Game Theory for Real-Time Synthesis: Decision, Approximation, and Randomness

Benjamin Monmege

Aix-Marseille Université

Habilitation à diriger des recherches, 29 avril 2022

Eugène Asarin	*Université de Paris, France (Examinateur)*
Béatrice Bérard	*Sorbonne Université, France (Présidente)*
Véronique Bruyère	*Université de Mons, Belgique (Rapporteuse)*
Marcin Jurdziński	*University of Warwick, UK (Rapporteur)*
Nicolas Markey	*CNRS, Irisa, France (Rapporteur)*
Pierre-Alain Reynier	*Aix-Marseille Université, France (Examinateur)*
Yann Vaxès	*Aix-Marseille Université, France (Examinateur)*

Formal methods for reliable critical software

Code & model-checking

Formal methods for reliable critical software

Game theory for synthesis

Code & model-checking

Controller player vs. environment player

Formal methods for reliable critical software

Code & model-checking

Game theory for synthesis

Controller player vs. environment player

Time constraints

Formal methods for reliable critical software

Code & model-checking

Game theory for synthesis

Controller player vs. environment player

Time constraints

Measure quality

Methodology

$\boxed{\text{Environment}} \quad \| \quad \boxed{\text{Controller??}} \quad \models \quad \text{Specif}$

Methodology

$\boxed{\text{Environment}} \quad \| \quad \boxed{\text{Controller??}} \quad \models \quad \text{Specif}$

Real-time requirements/environment $\implies$ real-time controller

Methodology

$\boxed{\text{Environment}} \quad \| \quad \boxed{\text{Controller??}} \quad \models \quad \text{Specif}$

Real-time requirements/environment $\Longrightarrow$ real-time controller

Among all *valid* controllers, choose a *cheap/efficient* one

Methodology

$\boxed{\text{Environment}} \quad \| \quad \boxed{\text{Controller??}} \quad \models \quad \text{Specif}$

Two-player game

Real-time requirements/environment $\Longrightarrow$ real-time controller

Among all *valid* controllers, choose a *cheap/efficient* one

Methodology

$\boxed{\text{Environment}} \quad \| \quad \boxed{\text{Controller??}} \quad \models \quad \text{Specif}$

Two-player game

Real-time requirements/environment $\Longrightarrow$ real-time controller

Two-player **timed** game

Among all *valid* controllers, choose a *cheap/efficient* one

Methodology

$\boxed{\text{Environment}} \quad \| \quad \boxed{\text{Controller??}} \quad \models \quad \text{Specif}$

Two-player game

Real-time requirements/environment $\Longrightarrow$ real-time controller

Two-player **timed** game

Among all *valid* controllers, choose a *cheap/efficient* one

Two-player **weighted** timed game

Methodology

$\boxed{\text{Environment}} \quad \| \quad \boxed{\text{Controller??}} \quad \models \quad \text{Specif}$

Two-player game

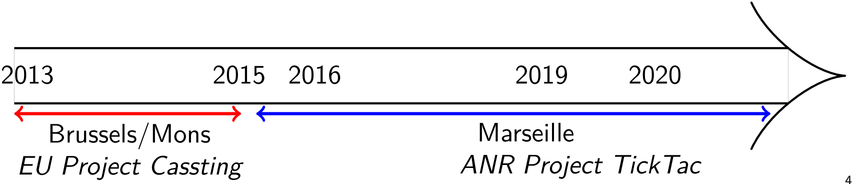
Real-time requirements/environment $\Longrightarrow$ real-time controller

Two-player **timed** game

Among all *valid* controllers, choose a *cheap/efficient* one

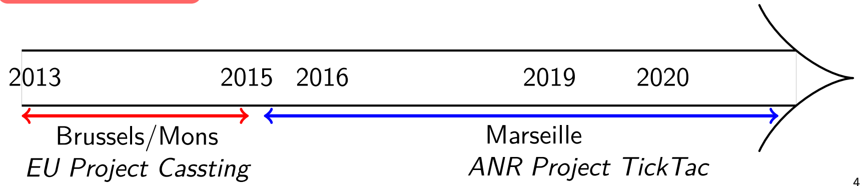
Two-player **weighted** timed game

Production/consumption of resources: **negative weights**


2013
2015
2016
2019
2020
Brussels/Mons
EU Project Cassting
Marseille
ANR Project TickTac

Timed games & $\leqslant 0$ weights

T. Brihaye
G. Geeraerts
S. K. Narayanan
L. Manasa
A. Trivedi



Untimed & total payoff

T. Brihaye
G. Geeraerts
A. Haddad

Timed games & $\leqslant 0$ weights

T. Brihaye
G. Geeraerts
S. K. Narayanan
L. Manasa
A. Trivedi

2013 2015 2016 2019 2020

Brussels/Mons
EU Project Cassting

Marseille
ANR Project TickTac

1 clock

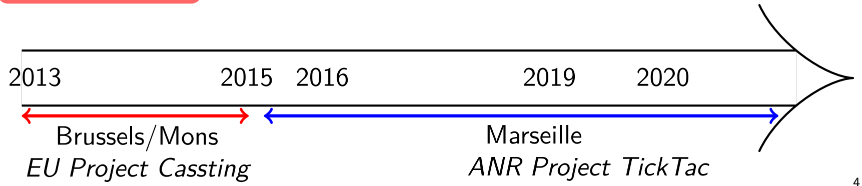
T. Brihaye
G. Geeraerts
A. Haddad
E. Lefaucheux

Untimed & total payoff

T. Brihaye
G. Geeraerts
A. Haddad

Timed games & $\leqslant 0$ weights

T. Brihaye
G. Geeraerts
S. K. Narayanan
L. Manasa
A. Trivedi



1 clock

T. Brihaye
G. Geeraerts
A. Haddad
E. Lefaucheux

Untimed & total payoff

T. Brihaye
G. Geeraerts
A. Haddad

Timed games & $\leqslant 0$ weights

T. Brihaye
G. Geeraerts
S. K. Narayanan
L. Manasa
A. Trivedi

Divergence, approximation, robustness

D. Busatto-Gaston (PhD)
P.-A. Reynier
O. Sankur

2013 2015 2016 2019 2020

Brussels/Mons
EU Project Cassting

Marseille
ANR Project TickTac

1 clock

T. Brihaye
G. Geeraerts
A. Haddad
E. Lefaucheux

Untimed & total payoff

T. Brihaye
G. Geeraerts
A. Haddad

Timed games & $\leqslant 0$ weights

T. Brihaye
G. Geeraerts
S. K. Narayanan
L. Manasa
A. Trivedi

Randomisation

J. Parreaux (PhD)
P.-A. Reynier

Divergence, approximation, robustness

D. Busatto-Gaston (PhD)
P.-A. Reynier
O. Sankur

2013 2015 2016 2019 2020

Brussels/Mons
EU Project Cassting

Marseille
ANR Project TickTac

1 clock

T. Brihaye
G. Geeraerts
A. Haddad
E. Lefaucheux

MITL

T. Brihaye
M. Estiévenart
G. Geeraerts
H.-M. Ho
A. Milchior
N. Sznajder

Untimed & total payoff

T. Brihaye
G. Geeraerts
A. Haddad

Randomisation

J. Parreaux (PhD)
P.-A. Reynier

Timed games & $\leqslant 0$ weights

T. Brihaye
G. Geeraerts
S. K. Narayanan
L. Manasa
A. Trivedi

Divergence, approximation, robustness

D. Busatto-Gaston (PhD)
P.-A. Reynier
O. Sankur

2013 2015 2016 2019 2020

Brussels/Mons
EU Project Cassting

Marseille
ANR Project TickTac

1 clock

T. Brihaye
G. Geeraerts
A. Haddad
E. Lefaucheux

Untimed & total payoff

T. Brihaye
G. Geeraerts
A. Haddad

Timed games & $\leqslant 0$ weights

T. Brihaye
G. Geeraerts
S. K. Narayanan
L. Manasa
A. Trivedi

MITL

T. Brihaye
M. Estiévenart
G. Geeraerts
H.-M. Ho
A. Milchior
N. Sznajder

Evolutionary Games

T. Brihaye
G. Geeraerts
M. Hallet
B. Quoitin

Randomisation

J. Parreaux (PhD)
P.-A. Reynier

Divergence, approximation, robustness

D. Busatto-Gaston (PhD)
P.-A. Reynier
O. Sankur

2013 2015 2016 2019 2020

Brussels/Mons
EU Project Cassting

Marseille
ANR Project TickTac

1 clock

T. Brihaye
G. Geeraerts
A. Haddad
E. Lefaucheux

Untimed & total payoff

T. Brihaye
G. Geeraerts
A. Haddad

Timed games & $\leqslant 0$ weights

T. Brihaye
G. Geeraerts
S. K. Narayanan
L. Manasa
A. Trivedi

MITL

T. Brihaye
M. Estiévenart
G. Geeraerts
H.-M. Ho
A. Milchior
N. Sznajder

Evolutionary Games

T. Brihaye
G. Geeraerts
M. Hallet
B. Quoitin

Divergence, approximation, robustness

D. Busatto-Gaston (PhD)
P.-A. Reynier
O. Sankur

Transducers & WA
ANR Project Delta

N. Baudru
L.-M. Dando
N. Lhote
T. Lopez (PhD)
P.-A. Reynier
J.-M. Talbot

Randomisation

J. Parreaux (PhD)
P.-A. Reynier

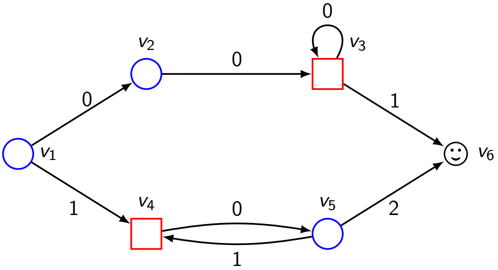
2013 2015 2016 2019 2020

Brussels/Mons
EU Project Cassting

Marseille
ANR Project TickTac

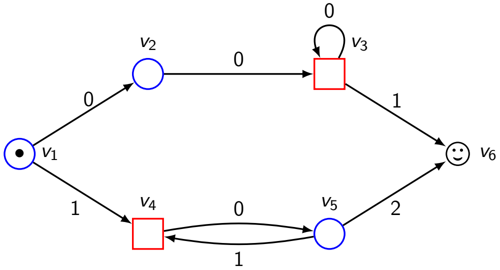
Part I : Weighted games

Weighted games



Weighted graph with
vertices partitioned between
2 players
+ reachability objective

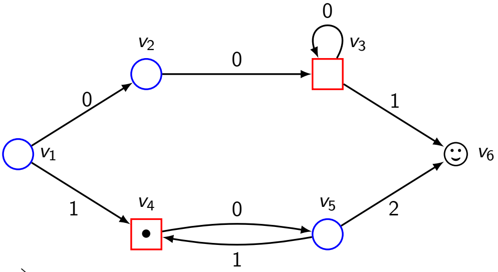
Weighted games



Weighted graph with vertices partitioned between 2 players + reachability objective

v_1

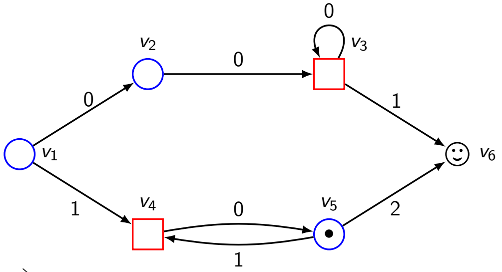
Weighted games



$v_1 \xrightarrow{\searrow} v_4$

Weighted graph with vertices partitioned between 2 players
+ reachability objective

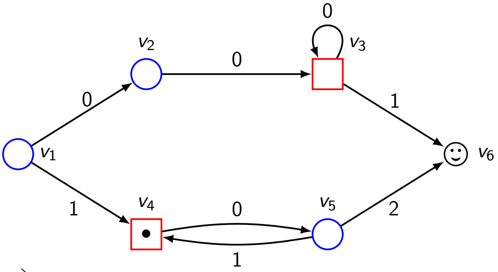
Weighted games



Weighted graph with
vertices partitioned between
2 players
+ reachability objective

$v_1 \xrightarrow{\searrow} v_4 \xrightarrow{\rightarrow} v_5$

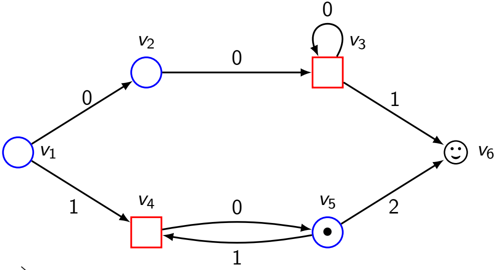
Weighted games



$v_1 \xrightarrow{\searrow} v_4 \xrightarrow{\rightarrow} v_5 \xrightarrow{\leftarrow} v_4$

Weighted graph with
vertices partitioned between
2 players
+ reachability objective

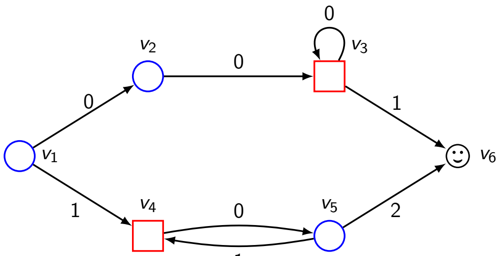
Weighted games



$v_1 \xrightarrow{\searrow} v_4 \xrightarrow{\rightarrow} v_5 \xrightarrow{\leftarrow} v_4 \xrightarrow{\rightarrow} v_5$

Weighted graph with
vertices partitioned between
2 players
+ reachability objective

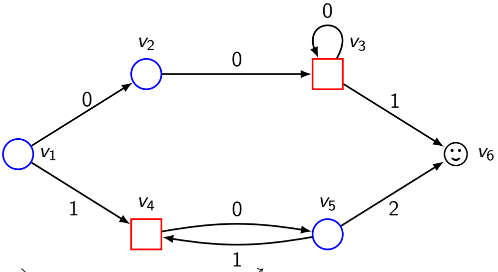
Weighted games



$$\color{blue}{v_1} \xrightarrow{\searrow} \color{red}{v_4} \xrightarrow{\rightarrow} \color{blue}{v_5} \xrightarrow{\leftarrow} \color{red}{v_4} \xrightarrow{\rightarrow} \color{blue}{v_5} \xrightarrow{\nearrow} ☺$$

Weighted graph with
vertices partitioned between
2 players
+ reachability objective

Weighted games

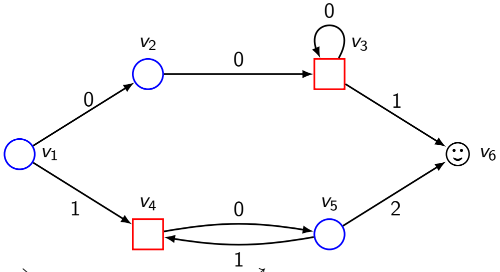


$$v_1 \xrightarrow{\searrow} v_4 \xrightarrow{\rightarrow} v_5 \xrightarrow{\leftarrow} v_4 \xrightarrow{\rightarrow} v_5 \xrightarrow{\nearrow} \smiley$$

$$1 \qquad +1 \qquad +2 \qquad = 4$$

Weighted graph with
vertices partitioned between
2 players
+ reachability objective

Weighted games



Weighted graph with vertices partitioned between 2 players + reachability objective

$$v_1 \xrightarrow{\searrow} v_4 \xrightarrow{\rightarrow} v_5 \xrightarrow{\leftarrow} v_4 \xrightarrow{\rightarrow} v_5 \xrightarrow{\nearrow} ☺$$

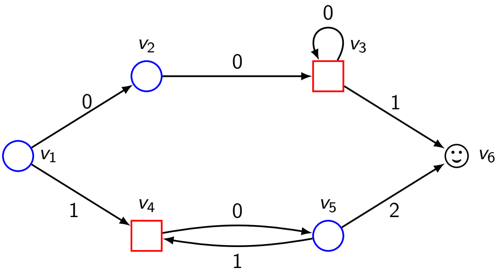
$$1 \quad +1 \quad +2 \quad = 4$$

$$v_1 \xrightarrow{\nearrow} v_2 \xrightarrow{\rightarrow} v_3 \xrightarrow{\circlearrowleft} v_3 \xrightarrow{\circlearrowleft} v_3 \quad \cdots$$

$$\cdots \quad = +\infty$$ (☺ not reached)

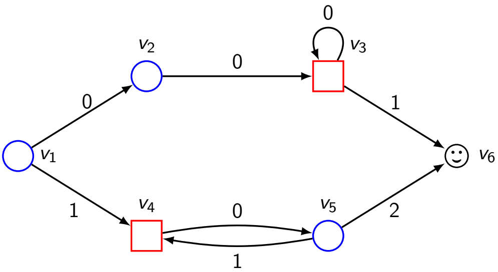
Weight of a path: $\begin{cases} +\infty & \text{if ☺ not reached} \\ \text{total weight until ☺} & \text{otherwise} \end{cases}$

Strategies and objectives



Strategy for a player: map finite executions to the transition to fire

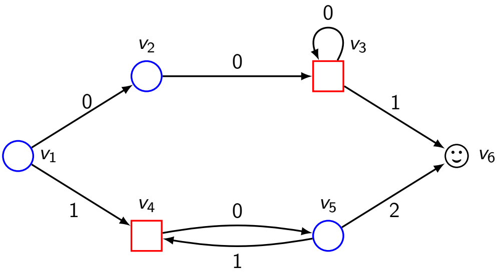
Strategies and objectives



Strategy for a player: map finite executions to the transition to fire

Objective of player ○: reach ☺ **and** minimise the weight
Objective of player □: avoid ☺ **or, if not possible,** maximise the weight

Strategies and objectives



Strategy for a player: map finite executions to the transition to fire

Objective of player ○: reach ☺ **and** minimise the weight
Objective of player □: avoid ☺ **or, if not possible,** maximise the weight

Main object of interest:
$$\mathrm{Val}(v) = \inf_{\sigma_{\mathrm{Min}} \in \mathrm{Strat}^{\mathrm{Min}}} \sup_{\sigma_{\mathrm{Max}} \in \mathrm{Strat}^{\mathrm{Max}}} \mathrm{Weight}(\mathrm{Exec}(v, \sigma_{\mathrm{Min}}, \sigma_{\mathrm{Max}})) \in \mathbb{Z} \cup \{\pm\infty\}$$
What weight can players guarantee? Following which strategies?

