Why Negatively-Priced Timed Games are Hard?

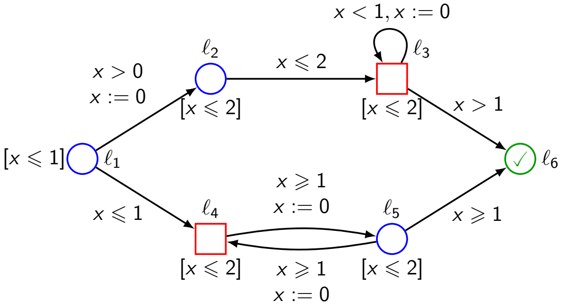
Dagstuhl Seminar on *Non-Zero-Sum Games and Control*

Benjamin Monmege
Université libre de Bruxelles, Belgium

Thomas Brihaye (UMons)
Gilles Geeraerts, Engel Lefaucheux (ULB)
Shankara Krishna, Lakshmi Manasa, Ashutosh Trivedi (IITB)

February 5, 2014

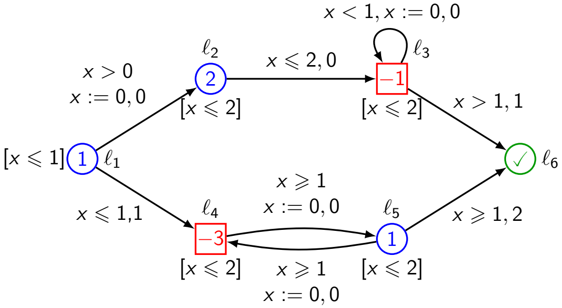
Priced Timed Games



Timed Automaton
with partition of states
between 2 players
+ reachability objective
+ rates in locations
+ costs over transitions

Semantics in terms of
infinite game with weights

Priced Timed Games

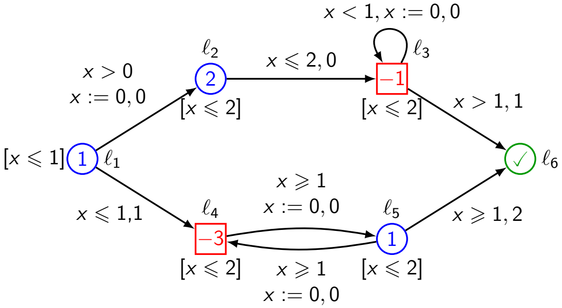


Timed Automaton
with partition of states
between 2 players
+ reachability objective
+ rates in locations
+ costs over transitions

Semantics in terms of
infinite game with weights

$(\ell_1, 0)$

Priced Timed Games

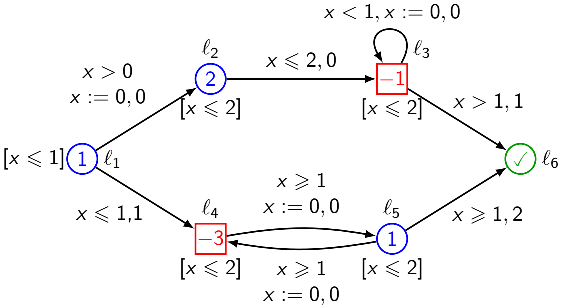


$(\ell_1, 0) \xrightarrow{0.4, \searrow} (\ell_4, 0.4)$

Timed Automaton
with partition of states
between 2 players
+ reachability objective
+ rates in locations
+ costs over transitions

Semantics in terms of
infinite game with weights

Priced Timed Games

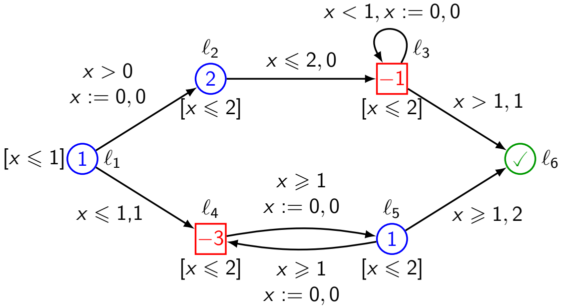


$(\ell_1, 0) \xrightarrow{0.4, \searrow} (\ell_4, 0.4) \xrightarrow{0.6, \rightarrow} (\ell_5, 0)$

Timed Automaton
with partition of states
between 2 players
+ reachability objective
+ rates in locations
+ costs over transitions

Semantics in terms of
infinite game with weights

Priced Timed Games

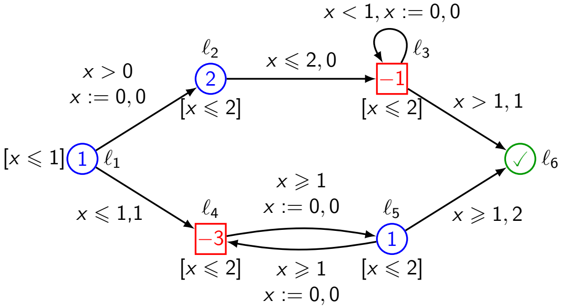


Timed Automaton
with partition of states
between 2 players
+ reachability objective
+ rates in locations
+ costs over transitions

Semantics in terms of
infinite game with weights

$(\ell_1, 0) \xrightarrow{0.4, \searrow} (\ell_4, 0.4) \xrightarrow{0.6, \rightarrow} (\ell_5, 0) \xrightarrow{1.5, \leftarrow} (\ell_4, 0) \xrightarrow{1.1, \rightarrow} (\ell_5, 0) \xrightarrow{2, \nearrow} (\checkmark, 2)$

Priced Timed Games



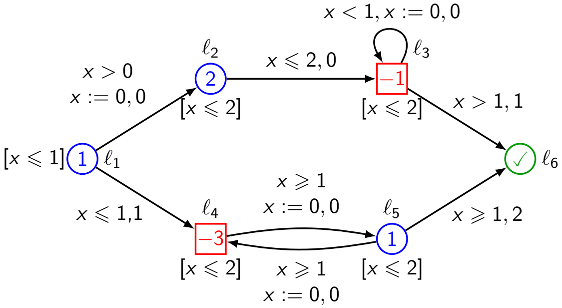
Timed Automaton
with partition of states
between 2 players
+ reachability objective
+ rates in locations
+ costs over transitions

Semantics in terms of
infinite game with weights

$$(\ell_1, 0) \xrightarrow{0.4, \searrow} (\ell_4, 0.4) \xrightarrow{0.6, \rightarrow} (\ell_5, 0) \xrightarrow{1.5, \leftarrow} (\ell_4, 0) \xrightarrow{1.1, \rightarrow} (\ell_5, 0) \xrightarrow{2, \nearrow} (\checkmark, 2)$$

$$0.4 + 1 \quad -3 \times 0.6 \quad +1.5 \quad -3 \times 1.1 \quad +2 \times 2 + 2 \quad = 3.8$$

