Synthesis of Robust Optimal Real-Time Systems

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

[1]University of Warsaw
[2]Aix–Marseille Université

MFCS 2024

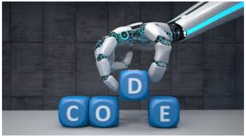
Motivation: game theory for synthesis

Classical approach

Check the correctness of a system

Game theory

Interaction between two antagonistic agents: environment and controller

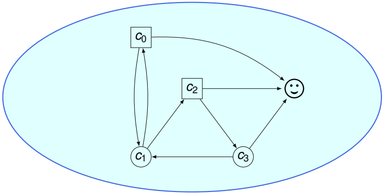


Code synthesis

Correct by construction: synthesis of controller

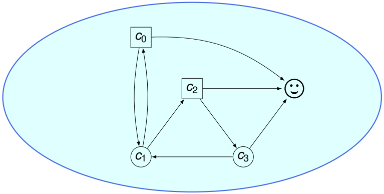
Different classes of games

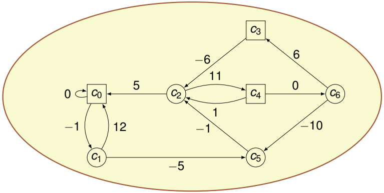
Qualitative games



Different classes of games

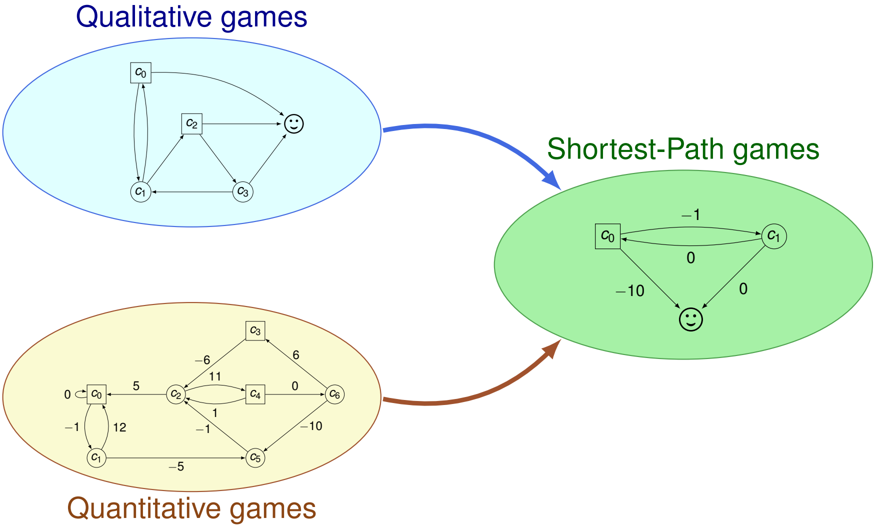
Qualitative games



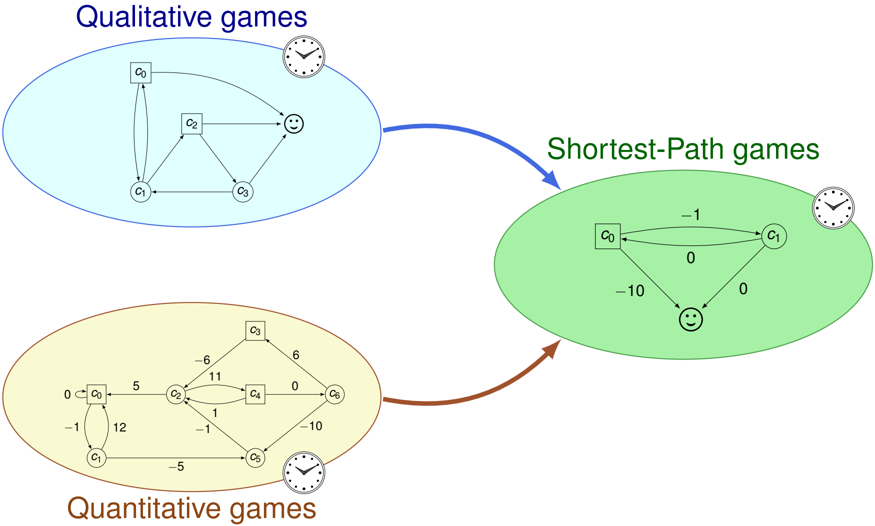


Quantitative games