State of the art

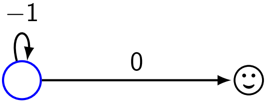
- one-player: shortest path in a weighted graph... polynomial algo.

State of the art

- one-player: shortest path in a weighted graph... polynomial algo.
- two players, $\geqslant 0$ weights: polynomial algo.
  (Khachiyan, Boros, Borys, Elbassioni, Gurvich, Rudolf, and Zhao 2008)

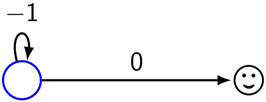
State of the art

- one-player: shortest path in a weighted graph... polynomial algo.
- two players, $\geqslant 0$ weights: polynomial algo.
  (Khachiyan, Boros, Borys, Elbassioni, Gurvich, Rudolf, and Zhao 2008)
- two players, arbitrary weights?



State of the art

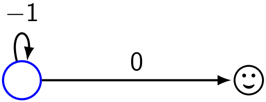
- one-player: shortest path in a weighted graph... polynomial algo.
- two players, $\geqslant 0$ weights: polynomial algo.
  (Khachiyan, Boros, Borys, Elbassioni, Gurvich, Rudolf, and Zhao 2008)
- two players, arbitrary weights?

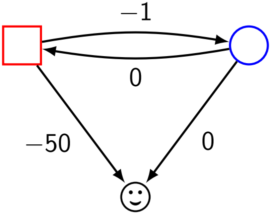


- Value $-\infty$: detection is as hard as solving parity games ($\mathbf{NP} \cap \mathbf{co\text{-}NP}$)

State of the art

- one-player: shortest path in a weighted graph... polynomial algo.
- two players, $\geqslant 0$ weights: polynomial algo.
  (Khachiyan, Boros, Borys, Elbassioni, Gurvich, Rudolf, and Zhao 2008)
- two players, arbitrary weights?

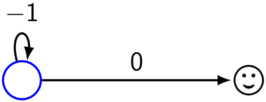


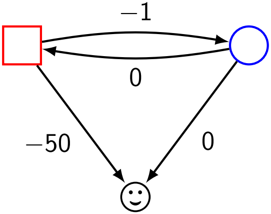


- Value $-\infty$: detection is as hard as solving parity games ($\mathbf{NP} \cap \mathbf{co\text{-}NP}$)

State of the art

- one-player: shortest path in a weighted graph... polynomial algo.
- two players, $\geqslant 0$ weights: polynomial algo.
  (Khachiyan, Boros, Borys, Elbassioni, Gurvich, Rudolf, and Zhao 2008)
- two players, arbitrary weights?





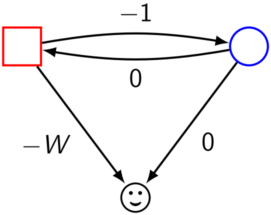
- Value $-\infty$: detection is as hard as solving parity games ($\mathbf{NP} \cap \mathbf{co\text{-}NP}$)
- ○ needs memory

Pseudo-polynomial time algorithm

Joint work with T. Brihaye, G. Geeraerts and A. Haddad

Value iteration algorithm: compute $\mathcal{F}^i(+\infty)$...

$$\mathcal{F}(\boldsymbol{x})_v = \begin{cases} \min\limits_{e=(v,a,v')\in E} \big(\mathrm{Weight}(e) + \boldsymbol{x}_{v'}\big) & \text{if } v \in V_{\mathrm{Min}} \\ \max\limits_{e=(v,a,v')\in E} \big(\mathrm{Weight}(e) + \boldsymbol{x}_{v'}\big) & \text{if } v \in V_{\mathrm{Max}} \end{cases}$$



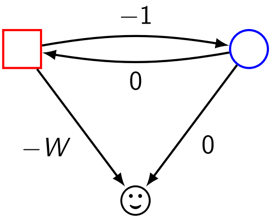
	$\square$	$\bigcirc$
horizon 0:	$+\infty$	$+\infty$

Pseudo-polynomial time algorithm

Joint work with T. Brihaye, G. Geeraerts and A. Haddad

Value iteration algorithm: compute $\mathcal{F}^i(+\infty)$...

$$\mathcal{F}(\boldsymbol{x})_v = \begin{cases} \min\limits_{e=(v,a,v')\in E} \big(\mathrm{Weight}(e) + \boldsymbol{x}_{v'}\big) & \text{if } v \in V_{\mathrm{Min}} \\ \max\limits_{e=(v,a,v')\in E} \big(\mathrm{Weight}(e) + \boldsymbol{x}_{v'}\big) & \text{if } v \in V_{\mathrm{Max}} \end{cases}$$



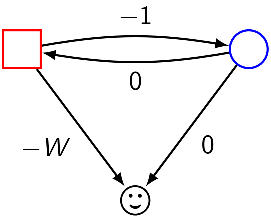
	□	○
horizon 0:	$+\infty$	$+\infty$
horizon 1:	$+\infty$	0

Pseudo-polynomial time algorithm

Joint work with T. Brihaye, G. Geeraerts and A. Haddad

Value iteration algorithm: compute $\mathcal{F}^i(+\infty)$...

$$\mathcal{F}(\boldsymbol{x})_v = \begin{cases} \min\limits_{e=(v,a,v')\in E} \big(\mathrm{Weight}(e) + \boldsymbol{x}_{v'}\big) & \text{if } v \in V_{\mathrm{Min}} \\ \max\limits_{e=(v,a,v')\in E} \big(\mathrm{Weight}(e) + \boldsymbol{x}_{v'}\big) & \text{if } v \in V_{\mathrm{Max}} \end{cases}$$



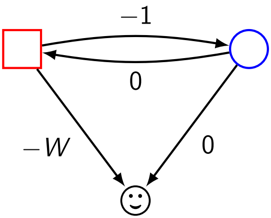
	□	○
horizon 0:	$+\infty$	$+\infty$
horizon 1:	$+\infty$	0
horizon 2:	-1	0

Pseudo-polynomial time algorithm

Joint work with T. Brihaye, G. Geeraerts and A. Haddad

Value iteration algorithm: compute $\mathcal{F}^i(+\infty)$...

$$\mathcal{F}(\boldsymbol{x})_v = \begin{cases} \min\limits_{e=(v,a,v')\in E} \big(\mathrm{Weight}(e) + \boldsymbol{x}_{v'}\big) & \text{if } v \in V_{\mathrm{Min}} \\ \max\limits_{e=(v,a,v')\in E} \big(\mathrm{Weight}(e) + \boldsymbol{x}_{v'}\big) & \text{if } v \in V_{\mathrm{Max}} \end{cases}$$



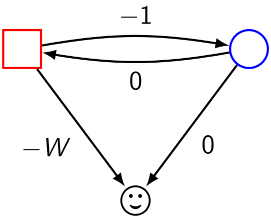
	□	○
horizon 0:	$+\infty$	$+\infty$
horizon 1:	$+\infty$	0
horizon 2:	-1	0
horizon 3:	-1	-1

Pseudo-polynomial time algorithm

Joint work with T. Brihaye, G. Geeraerts and A. Haddad

Value iteration algorithm: compute $\mathcal{F}^i(+\infty)$...

$$\mathcal{F}(\boldsymbol{x})_v = \begin{cases} \min\limits_{e=(v,a,v')\in E} \big(\mathrm{Weight}(e) + \boldsymbol{x}_{v'}\big) & \text{if } v \in V_{\mathsf{Min}} \\ \max\limits_{e=(v,a,v')\in E} \big(\mathrm{Weight}(e) + \boldsymbol{x}_{v'}\big) & \text{if } v \in V_{\mathsf{Max}} \end{cases}$$



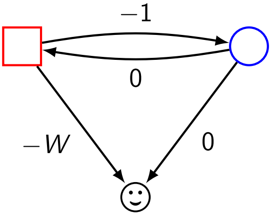
	□	○
horizon 0:	$+\infty$	$+\infty$
horizon 1:	$+\infty$	0
horizon 2:	-1	0
horizon 3:	-1	-1
horizon 4:	-2	-1

Pseudo-polynomial time algorithm

Joint work with T. Brihaye, G. Geeraerts and A. Haddad

Value iteration algorithm: compute $\mathcal{F}^i(+\infty)$...

$$\mathcal{F}(\boldsymbol{x})_v = \begin{cases} \min\limits_{e=(v,a,v')\in E} \big(\mathrm{Weight}(e) + \boldsymbol{x}_{v'}\big) & \text{if } v \in V_{\mathrm{Min}} \\ \max\limits_{e=(v,a,v')\in E} \big(\mathrm{Weight}(e) + \boldsymbol{x}_{v'}\big) & \text{if } v \in V_{\mathrm{Max}} \end{cases}$$



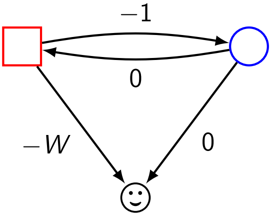
	$\square$	$\bigcirc$
horizon 0:	$+\infty$	$+\infty$
horizon 1:	$+\infty$	0
horizon 2:	-1	0
horizon 3:	-1	-1
horizon 4:	-2	-1
	$\dots$	$\dots$
horizon $2W+1$:	$-W$	$-W$

Pseudo-polynomial time algorithm

Joint work with T. Brihaye, G. Geeraerts and A. Haddad

Value iteration algorithm: compute $\mathcal{F}^i(+\infty)$...

$$\mathcal{F}(\boldsymbol{x})_v = \begin{cases} \min\limits_{e=(v,a,v')\in E} \big(\mathrm{Weight}(e) + \boldsymbol{x}_{v'}\big) & \text{if } v \in V_{\mathrm{Min}} \\ \max\limits_{e=(v,a,v')\in E} \big(\mathrm{Weight}(e) + \boldsymbol{x}_{v'}\big) & \text{if } v \in V_{\mathrm{Max}} \end{cases}$$



	□	○
horizon 0:	$+\infty$	$+\infty$
horizon 1:	$+\infty$	0
horizon 2:	-1	0
horizon 3:	-1	-1
horizon 4:	-2	-1
	...	...
horizon $2W+1$:	$-W$	$-W$
horizon $2W+2$:	$-W$	$-W$

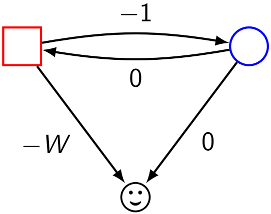
strategy of ○

Pseudo-polynomial time algorithm

Joint work with T. Brihaye, G. Geeraerts and A. Haddad

Value iteration algorithm: compute $\mathcal{F}^i(+\infty)$...

$$\mathcal{F}(\boldsymbol{x})_v = \begin{cases} \min\limits_{e=(v,a,v')\in E} \big(\mathrm{Weight}(e) + \boldsymbol{x}_{v'}\big) & \text{if } v \in V_{\mathrm{Min}} \\ \max\limits_{e=(v,a,v')\in E} \big(\mathrm{Weight}(e) + \boldsymbol{x}_{v'}\big) & \text{if } v \in V_{\mathrm{Max}} \end{cases}$$



	□	○
horizon 0:	$+\infty$	$+\infty$
horizon 1:	$+\infty$	0
horizon 2:	-1	0
horizon 3:	-1	-1
horizon 4:	-2	-1
	...	...
horizon $2W+1$:	$-W$	$-W$
horizon $2W+2$:	$-W$	$-W$

strategy of ○

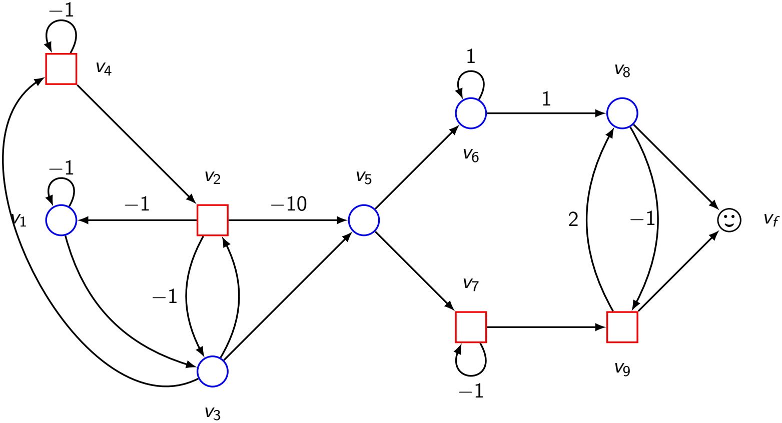
Theorem:

We can compute in pseudo-polynomial time the value of a weighted game, as well as optimal strategies: ○ may require (pseudo-polynomial) memory to play optimally, □ has optimal memoryless strategy.

Large polynomial fragment: divergent weighted games

Joint work with D. Busatto-Gaston and P.-A. Reynier

Divergence property (in the underlying graph):
Every cycle has total weight either $\leqslant -1$ **or** $\geqslant 1$

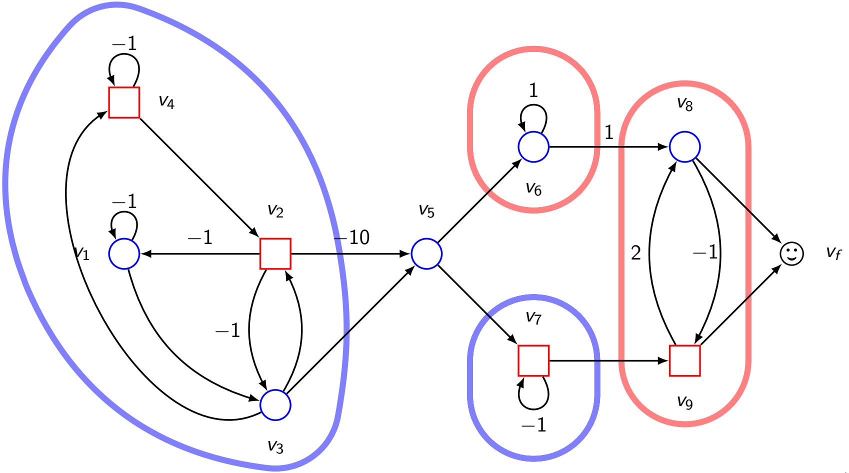


Large polynomial fragment: divergent weighted games

Joint work with D. Busatto-Gaston and P.-A. Reynier

Divergence property (in the underlying graph):
Every cycle has total weight either $\leqslant -1$ or $\geqslant 1$

Characterisation: all the simple cycles of an SCC have the same sign



Large polynomial fragment: divergent weighted games

Joint work with D. Busatto-Gaston and P.-A. Reynier

Divergence property (in the underlying graph):
Every cycle has total weight either $\leqslant -1$ or $\geqslant 1$

Characterisation: all the simple cycles of an SCC have the same sign

Theorem:

Deciding if a weighted game is divergent is in PTIME.

Large polynomial fragment: divergent weighted games

Joint work with D. Busatto-Gaston and P.-A. Reynier

Divergence property (in the underlying graph):
Every cycle has total weight either $\leqslant -1$ or $\geqslant 1$

Characterisation: all the simple cycles of an SCC have the same sign

Theorem:

Deciding if a weighted game is divergent is in PTIME.

Theorem:

We can compute in polynomial time the value of a divergent weighted game, as well as optimal strategies for both players.

Large polynomial fragment: divergent weighted games

Joint work with D. Busatto-Gaston and P.-A. Reynier

Divergence property (in the underlying graph):
Every cycle has total weight either $\leqslant -1$ or $\geqslant 1$

Characterisation: all the simple cycles of an SCC have the same sign

Theorem:

Deciding if a weighted game is divergent is in PTIME.

Theorem:

We can compute in polynomial time the value of a divergent weighted game, as well as optimal strategies for both players.

- Value computation SCC by SCC, bottom-up

Large polynomial fragment: divergent weighted games

Joint work with D. Busatto-Gaston and P.-A. Reynier

Divergence property (in the underlying graph):
Every cycle has total weight either $\leqslant -1$ **or** $\geqslant 1$

Characterisation: all the simple cycles of an SCC have the same sign

Theorem:

Deciding if a weighted game is divergent is in PTIME.

Theorem:

We can compute in polynomial time the value of a divergent weighted game, as well as optimal strategies for both players.

- Value computation SCC by SCC, bottom-up
- in **positive** SCC, the "value iteration" algo converges in linear time

Large polynomial fragment: divergent weighted games

Joint work with D. Busatto-Gaston and P.-A. Reynier

Divergence property (in the underlying graph):
Every cycle has total weight either $\leqslant -1$ or $\geqslant 1$

Characterisation: all the simple cycles of an SCC have the same sign

Theorem:

Deciding if a weighted game is divergent is in PTIME.

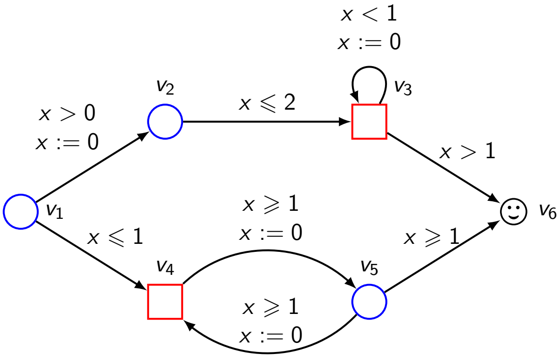
Theorem:

We can compute in polynomial time the value of a divergent weighted game, as well as optimal strategies for both players.

