_c^{\eta,\theta}(a\,\pi) = \int_{t \in I(c,a)} \eta_E(c)(a)\, \mathbb{P}_{c_1}^{\eta,\theta}(\pi)\, \mathrm{d}\eta_{\mathbb{R}^+}(c, a)(t)$$

Stochastic Timed Automata, N. Bertrand, P. Bouyer, T. Brihaye, Q. Menet, C. Baier, M. Grosser, and M. Jurdzinzki, 2014, Logical Methods in Computer Science

Existence of the expectation: $\mathbb{E}_c^{\eta,\theta}(\textbf{cost})$ $\;\eta$ Min $\;\theta$ Max

$\eta, \theta : C^* C \to \Delta(C)$

Distribution over possible choices

1. Edge a: finite distribution $\eta_E(c)$
2. Delay for a: infinite distribution: $\eta_{\mathbb{R}^+}(c, a)$

Measurability conditions on η and θ

Path $\quad \pi = (c, a_1 \dots a_n) = \{t_1, \dots, t_n \mid c \xrightarrow{t_1, a_1} \dots \xrightarrow{t_n, e_n}\}$

Probability of a path

$$\mathbb{P}_c^{\eta,\theta}(a\,\pi) = \int_{t \in I(c,a)} \eta_E(c)(a)\, \mathbb{P}_{c_1}^{\eta,\theta}(\pi)\, \mathrm{d}\eta_{\mathbb{R}^+}(c, a)(t)$$

Existence of the expectation: $\mathbb{E}_c^{\eta,\theta}(\textbf{cost})$ $\;(\eta)$ Min $\;\boxed{\theta}$ Max

$\eta, \theta : C^* C \to \Delta(C)$

Distribution over possible choices

1. Edge a: finite distribution $\eta_E(c)$
2. Delay for a: infinite distribution: $\eta_{\mathbb{R}^+}(c, a)$

⚠ Measurability conditions on η and θ

Path $\quad \pi = (c, a_1 \dots a_n) = \{t_1, \dots, t_n \mid c \xrightarrow{t_1, a_1} \dots \xrightarrow{t_n, e_n}\}$

Probability of a path

$$\mathbb{P}_c^{\eta,\theta}(a\,\pi) = \int_{t \in I(c,a)} \eta_E(c)(a)\, \mathbb{P}_{c_1}^{\eta,\theta}(\pi)\, \mathrm{d}\eta_{\mathbb{R}^+}(c, a)(t)$$

Expectation of **cost** in a path

$$\mathbb{E}_c^{\eta,\theta}(a\,\pi) = \int_{t \in I(c,a)} \eta_E(c)(a) \left[(t\, \mathrm{wt}(c) + \mathrm{wt}(a))\, \mathbb{P}_{c_1}^{\eta,\theta}(\pi) + \mathbb{E}_{c_1}^{\eta,\theta}(\pi) \right] \mathrm{d}\eta_{\mathbb{R}^+}(c, a)(t)$$

Existence of the expectation: $\mathbb{E}_c^{\eta,\theta}(\textbf{cost})$ η Min θ Max

$\eta, \theta : C^*C \to \Delta(C)$

Distribution over possible choices

1. Edge a: finite distribution $\eta_E(c)$
2. Delay for a: infinite distribution: $\eta_{\mathbb{R}^+}(c, a)$

Measurability conditions on η and θ

Path $\pi = (c, a_1 \dots a_n) = \{t_1, \dots, t_n \mid c \xrightarrow{t_1, a_1} \dots \xrightarrow{t_n, e_n}\}$

Probability of a path

$$\mathbb{P}_c^{\eta,\theta}(a\,\pi) = \int_{t \in I(c,a)} \eta_E(c)(a)\, \mathbb{P}_{c_1}^{\eta,\theta}(\pi)\, \mathrm{d}\eta_{\mathbb{R}^+}(c, a)(t)$$

Expectation of **cost** in a path

$$\mathbb{E}_c^{\eta,\theta}(a\,\pi) = \int_{t \in I(c,a)} \eta_E(c)(a) \left[(t\,\mathrm{wt}(c) + \mathrm{wt}(a))\, \mathbb{P}_{c_1}^{\eta,\theta}(\pi) + \mathbb{E}_{c_1}^{\eta,\theta}(\pi)\right]\, \mathrm{d}\eta_{\mathbb{R}^+}(c, a)(t)$$