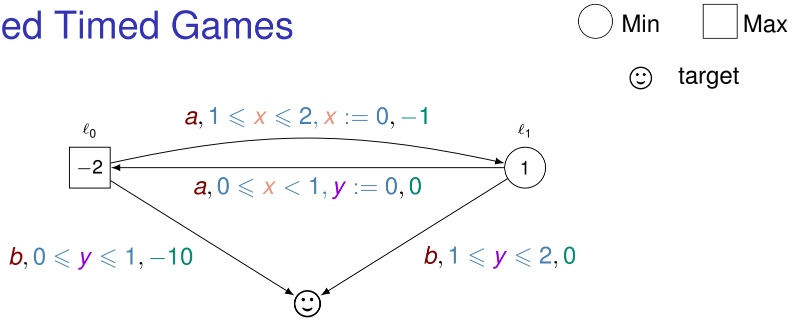
Different classes of games



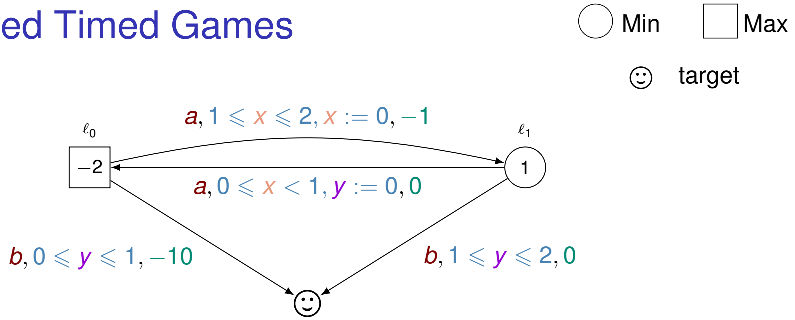
Different classes of games



Weighted Timed Games



Weighted Timed Games

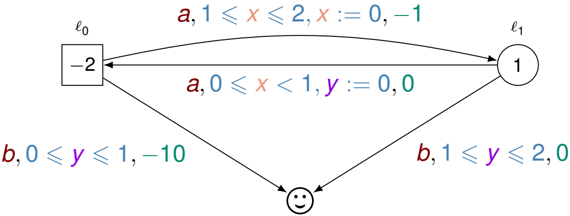


Play ρ $\left(\ell_1, \begin{bmatrix} x \mapsto 0 \\ y \mapsto 0 \end{bmatrix}\right)$

Weighted Timed Games

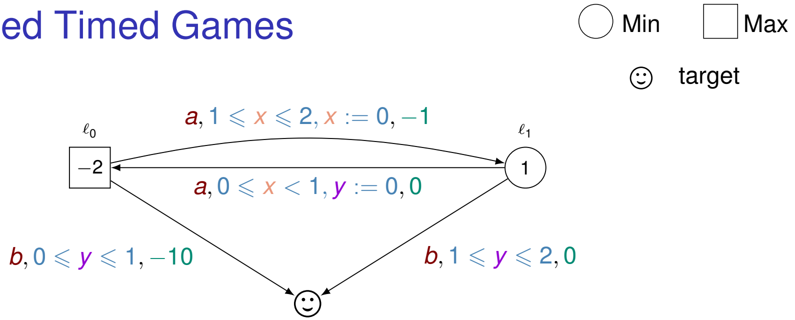
○ Min ☐ Max

☺ target



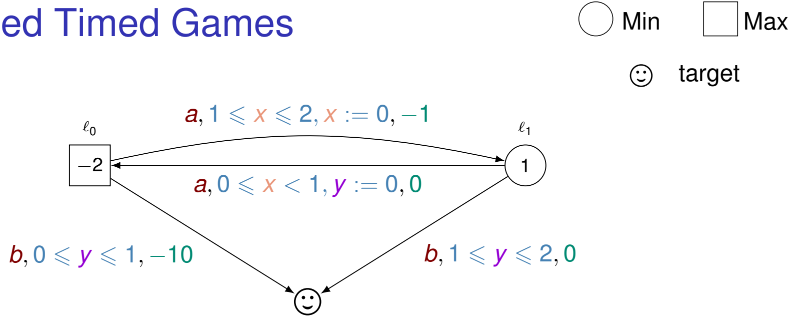
Play ρ $\quad (\ell_1, \begin{bmatrix} 0 \\ 0 \end{bmatrix})$

Weighted Timed Games



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

Weighted Timed Games

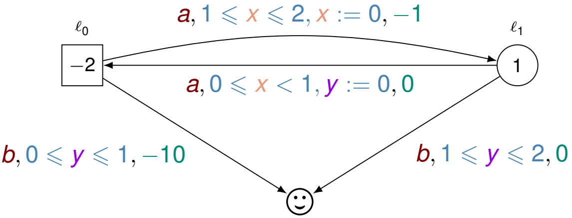


Play ρ $\quad (\ell_1, \begin{bmatrix} 0 \\ 0 \end{bmatrix}) \xrightarrow{0.5,\ a} (\ell_0, \begin{bmatrix} 0.5 \\ 0 \end{bmatrix})$