- Value computation SCC by SCC, bottom-up
- in **positive** SCC, the "value iteration" algo converges in linear time
- in **negative** SCC, detection of vertices of value $-\infty$ in polynomial time, and then the "value iteration" algo converges in linear time with initialisation at $-\infty$

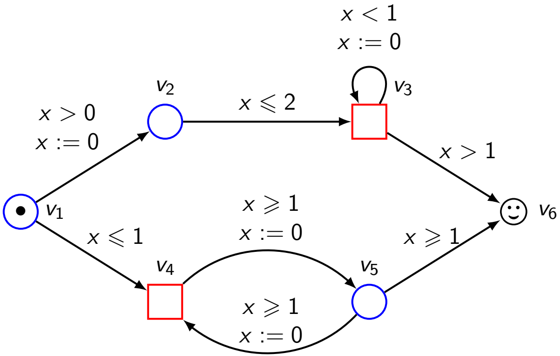
Part II : Weighted **timed** games

Weighted timed games



Timed automaton with
vertices partitioned between
2 players
$+$ reachability objective

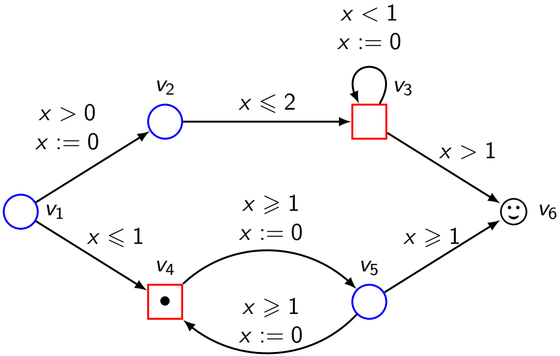
Weighted timed games



Timed automaton with vertices partitioned between 2 players
+ reachability objective

$(v_1, 0)$

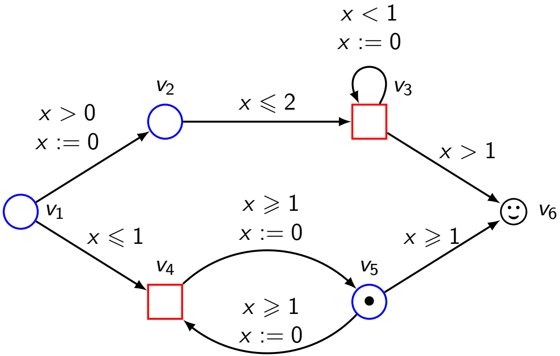
Weighted timed games



Timed automaton with vertices partitioned between 2 players + reachability objective

$$(v_1, 0) \xrightarrow{0.4, \searrow} (v_4, 0.4)$$

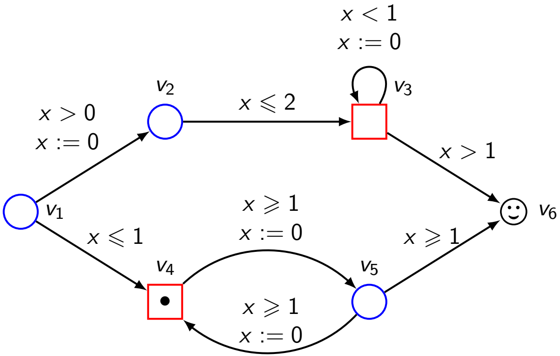
Weighted timed games



$(v_1, 0) \xrightarrow{0.4, \searrow} (v_4, 0.4) \xrightarrow{0.6, \rightarrow} (v_5, 0)$

Timed automaton with vertices partitioned between 2 players
+ reachability objective

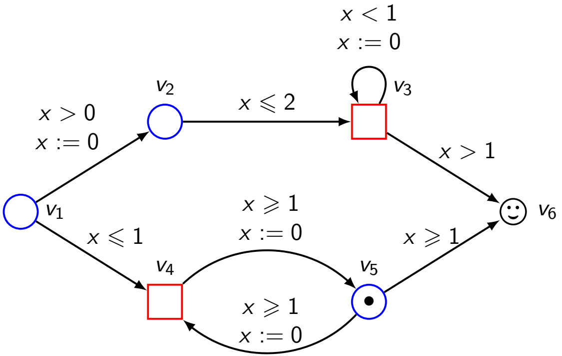
Weighted timed games



Timed automaton with vertices partitioned between 2 players + reachability objective

$$(v_1, 0) \xrightarrow{0.4, \searrow} (v_4, 0.4) \xrightarrow{0.6, \rightarrow} (v_5, 0) \xrightarrow{1.5, \leftarrow} (v_4, 0)$$

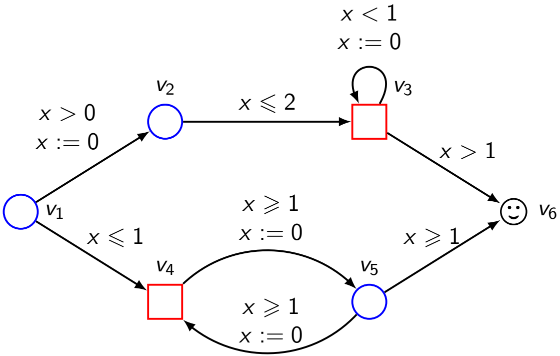
Weighted timed games



Timed automaton with vertices partitioned between 2 players + reachability objective

$$(v_1, 0) \xrightarrow{0.4, \searrow} (v_4, 0.4) \xrightarrow{0.6, \rightarrow} (v_5, 0) \xrightarrow{1.5, \leftarrow} (v_4, 0) \xrightarrow{1.1, \rightarrow} (v_5, 0)$$

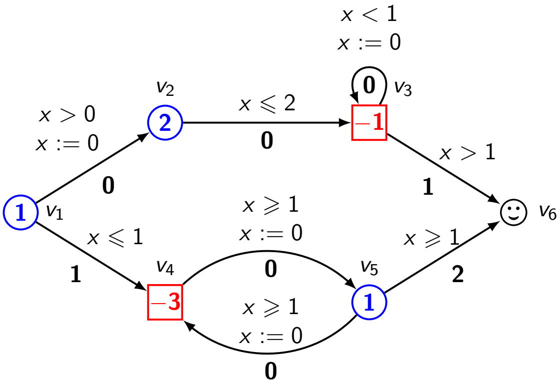
Weighted timed games



Timed automaton with vertices partitioned between 2 players + reachability objective

$$(v_1, 0) \xrightarrow{0.4, \searrow} (v_4, 0.4) \xrightarrow{0.6, \rightarrow} (v_5, 0) \xrightarrow{1.5, \leftarrow} (v_4, 0) \xrightarrow{1.1, \rightarrow} (v_5, 0) \xrightarrow{2, \nearrow} (☺, 2)$$

Weighted timed games

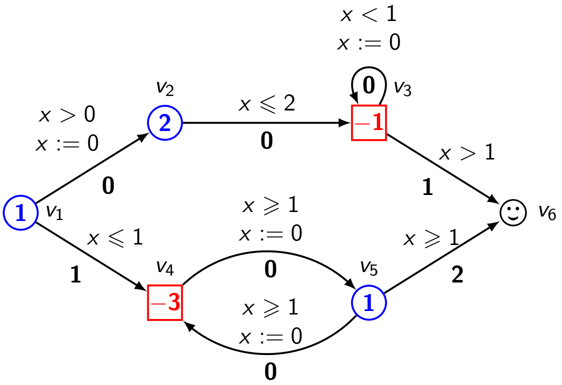


Timed automaton with vertices partitioned between 2 players
+ reachability objective
+ linear weights on vertices
+ discrete weights on transitions

$$(v_1, 0) \xrightarrow{0.4, \searrow} (v_4, 0.4) \xrightarrow{0.6, \rightarrow} (v_5, 0) \xrightarrow{1.5, \leftarrow} (v_4, 0) \xrightarrow{1.1, \rightarrow} (v_5, 0) \xrightarrow{2, \nearrow} (☺, 2)$$

$$\mathbf{1} \times 0.4 + \mathbf{1} \quad -\mathbf{3} \times 0.6 + \mathbf{0} \quad +\mathbf{1} \times 1.5 + \mathbf{0} \quad -\mathbf{3} \times 1.1 + \mathbf{0} \quad +\mathbf{1} \times 2 + \mathbf{2} \quad = 1.8$$

Weighted timed games



Timed automaton with vertices partitioned between 2 players
\+ reachability objective
\+ linear weights on vertices
\+ discrete weights on transitions

$$(v_1,0)\xrightarrow{0.4,\searrow}(v_4,0.4)\xrightarrow{0.6,\rightarrow}(v_5,0)\xrightarrow{1.5,\leftarrow}(v_4,0)\xrightarrow{1.1,\rightarrow}(v_5,0)\xrightarrow{2,\nearrow}(\smiley,2)$$

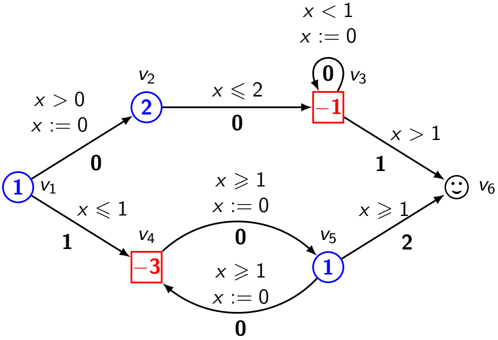
$$\mathbf{1}\times 0.4+\mathbf{1} \quad -\mathbf{3}\times 0.6+\mathbf{0} \quad +\mathbf{1}\times 1.5+\mathbf{0} \quad -\mathbf{3}\times 1.1+\mathbf{0} \quad +\mathbf{1}\times 2+\mathbf{2} \quad = 1.8$$

$$(v_1,0)\xrightarrow{0.2,\nearrow}(v_2,0)\xrightarrow{0.9,\rightarrow}(v_3,0.9)\xrightarrow{0.2,\circlearrowleft}(v_3,0)\xrightarrow{0.9,\circlearrowleft}(v_3,0)\quad\cdots$$

$$\mathbf{1}\times 0.2+\mathbf{0} \quad +\mathbf{2}\times 0.9+\mathbf{0} \quad -\mathbf{1}\times 0.2+\mathbf{0} \quad -\mathbf{1}\times 0.9+\mathbf{0} \quad \cdots \quad = +\infty$$

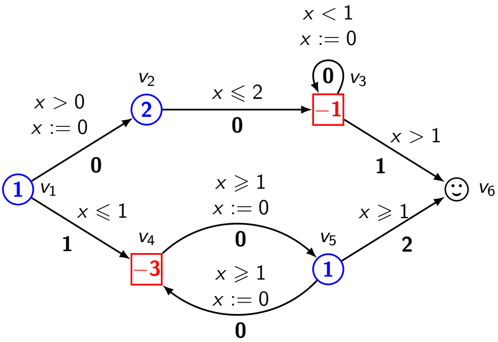
$$\text{Weight of an execution : }\begin{cases} +\infty & \text{if } \smiley \text{ not reached} \\ \text{total weight until } \smiley & \text{otherwise} \end{cases}$$

Strategies and objectives



Strategy for a player: map finite executions to a **delay** and a transition

Strategies and objectives



Strategy for a player: map finite executions to a **delay** and a transition

$$\mathsf{Val}(v,x) = \inf_{\sigma_{\mathsf{Min}} \in \mathsf{Strat}^{\mathsf{Min}}} \sup_{\sigma_{\mathsf{Max}} \in \mathsf{Strat}^{\mathsf{Max}}} \mathsf{Weight}(\mathsf{Exec}(v, x, \sigma_{\mathsf{Min}}, \sigma_{\mathsf{Max}})) \in \overline{\mathbb{R}}$$

State of the art

Decision problem: $\exists$ a strategy of $\bigcirc$ reaching ☺ with a weight $\leqslant K$?

State of the art

Decision problem: $\exists$ a strategy of ○ reaching ☺ with a weight $\leqslant K$?

- One-player case (**Weighted timed automata**): PSPACE-complete
  - Algorithm based on regions (Bouyer, Brinksma, and Larsen 2004; Bouyer, Brihaye, Bruyère, and Raskin 2007);
  - and hardness shown for timed automata with $\geqslant 2$ clocks (Fearnley and Jurdziński 2013; Haase, Ouaknine, and Worrell 2012)

State of the art

Decision problem: $\exists$ a strategy of $\bigcirc$ reaching ☺ with a weight $\leqslant K$?

- One-player case (**Weighted timed automata**): PSPACE-complete
  - Algorithm based on regions (Bouyer, Brinksma, and Larsen 2004; Bouyer, Brihaye, Bruyère, and Raskin 2007);
  - and hardness shown for timed automata with $\geqslant 2$ clocks (Fearnley and Jurdziński 2013; Haase, Ouaknine, and Worrell 2012)
- 2-player WTGs: undecidable (Brihaye, Bruyère, and Raskin 2005; Bouyer, Brihaye, and Markey 2006), even with only $\geqslant 0$ weights and 3 clocks (only 2 clocks needed with arbitrary weights (Brihaye, Geeraerts, Narayanan Krishna, Manasa, Monmege, and Trivedi 2014))

State of the art

Decision problem: $\exists$ a strategy of $\bigcirc$ reaching ☺ with a weight $\leqslant K$?

- One-player case (**Weighted timed automata**): PSPACE-complete
  - Algorithm based on regions (Bouyer, Brinksma, and Larsen 2004; Bouyer, Brihaye, Bruyère, and Raskin 2007);
  - and hardness shown for timed automata with $\geqslant 2$ clocks (Fearnley and Jurdziński 2013; Haase, Ouaknine, and Worrell 2012)
- 2-player WTGs: undecidable (Brihaye, Bruyère, and Raskin 2005; Bouyer, Brihaye, and Markey 2006), even with only $\geqslant 0$ weights and 3 clocks (only 2 clocks needed with arbitrary weights (Brihaye, Geeraerts, Narayanan Krishna, Manasa, Monmege, and Trivedi 2014))
- Decidability results for WTGs with arbitrary weights?

State of the art: one clock, $\geqslant 0$ weights

(Fearnley, Ibsen-Jensen, and Savani 2020): PSPACE-hard

(Bouyer, Larsen, Markey, and Rasmussen 2006; Rutkowski 2011; Hansen, Ibsen-Jensen, and Miltersen 2013): exponential time algo

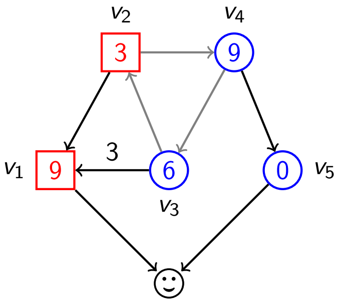
- simplification of 1-clock WTGs:
  - clock bounded by 1, no guards, no resets

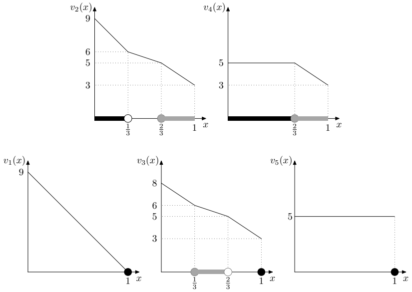
State of the art: one clock, $\geqslant 0$ weights

(Fearnley, Ibsen-Jensen, and Savani 2020): PSPACE-hard

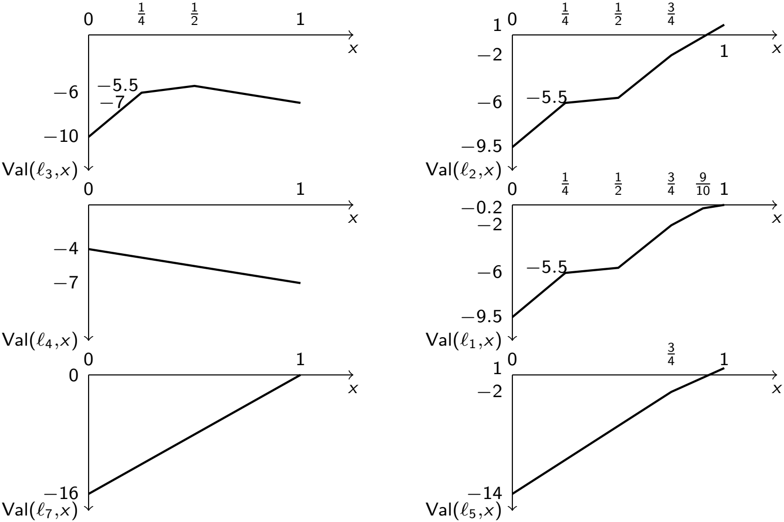
(Bouyer, Larsen, Markey, and Rasmussen 2006; Rutkowski 2011; Hansen, Ibsen-Jensen, and Miltersen 2013): exponential time algo

- simplification of 1-clock WTGs:
  - clock bounded by 1, no guards, no resets
- for simple WTGs: compute value functions $\mathrm{Val}(v, x)$.