Existence of the expectation: $\mathbb{E}_c^{\eta,\theta}(\textbf{cost})$ η Min θ Max

$\eta, \theta : C^* C \to \Delta(C)$

Distribution over possible choices

1. Edge a: finite distribution $\eta_E(c)$
2. Delay for a: infinite distribution: $\eta_{\mathbb{R}^+}(c, a)$

⚠ Measurability conditions on η and θ

Path $\quad \pi = (c, a_1 \dots a_n) = \{t_1, \dots, t_n \mid c \xrightarrow{t_1, a_1} \dots \xrightarrow{t_n, e_n}\}$

Expectation of **cost**

$$\mathbb{E}_c^{\eta,\theta}(\textbf{cost}) = \sum_{\pi} \mathbb{E}_c^{\eta,\theta}(\pi)$$

Existence of the expectation: $\mathbb{E}_c^{\eta,\theta}(\mathbf{cost})$

η Min θ Max

$\eta, \theta : C^* C \to \Delta(C)$

Distribution over possible choices

1. Edge a: finite distribution $\eta_E(c)$
2. Delay for a: infinite distribution: $\eta_{\mathbb{R}^+}(c, a)$

Measurability conditions on η and θ

Path $\pi = (c, a_1 \dots a_n) = \{t_1, \dots, t_n \mid c \xrightarrow{t_1, a_1} \dots \xrightarrow{t_n, e_n}\}$

Expectation of **cost**

$$\mathbb{E}_c^{\eta,\theta}(\mathbf{cost}) = \sum_{\pi} \mathbb{E}_c^{\eta,\theta}(\pi)$$

Restrictions on strategies for Min

Existence of the expectation: $\mathbb{E}_c^{\eta,\theta}(\mathbf{cost})$ η Min θ Max

$\eta, \theta : C^*C \to \Delta(C)$

Distribution over possible choices

1. Edge a: finite distribution $\eta_E(c)$
2. Delay for a: infinite distribution: $\eta_{\mathbb{R}^+}(c, a)$

Measurability conditions on η and θ

Path $\pi = (c, a_1 \dots a_n) = \{t_1, \dots, t_n \mid c \xrightarrow{t_1, a_1} \dots \xrightarrow{t_n, e_n}\}$

Expectation of **cost**

$$\mathbb{E}_c^{\eta,\theta}(\mathbf{cost}) = \sum_{\pi} \mathbb{E}_c^{\eta,\theta}(\pi)$$

Restrictions on strategies for Min

- For all θ, $\mathbb{P}_c^{\eta,\theta}(\diamond☺) = 1$

Existence of the expectation: $\mathbb{E}_c^{\eta,\theta}(\textbf{cost})$ $\circled{\eta}$ Min $\boxed{\theta}$ Max

$\eta, \theta : C^*C \to \Delta(C)$

Distribution over possible choices

1. Edge a: finite distribution $\eta_E(c)$
2. Delay for a: infinite distribution: $\eta_{\mathbb{R}^+}(c, a)$

Measurability conditions on η and θ

Path $\pi = (c, a_1 \dots a_n) = \{t_1, \dots, t_n \mid c \xrightarrow{t_1, a_1} \dots \xrightarrow{t_n, e_n}\}$

Expectation of **cost**

$$\mathbb{E}_c^{\eta,\theta}(\textbf{cost}) = \sum_{\substack{\pi \\ \pi \models \diamond \text{☺}}} \mathbb{E}_c^{\eta,\theta}(\pi)$$

Restrictions on strategies for Min

- For all θ, $\mathbb{P}_c^{\eta,\theta}(\diamond \text{☺}) = 1$

Existence of the expectation: $\mathbb{E}_c^{\eta,\theta}(\mathbf{cost})$ $\textcircled{\eta}$ Min $\boxed{\theta}$ Max

$\eta, \theta : C^*C \to \Delta(C)$

Distribution over possible choices

1. Edge a: finite distribution $\eta_E(c)$
2. Delay for a: infinite distribution: $\eta_{\mathbb{R}^+}(c, a)$

⚠ Measurability conditions on η and θ

Path $\pi = (c, a_1 \dots a_n) = \{t_1, \dots, t_n \mid c \xrightarrow{t_1, a_1} \dots \xrightarrow{t_n, e_n}\}$

Expectation of **cost**

$$\mathbb{E}_c^{\eta,\theta}(\mathbf{cost}) = \sum_{\substack{\pi \\ \pi \models \diamond ☺}} \mathbb{E}_c^{\eta,\theta}(\pi)$$

Convergence ?