Weighted Timed Games

◯ Min ☐ Max

☺ target



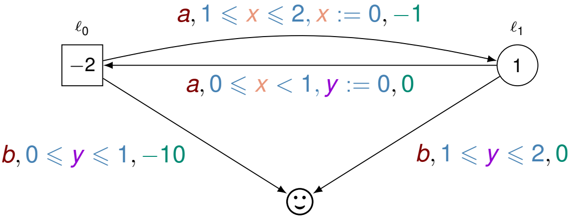
Play ρ

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

Weighted Timed Games

○ Min □ Max

☺ target



Play ρ

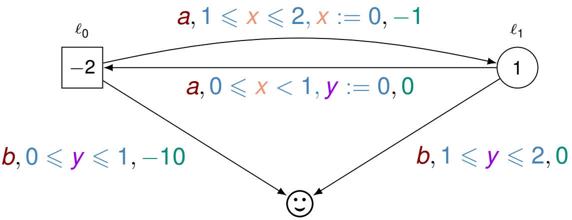
$$(\ell_1, \begin{bmatrix}0\\0\end{bmatrix}) \xrightarrow{0.5,\ a} (\ell_0, \begin{bmatrix}0.5\\0\end{bmatrix}) \xrightarrow{1.25,\ a} (\ell_1, \begin{bmatrix}0\\1.25\end{bmatrix}) \xrightarrow{1/3,\ b} (☺, \begin{bmatrix}1/3\\19/12\end{bmatrix})$$

$+ \qquad +$

Weighted Timed Games

○ Min □ Max

☺ target



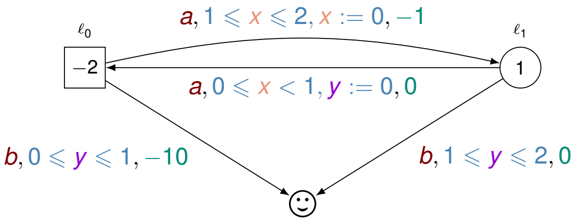
Play ρ $\quad (\ell_1, \begin{bmatrix} 0 \\ 0 \end{bmatrix}) \xrightarrow{0.5,\ a} (\ell_0, \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}) \xrightarrow{1.25,\ a} (\ell_1, \begin{bmatrix} 0 \\ 1.25 \end{bmatrix}) \xrightarrow{1/3,\ b} (☺, \begin{bmatrix} 1/3 \\ 19/12 \end{bmatrix})$

$0 \quad + \quad +$

Weighted Timed Games

Min Max

target

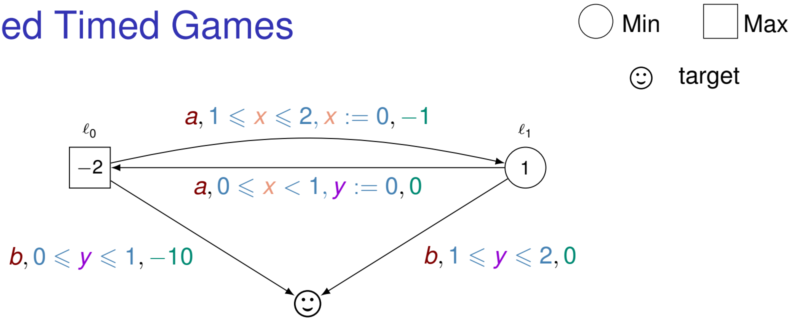


Play ρ

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

$$1 \times 0.5 + 0 \quad + \quad +$$

Weighted Timed Games

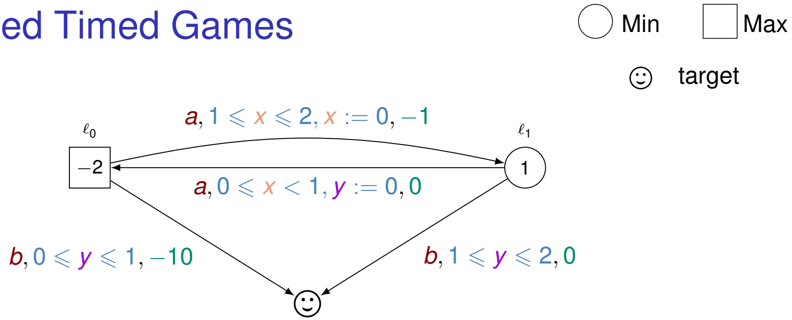


Play ρ

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

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

Weighted Timed Games



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

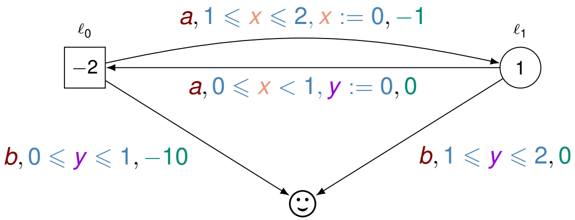
Deterministic strategy

Choose an edge and a delay

Weighted Timed Games

○ Min □ Max

☺ target



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 $t = \frac{1}{3}$

How Min wins in a weighted timed game?