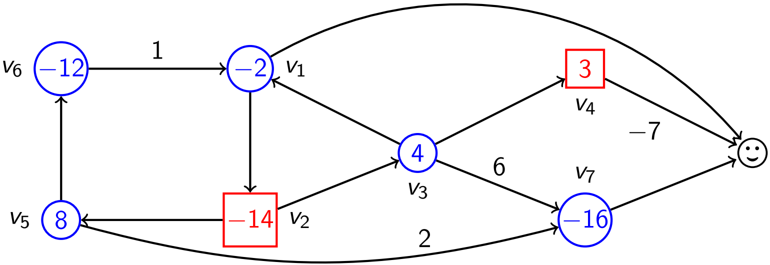


Simple WTGs with arbitrary weights



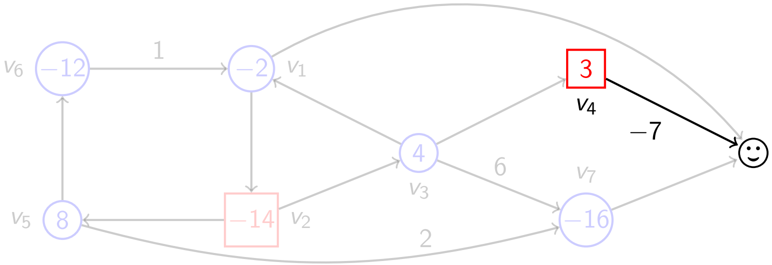
Simple WTGs with arbitrary weights

Joint work with T. Brihaye, G. Geeraerts, A. Haddad and E. Lefaucheux



Simple WTGs with arbitrary weights

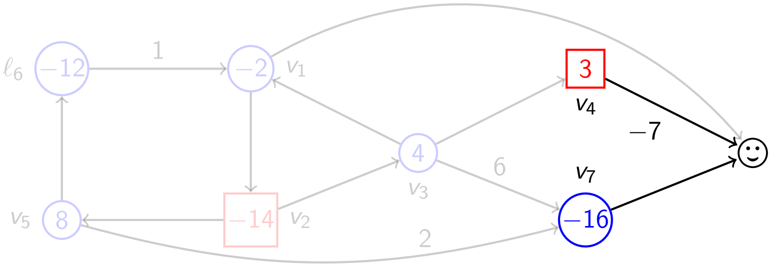
Joint work with T. Brihaye, G. Geeraerts, A. Haddad and E. Lefaucheux



$$\mathsf{Val}(v_4, x) = \sup_{0 \leqslant t \leqslant 1-x} 3t - 7 = 3(1-x) - 7 = -3x - 4$$

Simple WTGs with arbitrary weights

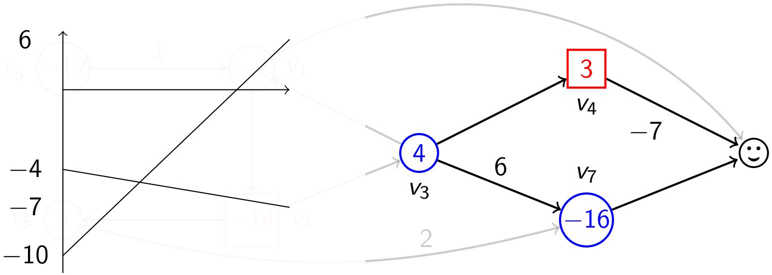
Joint work with T. Brihaye, G. Geeraerts, A. Haddad and E. Lefaucheux



$$\mathrm{Val}(v_4, x) = -3x - 4, \qquad \mathrm{Val}(v_7, x) = -16(1 - x)$$

Simple WTGs with arbitrary weights

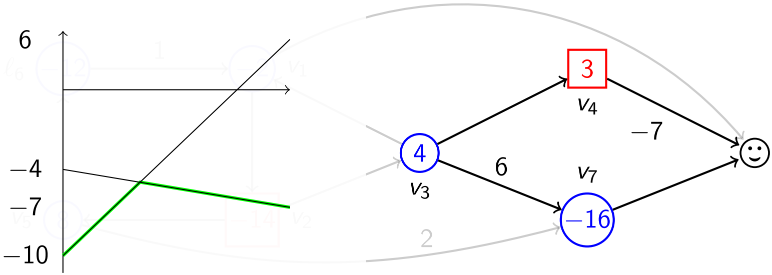
Joint work with T. Brihaye, G. Geeraerts, A. Haddad and E. Lefaucheux



$$\mathrm{Val}(v_4, x) = -3x - 4, \qquad \mathrm{Val}(v_7, x) = -16(1 - x),$$

Simple WTGs with arbitrary weights

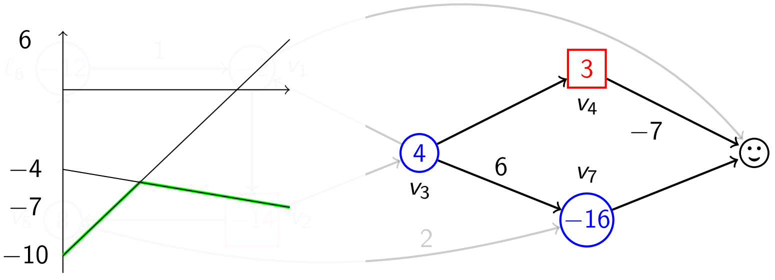
Joint work with T. Brihaye, G. Geeraerts, A. Haddad and E. Lefaucheux



$$\mathsf{Val}(v_4, x) = -3x - 4, \qquad \mathsf{Val}(v_7, x) = -16(1 - x),$$
$$\mathsf{Val}(v_3, x) = \min(-3x - 4, -16(1 - x) + 6)$$

Simple WTGs with arbitrary weights

Joint work with T. Brihaye, G. Geeraerts, A. Haddad and E. Lefaucheux



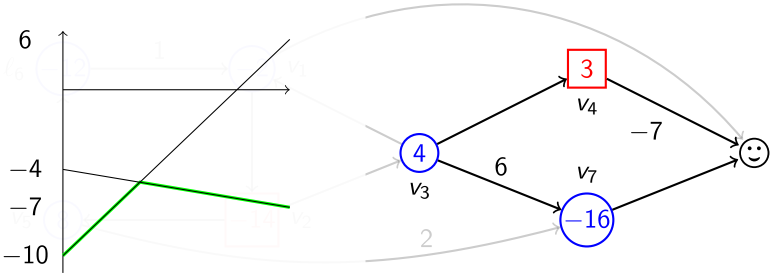
$$\mathrm{Val}(v_4, x) = -3x - 4, \qquad \mathrm{Val}(v_7, x) = -16(1-x),$$
$$\mathrm{Val}(v_3, x) = \min(-3x - 4, -16(1-x) + 6)$$

Theorem:

For every simple WTG, all value functions are piecewise affine, with at most an exponential number of cutpoints, and can be computed in exponential time.

Simple WTGs with arbitrary weights

Joint work with T. Brihaye, G. Geeraerts, A. Haddad and E. Lefaucheux



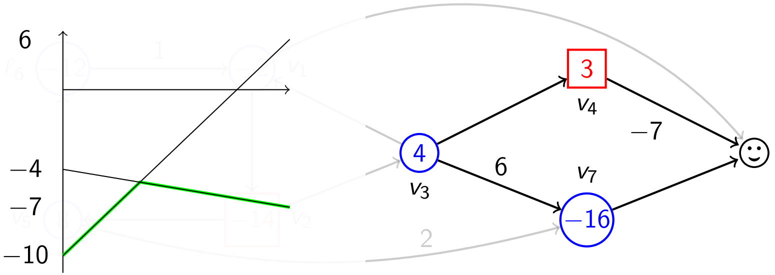
$$\mathrm{Val}(v_4, x) = -3x - 4, \qquad \mathrm{Val}(v_7, x) = -16(1-x),$$
$$\mathrm{Val}(v_3, x) = \min(-3x - 4, -16(1-x) + 6)$$

Theorem: NEW!

For every simple WTG, all value functions are piecewise affine, with at most a **pseudo-polynomial** number of cutpoints, and can be computed in **pseudo-polynomial** time.

Simple WTGs with arbitrary weights

Joint work with T. Brihaye, G. Geeraerts, A. Haddad and E. Lefaucheux



Theorem: NEW!

For every simple WTG, all value functions are piecewise affine, with at most a **pseudo-polynomial** number of cutpoints, and can be computed in **pseudo-polynomial** time.

For general 1-clock WTGs?

- removing guards: previously used techniques work!
- removing resets: previously, bound the number of resets...

One-clock WTG with arbitrary weights NEW!

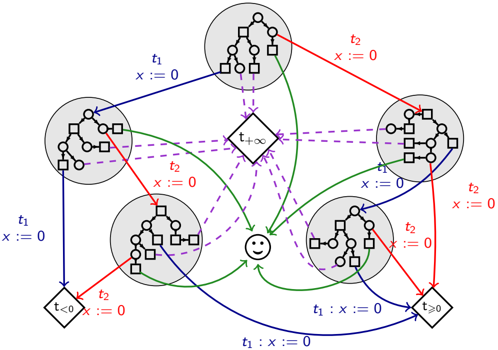
Joint work with J. Parreaux and P.-A. Reynier

New idea: limit the number of resets (to at most once for each transition), after having blown up exponentially the WTG

One-clock WTG with arbitrary weights NEW!

Joint work with J. Parreaux and P.-A. Reynier

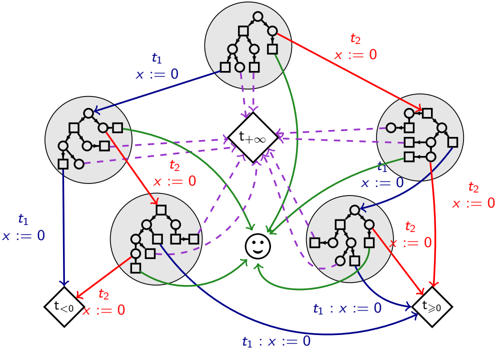
New idea: limit the number of resets (to at most once for each transition), after having blown up exponentially the WTG



One-clock WTG with arbitrary weights NEW!

Joint work with J. Parreaux and P.-A. Reynier

New idea: limit the number of resets (to at most once for each transition), after having blown up exponentially the WTG



Theorem:

For every 1-clock WTG, all value functions can be computed in time exponential in the number of locations and in the largest transition weight, and polynomial in other weights.

State of the art: $\geqslant 0$ weights

$\geqslant 0$ **weights and strictly non-Zeno-cost cycles**: 2-exp algo
(Bouyer, Cassez, Fleury, and Larsen 2004; Alur, Bernadsky, and Madhusudan 2004)

Value iteration algorithm: compute $\mathcal{F}^i(+\infty)$...

$$\mathcal{F}(\boldsymbol{x})_{(v,\nu)} = \begin{cases} \sup\limits_{(v,\nu)\xrightarrow{d,t}(v',\nu')} \big(d \times \mathsf{Weight}(v) + \mathsf{Weight}(t) + \boldsymbol{x}_{(v',\nu')}\big) & \text{if } v \in V_{\mathsf{Max}} \\ \inf\limits_{(v,\nu)\xrightarrow{d,t}(v',\nu')} \big(d \times \mathsf{Weight}(v) + \mathsf{Weight}(t) + \boldsymbol{x}_{(v',\nu')}\big) & \text{if } v \in V_{\mathsf{Min}} \end{cases}$$

Extension to negative weights

Joint work with D. Busatto-Gaston and P.-A. Reynier

Divergence property:
Every execution following a cycle of the region automaton has a total weight either $\leqslant -1$ or $\geqslant 1$

Extension to negative weights

Joint work with D. Busatto-Gaston and P.-A. Reynier

Divergence property:
Every execution following a cycle of the region automaton has a total weight either $\leqslant -1$ or $\geqslant 1$

Characterisation: all simple cycles of an SCC of the region automaton have the same sign

Extension to negative weights

Joint work with D. Busatto-Gaston and P.-A. Reynier

Divergence property:
Every execution following a cycle of the region automaton has a total weight either $\leqslant -1$ or $\geqslant 1$

Characterisation: all simple cycles of an SCC of the region automaton have the same sign

Theorem:

Deciding if a WTG is divergent is **PSPACE**-complete.

Extension to negative weights

Joint work with D. Busatto-Gaston and P.-A. Reynier

Divergence property:
Every execution following a cycle of the region automaton has a total weight either $\leqslant -1$ or $\geqslant 1$

Characterisation: all simple cycles of an SCC of the region automaton have the same sign

Theorem:

Deciding if a WTG is divergent is **PSPACE**-complete.

Theorem:

The value problem on divergent WTG is in 3-**EXP**, and is **EXP**-hard.

What about cycles of weight $= 0$?

- Adding cycles of weight $= 0$ to divergent WTG: **undecidable** but *approximable* (Bouyer, Jaziri, and Markey 2015)

What about cycles of weight $= 0$?

- Adding cycles of weight $= 0$ to divergent WTG: **undecidable** but *approximable* (Bouyer, Jaziri, and Markey 2015)

Joint work with D. Busatto-Gaston and P.-A. Reynier

Almost-divergent WTG: every SCC of the region automaton is

$$\text{either } (\geqslant 1 \text{ or } = 0), \qquad \text{or } (\leqslant -1 \text{ or } = 0)$$

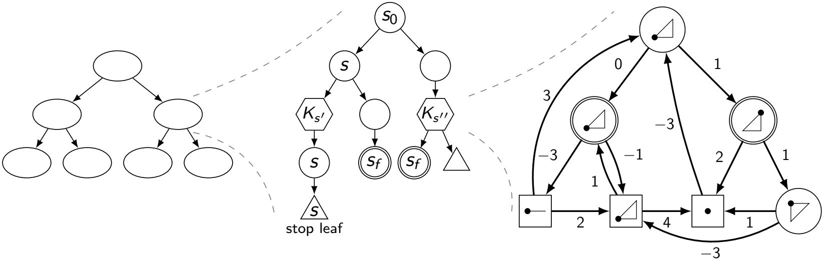
What about cycles of weight $= 0$?

- Adding cycles of weight $= 0$ to divergent WTG: **undecidable** but *approximable* (Bouyer, Jaziri, and Markey 2015)

Joint work with D. Busatto-Gaston and P.-A. Reynier

Almost-divergent WTG: every SCC of the region automaton is

$$\text{either } (\geqslant 1 \text{ or } = 0), \qquad \text{or } (\leqslant -1 \text{ or } = 0)$$



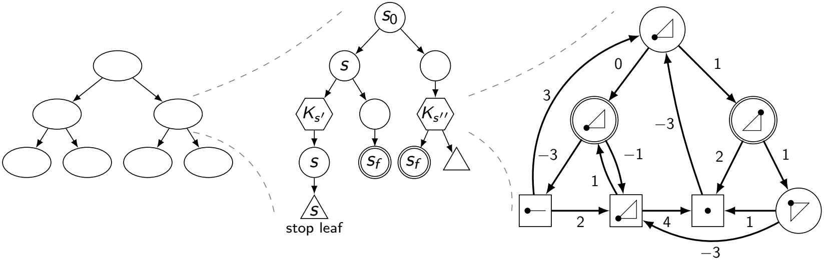
What about cycles of weight $= 0$?

- Adding cycles of weight $= 0$ to divergent WTG: **undecidable** but *approximable* (Bouyer, Jaziri, and Markey 2015)

Joint work with D. Busatto-Gaston and P.-A. Reynier

Almost-divergent WTG: every SCC of the region automaton is

$$\text{either } (\geqslant 1 \text{ or } = 0), \qquad \text{or } (\leqslant -1 \text{ or } = 0)$$



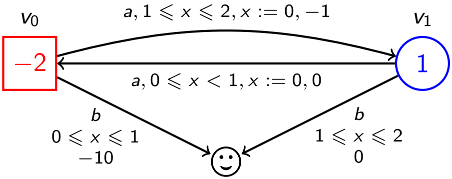
Theorem:

Approximation is decidable for almost-divergent WTGs: (semi-)symbolic algorithm that does not rely on an a-priori discretisation of the regions with a fixed granularity $1/N$.

Part III : Trade memory for randomisation

How to define stochastic strategies?

(Bertrand, Bouyer, Brihaye, Menet, Baier, Grösser, and Jurdziński 2014)

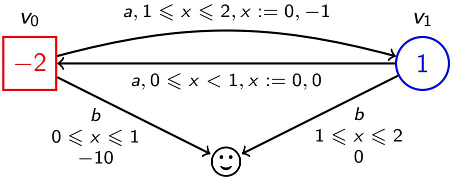


Deterministic strategy

Choose an edge and a delay

How to define stochastic strategies?

(Bertrand, Bouyer, Brihaye, Menet, Baier, Grösser, and Jurdziński 2014)



Deterministic strategy

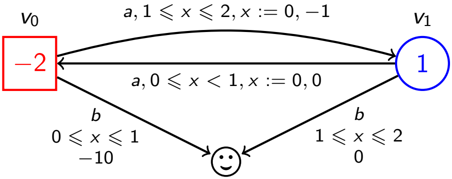
Choose an edge and a delay

In $(v_1, 0)$

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

How to define stochastic strategies?

(Bertrand, Bouyer, Brihaye, Menet, Baier, Grösser, and Jurdziński 2014)



Deterministic strategy

Choose an edge and a delay

In $(v_1, 0)$

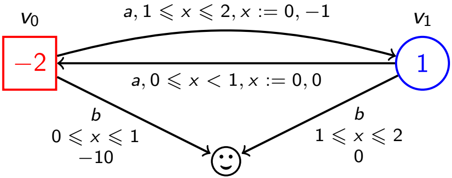
Choose a with $t = \frac{1}{3}$

Probabilistic strategy

Distribution over possible choices

How to define stochastic strategies?

(Bertrand, Bouyer, Brihaye, Menet, Baier, Grösser, and Jurdziński 2014)



Deterministic strategy

Choose an edge and a delay

In $(v_1, 0)$

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

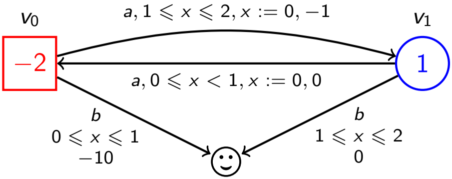
Probabilistic strategy

Distribution over possible choices

1. Edge: finite distribution

How to define stochastic strategies?

(Bertrand, Bouyer, Brihaye, Menet, Baier, Grösser, and Jurdziński 2014)



Deterministic strategy

Choose an edge and a delay

In $(v_1, 0)$

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

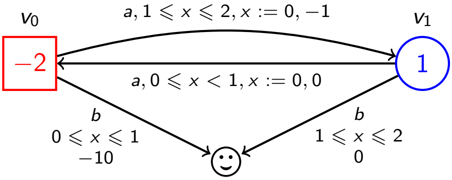
Probabilistic strategy

Distribution over possible choices

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

How to define stochastic strategies?