Priced Timed Games



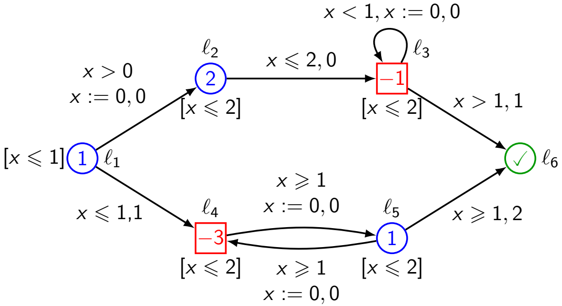
Timed Automaton
with partition of states
between 2 players
+ reachability objective
+ rates in locations
+ costs over transitions

Semantics in terms of
infinite game with weights

$$(\ell_1,0)\xrightarrow[0.4+1]{0.4,\searrow}(\ell_4,0.4)\xrightarrow[-3\times 0.6]{0.6,\rightarrow}(\ell_5,0)\xrightarrow[+1.5]{1.5,\leftarrow}(\ell_4,0)\xrightarrow[-3\times 1.1]{1.1,\rightarrow}(\ell_5,0)\xrightarrow[+2\times 2+2]{2,\nearrow}(\checkmark,2) \quad = 3.8$$

$$(\ell_1,0)\xrightarrow[0.2]{0.2,\nearrow}(\ell_2,0)\xrightarrow[+0.9]{0.9,\rightarrow}(\ell_3,0.9)\xrightarrow[-0.2]{0.2,\circlearrowleft}(\ell_3,0)\xrightarrow[-0.9]{0.9,\circlearrowleft}(\ell_3,0)\quad \cdots \quad = +\infty \ (\checkmark \text{ not reached})$$

Priced Timed Games



Timed Automaton
with partition of states
between 2 players
+ reachability objective
+ rates in locations
+ costs over transitions

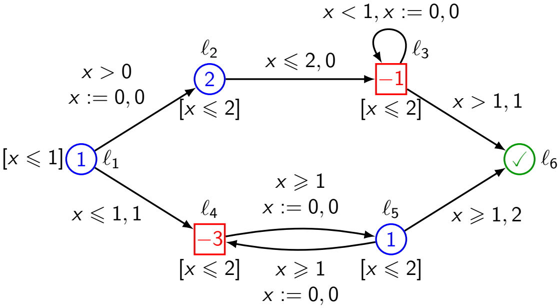
Semantics in terms of
infinite game with weights

$$(\ell_1,0)\xrightarrow[0.4+1]{0.4,\searrow}(\ell_4,0.4)\xrightarrow[-3\times 0.6]{0.6,\rightarrow}(\ell_5,0)\xrightarrow[+1.5]{1.5,\leftarrow}(\ell_4,0)\xrightarrow[-3\times 1.1]{1.1,\rightarrow}(\ell_5,0)\xrightarrow[+2\times 2+2]{2,\nearrow}(\checkmark,2) \quad = 3.8$$

$$(\ell_1,0)\xrightarrow[0.2]{0.2,\nearrow}(\ell_2,0)\xrightarrow[+0.9]{0.9,\rightarrow}(\ell_3,0.9)\xrightarrow[-0.2]{0.2,\circlearrowleft}(\ell_3,0)\xrightarrow[-0.9]{0.9,\circlearrowleft}(\ell_3,0)\quad \cdots \quad = +\infty \text{ } (\checkmark \text{ not reached})$$

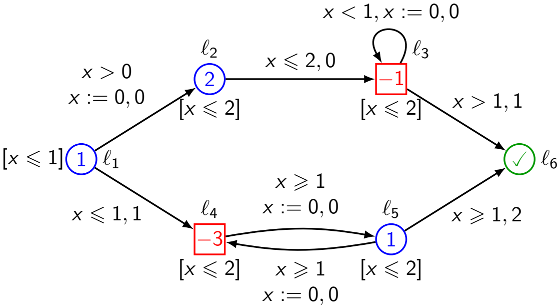
Cost of a play: $\begin{cases} +\infty & \text{if } \checkmark \text{ not reached} \\ \text{total payoff up to } \checkmark & \text{otherwise} \end{cases}$

Strategies and objectives



Strategy for each player: mapping of finite runs to a delay and an action

Strategies and objectives

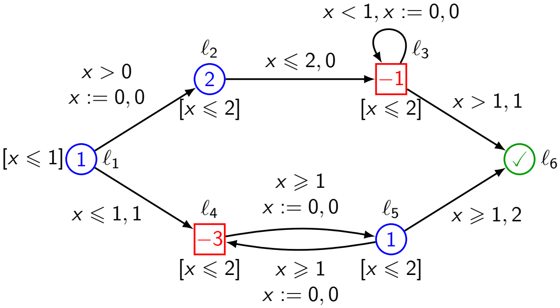


Strategy for each player: mapping of finite runs to a delay and an action

Goal of player ○: reach ✓ **and** minimize the cost

Goal of player □: avoid ✓ **or, if not possible,** maximize the cost

Strategies and objectives



Strategy for each player: mapping of finite runs to a delay and an action

Goal of player ○: reach ✓ **and** minimize the cost
Goal of player □: avoid ✓ **or, if not possible,** maximize the cost

Main object of interest:

$$\overline{\mathsf{Val}}(\ell, v) = \inf_{\sigma_\bigcirc \in \mathsf{Strat}_\bigcirc} \sup_{\sigma_\square \in \mathsf{Strat}_\square} \mathsf{Wt}(\mathsf{Play}((\ell, v), \sigma_\bigcirc, \sigma_\square)) \in \mathbf{R} \cup \{-\infty, +\infty\}$$

What player ○ can guarantee as a payoff? and design *good* strategies

State of the art

$\mathsf{F}_{\leqslant K}$✓: $\exists$ a strategy in the PTG (priced timed game) for player ○ reaching ✓ with a cost $\leqslant K$?

State of the art

$F_{\leqslant K}\checkmark$: $\exists$ a strategy in the PTG (priced timed game) for player ○ reaching ✓ with a cost $\leqslant K$?

- One-player case (**Priced timed automata**): optimal reachability problem is PSPACE-complete
  - Algorithm based on regions [Bouyer, Brihaye, Bruyère, and Raskin, 2007];
  - and hardness shown for timed automata with at least 2 clocks [Fearnley and Jurdziński, 2013, Haase, Ouaknine, and Worrell, 2012]

State of the art

$\mathsf{F}_{\leqslant K}\checkmark$: $\exists$ a strategy in the PTG (priced timed game) for player ○ reaching ✓ with a cost $\leqslant K$?