σ Min

τ Max

How Min wins in a weighted timed game?

σ Min

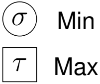
τ Max

Value

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

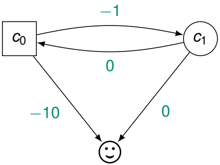
Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

How Min wins in a weighted timed game?



Value

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



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

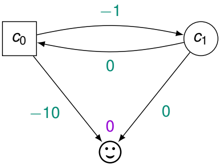
How Min wins in a weighted timed game?

σ Min

τ Max

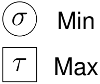
Value

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



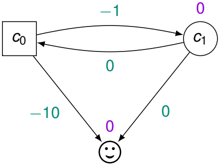
Pseudopolynomial iterative algorithm to solve total-payoff games and min-cost reachability games, T. Brihaye, G. Geeraerts, A. Haddad and B. Monmege, 2016, Acta Informatica

How Min wins in a weighted timed game?



Value

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



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

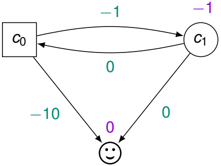
How Min wins in a weighted timed game?

σ Min

τ Max

Value

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



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

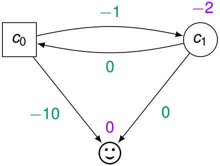
How Min wins in a weighted timed game?

σ Min

τ Max

Value

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



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

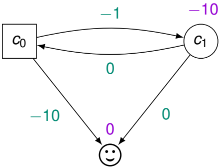
How Min wins in a weighted timed game?

σ Min

τ Max

Value

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



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

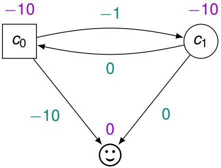
How Min wins in a weighted timed game?

σ Min

τ Max

Value

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



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

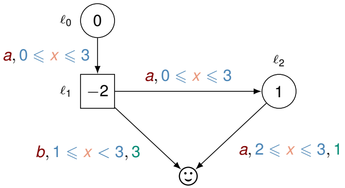
How Min wins in a weighted timed game?

σ Min

τ Max

Value

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



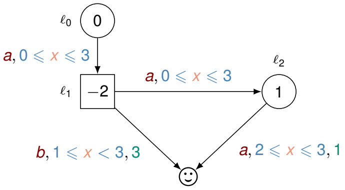
How Min wins in a weighted timed game?

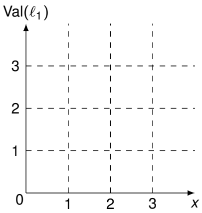
σ Min

τ Max

Value

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





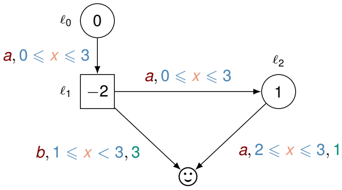
How Min wins in a weighted timed game?

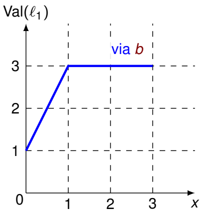
σ Min

τ Max

Value

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





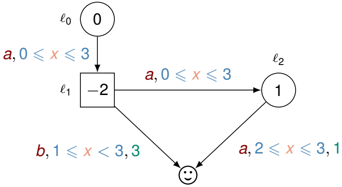
How Min wins in a weighted timed game?

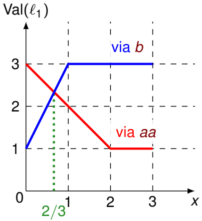
σ Min

τ Max

Value

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





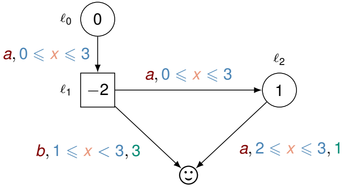
How Min wins in a weighted timed game?

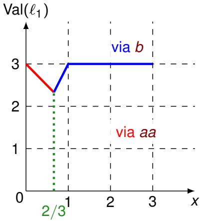
σ Min

τ Max

Value

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





How Min wins in a weighted timed game?