(Bertrand, Bouyer, Brihaye, Menet, Baier, Grösser, and Jurdziński 2014)



Deterministic strategy

Choose an edge and a delay

In $(v_1, 0)$

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

Probabilistic strategy

Distribution over possible choices

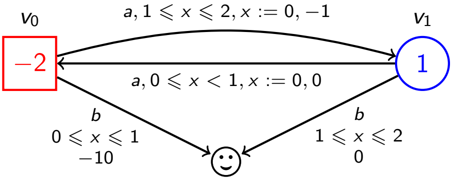
1. Edge: finite distribution
2. Delay: infinite distribution

In $(v_1, 0)$

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

How to define stochastic strategies?

(Bertrand, Bouyer, Brihaye, Menet, Baier, Grösser, and Jurdziński 2014)



Deterministic strategy

Choose an edge and a delay

In $(v_1, 0)$

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

Probabilistic strategy

Distribution over possible choices

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

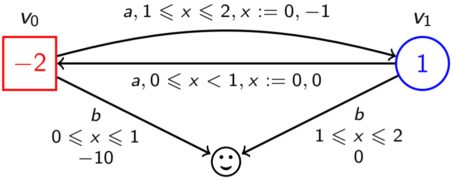
In $(v_1, 0)$

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

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

How to define stochastic strategies?

(Bertrand, Bouyer, Brihaye, Menet, Baier, Grösser, and Jurdziński 2014)



Deterministic strategy

Choose an edge and a delay

In $(v_1, 0)$

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

Probabilistic strategy

Distribution over possible choices

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

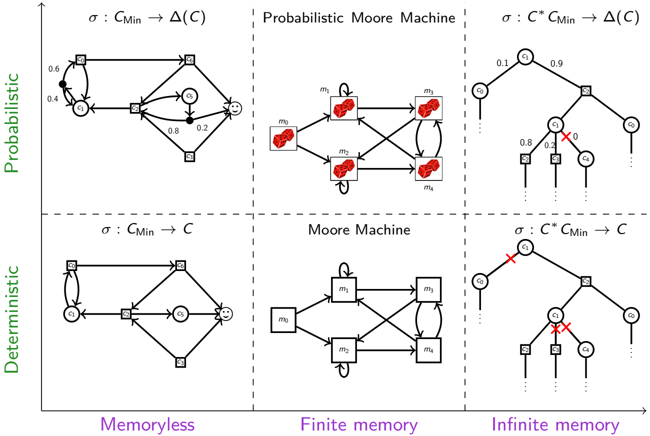
In $(v_1, 0)$

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

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

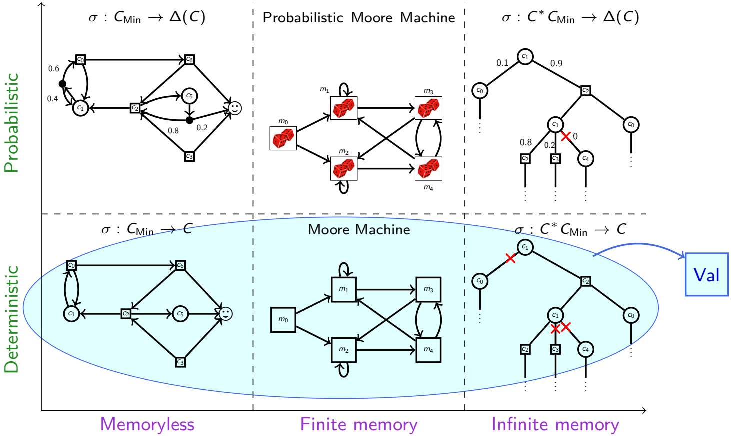
Trade memory for randomisation

Joint work with J. Parreaux and P.-A. Reynier



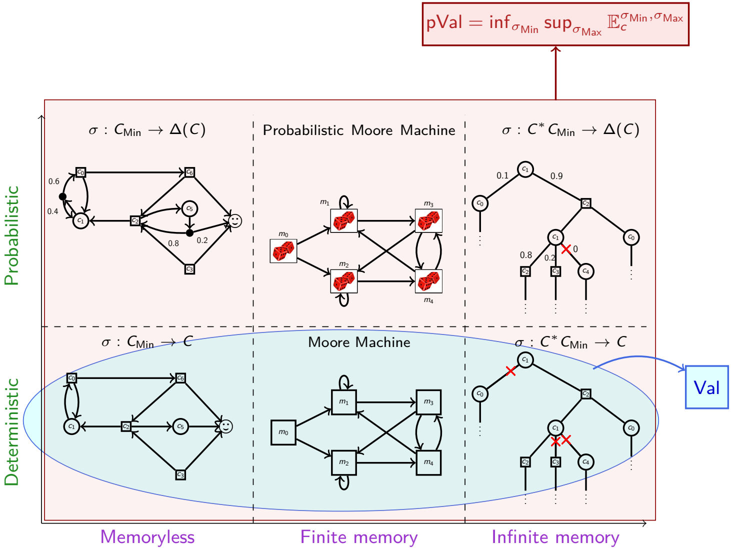
Trade memory for randomisation

Joint work with J. Parreaux and P.-A. Reynier



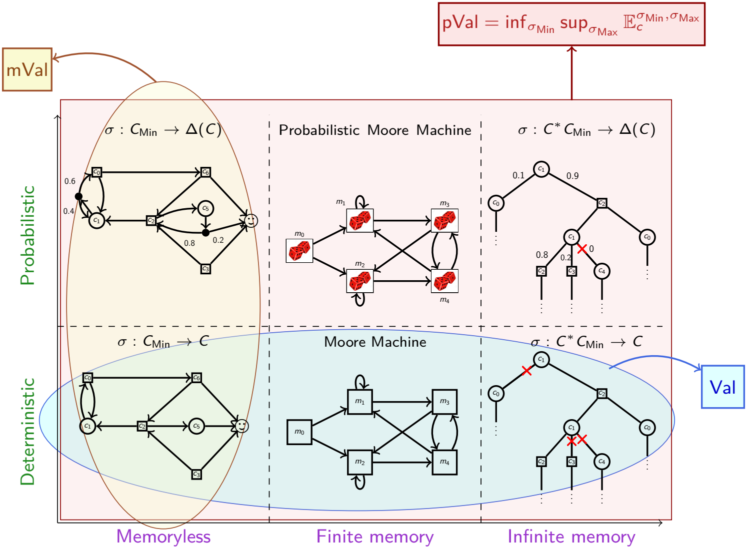
Trade memory for randomisation

Joint work with J. Parreaux and P.-A. Reynier



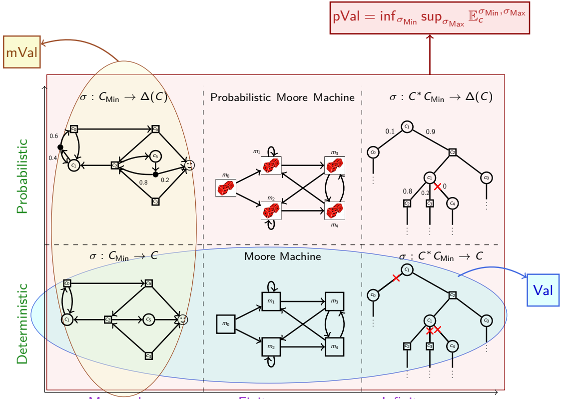
Trade memory for randomisation

Joint work with J. Parreaux and P.-A. Reynier



Trade memory for randomisation

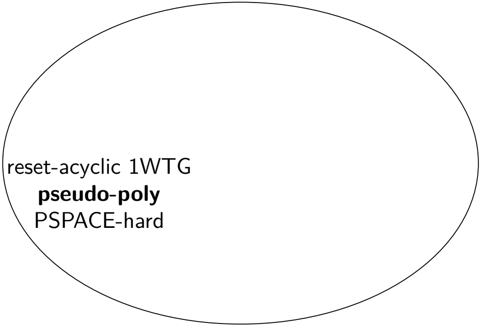
Joint work with J. Parreaux and P.-A. Reynier



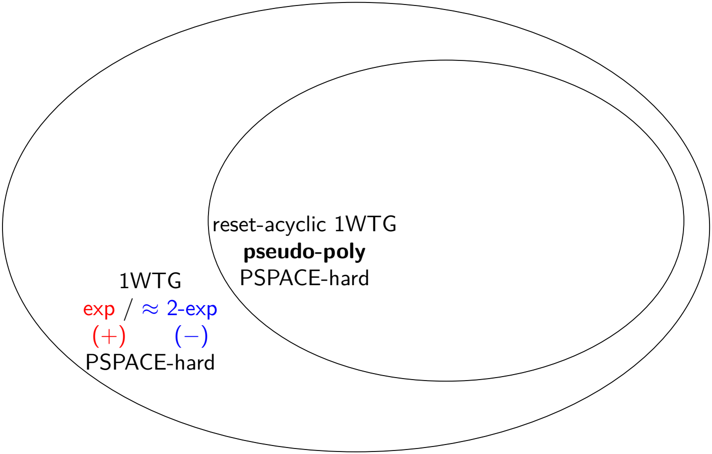
Theorem:

Val = pVal = mVal in weighted (untimed) games and in divergent WTG.

Conclusion

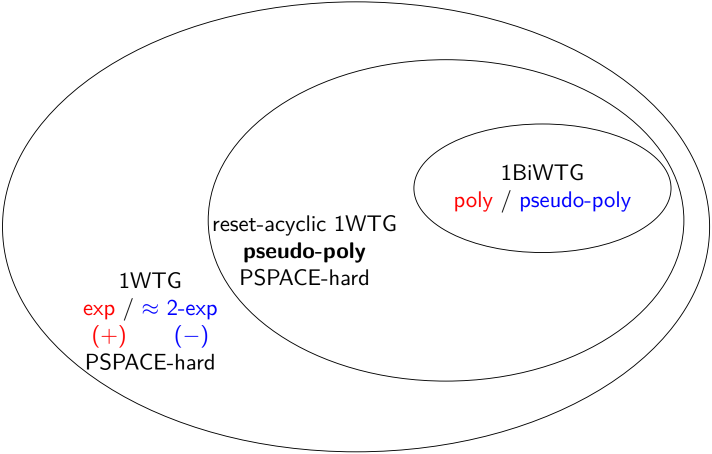


Conclusion



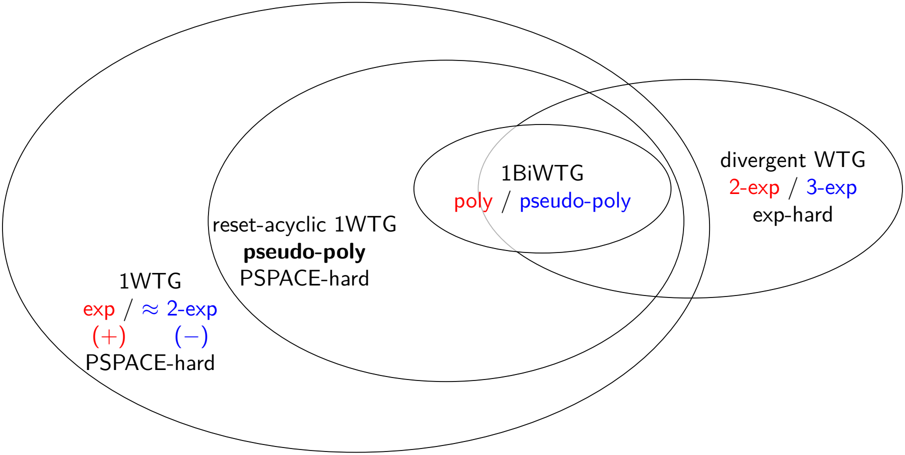
reset-acyclic 1WTG
pseudo-poly
PSPACE-hard
1WTG
exp / $\approx$ 2-exp
$(+)$ $(-)$
PSPACE-hard

Conclusion

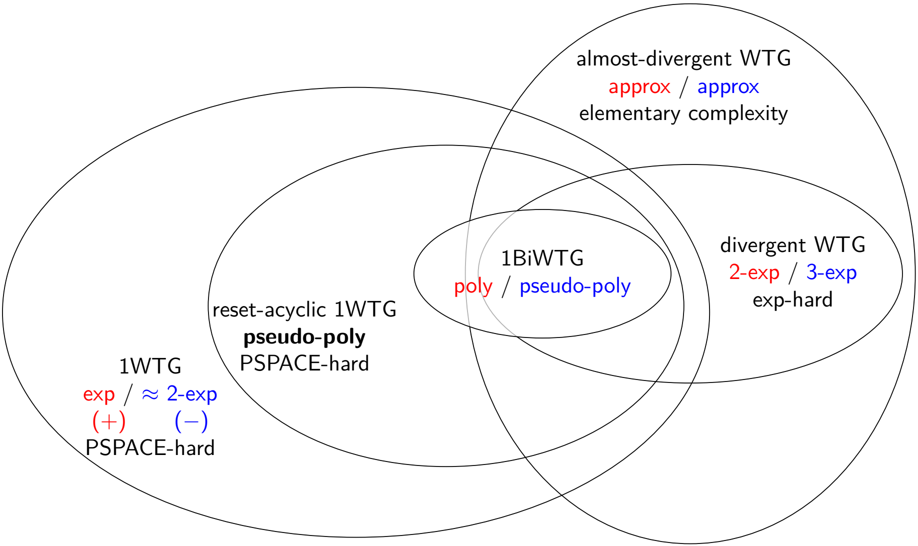


Joint work with T. Brihaye, G. Geeraerts, S. K. Narayanan, L. Manasa and A. Trivedi

Conclusion

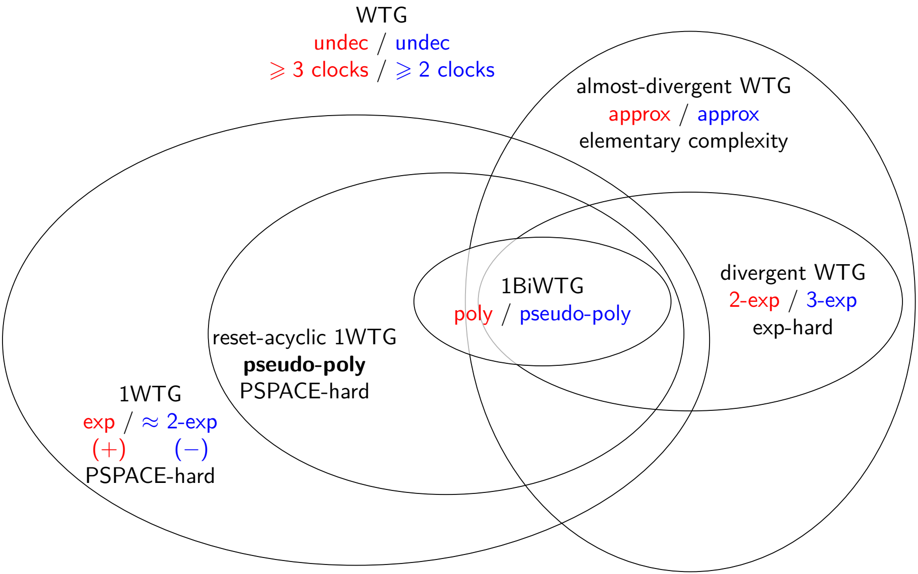

1BiWTG
poly / pseudo-poly
divergent WTG
2-exp / 3-exp
exp-hard
reset-acyclic 1WTG
pseudo-poly
PSPACE-hard
1WTG
exp / ≈ 2-exp
(+) (−)
PSPACE-hard

Conclusion



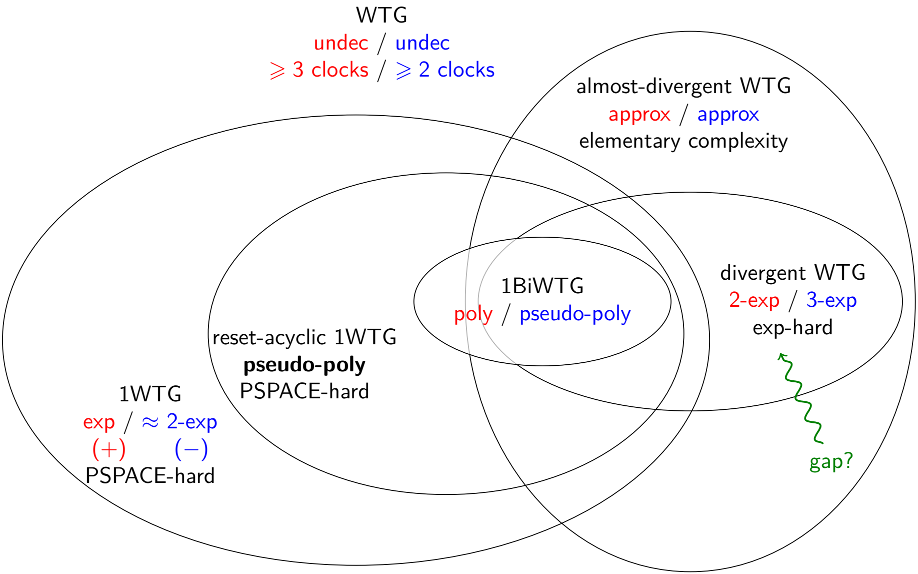
almost-divergent WTG
approx / approx
elementary complexity
divergent WTG
2-exp / 3-exp
exp-hard
1BiWTG
poly / pseudo-poly
reset-acyclic 1WTG
pseudo-poly
PSPACE-hard
1WTG
exp / ≈ 2-exp
(+) (−)
PSPACE-hard