- One-player case (**Priced timed automata**): optimal reachability problem is PSPACE-complete
  - Algorithm based on regions [Bouyer, Brihaye, Bruyère, and Raskin, 2007];
  - and hardness shown for timed automata with at least 2 clocks [Fearnley and Jurdziński, 2013, Haase, Ouaknine, and Worrell, 2012]
- 2-player PTGs: undecidable [Brihaye, Bruyère, and Raskin, 2005, Bouyer, Brihaye, and Markey, 2006a], even with only non-negative costs and 3 clocks

State of the art

$F_{\leqslant K}\checkmark$: $\exists$ a strategy in the PTG (priced timed game) for player ○ reaching ✓ with a cost $\leqslant K$?

- One-player case (**Priced timed automata**): optimal reachability problem is PSPACE-complete
  - Algorithm based on regions [Bouyer, Brihaye, Bruyère, and Raskin, 2007];
  - and hardness shown for timed automata with at least 2 clocks [Fearnley and Jurdziński, 2013, Haase, Ouaknine, and Worrell, 2012]
- 2-player PTGs: undecidable [Brihaye, Bruyère, and Raskin, 2005, Bouyer, Brihaye, and Markey, 2006a], even with only non-negative costs and 3 clocks
- PTGs with **non-negative costs and strictly non-Zeno cost cycles**: exponential algorithm [Bouyer, Cassez, Fleury, and Larsen, 2004, Alur, Bernadsky, and Madhusudan, 2004]

State of the art

$F_{\leqslant K}\checkmark$: $\exists$ a strategy in the PTG (priced timed game) for player ○ reaching ✓ with a cost $\leqslant K$?

- One-player case (**Priced timed automata**): optimal reachability problem is PSPACE-complete
  - Algorithm based on regions [Bouyer, Brihaye, Bruyère, and Raskin, 2007];
  - and hardness shown for timed automata with at least 2 clocks [Fearnley and Jurdziński, 2013, Haase, Ouaknine, and Worrell, 2012]
- 2-player PTGs: undecidable [Brihaye, Bruyère, and Raskin, 2005, Bouyer, Brihaye, and Markey, 2006a], even with only non-negative costs and 3 clocks
- PTGs with **non-negative costs and strictly non-Zeno cost cycles**: exponential algorithm [Bouyer, Cassez, Fleury, and Larsen, 2004, Alur, Bernadsky, and Madhusudan, 2004]
- **One-clock** PTGs with **non-negative costs**: exponential algorithm [Bouyer, Larsen, Markey, and Rasmussen, 2006b, Rutkowski, 2011, Hansen, Ibsen-Jensen, and Miltersen, 2013]

State of the art

$F_{\leqslant K}$✓: ∃ a strategy in the PTG (priced timed game) for player ○ reaching ✓ with a cost $\leqslant K$?

- One-player case (**Priced timed automata**): optimal reachability problem is PSPACE-complete
  - Algorithm based on regions [Bouyer, Brihaye, Bruyère, and Raskin, 2007];
  - and hardness shown for timed automata with at least 2 clocks [Fearnley and Jurdziński, 2013, Haase, Ouaknine, and Worrell, 2012]
- 2-player PTGs: undecidable [Brihaye, Bruyère, and Raskin, 2005, Bouyer, Brihaye, and Markey, 2006a], even with only non-negative costs and 3 clocks
- PTGs with **non-negative costs and strictly non-Zeno cost cycles**: exponential algorithm [Bouyer, Cassez, Fleury, and Larsen, 2004, Alur, Bernadsky, and Madhusudan, 2004]
- **One-clock** PTGs with **non-negative costs**: exponential algorithm [Bouyer, Larsen, Markey, and Rasmussen, 2006b, Rutkowski, 2011, Hansen, Ibsen-Jensen, and Miltersen, 2013]

This talk: **PTGs with negative costs**

Undecidability results: less clocks...

- Known: $F_{\leqslant K}$✓ undecidable for 3 or more clocks

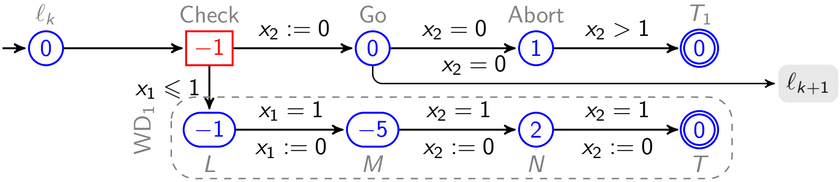
Proof by reduction of 2-counter machines: $x_1 = \frac{1}{2^{c_1}}$, $x_2 = \frac{1}{3^{c_2}}$, x_3 for work

Theorem:

$F_{\leqslant K}$✓ undecidable for PTGs with 2 or more clocks
idem for $F_{\geqslant K}$✓, $F_{>K}$✓, $F_{=K}$✓, $F_{<K}$✓

New encoding: $x_1 = \frac{1}{5^{c_1} 7^{c_2}}$, x_2 for work

Simulation of "ℓ_k : `decrement` c_1; `goto` ℓ_{k+1}" for Reach(= 1)



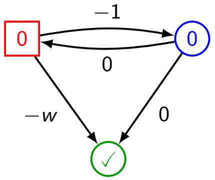
Regain decidability
in the presence of negative prices?

More complex when negative costs

- Value $-\infty$: detection is as hard as mean-payoff. No hope for complexity better than $\mathbf{NP} \cap \mathbf{co\text{-}NP}$, or pseudo-polynomial

More complex when negative costs

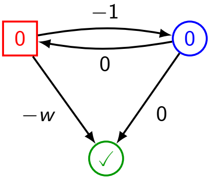
- Value $-\infty$: detection is as hard as mean-payoff. No hope for complexity better than **NP** $\cap$ **co-NP**, or pseudo-polynomial
- Memory complexity
  - Player ○ needs memory, even in the untimed setting



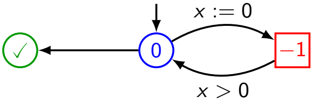
More complex when negative costs

- Value $-\infty$: detection is as hard as mean-payoff. No hope for complexity better than **NP** ∩ **co-NP**, or pseudo-polynomial

- Memory complexity
  - Player ○ needs memory, even in the untimed setting



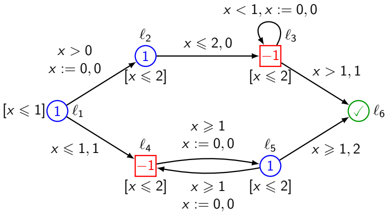
  - Player □ may require infinite memory



Building over the corner-point abstraction

Only known decidable fragment with negative weights: 1-player case (priced timed automata)