Restrictions on strategies for Min

- For all θ, $\mathbb{P}_c^{\eta,\theta}(\diamond ☺) = 1$

Existence of the expectation: $\mathbb{E}_c^{\eta,\theta}(\textbf{cost})$

η Min θ Max

$\eta, \theta : C^*C \to \Delta(C)$

Distribution over possible choices

1. Edge a: finite distribution $\eta_E(c)$
2. Delay for a: infinite distribution: $\eta_{\mathbb{R}^+}(c, a)$

Measurability conditions on η and θ

Path $\pi = (c, a_1 \dots a_n) = \{t_1, \dots, t_n \mid c \xrightarrow{t_1, a_1} \dots \xrightarrow{t_n, e_n}\}$

Expectation of **cost**

$$\mathbb{E}_c^{\eta,\theta}(\textbf{cost}) = \sum_{\substack{\pi \\ \pi \models \diamond ☺}} \mathbb{E}_c^{\eta,\theta}(\pi)$$

Convergence ?

$|\mathbb{E}_c^{\eta,\theta}(\pi)|$

Restrictions on strategies for Min

- For all θ, $\mathbb{P}_c^{\eta,\theta}(\diamond ☺) = 1$

Existence of the expectation: $\mathbb{E}_c^{\eta,\theta}(\textbf{cost})$ $\quad$ η Min $\quad$ θ Max

$\eta, \theta : C^*C \to \Delta(C)$

Distribution over possible choices

1. Edge a: finite distribution $\eta_E(c)$
2. Delay for a: infinite distribution: $\eta_{\mathbb{R}^+}(c, a)$

Measurability conditions on η and θ

Path $\quad \pi = (c, a_1 \ldots a_n) = \{t_1, \ldots, t_n \mid c \xrightarrow{t_1, a_1} \ldots \xrightarrow{t_n, e_n}\}$

Expectation of **cost**

$$\mathbb{E}_c^{\eta,\theta}(\textbf{cost}) = \sum_{\substack{\pi \\ \pi \models \diamond \smiley}} \mathbb{E}_c^{\eta,\theta}(\pi)$$

Convergence ?

$$|\mathbb{E}_c^{\eta,\theta}(\pi)| \leqslant k|\pi|\ \alpha^{-|\pi|}$$

Restrictions on strategies for Min

- For all θ, $\mathbb{P}_c^{\eta,\theta}(\diamond \smiley) = 1$

Existence of the expectation: $\mathbb{E}_c^{\eta,\theta}(\mathbf{cost})$ ⃝η Min □θ Max

$\eta, \theta : C^* C \to \Delta(C)$

Distribution over possible choices

1. Edge a: finite distribution $\eta_E(c)$
2. Delay for a: infinite distribution: $\eta_{\mathbb{R}^+}(c, a)$

Measurability conditions on η and θ

Path $\pi = (c, a_1 \dots a_n) = \{t_1, \dots, t_n \mid c \xrightarrow{t_1, a_1} \dots \xrightarrow{t_n, e_n}\}$

Expectation of **cost**

$$\mathbb{E}_c^{\eta,\theta}(\mathbf{cost}) = \sum_{\substack{\pi \\ \pi \models \diamond \smiley}} \mathbb{E}_c^{\eta,\theta}(\pi)$$

Convergence ?

$$|\mathbb{E}_c^{\eta,\theta}(\pi)| \leqslant \underbrace{k|\pi|}_{|\mathbf{cost}(\pi)|} \quad \alpha^{-|\pi|}$$

Restrictions on strategies for Min

- For all θ, $\mathbb{P}_c^{\eta,\theta}(\diamond \smiley) = 1$

Existence of the expectation: $\mathbb{E}_c^{\eta,\theta}(\textbf{cost})$ $\bigcirc\!\!\!\!\eta$ Min $\boxed{\theta}$ Max

$\eta, \theta : C^* C \to \Delta(C)$

Distribution over possible choices

1. Edge a: finite distribution $\eta_E(c)$
2. Delay for a: infinite distribution: $\eta_{\mathbb{R}^+}(c, a)$

⚠ Measurability conditions on η and θ

Path $\quad \pi = (c, a_1 \dots a_n) = \{t_1, \dots, t_n \mid c \xrightarrow{t_1, a_1} \dots \xrightarrow{t_n, e_n}\}$