Conclusion

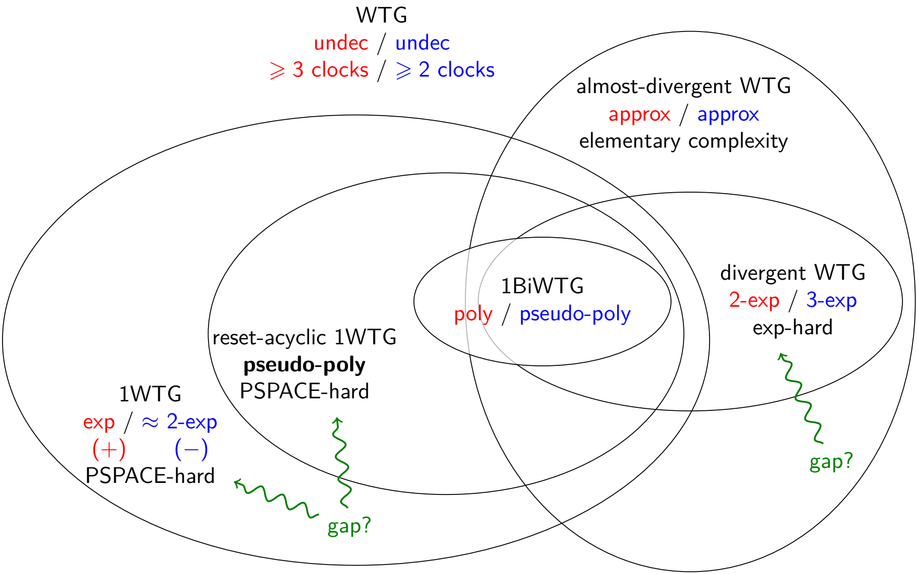


WTG
undec / undec
$\geqslant 3$ clocks / $\geqslant 2$ clocks
almost-divergent WTG
approx / approx
elementary complexity
1BiWTG
poly / pseudo-poly
divergent WTG
2-exp / 3-exp
exp-hard
reset-acyclic 1WTG
pseudo-poly
PSPACE-hard
1WTG
exp / $\approx$ 2-exp
(+) (−)
PSPACE-hard

Conclusion

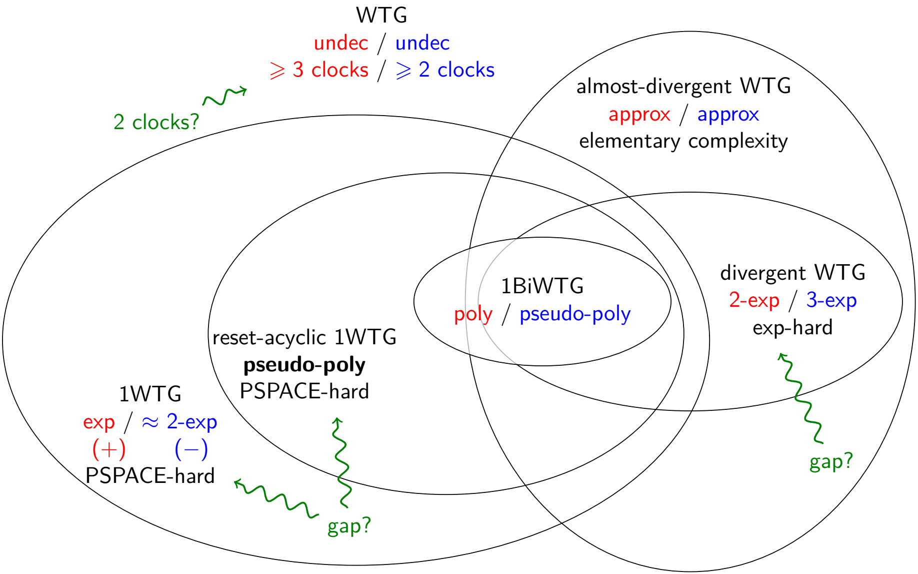

WTG
undec / undec
⩾ 3 clocks / ⩾ 2 clocks
almost-divergent WTG
approx / approx
elementary complexity
1BiWTG
poly / pseudo-poly
divergent WTG
2-exp / 3-exp
exp-hard
reset-acyclic 1WTG
pseudo-poly
PSPACE-hard
1WTG
exp / ≈ 2-exp
(+) (−)
PSPACE-hard
gap?

Conclusion



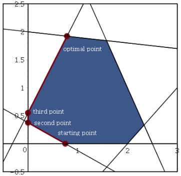
WTG
undec / undec
⩾ 3 clocks / ⩾ 2 clocks
almost-divergent WTG
approx / approx
elementary complexity
1BiWTG
poly / pseudo-poly
divergent WTG
2-exp / 3-exp
exp-hard
reset-acyclic 1WTG
pseudo-poly
PSPACE-hard
1WTG
exp / ≈ 2-exp
(+) (−)
PSPACE-hard
gap?
gap?

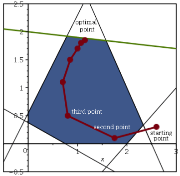
Conclusion



WTG
undec / undec
$\geqslant 3$ clocks / $\geqslant 2$ clocks
2 clocks?
almost-divergent WTG
approx / approx
elementary complexity
1BiWTG
poly / pseudo-poly
divergent WTG
2-exp / 3-exp
exp-hard
reset-acyclic 1WTG
pseudo-poly
PSPACE-hard
1WTG
exp / $\approx$ 2-exp
$(+)$ $(-)$
PSPACE-hard
gap?
gap?

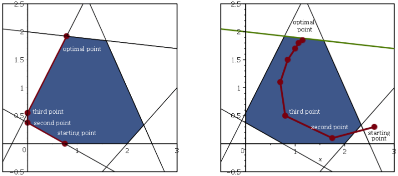
Perspectives



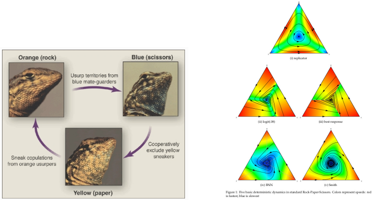


Poly-time algorithms in weighted games

Perspectives

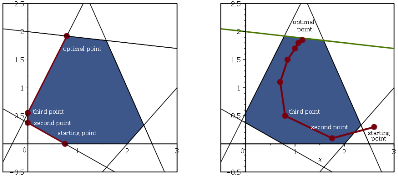


Poly-time algorithms in weighted games

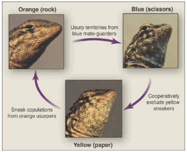


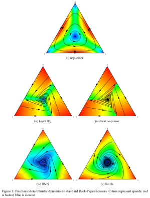
Evolutionary game theory on graphs

Perspectives



Poly-time algorithms in weighted games





Evolutionary game theory on graphs

Play with less visibility:

- robustness to environmental perturbations
- randomisation with interval of delays
- incomplete information

Appendix

Back to appendix

Case study

	Peak-hour 	Offpeak-hour 
	15 c€/kWh	12 c€/kWh
rate:	total power $\times$ 15 c€/h	total power $\times$ 12 c€/h

states to record which device is on/off: computation of the total power

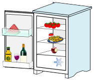
Back to appendix

Case study

	Peak-hour 	Offpeak-hour 
	15 c€/kWh	12 c€/kWh
rate:	total power $\times$ 15 c€/h	total power $\times$ 12 c€/h

states to record which device is on/off: computation of the total power

Power consumption:

-  100W (1.5 c€/h in peak-hour, 1.2 c€/h in offpeak-hour)
-  2500W (37.5 c€/h in peak-hour, 30 c€/h in offpeak-hour)
-  2000W (24 c€/h in offpeak-hour)

Back to appendix

Case study

	Peak-hour 	Offpeak-hour 	Solar panels 
	15 c€/kWh	12 c€/kWh	Reselling: 20 c€/kWh
rate:	total power $\times$ 15 c€/h	total power $\times$ 12 c€/h	-0.5×20 c€/h

states to record which device is on/off: computation of the total power

Back to appendix

Case study

	Peak-hour 	Offpeak-hour 	Solar panels 
	15 c€/kWh	12 c€/kWh	Reselling: 20 c€/kWh
rate:	total power $\times$ 15 c€/h	total power $\times$ 12 c€/h	-0.5×20 c€/h

states to record which device is on/off: computation of the total power

Environment: user profile, weather profile  / 
Controller: chooses contract (discrete cost for the monthly subscription) and exact consumption (what, when...)

Back to appendix

Case study

	Peak-hour 	Offpeak-hour 	Solar panels 
	15 c€/kWh	12 c€/kWh	Reselling: 20 c€/kWh
rate:	total power $\times$ 15 c€/h	total power $\times$ 12 c€/h	-0.5×20 c€/h

states to record which device is on/off: computation of the total power

Environment: user profile, weather profile  / 
Controller: chooses contract (discrete cost for the monthly subscription) and exact consumption (what, when...)

Goal: optimise the energy consumption based on the cost

Back to appendix

Case study

	Peak-hour 	Offpeak-hour 	Solar panels 
	15 c€/kWh	12 c€/kWh	Reselling: 20 c€/kWh
rate:	total power $\times$ 15 c€/h	total power $\times$ 12 c€/h	-0.5×20 c€/h

states to record which device is on/off: computation of the total power

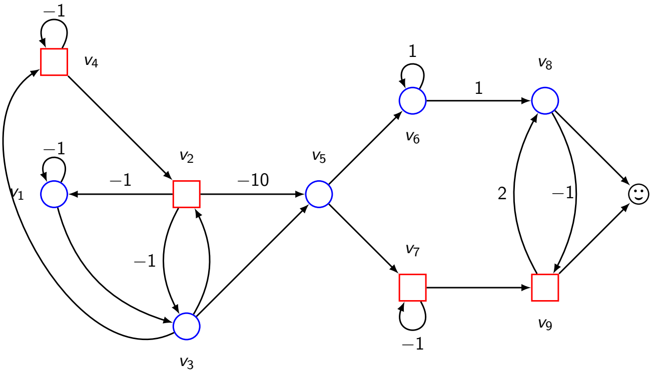
Environment: user profile, weather profile  / 
Controller: chooses contract (discrete cost for the monthly subscription) and exact consumption (what, when...)

Goal: optimise the energy consumption based on the cost

Solution 1: discretisation of time, resolution via a *weighted game*
Solution 2: thin time behaviours, resolution via a *weighted timed game*
Solution 3: allow for *randomisation* in the behaviours?