σ Min

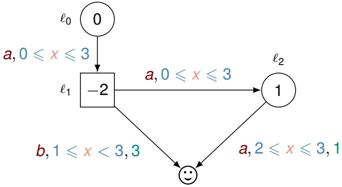
τ Max

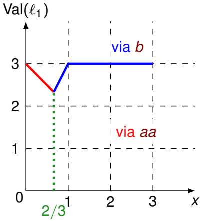
Value

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

Optimal strategy for Min

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





How Min wins in a weighted timed game?

σ Min

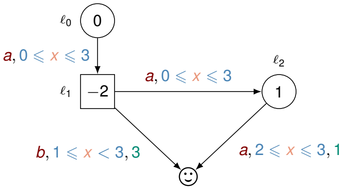
τ Max

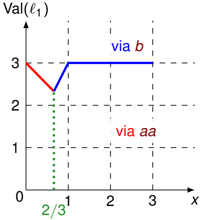
Value

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

Optimal strategy for Min

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





Infinite precision

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

How Min wins in a weighted timed game?

σ Min

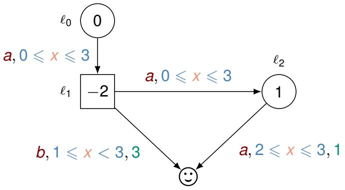
τ Max

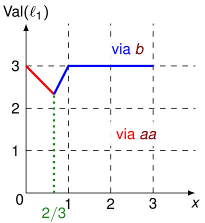
Value

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

Optimal strategy for Min

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





Infinite precision

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

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

How Min wins in a weighted timed game?

σ Min

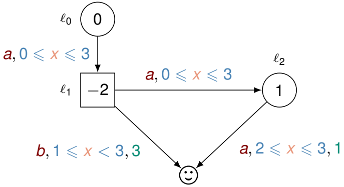
τ Max

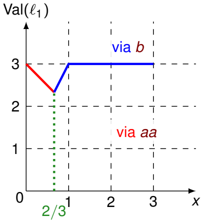
Value

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

Optimal strategy for Min

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





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

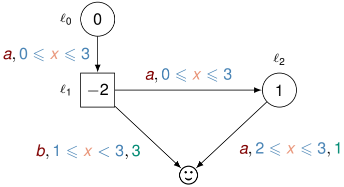
How Min wins in a weighted timed game?

σ Min

τ Max

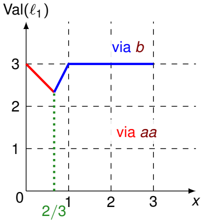
Value

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



Optimal strategy for Min

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



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

Can Min play without the infinite precision on its strategies?

Robustness in weighted timed games

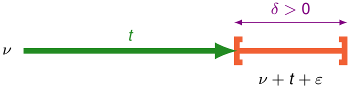
$$\nu \xrightarrow{t} \nu + t$$

Robustness in weighted timed games

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

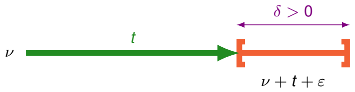
Robustness in weighted timed games

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

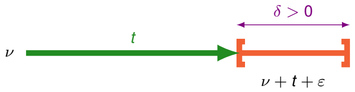


Fixed-δ semantics

Check the guard after the perturbation:

Robustness in weighted timed games

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

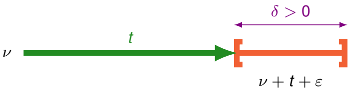


Fixed-δ semantics

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

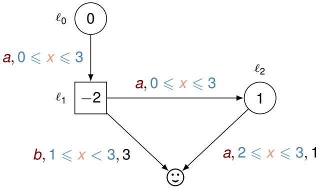
Robustness in weighted timed games

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

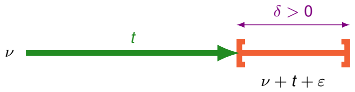


Fixed-δ play when $\delta = \frac{1}{3}$

$(\ell_0, 0)$

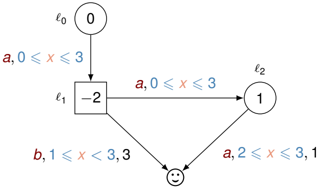
Robustness in weighted timed games

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

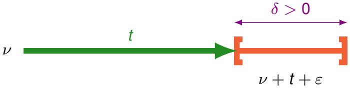


Fixed-δ play when $\delta = \frac{1}{3}$

$$(\ell_0, 0) \xrightarrow{2/3,\ a}$$

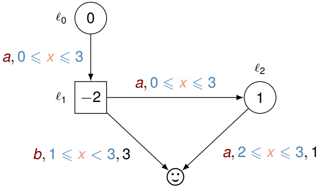
Robustness in weighted timed games

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

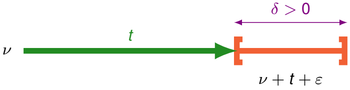


Fixed-δ play when $\delta = \frac{1}{3}$

$$(\ell_0, 0) \xrightarrow{2/3,\ a} \overset{1/3}{\rightsquigarrow} (\ell_1, 1)$$