- Main tool: refinement of regions similar to corner point abstraction

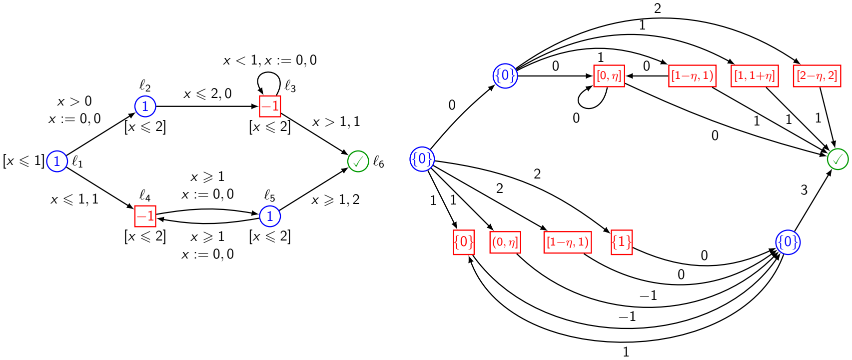


regions: $\{0\}, (0, 1), \{1\}, (1, 2), \{2\}, (2, +\infty)$
regions refined with corner information:
$\{0\}, (0, \eta), (1 - \eta, 1), \{1\}, (1, 1 + \eta), (2 - \eta, 2), \{2\}, (2, +\infty)$

Building over the corner-point abstraction

Only known decidable fragment with negative weights: 1-player case (priced timed automata)

- Main tool: refinement of regions similar to corner point abstraction

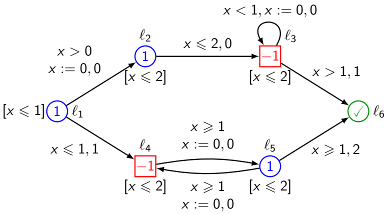


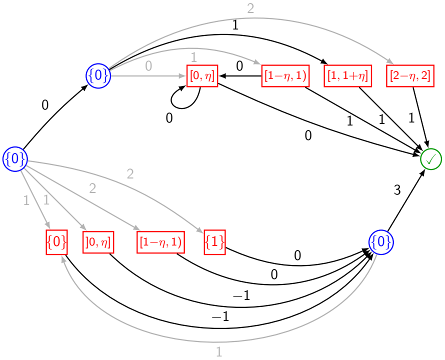
regions: $\{0\}, (0, 1), \{1\}, (1, 2), \{2\}, (2, +\infty)$
regions refined with corner information:
$\{0\}, (0, \eta), (1 - \eta, 1), \{1\}, (1, 1 + \eta), (2 - \eta, 2), \{2\}, (2, +\infty)$

Building over the corner-point abstraction

Only known decidable fragment with negative weights: 1-player case (priced timed automata)

- Main tool: refinement of regions similar to corner point abstraction

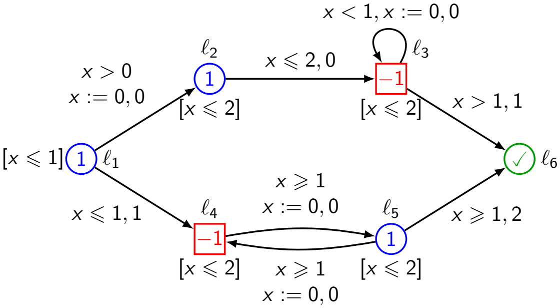




regions: $\{0\}, (0,1), \{1\}, (1,2), \{2\}, (2,+\infty)$
regions refined with corner information:
$\{0\}, (0,\eta), (1-\eta,1), \{1\}, (1,1+\eta), (2-\eta,2), \{2\}, (2,+\infty)$

One-clock Bi-Valued PTGs (1BPTGs)

Assumption: rates of locations $\{p^-, p^+\}$ included in $\{0, +d, -d\}$ ($d \in \mathbf{N}$) (no assumption on costs of transitions)



- Techniques of [Bouyer, Cassez, Fleury, and Larsen, 2004, Alur, Bernadsky, and Madhusudan, 2004] not applicable, e.g., because of Zeno costs cycles
- Exponential algorithms of [Bouyer, Larsen, Markey, and Rasmussen, 2006b, Rutkowski, 2011, Hansen, Ibsen-Jensen, and Miltersen, 2013] not working because of presence of negative costs

Results

Theorem:

- Computation of the value $\overline{\mathsf{Val}}(\ell, v)$ of states of a 1BPTG in pseudo-polynomial time
- Synthesis of ε-optimal strategies for player $\bigcirc$ in pseudo-polynomial time

Theorem: Non-negative case

In case of a 1BPTG with only non-negative costs, all complexities drop down to polynomial.

First idea: symmetrize the viewpoint

Value for player $\bigcirc$: $\overline{\mathsf{Val}}(\ell, v) = \inf_{\sigma_\bigcirc \in \mathsf{Strat}_\bigcirc} \sup_{\sigma_\square \in \mathsf{Strat}_\square} \mathsf{Wt}(\mathsf{Play}((\ell, v), \sigma_\bigcirc, \sigma_\square))$

Value for player $\square$: $\underline{\mathsf{Val}}(\ell, v) = \sup_{\sigma_\square \in \mathsf{Strat}_\square} \inf_{\sigma_\bigcirc \in \mathsf{Strat}_\bigcirc} \mathsf{Wt}(\mathsf{Play}((\ell, v), \sigma_\bigcirc, \sigma_\square))$

How to compare them? $\underline{\mathsf{Val}}(\ell, v) \leqslant \overline{\mathsf{Val}}(\ell, v)$

First idea: symmetrize the viewpoint

Value for player $\bigcirc$: $\overline{\mathsf{Val}}(\ell, v) = \inf_{\sigma_\bigcirc \in \mathsf{Strat}_\bigcirc} \sup_{\sigma_\square \in \mathsf{Strat}_\square} \mathsf{Wt}(\mathsf{Play}((\ell, v), \sigma_\bigcirc, \sigma_\square))$

Value for player $\square$: $\underline{\mathsf{Val}}(\ell, v) = \sup_{\sigma_\square \in \mathsf{Strat}_\square} \inf_{\sigma_\bigcirc \in \mathsf{Strat}_\bigcirc} \mathsf{Wt}(\mathsf{Play}((\ell, v), \sigma_\bigcirc, \sigma_\square))$

How to compare them? $\underline{\mathsf{Val}}(\ell, v) \leqslant \overline{\mathsf{Val}}(\ell, v)$

Theorem: (continued)

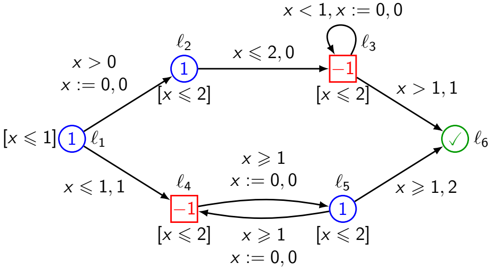
- *Minmax theorem*: 1BPGs are determined, i.e., $\underline{\mathsf{Val}}(\ell, v) = \overline{\mathsf{Val}}(\ell, v)$
- Synthesis of ε-optimal strategies for player $\square$ in pseudo-polynomial time (and polynomial in case of non-negative weights)