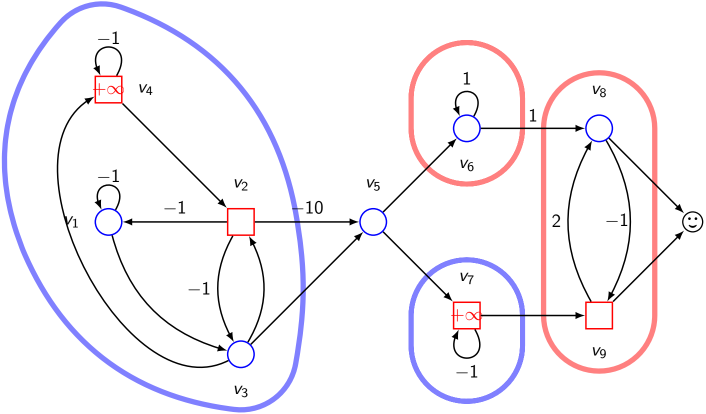
▸ Back to appendix

Example of divergent weighted game



Min = ○, Max = □

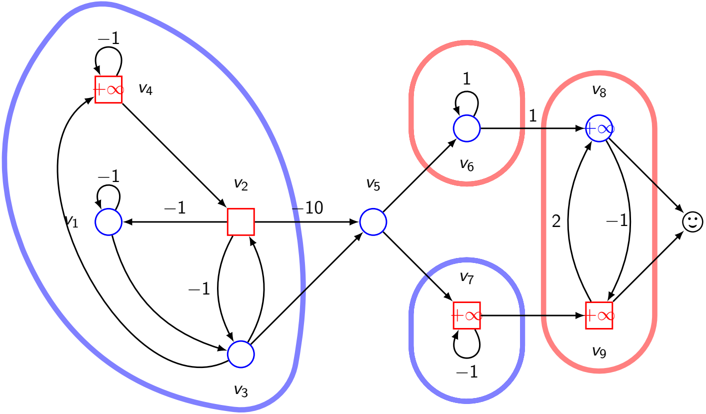
Example of divergent weighted game



Min = ◯, Max = □

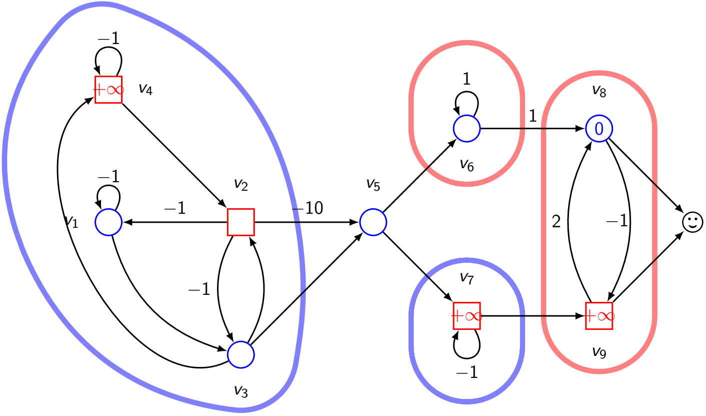
Back to appendix

Example of divergent weighted game



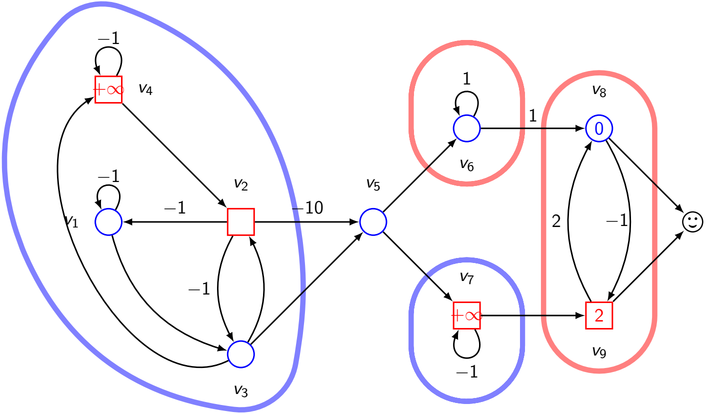
Min = ○, Max = □

Example of divergent weighted game



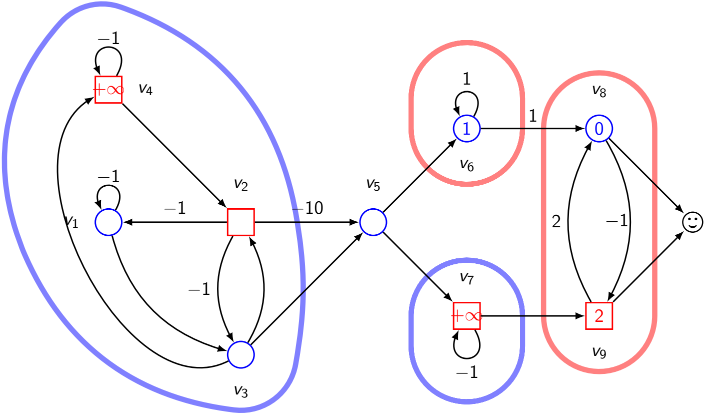
Min = ○, Max = □

Example of divergent weighted game



Min = ○, Max = □

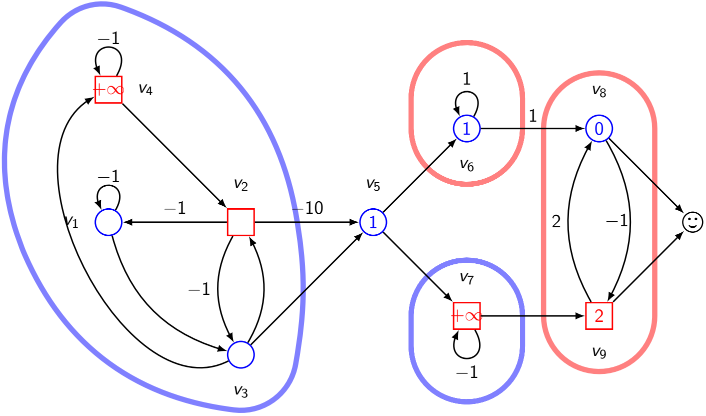
Example of divergent weighted game



Min = ◯, Max = □

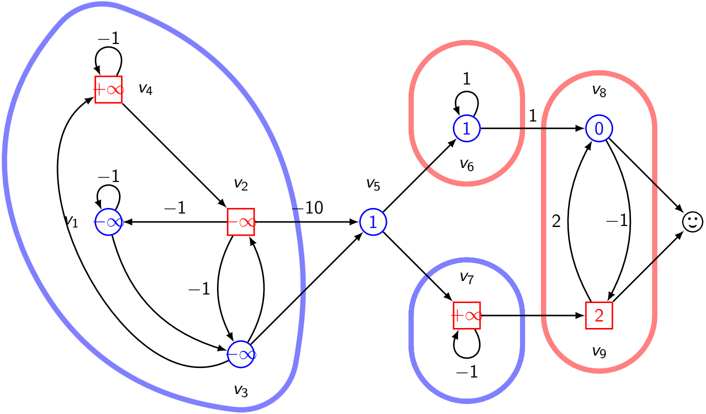
▸ Back to appendix

Example of divergent weighted game



Min = ○, Max = □

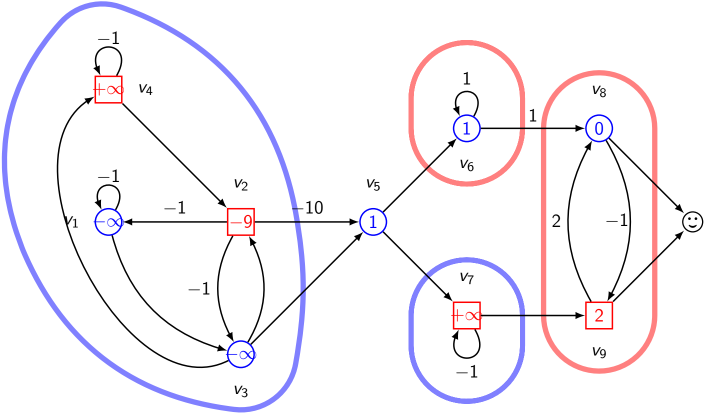
Example of divergent weighted game



Min = ○, Max = □

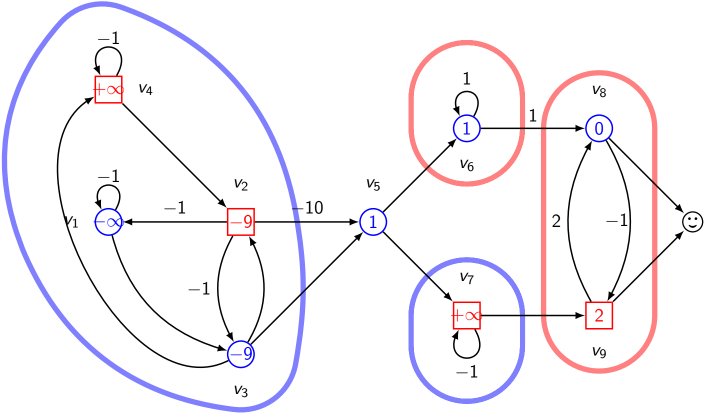
Back to appendix

Example of divergent weighted game



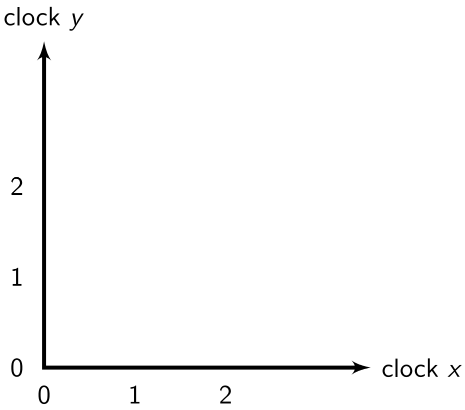
Min = ○, Max = □

Example of divergent weighted game



Min = ○, Max = □

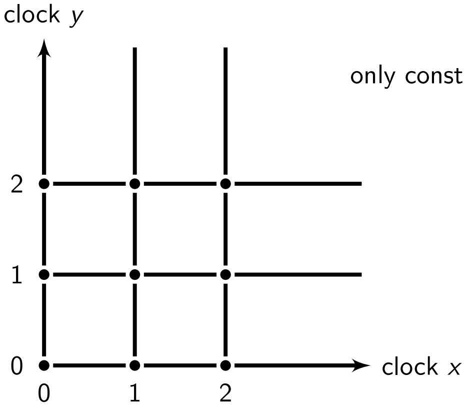
A fundamental tool: region abstraction



Min = ◯, Max = □

Back to appendix

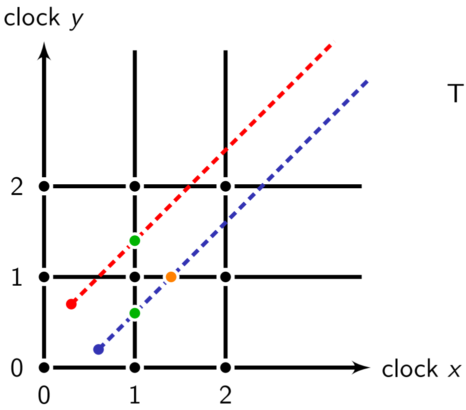
A fundamental tool: region abstraction



only constraints: $x \sim c$ with $c \in \{0, 1, 2\}$
$y \sim c$ with $c \in \{0, 1, 2\}$

- compatibility between regions and guards

A fundamental tool: region abstraction

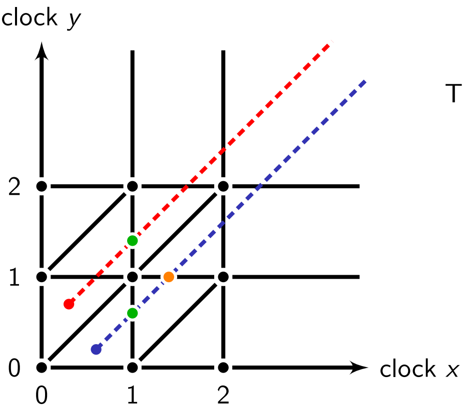


The path 

- can be fired from ●
- cannot be fired from ●

- compatibility between regions and guards
- compatibility between regions and delays

A fundamental tool: region abstraction

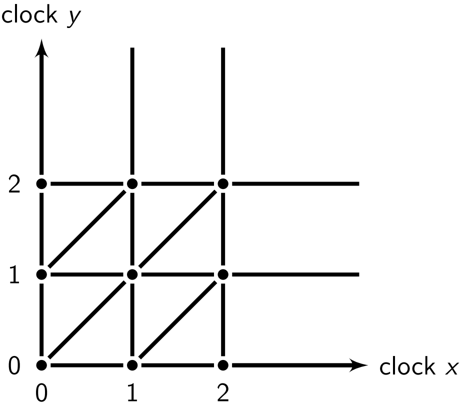


The path 

- can be fired from ●
- cannot be fired from ●

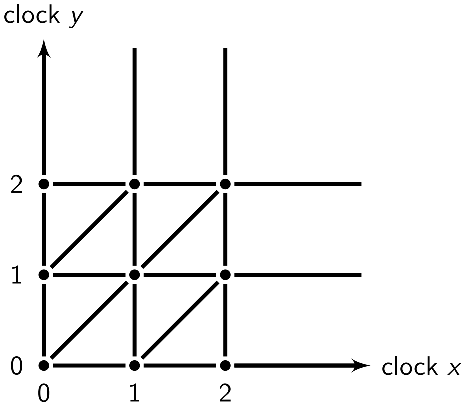
- compatibility between regions and guards
- compatibility between regions and delays

A fundamental tool: region abstraction



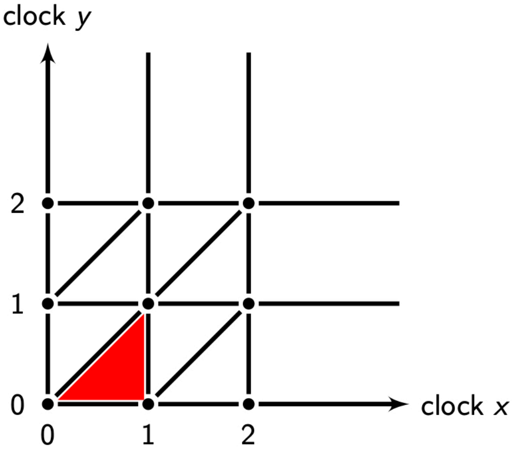
- compatibility between regions and guards
- compatibility between regions and delays

A fundamental tool: region abstraction



- compatibility between regions and guards
- compatibility between regions and delays
- $\rightarrow$ equivalence relation of finite index

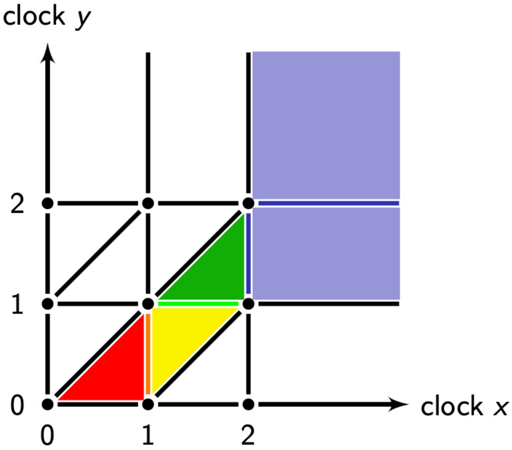
A fundamental tool: region abstraction



- region R defined by:

$$\begin{cases} 0 < x < 1 \\ 0 < y < 1 \\ y < x \end{cases}$$

A fundamental tool: region abstraction

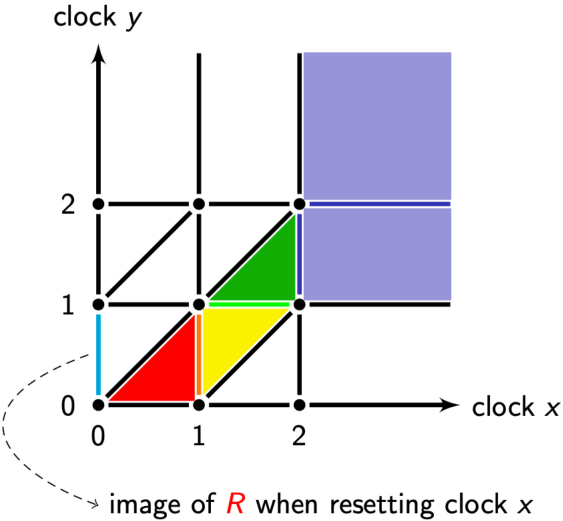


- region R defined by:

$$\begin{cases} 0 < x < 1 \\ 0 < y < 1 \\ y < x \end{cases}$$

- time successors of R

A fundamental tool: region abstraction



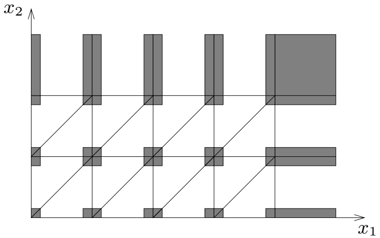
- region R defined by:

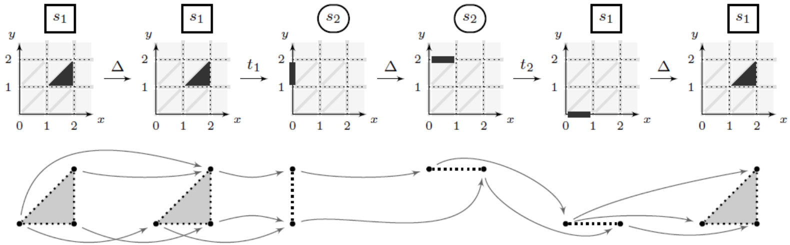
$$\begin{cases} 0 < x < 1 \\ 0 < y < 1 \\ y < x \end{cases}$$

- time successors of R

Corner-point abstraction

- Main tool to solve one-player WTG: refinement of regions via corner point abstraction / ε-graph (Bouyer, Brinksma, and Larsen 2004; Bouyer, Brihaye, Bruyère, and Raskin 2007)





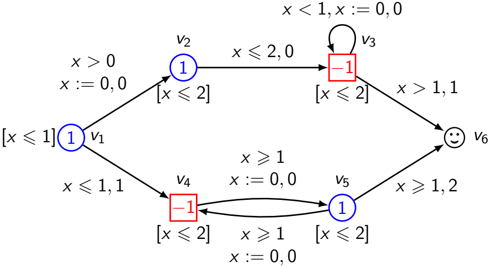
Min = ○, Max = □

Back to appendix

One-clock Bi-WTGs (1BiWTGs)

Joint work with T. Brihaye, G. Geeraerts, S. K. Narayanan, L. Manasa and A. Trivedi

Weights of locations $\{p^-, p^+\}$ included in $\{0, +d, -d\}$, $d \in \mathbb{N}$



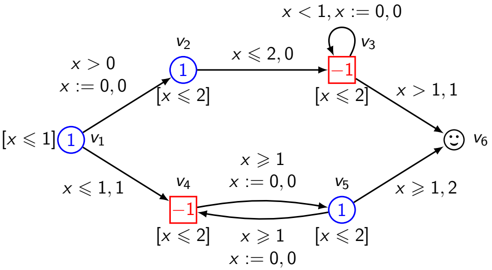
Min = ○, Max = □

Back to appendix

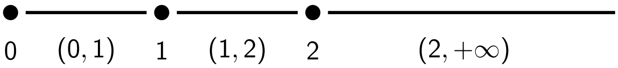
One-clock Bi-WTGs (1BiWTGs)

Joint work with T. Brihaye, G. Geeraerts, S. K. Narayanan, L. Manasa and A. Trivedi

Weights of locations $\{p^-, p^+\}$ included in $\{0, +d, -d\}$, $d \in \mathbb{N}$



Region abstraction:



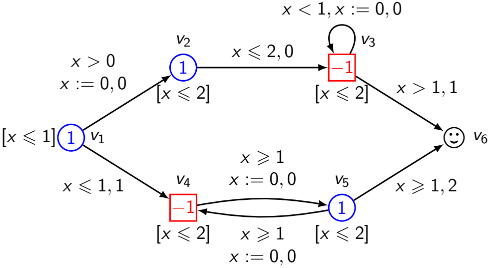
Min = ○, Max = □

Back to appendix

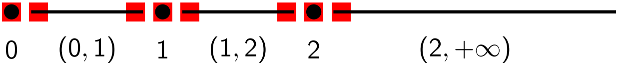
One-clock Bi-WTGs (1BiWTGs)

Joint work with T. Brihaye, G. Geeraerts, S. K. Narayanan, L. Manasa and A. Trivedi

Weights of locations $\{p^-, p^+\}$ included in $\{0, +d, -d\}$, $d \in \mathbb{N}$



Corner-point abstraction:



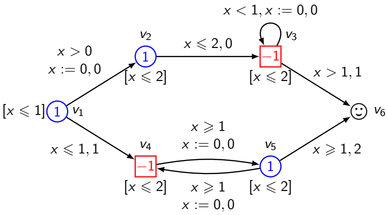
Min = ○, Max = □

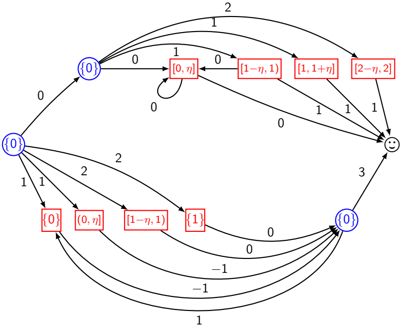
Back to appendix

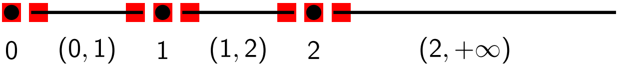
One-clock Bi-WTGs (1BiWTGs)

Joint work with T. Brihaye, G. Geeraerts, S. K. Narayanan, L. Manasa and A. Trivedi

Weights of locations $\{p^-, p^+\}$ included in $\{0, +d, -d\}$, $d \in \mathbb{N}$





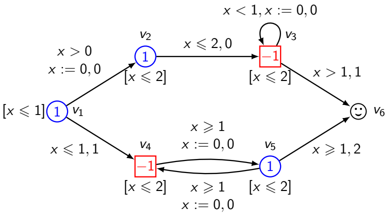


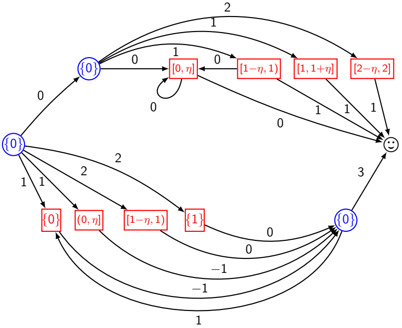
Min = ○, Max = □

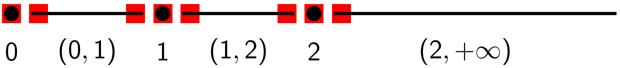
One-clock Bi-WTGs (1BiWTGs)

Joint work with T. Brihaye, G. Geeraerts, S. K. Narayanan, L. Manasa and A. Trivedi

Weights of locations $\{p^-, p^+\}$ included in $\{0, +d, -d\}$, $d \in \mathbb{N}$







Theorem:

Computation of the values of a 1BiWTG and synthesis of ε-optimal strategies in pseudo-polynomial time (polynomial time if $\geqslant 0$ weights only).

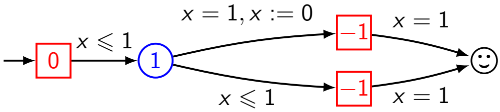
Min = ○, Max = □

Back to appendix

1BiWTG: maximal fragment for corner-point abstraction

Generalisation by E. Lefaucheux: two rates $\{p^-, p^+\}$ included in $\{0, +d, -c\}$ ($d, c \in \mathbb{N}$)
In more general settings, players may need to play far from corners...

- With 3 weights in $\{-1, 0, +1\}$: value 1/2...



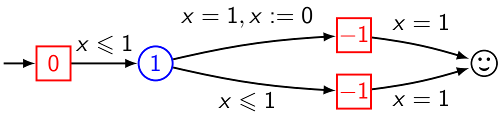
Min = ○, Max = □

Back to appendix

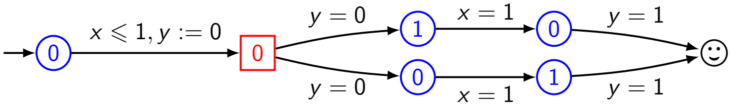
1BiWTG: maximal fragment for corner-point abstraction

Generalisation by E. Lefaucheux: two rates $\{p^-, p^+\}$ included in $\{0, +d, -c\}$ ($d, c \in \mathbb{N}$)
In more general settings, players may need to play far from corners...

- With 3 weights in $\{-1, 0, +1\}$: value 1/2...



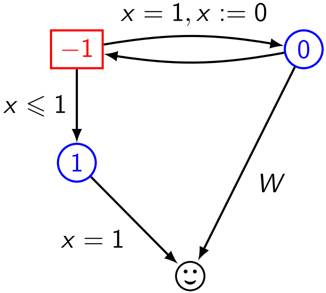
- With 2 weights in $\{-1, 0, +1\}$ but 2 clocks: value 1/2...