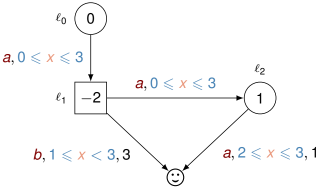
Robustness in weighted timed games

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

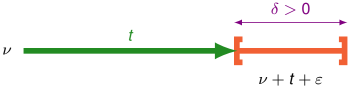


Fixed-δ play when $\delta = \frac{1}{3}$

$$(\ell_0, 0) \xrightarrow{2/3,\ a} \overset{1/3}{\rightsquigarrow} (\ell_1, 1) \xrightarrow{0,\ b} (☺, 1)$$

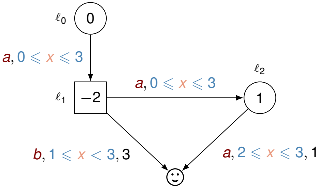
Robustness in weighted timed games

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



Fixed-δ play when $\delta = \frac{1}{3}$

$$(\ell_0, 0) \xrightarrow{2/3,\ a} \overset{1/3}{\rightsquigarrow} (\ell_1, 1) \xrightarrow{0,\ b} (\text{☺}, 1)$$

for a weight of 3

Robust values problem

Value problem: deciding if a value is at most equal to λ?

	WTG			
Val	undecidable			

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

Robust values problem

Value problem: deciding if a value is at most equal to λ?

	WTG	acyclic	divergent	1-clock
Val	undecidable	EXPTIME	3-EXPTIME	2-EXPTIME

Divergence

All SCCs contain only cycles with a weight $\leqslant -1$ or $\geqslant 1$.

A Theory of Timed Automata, R. Alur and D. Dill, 1994, Theoretical Computer Science

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

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

Decidability of One-Clock WTG with Arbitrary Weights, B. Monmege, J. Parreaux, and P.-A. Reynier, 2022, CONCUR

Robust values problem

Problem 1: δ is fixed and known

Value problem: deciding if a value is at most equal to λ?

	WTG	acyclic	divergent	1-clock
Val	undecidable	EXPTIME	3-EXPTIME	2-EXPTIME

Divergence

All SCCs contain only cycles with a weight $\leqslant -1$ or $\geqslant 1$.

Robust values problem

χ Min ζ Max

Problem 1: δ is fixed and known

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

Value problem: deciding if a value is at most equal to λ?

	WTG	acyclic	divergent	1-clock
Val	undecidable	EXPTIME	3-EXPTIME	2-EXPTIME

Divergence

All SCCs contain only cycles with a weight $\leqslant -1$ or $\geqslant 1$.

Robust values problem

χ Min ζ Max

Problem 1: δ is fixed and known

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

Value problem: deciding if a value is at most equal to λ?

	WTG	acyclic	divergent	1-clock
Val	undecidable	EXPTIME	3-EXPTIME	2-EXPTIME
rVal^{δ}	undecidable			decidable (in $\mathbb{N}$)

Divergence

All SCCs contain only cycles with a weight $\leqslant -1$ or $\geqslant 1$.

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

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

Robust values problem

χ Min ζ Max

Problem 1: δ is fixed and known

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

Value problem: deciding if a value is at most equal to λ?

	WTG	acyclic	divergent	1-clock
Val	undecidable	EXPTIME	3-EXPTIME	2-EXPTIME
rVal^{δ}	undecidable	decidable	decidable	decidable (in $\mathbb{N}$)

Divergence

All SCCs contain only cycles with a weight $\leqslant -1$ or $\geqslant 1$.

Robust values problem

χ Min ζ Max

Problem 1: δ is fixed and known

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

Value problem: deciding if a value is at most equal to λ?

	WTG	acyclic	divergent	1-clock
Val	undecidable	EXPTIME	3-EXPTIME	2-EXPTIME
rVal$^{\delta}$	undecidable	decidable	decidable	decidable (in $\mathbb{N}$)

Divergence

All SCCs contain only cycles with a weight $\leqslant -1$ or $\geqslant 1$.

💡 Encoding fixed-δ semantics into exact one

Robust values problem

χ Min ζ Max

Problem 1: δ is fixed and known

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

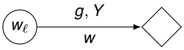
Value problem: deciding if a value is at most equal to λ?