Expectation of **cost**

$$\mathbb{E}_c^{\eta,\theta}(\textbf{cost}) = \sum_{\substack{\pi \\ \pi \models \diamond ☺}} \mathbb{E}_c^{\eta,\theta}(\pi)$$

Convergence ?

$$|\mathbb{E}_c^{\eta,\theta}(\pi)| \leqslant \underbrace{k|\pi|}_{|\textbf{cost}(\pi)|} \underbrace{\alpha^{-|\pi|}}_{\mathbb{P}_c^{\eta,\theta}(\pi)}$$

Restrictions on strategies for Min

- For all θ, $\mathbb{P}_c^{\eta,\theta}(\diamond ☺) = 1$

Existence of the expectation: $\mathbb{E}_c^{\eta,\theta}(\textbf{cost})$

η Min θ Max

$\eta, \theta : C^* C \to \Delta(C)$

Distribution over possible choices

1. Edge a: finite distribution $\eta_E(c)$
2. Delay for a: infinite distribution: $\eta_{\mathbb{R}^+}(c, a)$

Measurability conditions on η and θ

Path $\pi = (c, a_1 \dots a_n) = \{t_1, \dots, t_n \mid c \xrightarrow{t_1, a_1} \dots \xrightarrow{t_n, e_n}\}$

Expectation of **cost**

$$\mathbb{E}_c^{\eta,\theta}(\textbf{cost}) = \sum_{\substack{\pi \\ \pi \models \diamond ☺}} \mathbb{E}_c^{\eta,\theta}(\pi)$$

Convergence ?

$$|\mathbb{E}_c^{\eta,\theta}(\pi)| \leqslant \underbrace{k|\pi|}_{|\textbf{cost}(\pi)|} \; \underbrace{\alpha^{-|\pi|}}_{\mathbb{P}_c^{\eta,\theta}(\pi)}$$

Restrictions on strategies for Min

- For all θ, $\mathbb{P}_c^{\eta,\theta}(\diamond ☺) = 1$
- ☺ must be reached quickly enough

Computation of the expectation $\mathbb{E}_c^{\eta_p,\tau}(\mathbf{cost})$

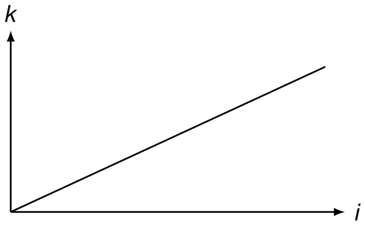
$$\mathbb{E}_c^{\eta_p,\tau}(\mathbf{cost}) = \sum_{\substack{\rho \\ \rho \models \diamond ☺}} \mathbf{cost}(\rho)\, \mathbb{P}(\rho)$$

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

Computation of the expectation $\mathbb{E}_c^{\eta_p,\tau}(\textbf{cost})$

$$\mathbb{E}_c^{\eta_p,\tau}(\textbf{cost}) = \sum_{\substack{\rho \\ \rho \models \diamond ☺}} \textbf{cost}(\rho)\,\mathbb{P}(\rho) = \quad + \quad +$$

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



k size of play reaching the target

i number of choices given by σ_2

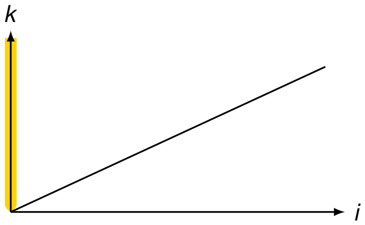
Computation of the expectation $\mathbb{E}_c^{\eta_p,\tau}(\mathbf{cost})$

$$\mathbb{E}_c^{\eta_p,\tau}(\mathbf{cost}) = \sum_{\substack{\rho \\ \rho \models \diamond \smiley}} \mathbf{cost}(\rho)\,\mathbb{P}(\rho) = \mathbb{E} + \quad +$$

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

Yellow zone

All plays conforming to σ_1



k size of play reaching the target

i number of choices given by σ_2

Computation of the expectation $\mathbb{E}_c^{\eta_p,\tau}(\mathbf{cost})$

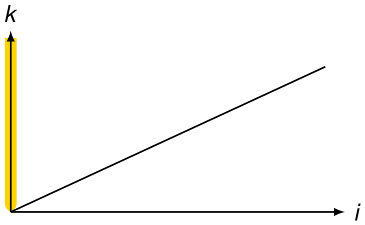
$$\mathbb{E}_c^{\eta_p,\tau}(\mathbf{cost}) = \sum_{\substack{\rho \\ \rho \models \diamond ☺}} \mathbf{cost}(\rho)\, \mathbb{P}(\rho) = \mathbb{E} + \quad +$$