Min = ○, Max = □

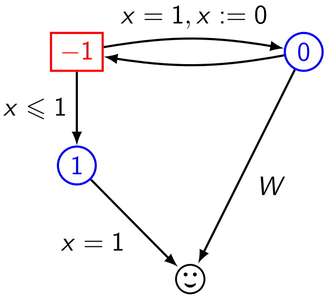
▸ Back to appendix

Bounding the number of resets needed is not easy



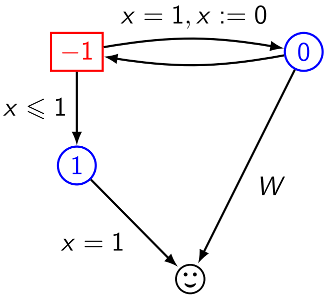
Min = ○, Max = □

Bounding the number of resets needed is not easy



Player ○ can guarantee (i.e., ensure to be below) value ε for all $\varepsilon > 0$...

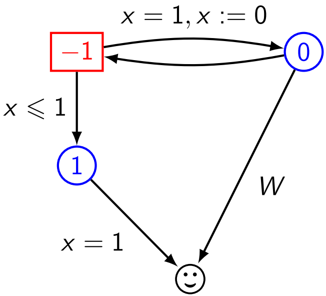
Bounding the number of resets needed is not easy



Player ◯ can guarantee (i.e., ensure to be below) value ε for all $\varepsilon > 0$...

... but cannot obtain 0: hence, no optimal strategy...

Bounding the number of resets needed is not easy



Player ○ can guarantee (i.e., ensure to be below) value ε for all $\varepsilon > 0$...

... but cannot obtain 0: hence, no optimal strategy...

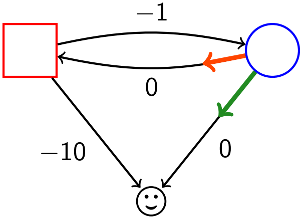
... moreover, to obtain ε, ○ needs to loop at least $W + \lceil 1/\log \varepsilon \rceil$ times before reaching ☺

Randomisation emulates memory

Let $(\sigma^1_{\text{Min}}, \sigma^2_{\text{Min}}, K)$ be an optimal switching strategy,

Randomisation emulates memory

Let $(\sigma^1_{\mathsf{Min}}, \sigma^2_{\mathsf{Min}}, K)$ be an optimal switching strategy,

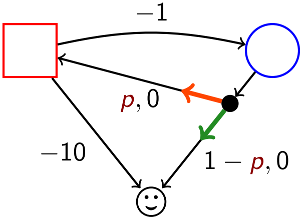


Min = ○, Max = □

Randomisation emulates memory

Let $(\sigma^1_{\mathsf{Min}}, \sigma^2_{\mathsf{Min}}, K)$ be an optimal switching strategy, for all $p \in (0, 1)$,

$$\eta^p = p \times \sigma^1_{\mathsf{Min}} + (1 - p) \times \sigma^2_{\mathsf{Min}}$$



Min = ○, Max = □

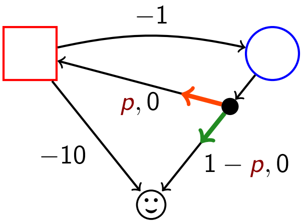
Back to appendix

Randomisation emulates memory

Let $(\sigma^1_{\mathsf{Min}}, \sigma^2_{\mathsf{Min}}, K)$ be an optimal switching strategy, for all $p \in (0,1)$,

$$\eta^p = p \times \sigma^1_{\mathsf{Min}} + (1-p) \times \sigma^2_{\mathsf{Min}}$$

- For all σ_{Max}, $\mathbb{P}_v^{\eta^p, \sigma_{\mathsf{Max}}}(\diamond ☺) = 1$
- For all σ_{Max}, $\mathbb{E}_v^{\eta^p, \sigma_{\mathsf{Max}}} < \infty$
- Max has a best response σ_{Max} memoryless and deterministic



Min = ○, Max = □

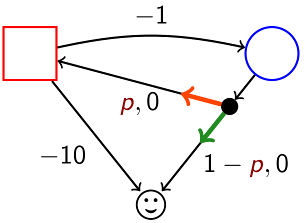
▸ Back to appendix

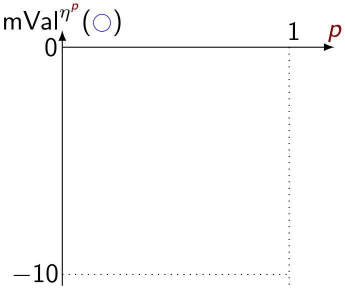
Randomisation emulates memory

Let $(\sigma^1_{\mathsf{Min}}, \sigma^2_{\mathsf{Min}}, K)$ be an optimal switching strategy, for all $p \in (0,1)$,

$$\eta^p = p \times \sigma^1_{\mathsf{Min}} + (1-p) \times \sigma^2_{\mathsf{Min}}$$

- For all σ_{Max}, $\mathbb{P}_v^{\eta^p, \sigma_{\mathsf{Max}}}(\diamond ☺) = 1$
- For all σ_{Max}, $\mathbb{E}_v^{\eta^p, \sigma_{\mathsf{Max}}} < \infty$
- Max has a best response σ_{Max} memoryless and deterministic





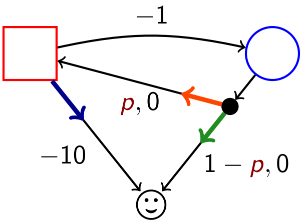
Min = ○, Max = □

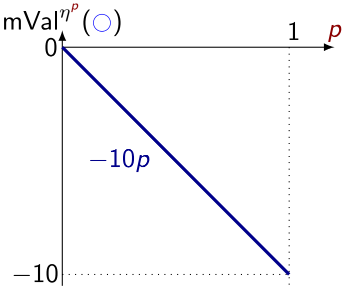
Randomisation emulates memory

Let $(\sigma^1_{\mathsf{Min}}, \sigma^2_{\mathsf{Min}}, K)$ be an optimal switching strategy, for all $p \in (0,1)$,

$$\eta^p = p \times \sigma^1_{\mathsf{Min}} + (1-p) \times \sigma^2_{\mathsf{Min}}$$

- For all σ_{Max}, $\mathbb{P}^{\eta^p,\sigma_{\mathsf{Max}}}_v(\Diamond ☺) = 1$
- For all σ_{Max}, $\mathbb{E}^{\eta^p,\sigma_{\mathsf{Max}}}_v < \infty$
- Max has a best response σ_{Max} memoryless and deterministic



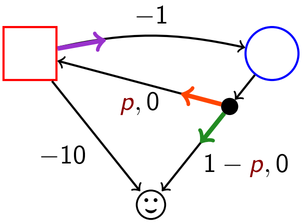


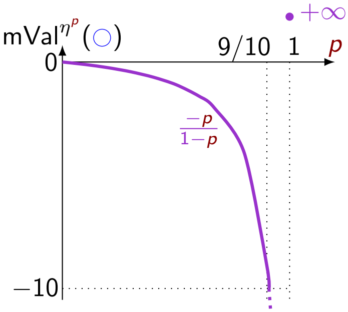
Randomisation emulates memory

Let $(\sigma^1_{\mathsf{Min}}, \sigma^2_{\mathsf{Min}}, K)$ be an optimal switching strategy, for all $p \in (0,1)$,

$$\eta^p = p \times \sigma^1_{\mathsf{Min}} + (1-p) \times \sigma^2_{\mathsf{Min}}$$

- For all σ_{Max}, $\mathbb{P}_v^{\eta^p, \sigma_{\mathsf{Max}}}(\Diamond ☺) = 1$
- For all σ_{Max}, $\mathbb{E}_v^{\eta^p, \sigma_{\mathsf{Max}}} < \infty$
- Max has a best response σ_{Max} memoryless and deterministic





Min = ○, Max = □

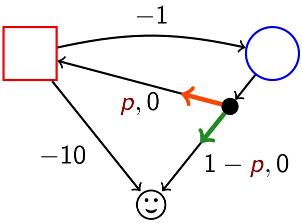
Back to appendix

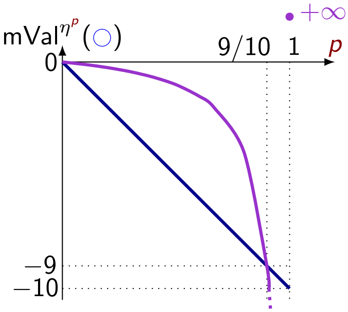
Randomisation emulates memory

Let $(\sigma^1_{\text{Min}}, \sigma^2_{\text{Min}}, K)$ be an optimal switching strategy, for all $p \in (0,1)$,

$$\eta^p = p \times \sigma^1_{\text{Min}} + (1-p) \times \sigma^2_{\text{Min}}$$

- For all σ_{Max}, $\mathbb{P}_v^{\eta^p,\sigma_{\text{Max}}}(\diamond ☺) = 1$
- For all σ_{Max}, $\mathbb{E}_v^{\eta^p,\sigma_{\text{Max}}} < \infty$
- Max has a best response σ_{Max} memoryless and deterministic





Min = ○, Max = □

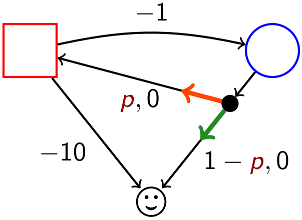
Back to appendix

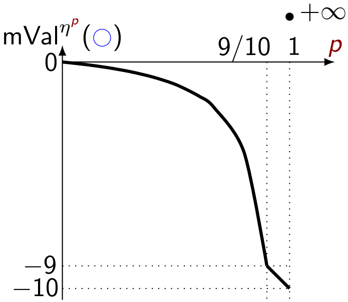
Randomisation emulates memory

Let $(\sigma^1_{\text{Min}}, \sigma^2_{\text{Min}}, K)$ be an optimal switching strategy, for all $p \in (0,1)$,

$$\eta^p = p \times \sigma^1_{\text{Min}} + (1-p) \times \sigma^2_{\text{Min}}$$

- For all σ_{Max}, $\mathbb{P}_v^{\eta^p, \sigma_{\text{Max}}}(\Diamond ☺) = 1$
- For all σ_{Max}, $\mathbb{E}_v^{\eta^p, \sigma_{\text{Max}}} < \infty$
- Max has a best response σ_{Max} memoryless and deterministic



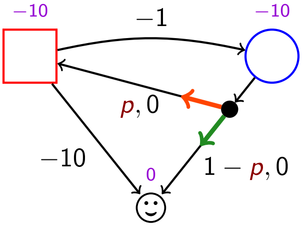


▸ Back to appendix

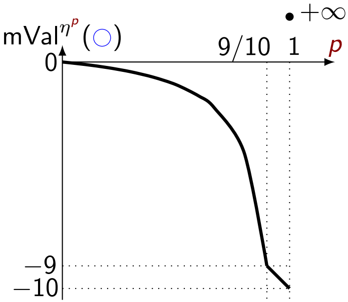
Randomisation emulates memory

Let $(\sigma^1_{\text{Min}}, \sigma^2_{\text{Min}}, K)$ be an optimal switching strategy, for all $p \in (0,1)$,

$$\eta^p = p \times \sigma^1_{\text{Min}} + (1-p) \times \sigma^2_{\text{Min}}$$



- For all σ_{Max}, $\mathbb{P}_v^{\eta^p, \sigma_{\text{Max}}}(\diamond ☺) = 1$
- For all σ_{Max}, $\mathbb{E}_v^{\eta^p, \sigma_{\text{Max}}} < \infty$
- Max has a best response σ_{Max} memoryless and deterministic



$$\lim_{\substack{p \to 1 \\ p < 1}} \text{mVal}^{\eta^p} \leqslant \text{Val}$$

Min $= \bigcirc$, Max $= \square$

Back to appendix

References I

- Alur, Rajeev, Mikhail Bernadsky, and P. Madhusudan (2004). "Optimal Reachability for Weighted Timed Games". In: *Proceedings of the 31st International Colloquium on Automata, Languages and Programming (ICALP'04)*. Vol. 3142. LNCS. Springer, pp. 122–133.
- Bertrand, Nathalie, Patricia Bouyer, Thomas Brihaye, Quentin Menet, Christel Baier, Marcus Grösser, and Marcin Jurdziński (2014). "Stochastic Timed Automata". In: *Log. Methods Comput. Sci.* 10.4.
- Bouyer, Patricia, Thomas Brihaye, Véronique Bruyère, and Jean-François Raskin (2007). "On the Optimal Reachability Problem of Weighted Timed Automata". In: *Formal Methods in System Design* 31.2, pp. 135–175.
- Bouyer, Patricia, Thomas Brihaye, and Nicolas Markey (2006). "Improved Undecidability Results on Weighted Timed Automata". In: *Information Processing Letters* 98.5, pp. 188–194.
- Bouyer, Patricia, Ed Brinksma, and Kim G. Larsen (2004). "Staying Alive as Cheaply as Possible". In: *Hybrid Systems: Computation and Control*. Springer, pp. 203–218.

References II

Bouyer, Patricia, Franck Cassez, Emmanuel Fleury, and Kim G. Larsen (2004). “Optimal Strategies in Priced Timed Game Automata”. In: *Proceedings of the 24th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'04)*. Vol. 3328. LNCS. Springer, pp. 148–160.

Bouyer, Patricia, Samy Jaziri, and Nicolas Markey (2015). “On the Value Problem in Weighted Timed Games”. In: *Proceedings of the 26th International Conference on Concurrency Theory (CONCUR'15)*. Vol. 42. Leibniz International Proceedings in Informatics. Leibniz-Zentrum für Informatik, pp. 311–324. DOI: 10.4230/LIPIcs.CONCUR.2015.311.

Bouyer, Patricia, Kim G. Larsen, Nicolas Markey, and Jacob Illum Rasmussen (2006). “Almost Optimal Strategies in One-Clock Priced Timed Games”. In: *Proceedings of the 26th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'06)*. Vol. 4337. Lecture Notes in Computer Science. Springer, pp. 345–356.

References III

Brihaye, Thomas, Véronique Bruyère, and Jean-François Raskin (2005). "On Optimal Timed Strategies". In: *Proceedings of the Third international conference on Formal Modeling and Analysis of Timed Systems (FORMATS'05)*. Vol. 3829. Lecture Notes in Computer Science. Springer, pp. 49–64.

Brihaye, Thomas, Gilles Geeraerts, Axel Haddad, Engel Lefaucheux, and Benjamin Monmege (2015). "Simple Priced Timed Games Are Not That Simple". In: *Proceedings of the 35th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'15)*. Vol. 45. LIPIcs. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, pp. 278–292.

Brihaye, Thomas, Gilles Geeraerts, Axel Haddad, and Benjamin Monmege (2016). "Pseudopolynomial Iterative Algorithm to Solve Total-Payoff Games and Min-Cost Reachability Games". In: *Acta Informatica*. DOI: 10.1007/s00236-016-0276-z.

References IV

Brihaye, Thomas, Gilles Geeraerts, Shankara Narayanan Krishna, Lakshmi Manasa, Benjamin Monmege, and Ashutosh Trivedi (2014). "Adding Negative Prices to Priced Timed Games". In: *Proceedings of the 25th International Conference on Concurrency Theory (CONCUR'14)*. Vol. 8704. Springer, pp. 560–575. DOI: 10.1007/978-3-662-44584-6_38.

Busatto-Gaston, Damien, Benjamin Monmege, and Pierre-Alain Reynier (Apr. 2017). "Optimal Reachability in Divergent Weighted Timed Games". In: *Proceedings of the 20th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS'17)*. Ed. by Javier Esparza and Andrzej S. Murawski. Vol. 10203. Lecture Notes in Computer Science. Uppsala, Sweden: Springer, pp. 162–178. DOI: 10.1007/978-3-662-54458-7_10.

References V

Fearnley, John, Rasmus Ibsen-Jensen, and Rahul Savani (2020). "One-Clock Priced Timed Games are PSPACE-hard". In: *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Sciences (LICS'20)*. ACM, pp. 397–409. DOI: 10.1145/3373718.3394772.

Fearnley, John and Marcin Jurdziński (2013). "Reachability in Two-Clock Timed Automata Is PSPACE-Complete". In: *Proceedings of ICALP'13*. Vol. 7966. Lecture Notes in Computer Science. Springer, pp. 212–223.

Haase, Christoph, Joël Ouaknine, and James Worrell (2012). "On the Relationship Between Reachability Problems in Timed and Counter Automata". In: *Proceedings of RP'12*, pp. 54–65.

Hansen, Thomas Dueholm, Rasmus Ibsen-Jensen, and Peter Bro Miltersen (2013). "A Faster Algorithm for Solving One-Clock Priced Timed Games". In: *Proceedings of the 24th International Conference on Concurrency Theory (CONCUR'13)*. Vol. 8052. LNCS. Springer, pp. 531–545.

References VI

- Khachiyan, Leonid, Endre Boros, Konrad Borys, Khaled Elbassioni, Vladimir Gurvich, Gabor Rudolf, and Jihui Zhao (2008). "On Short Paths Interdiction Problems: Total and Node-Wise Limited Interdiction". In: *Theory of Computing Systems* 43.2, pp. 204–233. DOI: 10.1007/s00224-007-9025-6.
- Monmege, Benjamin, Julie Parreaux, and Pierre-Alain Reynier (Sept. 2020). "Reaching Your Goal Optimally by Playing at Random with No Memory". In: *Proceedings of the 31st International Conference on Concurrency Theory (CONCUR 2020)*. Ed. by Igor Konnov and Laura Kovács. Vol. 171. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 26:1–26:21. DOI: 10.4230/LIPIcs.CONCUR.2020.26.
- Monmege, Benjamin, Julie Parreaux, and Pierre-Alain Reynier (2021). "Playing Stochastically in Weighted Timed Games to Emulate Memory". In: *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*. Ed. by Nikhil Bansal, Emanuela Merelli, and James Worrell. Vol. 198. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 137:1–137:17. DOI: 10.4230/LIPIcs.ICALP.2021.137.

References VII

Rutkowski, Michał (2011). "Two-Player Reachability-Price Games on Single-Clock Timed Automata". In: *Proceedings of the Ninth Workshop on Quantitative Aspects of Programming Languages (QAPL'11)*. Vol. 57. Electronic Proceedings in Theoretical Computer Science, pp. 31–46.