	WTG	acyclic	divergent	1-clock
Val	undecidable	EXPTIME	3-EXPTIME	2-EXPTIME
rVal^{δ}	undecidable	decidable	decidable	decidable (in $\mathbb{N}$)

Divergence

All SCCs contain only cycles with a weight $\leqslant -1$ or $\geqslant 1$.

Encoding fixed-δ semantics into exact one



Robust values problem

χ Min $\quad$ ζ Max

Problem 1: δ is fixed and known

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

Value problem: deciding if a value is at most equal to λ?

	WTG	acyclic	divergent	1-clock
Val	undecidable	EXPTIME	3-EXPTIME	2-EXPTIME
rVal^{δ}	undecidable	decidable	decidable	decidable (in $\mathbb{N}$)

Divergence

All SCCs contain only cycles with a weight $\leqslant -1$ or $\geqslant 1$.

Encoding fixed-δ semantics into exact one

Need a new clock

Robust values problem

χ Min $\quad$ ζ Max

Problem 1: δ is fixed and known

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

Problem 2: δ tends to 0

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

Value problem: deciding if a value is at most equal to λ?

	WTG	acyclic	divergent	1-clock
Val	undecidable	EXPTIME	3-EXPTIME	2-EXPTIME
rVal^{δ}	undecidable	decidable	decidable	decidable (in $\mathbb{N}$)

Divergence

All SCCs contain only cycles with a weight $\leqslant -1$ or $\geqslant 1$.

Encoding fixed-δ semantics into exact one

Need a new clock

Robust values problem

χ Min ζ Max

Problem 1: δ is fixed and known

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

Problem 2: δ tends to 0

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

Value problem: deciding if a value is at most equal to λ?

	WTG	acyclic	divergent	1-clock
Val	undecidable	EXPTIME	3-EXPTIME	2-EXPTIME
rVal^{δ}	undecidable	decidable	decidable	decidable (in $\mathbb{N}$)
rVal	undecidable			

Divergence

All SCCs contain only cycles with a weight $\leqslant -1$ or $\geqslant 1$.

Encoding fixed-δ semantics into exact one

Need a new clock

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

Robust values problem

χ Min ζ Max

Problem 1: δ is fixed and known

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

Problem 2: δ tends to 0

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

Value problem: deciding if a value is at most equal to λ?

	WTG	acyclic	divergent	1-clock
Val	undecidable	EXPTIME	3-EXPTIME	2-EXPTIME
rVal^{δ}	undecidable	decidable	decidable	decidable (in $\mathbb{N}$)
rVal	undecidable	decidable		

Divergence

All SCCs contain only cycles with a weight $\leqslant -1$ or $\geqslant 1$.

Encoding fixed-δ semantics into exact one

Need a new clock

Robust values problem

χ Min ζ Max

Problem 1: δ is fixed and known

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

Problem 2: δ tends to 0

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

Value problem: deciding if a value is at most equal to λ?

	WTG	acyclic	divergent	1-clock
Val	undecidable	EXPTIME	3-EXPTIME	2-EXPTIME
rVal^{δ}	undecidable	decidable	decidable	decidable (in $\mathbb{N}$)
rVal	undecidable	decidable	decidable	

Divergence

All SCCs contain only cycles with a weight $\leqslant -1$ or $\geqslant 1$.

Encoding fixed-δ semantics into exact one

Need a new clock

The robust value when δ tends to 0

Fixpoint symbolic computation

The robust value when δ tends to 0

Fixpoint symbolic computation

A combination of two existing methods

The robust value when δ tends to 0

Fixpoint symbolic computation

A combination of two existing methods

Cells

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

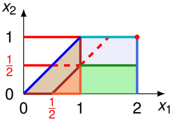
The robust value when δ tends to 0

Fixpoint symbolic computation

A combination of two existing methods

Cells

Affine equations:
$y = \sum_i a_i\, x_i + b$



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

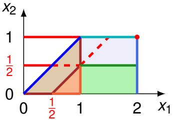
The robust value when δ tends to 0

Fixpoint symbolic computation

A combination of two existing methods

Cells

Affine equations:
$y = \sum_i a_i x_i + b$



Shrunk DBM

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

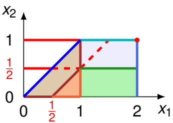
The robust value when δ tends to 0

Fixpoint symbolic computation

A combination of two existing methods

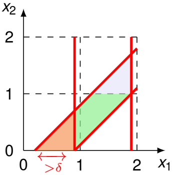
Cells

Affine equations:
$y = \sum_i a_i\, x_i + b$



Shrunk DBM

Matrix: $M - \delta\, P$
where $\delta \to 0$



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

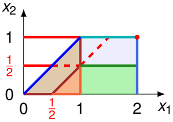
The robust value when δ tends to 0

Fixpoint symbolic computation

A combination of two existing methods

Cells

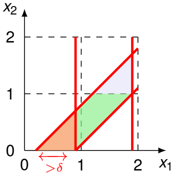
Affine equations:
$y = \sum_i a_i x_i + b$