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

Yellow zone

All plays conforming to σ_1

$\mathbf{cost}(\rho) \leqslant \mathrm{dVal}^{\langle \sigma_1, \sigma_2, K \rangle}(c)$



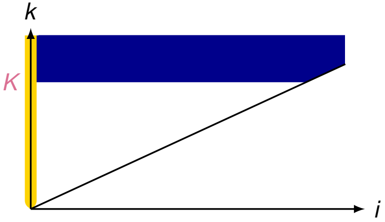
k size of play reaching the target

i number of choices given by σ_2

Computation of the expectation $\mathbb{E}_c^{\eta_p,\tau}(\mathbf{cost})$

$$\mathbb{E}_c^{\eta_p,\tau}(\mathbf{cost}) = \sum_{\substack{\rho \\ \rho \models \diamond ☺}} \mathbf{cost}(\rho)\,\mathbb{P}(\rho) = \mathbb{E} + \mathbb{E} +$$

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



Yellow zone

All plays conforming to σ_1

$\mathbf{cost}(\rho) \leqslant \mathsf{dVal}^{\langle \sigma_1, \sigma_2, K\rangle}(c)$

Blue zone

Plays with many negative cycles

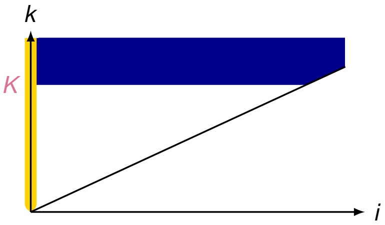
k size of play reaching the target

i number of choices given by σ_2

Computation of the expectation $\mathbb{E}_c^{\eta_p,\tau}(\mathbf{cost})$

$$\mathbb{E}_c^{\eta_p,\tau}(\mathbf{cost}) = \sum_{\substack{\rho \\ \rho \models \Diamond \smiley}} \mathbf{cost}(\rho)\,\mathbb{P}(\rho) = \mathbb{E} + \mathbb{E} +$$

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



Yellow zone

All plays conforming to σ_1

$\mathbf{cost}(\rho) \leqslant \mathrm{dVal}^{\langle \sigma_1, \sigma_2, K \rangle}(c)$

Blue zone

Plays with many negative cycles

$\mathbf{cost}(\rho) \leqslant \mathrm{dVal}^{\langle \sigma_1, \sigma_2, K \rangle}(c)$

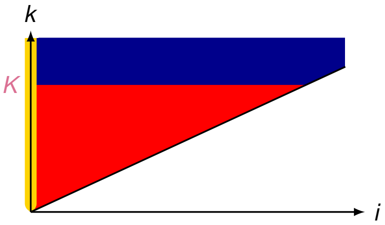
k size of play reaching the target

i number of choices given by σ_2

Computation of the expectation $\mathbb{E}_c^{\eta_p,\tau}(\mathbf{cost})$

$$\mathbb{E}_c^{\eta_p,\tau}(\mathbf{cost}) = \sum_{\substack{\rho \\ \rho \models \Diamond \smiley}} \mathbf{cost}(\rho)\,\mathbb{P}(\rho) = \mathbb{E} + \mathbb{E} + \mathbb{E}$$

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



k size of play reaching the target

i number of choices given by σ_2

Yellow zone

All plays conforming to σ_1

$\mathbf{cost}(\rho) \leqslant \mathsf{dVal}^{\langle \sigma_1, \sigma_2, K \rangle}(c)$

Blue zone

Plays with many negative cycles

$\mathbf{cost}(\rho) \leqslant \mathsf{dVal}^{\langle \sigma_1, \sigma_2, K \rangle}(c)$

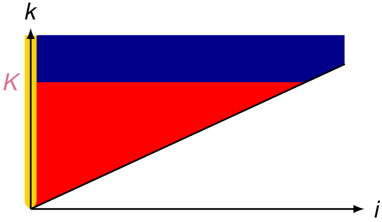
Red zone

Rest of plays

Computation of the expectation $\mathbb{E}_c^{\eta_p,\tau}(\textbf{cost})$

$$\mathbb{E}_c^{\eta_p,\tau}(\textbf{cost}) = \sum_{\substack{\rho \\ \rho \models \diamond ☺}} \textbf{cost}(\rho)\,\mathbb{P}(\rho) = \mathbb{E} + \mathbb{E} + \mathbb{E}$$