Sketch of proof

1. **Reduce the space of strategies in the 1BPTG**
   - restrict to uniform strategies w.r.t. timed behaviors
2. **Build a finite priced game $\mathcal{G}$**
   - based on a refinement of the region abstraction
3. **Study $\mathcal{G}$**
4. **Lift results of $\mathcal{G}$ to the original 1BPTG**

1. Reduce the space of strategies

Intuition: no need for both players to play far from borders of regions



Regions:
$\{0\}, (0,1), \{1\}, (1,2), \{2\}, (2,+\infty)$

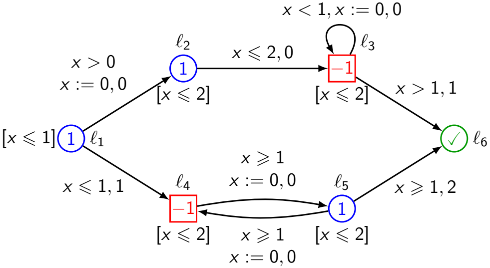
Player ○ wants to leave as soon as possible a state with rate p^+, and wants to stay as long as possible in a state with rate p^-: so, he will always play η-close to a border...

Lemma:

Both players can play arbitrarily close to borders w.l.o.g.: for every η

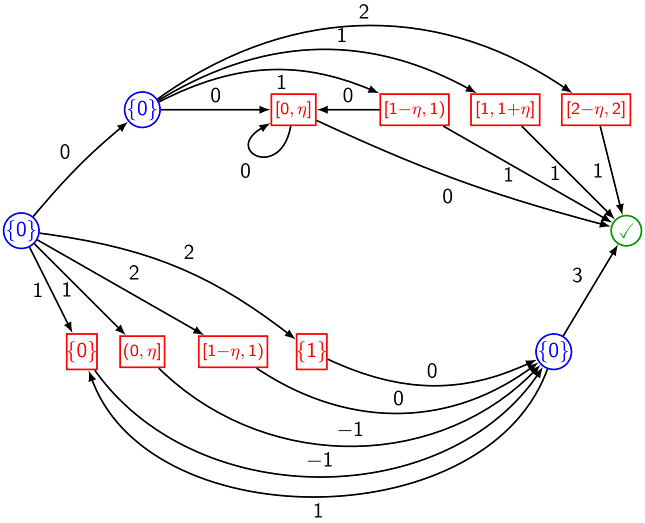
$$\underline{\mathsf{Val}}^\eta(\ell, v) \leqslant \underline{\mathsf{Val}}(\ell, v) \quad \leqslant \quad \overline{\mathsf{Val}}(\ell, v) \leqslant \overline{\mathsf{Val}}^\eta(\ell, v)$$

2. Finite priced game abstraction



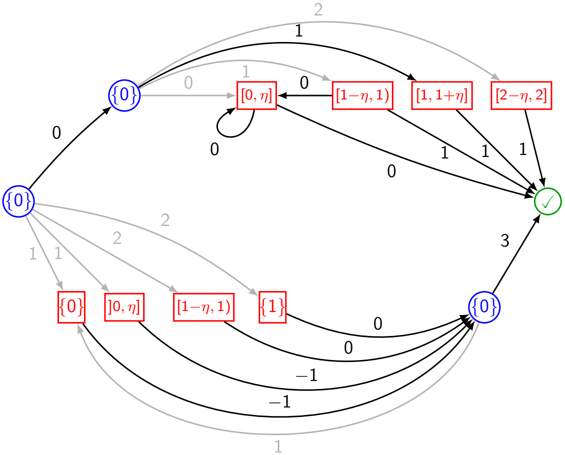
η-regions: $\{0\}, (0, \eta), (1-\eta, 1), \{1\}, (1, 1+\eta), (2-\eta, 2), \{2\}, (2, +\infty)$

2. Finite priced game abstraction



3. Study $\mathcal{G}$: values, optimal strategies of a min-cost reachability game

[Brihaye, Geeraerts, Haddad, and Monmege, 2014]



Optimal value: $\text{Val}_{\mathcal{G}}(\ell_1, \{0\}) = +2$ (for both players)

4. Lift results to the original 1BPTG

Reconstruct strategies in the 1BPTG from optimal strategies of $\mathcal{G}$

Lemma:

For all $\varepsilon > 0$, there exists $\eta > 0$ such that:

$$\mathsf{Val}_{\mathcal{G}}(\ell, \{0\}) - \varepsilon \leqslant \underline{\mathsf{Val}}^{\eta}(\ell, 0) \leqslant \underline{\mathsf{Val}}(\ell, 0) \leqslant \overline{\mathsf{Val}}(\ell, 0) \leqslant \overline{\mathsf{Val}}^{\eta}(\ell, 0) \leqslant \mathsf{Val}_{\mathcal{G}}(\ell, \{0\}) + \varepsilon$$

4. Lift results to the original 1BPTG

Reconstruct strategies in the 1BPTG from optimal strategies of $\mathcal{G}$

Lemma:

For all $\varepsilon > 0$, there exists $\eta > 0$ such that:

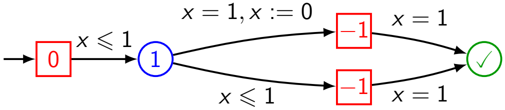
$\mathsf{Val}_{\mathcal{G}}(\ell, \{0\}) - \varepsilon \leqslant \underline{\mathsf{Val}}^{\eta}(\ell, 0) \leqslant \underline{\mathsf{Val}}(\ell, 0) \leqslant \overline{\mathsf{Val}}(\ell, 0) \leqslant \overline{\mathsf{Val}}^{\eta}(\ell, 0) \leqslant \mathsf{Val}_{\mathcal{G}}(\ell, \{0\}) + \varepsilon$

- So $\underline{\mathsf{Val}}(\ell, 0) = \overline{\mathsf{Val}}(\ell, 0)$, i.e., determination
- ε-optimal strategies for both players
  - Finite memory for player ○ (finite memory in finite priced games)
  - Infinite memory for player □ (even though memoryless in finite priced games), because it needs to ensure convergence of its differences between the 1BPTG and $\mathcal{G}$
- Overall complexity: pseudo-polynomial (polynomial if non-negative costs) in the size of $\mathcal{G}$, which is polynomial in the 1BPTG (because 1 clock)

1BPTG: maximal fragment for corner-point abstraction

Players may need to play far from corners...