Shrunk cells

Shrunk DBM

Matrix: $M - \delta P$
where $\delta \to 0$



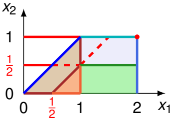
The robust value when δ tends to 0

Fixpoint symbolic computation

A combination of two existing methods

Cells

Affine equations:
$y = \sum_i a_i x_i + b$

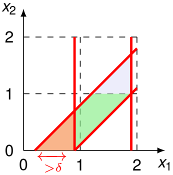


Shrunk cells

Affine equations:
$y = \sum_i a_i x_i + b$

Shrunk DBM

Matrix: $M - \delta P$
where $\delta \to 0$



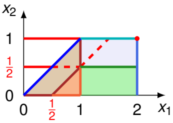
The robust value when δ tends to 0

Fixpoint symbolic computation

A combination of two existing methods

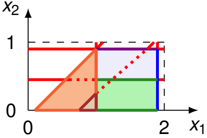
Cells

Affine equations:
$y = \sum_i a_i x_i + b$



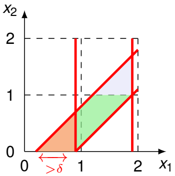
Shrunk cells

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



Shrunk DBM

Matrix: $M - \delta P$
where $\delta \to 0$



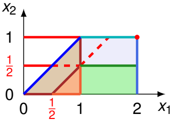
The robust value when δ tends to 0

Fixpoint symbolic computation

A combination of two existing methods

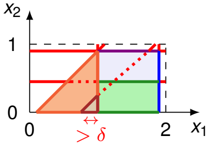
Cells

Affine equations:
$y = \sum_i a_i\, x_i + b$



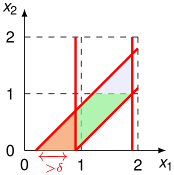
Shrunk cells

Affine equations:
$y = \sum_i a_i\, x_i + b + c\,\delta$
where $\delta \to 0$



Shrunk DBM

Matrix: $M - \delta\, P$
where $\delta \to 0$



Conclusion

	WTG	acyclic	divergent
rVal^{δ}	undecidable	decidable	decidable
rVal	undecidable	decidable	decidable

Conclusion

	WTG	acyclic	divergent
rVal^{δ}	undecidable	decidable	decidable
rVal	undecidable	decidable	decidable

Decidability algorithms' principle

Conclusion

	WTG	acyclic	divergent
rVal^{δ}	undecidable	decidable	decidable
rVal	undecidable	decidable	decidable

Decidability algorithms' principle

- Encoding fixed-δ semantics into already known semantics (like the exact one)

Conclusion

	WTG	acyclic	divergent
rVal^{δ}	undecidable	decidable	decidable
rVal	undecidable	decidable	decidable

Decidability algorithms' principle

- Encoding fixed-δ semantics into already known semantics (like the exact one)
- Adaptation of cells and shrunk DBM

Conclusion

	WTG	acyclic	divergent
rVal^{δ}	undecidable	decidable	decidable
rVal	undecidable	decidable	decidable

Decidability algorithms' principle

- Encoding fixed-δ semantics into already known semantics (like the exact one)
- Adaptation of cells and shrunk DBM

Perspectives

Conclusion

	WTG	acyclic	divergent
rVal^δ	undecidable	decidable	decidable
rVal	undecidable	decidable	decidable

Decidability algorithms' principle

- Encoding fixed-δ semantics into already known semantics (like the exact one)
- Adaptation of cells and shrunk DBM

Perspectives

- Others classes of weighted timed games like the 1-clock games

Conclusion

	WTG	acyclic	divergent
rVal^{δ}	undecidable	decidable	decidable
rVal	undecidable	decidable	decidable

Decidability algorithms' principle

- Encoding fixed-δ semantics into already known semantics (like the exact one)
- Adaptation of cells and shrunk DBM

Perspectives

- Others classes of weighted timed games like the 1-clock games
- Robustness with probabilistic strategies

Conclusion

	WTG	acyclic	divergent
rVal^{δ}	undecidable	decidable	decidable
rVal	undecidable	decidable	decidable

Decidability algorithms' principle

- Encoding fixed-δ semantics into already known semantics (like the exact one)
- Adaptation of cells and shrunk DBM

Perspectives

- Others classes of weighted timed games like the 1-clock games
- Robustness with probabilistic strategies

Thank you! Questions?