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



Yellow zone

All plays conforming to σ_1

$\textbf{cost}(\rho) \leqslant \mathrm{dVal}^{\langle \sigma_1, \sigma_2, K\rangle}(c)$

Blue zone

Plays with many negative cycles

$\textbf{cost}(\rho) \leqslant \mathrm{dVal}^{\langle \sigma_1, \sigma_2, K\rangle}(c)$

k size of play reaching the target

i number of choices given by σ_2

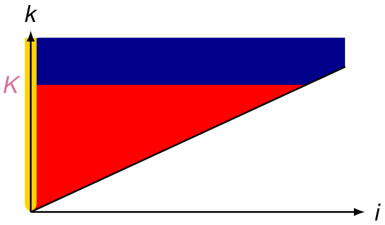
Red zone

Rest of plays $\quad \mathbb{E} \underset{\substack{p \to 1 \\ p<1}}{\longrightarrow} 0$

Computation of the expectation $\mathbb{E}_c^{\eta_p,\tau}(\textbf{cost})$

$$\mathbb{E}_c^{\eta_p,\tau}(\textbf{cost}) = \sum_{\substack{\rho \\ \rho \models \diamond ☺}} \textbf{cost}(\rho)\,\mathbb{P}(\rho) = \mathbb{E} + \mathbb{E} + \mathbb{E}$$

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



k size of play reaching the target

i number of choices given by σ_2

Yellow zone

All plays conforming to σ_1

$\textbf{cost}(\rho) \leqslant \mathrm{dVal}^{\langle \sigma_1, \sigma_2, K\rangle}(c)$

Blue zone

Plays with many negative cycles

$\textbf{cost}(\rho) \leqslant \mathrm{dVal}^{\langle \sigma_1, \sigma_2, K\rangle}(c)$

$$\lim_{\substack{p \to 1 \\ p < 1}} \mathbb{E} + \mathbb{E} \leqslant \mathrm{dVal}^{\langle \sigma_1, \sigma_2 K\rangle}(c)$$

Red zone

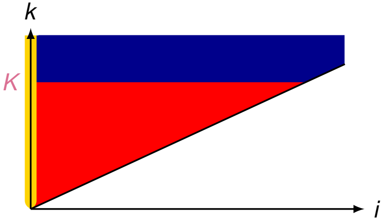
Rest of plays

$$\mathbb{E} \underset{\substack{p \to 1 \\ p < 1}}{\longrightarrow} 0$$

Computation of the expectation $\mathbb{E}_c^{\eta_p,\tau}(\textbf{cost})$

$$\mathbb{E}_c^{\eta_p,\tau}(\textbf{cost}) = \sum_{\substack{\rho \\ \rho \models \diamond \smiley}} \textbf{cost}(\rho)\,\mathbb{P}(\rho) = \mathbb{E} + \mathbb{E} + \mathbb{E} \Rightarrow \lim_{\substack{p \to 1 \\ p<1}} \mathbb{E}_c^{\eta_p,\tau}(\textbf{cost}) \leqslant \mathsf{dVal}^{\langle \sigma_1, \sigma_2, K\rangle}(c)$$

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



k size of play reaching the target

i number of choices given by σ_2

Yellow zone

All plays conforming to σ_1

$\textbf{cost}(\rho) \leqslant \mathsf{dVal}^{\langle \sigma_1, \sigma_2, K\rangle}(c)$

Blue zone

Plays with many negative cycles

$\textbf{cost}(\rho) \leqslant \mathsf{dVal}^{\langle \sigma_1, \sigma_2, K\rangle}(c)$

$$\lim_{\substack{p \to 1 \\ p<1}} \mathbb{E} + \mathbb{E} \leqslant \mathsf{dVal}^{\langle \sigma_1, \sigma_2 K\rangle}(c)$$

Red zone

Rest of plays

$$\mathbb{E} \underset{\substack{p \to 1 \\ p<1}}{\longrightarrow} 0$$

Robust reachability

Deciding if exists $\delta > 0$ such that Min reaches ☺ when Max perturbs with $[0, \delta]$?