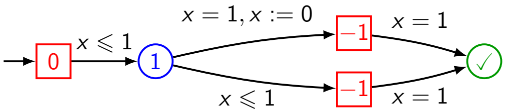
- With 3 weights in $\{-1, 0, +1\}$: value $1/2$...



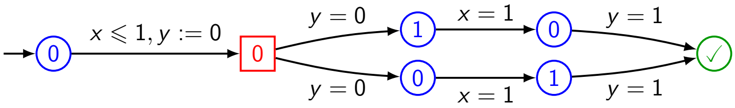
1BPTG: maximal fragment for corner-point abstraction

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 $1/2$...



Inspired by other previous techniques for 1-clock PTGs?

[Hansen, Ibsen-Jensen, and Miltersen, 2013]: strategy improvement algorithm

[Bouyer, Larsen, Markey, and Rasmussen, 2006b, Rutkowski, 2011]: iterative elimination of locations

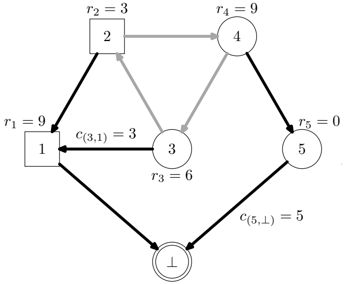
- precomputation: polynomial-time cascade of simplification of 1-clock PTGs into simple 1-clock PTGs (SPTGs)
  - clock bounded by 1, no guards/invariants, no resets

Inspired by other previous techniques for 1-clock PTGs?

[Hansen, Ibsen-Jensen, and Miltersen, 2013]: strategy improvement algorithm

[Bouyer, Larsen, Markey, and Rasmussen, 2006b, Rutkowski, 2011]: iterative elimination of locations

- precomputation: polynomial-time cascade of simplification of 1-clock PTGs into simple 1-clock PTGs (SPTGs)
  - clock bounded by 1, no guards/invariants, no resets

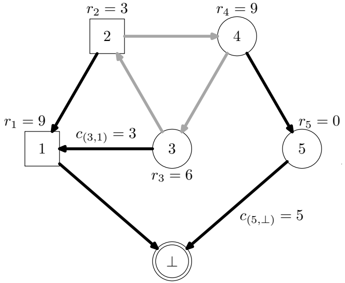


Inspired by other previous techniques for 1-clock PTGs?

[Hansen, Ibsen-Jensen, and Miltersen, 2013]: strategy improvement algorithm

[Bouyer, Larsen, Markey, and Rasmussen, 2006b, Rutkowski, 2011]: iterative elimination of locations

- precomputation: polynomial-time cascade of simplification of 1-clock PTGs into simple 1-clock PTGs (SPTGs)
  - clock bounded by 1, no guards/invariants, no resets
- for SPTGs: compute value functions $\underline{\mathsf{Val}}(\ell, x) = \overline{\mathsf{Val}}(\ell, x)$, $\forall \ell$ loc.

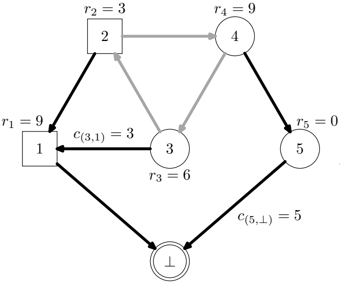


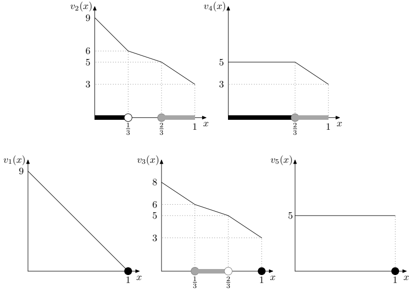
Inspired by other previous techniques for 1-clock PTGs?

[Hansen, Ibsen-Jensen, and Miltersen, 2013]: strategy improvement algorithm

[Bouyer, Larsen, Markey, and Rasmussen, 2006b, Rutkowski, 2011]: iterative elimination of locations

- precomputation: polynomial-time cascade of simplification of 1-clock PTGs into simple 1-clock PTGs (SPTGs)
  - clock bounded by 1, no guards/invariants, no resets
- for SPTGs: compute value functions $\underline{\mathsf{Val}}(\ell, x) = \overline{\mathsf{Val}}(\ell, x)$, $\forall \ell$ loc.



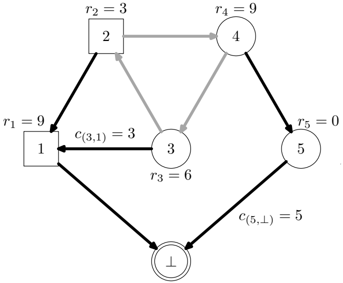


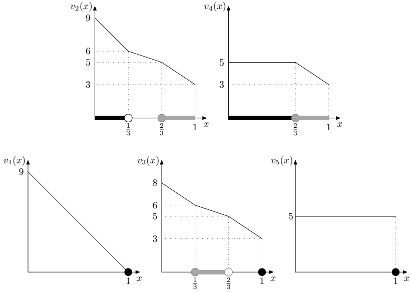
Inspired by other previous techniques for 1-clock PTGs?

[Hansen, Ibsen-Jensen, and Miltersen, 2013]: strategy improvement algorithm

[Bouyer, Larsen, Markey, and Rasmussen, 2006b, Rutkowski, 2011]: iterative elimination of locations

- precomputation: polynomial-time cascade of simplification of 1-clock PTGs into simple 1-clock PTGs (SPTGs)
  - clock bounded by 1, no guards/invariants, no resets
- for SPTGs: compute value functions $\underline{\mathsf{Val}}(\ell, x) = \overline{\mathsf{Val}}(\ell, x)$, $\forall \ell$ loc.





Value functions: continuous, non-increasing, piecewise linear functions with at most exponential number of *event points*

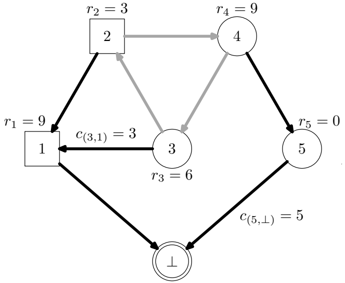
$+\ \varepsilon$-optimal positional strategies

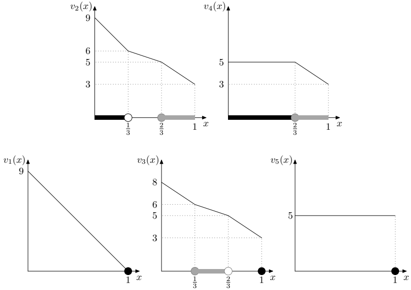
Inspired by other previous techniques for 1-clock PTGs?