Robust reachability

Deciding if exists $\delta > 0$ such that Min reaches ☺ when Max perturbs with $[0, \delta]$?

Theorem (SUBMITTED): Robust reachability problem is EXPTIME-complete

Robust reachability

Deciding if exists $\delta > 0$ such that Min reaches ☺ when Max perturbs with $[0, \delta]$?

Theorem (SUBMITTED): Robust reachability problem is EXPTIME-complete

- hardness

Probabilistic Robust Timed Game, Y. Oualhadj, PA. Reynier, and O. Sankur, 2014, CONCUR

Robust reachability

Deciding if exists $\delta > 0$ such that Min reaches ☺ when Max perturbs with $[0, \delta]$?

Theorem (SUBMITTED): Robust reachability problem is EXPTIME-complete

- hardness
- easiness:

Robust reachability

Deciding if exists $\delta > 0$ such that Min reaches ☺ when Max perturbs with $[0, \delta]$?

Theorem (SUBMITTED): Robust reachability problem is EXPTIME-complete

- hardness
- easiness: encoding the fixed-δ semantics into excess one

Excessive semantics

Check the guard before the perturbation:

Robust reachability

Deciding if exists $\delta > 0$ such that Min reaches ☺ when Max perturbs with $[0, \delta]$?

Theorem (SUBMITTED): Robust reachability problem is EXPTIME-complete

- hardness
- easiness: encoding the fixed-δ semantics into excess one

Excessive semantics

Check the guard before the perturbation: $\nu + t$ satisfies the guard

Robust reachability

Deciding if exists $\delta > 0$ such that Min reaches ☺ when Max perturbs with $[0, \delta]$?

Theorem (SUBMITTED): Robust reachability problem is EXPTIME-complete

- hardness
- easiness: encoding the fixed-δ semantics into excess one

Excessive semantics

Check the guard before the perturbation: $\nu + t$ satisfies the guard
Robust reachability under the excessive semantics is EXPTIME-complete

---

Robust Controller Synthesis in Timed Automata, O. Sankur, P. Bouyer, N.s Markey, and PA. Reynier, 2013, CONCUR

Robust Reachability in Timed Automata: Game-Based Approach, P. Bouyer, N. Markey, and O. Sankur, 2015, TCS

Robust reachability

Deciding if exists $\delta > 0$ such that Min reaches ☺ when Max perturbs with $[0, \delta]$?

Theorem (SUBMITTED): Robust reachability problem is EXPTIME-complete

- hardness
- easiness: encoding the fixed-δ semantics into excess one

Excessive semantics

Check the guard before the perturbation: $\nu + t$ satisfies the guard
Robust reachability under the excessive semantics is EXPTIME-complete

Encoding fixed-δ semantics into the excess one

Robust reachability

Deciding if exists $\delta > 0$ such that Min reaches ☺ when Max perturbs with $[0, \delta]$?

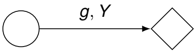
Theorem (SUBMITTED): Robust reachability problem is EXPTIME-complete

- hardness
- easiness: encoding the fixed-δ semantics into excess one

Excessive semantics

Check the guard before the perturbation: $\nu + t$ satisfies the guard
Robust reachability under the excessive semantics is EXPTIME-complete

Encoding fixed-δ semantics into the excess one



Robust reachability

Deciding if exists $\delta > 0$ such that Min reaches ☺ when Max perturbs with $[0, \delta]$?

Theorem (SUBMITTED): Robust reachability problem is EXPTIME-complete

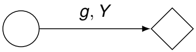
- hardness
- easiness: encoding the fixed-δ semantics into excess one

Excessive semantics

Check the guard before the perturbation: $\nu + t$ satisfies the guard
Robust reachability under the excessive semantics is EXPTIME-complete

Encoding fixed-δ semantics into the excess one

Max controls a posteriori the delay chosen by Min



Robust reachability

Deciding if exists $\delta > 0$ such that Min reaches ☺ when Max perturbs with $[0, \delta]$?

Theorem (SUBMITTED): Robust reachability problem is EXPTIME-complete

- hardness
- easiness: encoding the fixed-δ semantics into excess one

Excessive semantics

Check the guard before the perturbation: $\nu + t$ satisfies the guard
Robust reachability under the excessive semantics is EXPTIME-complete

Encoding fixed-δ semantics into the excess one

Max controls a posteriori the delay chosen by Min