[Hansen, Ibsen-Jensen, and Miltersen, 2013]: strategy improvement algorithm

[Bouyer, Larsen, Markey, and Rasmussen, 2006b, Rutkowski, 2011]: iterative elimination of locations

- precomputation: polynomial-time cascade of simplification of 1-clock PTGs into simple 1-clock PTGs (SPTGs) Difficult with negative prices
  - clock bounded by 1, no guards/invariants, no resets
- for SPTGs: compute value functions $\underline{\mathsf{Val}}(\ell, x) = \overline{\mathsf{Val}}(\ell, x)$, $\forall \ell$ loc.





Value functions: continuous, non-increasing, piecewise linear functions with at most exponential number of *event points*

+ ε-optimal positional strategies

Inspired by other previous techniques for 1-clock PTGs?

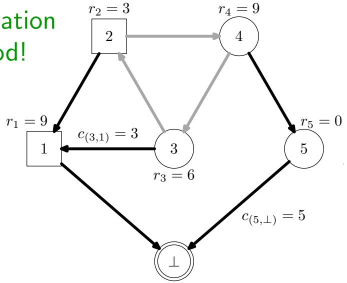
[Hansen, Ibsen-Jensen, and Miltersen, 2013]: strategy improvement algorithm

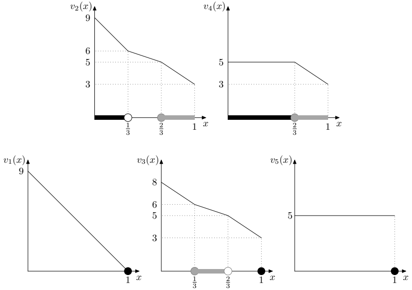
[Bouyer, Larsen, Markey, and Rasmussen, 2006b, Rutkowski, 2011]: iterative elimination of locations

- precomputation: polynomial-time cascade of simplification of 1-clock PTGs into simple 1-clock PTGs (SPTGs)
  - clock bounded by 1, no guards/invariants, no resets
- for SPTGs: compute value functions $\underline{\mathrm{Val}}(\ell, x) = \overline{\mathrm{Val}}(\ell, x)$, $\forall \ell$ loc.

Difficult with negative prices

OK with elimination method!





Value functions: continuous, non-increasing, piecewise linear functions with at most exponential number of *event points*

$+\ \varepsilon$-optimal positional strategies

Inspired by other previous techniques for 1-clock PTGs?

[Hansen, Ibsen-Jensen, and Miltersen, 2013]: strategy improvement algorithm

[Bouyer, Larsen, Markey, and Rasmussen, 2006b, Rutkowski, 2011]: iterative elimination of locations

- precomputation: polynomial-time cascade of simplification of 1-clock PTGs into simple 1-clock PTGs (SPTGs) — Difficult with negative prices
  - clock bounded by 1, no guards/invariants, no resets
- for SPTGs: compute value functions $\underline{\mathsf{Val}}(\ell, x) = \overline{\mathsf{Val}}(\ell, x)$, $\forall \ell$ loc.

OK with elimination method!

One important ingredient

Every PTG is determined.

Proof using the extension of Martin's determinacy result for games with arbitrary branching [Martin, 1990]

$\Rightarrow$ **exponential time algorithm for reset-acyclic 1-clock PTGs (with arbitrary weights)**

Value functions: continuous, non-increasing, piecewise linear functions with at most exponential number of *event points*

$+$ ε-optimal positional strategies

Inspired by other previous techniques for 1-clock PTGs?

[Hansen, Ibsen-Jensen, and Miltersen, 2013]: strategy improvement algorithm

[Bouyer, Larsen, Markey, and Rasmussen, 2006b, Rutkowski, 2011]: iterative elimination of locations

- precomputation: polynomial-time cascade of simplification of 1-clock PTGs into simple 1-clock PTGs (SPTGs) Difficult with negative prices
  - clock bounded by 1, no guards/invariants, no resets
- for SPTGs: compute value functions $\underline{\mathsf{Val}}(\ell, x) = \overline{\mathsf{Val}}(\ell, x)$, $\forall \ell$ loc.

OK with elimination method!

One important ingredient

Every PTG is determined.

Proof using the extension of Martin's determinacy result for games with arbitrary branching [Martin, 1990]

$\Rightarrow$ **exponential time algorithm for reset-acyclic 1-clock PTGs (with arbitrary weights)**

Value functions: continuous, non-increasing, piecewise linear functions with at most exponential number of *event points*

$+$ ε-optimal positional strategies

Inspired by other previous techniques for 1-clock PTGs?

[Hansen, Ibsen-Jensen, and Miltersen, 2013]: strategy improvement algorithm

[Bouyer, Larsen, Markey, and Rasmussen, 2006b, Rutkowski, 2011]: iterative elimination of locations

- precomputation: polynomial-time cascade of simplification of 1-clock PTGs into simple 1-clock PTGs (SPTGs)
  - clock bounded by 1, no guards/invariants, no resets

Difficult with negative prices

- for SPTGs: compute value functions $\underline{\mathsf{Val}}(\ell, x) = \overline{\mathsf{Val}}(\ell, x)$, $\forall \ell$ loc.

OK with elimination method!

One important ingredient

Every PTG is determined.

Proof using the extension of Martin's determinacy result for games with arbitrary branching [Martin, 1990]

$\Rightarrow$ **exponential time algorithm for reset-acyclic 1-clock PTGs (with arbitrary weights)**

Value functions: continuous, non-increasing, piecewise linear functions with at most exponential number of *event points*

$+$ ε-optimal positional strategies

Inspired by other previous techniques for 1-clock PTGs?

[Hansen, Ibsen-Jensen, and Miltersen, 2013]: strategy improvement algorithm

[Bouyer, Larsen, Markey, and Rasmussen, 2006b, Rutkowski, 2011]: iterative elimination of locations

- precomputation: polynomial-time cascade of simplification of 1-clock PTGs into simple 1-clock PTGs (SPTGs) Difficult with negative prices
  - clock bounded by 1, no guards/invariants, no resets
- for SPTGs: compute value functions $\underline{\text{Val}}(\ell, x) = \overline{\text{Val}}(\ell, x)$, $\forall \ell$ loc.

OK with elimination method!

One important ingredient

Every PTG is determined.

Proof using the extension of Martin's determinacy result for games with arbitrary branching [Martin, 1990]

$\Rightarrow$ **exponential time algorithm for reset-acyclic 1-clock PTGs (with arbitrary weights)**

Value functions: continuous, non-increasing, piecewise linear functions with at most exponential number of *event points* ??

+ ε-optimal positional strategies

Inspired by other previous techniques for 1-clock PTGs?

[Hansen, Ibsen-Jensen, and Miltersen, 2013]: strategy improvement algorithm

[Bouyer, Larsen, Markey, and Rasmussen, 2006b, Rutkowski, 2011]: iterative elimination of locations

- precomputation: polynomial-time cascade of simplification of 1-clock PTGs into simple 1-clock PTGs (SPTGs) Difficult with
  - clock bounded by 1, no guards/invariants, no resets negative prices
- for SPTGs: compute value functions $\underline{\mathsf{Val}}(\ell, x) = \overline{\mathsf{Val}}(\ell, x)$, $\forall \ell$ loc.

OK with elimination method!

One important ingredient

Every PTG is determined.

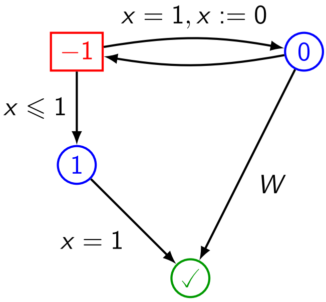
Proof using the extension of Martin's determinacy result for games with arbitrary branching [Martin, 1990]

$\Rightarrow$ **exponential time algorithm for reset-acyclic 1-clock PTGs (with arbitrary weights)**

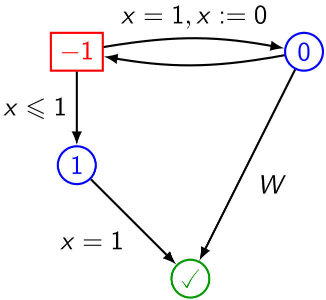
Value functions: continuous, non-increasing, piecewise linear functions with at most exponential number of *event points* ??

+ ε-optimal positional strategies Do not always exist!

ε-optimal positional strategies do not always exist

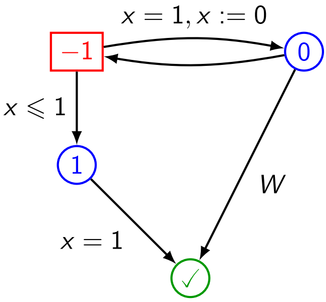


ε-optimal positional strategies do not always exist



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

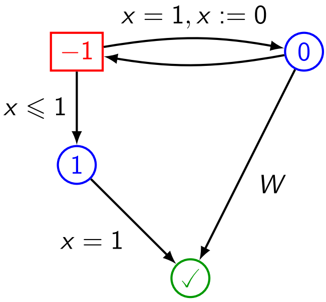
ε-optimal positional strategies do not always exist



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

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

ε-optimal positional strategies do not always exist



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

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

... yet, to obtain ε, ○ needs to loop at most $W \cdot \lceil 1/\varepsilon \rceil - 1$ times before reaching ✓!

Summary and Future Work

Results

- Refined undecidability results due to the presence of negative costs
- Decidable fragment of 1BPTGs in pseudo-polynomial time, with finite memory for ○, infinite memory for □. Obtained by lifting of *corner point abstraction* to game setting
- Extension of iterative elimination for reset-acyclic 1-clock PTGs

Summary and Future Work

Results

- Refined undecidability results due to the presence of negative costs
- Decidable fragment of 1BPTGs in pseudo-polynomial time, with finite memory for ○, infinite memory for □. Obtained by lifting of *corner point abstraction* to game setting
- Extension of iterative elimination for reset-acyclic 1-clock PTGs

- Ongoing work: final extension of the result for all 1-clock PTGs
  - How to deal with resets in presence of negative prices?
  - With which complexity?
- Implementation and test of different algorithms on real instances

Summary and Future Work

Results

- Refined undecidability results due to the presence of negative costs
- Decidable fragment of 1BPTGs in pseudo-polynomial time, with finite memory for ○, infinite memory for □. Obtained by lifting of *corner point abstraction* to game setting
- Extension of iterative elimination for reset-acyclic 1-clock PTGs

- Ongoing work: final extension of the result for all 1-clock PTGs
  - How to deal with resets in presence of negative prices?
  - With which complexity?
- Implementation and test of different algorithms on real instances

Thank you for your attention

References I

Rajeev Alur, Mikhail Bernadsky, and P. Madhusudan. Optimal reachability for weighted timed games. In *Proceedings of the 31st International Colloquium on Automata, Languages and Programming (ICALP'04)*, volume 3142 of *Lecture Notes in Computer Science*, pages 122–133. Springer, 2004.

Patricia Bouyer, Franck Cassez, Emmanuel Fleury, and Kim G. Larsen. Optimal strategies in priced timed game automata. In *Proceedings of the 24th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'04)*, volume 3328 of *Lecture Notes in Computer Science*, pages 148–160. Springer, 2004.

Patricia Bouyer, Thomas Brihaye, and Nicolas Markey. Improved undecidability results on weighted timed automata. *Information Processing Letters*, 98(5):188–194, 2006a.

Patricia Bouyer, Kim G. Larsen, Nicolas Markey, and Jacob Illum Rasmussen. 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)*, volume 4337 of *Lecture Notes in Computer Science*, pages 345–356. Springer, 2006b.

Patricia Bouyer, Thomas Brihaye, Véronique Bruyère, and Jean-François Raskin. On the optimal reachability problem of weighted timed automata. *Formal Methods in System Design*, 31(2):135–175, 2007.

References II

Thomas Brihaye, Véronique Bruyère, and Jean-François Raskin. On optimal timed strategies. In *Proceedings of the Third international conference on Formal Modeling and Analysis of Timed Systems (FORMATS'05)*, volume 3829 of *Lecture Notes in Computer Science*, pages 49–64. Springer, 2005.

Thomas Brihaye, Gilles Geeraerts, Axel Haddad, and Benjamin Monmege. To reach or not to reach? Efficient algorithms for total-payoff games. Research Report 1407.5030, arXiv, July 2014. URL `http://arxiv.org/abs/1407.5030`.

John Fearnley and Marcin Jurdziński. Reachability in two-clock timed automata is PSPACE-complete. In *Proceedings of ICALP'13*, volume 7966 of *Lecture Notes in Computer Science*, pages 212–223. Springer, 2013.

Christoph Haase, Joël Ouaknine, and James Worrell. On the relationship between reachability problems in timed and counter automata. In *Proceedings of RP'12*, pages 54–65, 2012.

Thomas Dueholm Hansen, Rasmus Ibsen-Jensen, and Peter Bro Miltersen. A faster algorithm for solving one-clock priced timed games. In *Proceedings of the 24th International Conference on Concurrency Theory (CONCUR'13)*, volume 8052 of *Lecture Notes in Computer Science*, pages 531–545. Springer, 2013.

Donald A. Martin. An extension of borel determinacy. *Annals of Pure and Applied Logic*, 49:279–293, 1990.

Michał Rutkowski. Two-player reachability-price games on single-clock timed automata. In *Proceedings of the Ninth Workshop on Quantitative Aspects of Programming Languages (QAPL'11)*, volume 57 of *Electronic Proceedings in Theoretical Computer Science*, pages 31–46, 2011.