Simple Priced Timed Games are not that simple

FSTTCS 2015, Bangalore

Benjamin Monmege
Aix-Marseille Université, LIF, France

Thomas Brihaye (UMons), Gilles Geeraerts (ULB), Engel Lefaucheux (ENS Cachan), Axel Haddad (UMons)

December 17, 2015
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

Cost of a play:
{\begin{align*}
 & +\infty \quad \text{if not reached} \\
 & \text{total payoff up to} \quad \text{otherwise}
\end{align*}}
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**

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

\[
(\ell_1, 0) \xrightarrow{0.4, \leftarrow} (\ell_4, 0.4)
\]
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

Cost of a play:
\{ \begin{align*}
\infty & \text{ if not reached} \\
\text{total payoff up to otherwise}
\end{align*} \}

\[ (\ell_1, 0) \xrightarrow{0.4} (\ell_4, 0.4) \xrightarrow{0.6} (\ell_5, 0) \]
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

\((\ell_1, 0) \xrightarrow{0.4, \downarrow} (\ell_4, 0.4) \xrightarrow{0.6, \rightarrow} (\ell_5, 0) \xrightarrow{1.5, \leftarrow} (\ell_4, 0) \xrightarrow{1.1, \rightarrow} (\ell_5, 0) \xrightarrow{2, \uparrow} (\checkmark, 2)\)
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

\[
\begin{align*}
(l_1, 0) &\xrightarrow{0.4} (l_4, 0.4) \xrightarrow{0.6} (l_5, 0) \xrightarrow{1.5} (l_4, 0) \xrightarrow{1.1} (l_5, 0) \xrightarrow{2} (\checkmark, 2) \\
0.4 + 1 &\quad -3 \times 0.6 \quad +1.5 \quad -3 \times 1.1 \quad +2 \times 2 + 2 \quad = 3.8
\end{align*}
\]
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

\[
\begin{align*}
\ell_1 & \xrightarrow{(0.4, \rightarrow)} \ell_4 \xrightarrow{(0.4)} \ell_5 \xrightarrow{1.5, \leftarrow} (\ell_4, 0) \xrightarrow{1.1} (\ell_5, 0) \xrightarrow{2, \rightarrow} (\checkmark, 2) \\
& \quad \text{Cost of a play:} \quad +\infty \text{ if not reached, otherwise total payoff up to}
\end{align*}
\]

\[
\begin{align*}
\ell_1 & \xrightarrow{(0.2, \rightarrow)} \ell_2 \xrightarrow{(0.9)} \ell_3 \xrightarrow{(0.2, \varnothing)} (\ell_3, 0) \xrightarrow{(0.9, \varnothing)} (\ell_3, 0) \xrightarrow{(0.9, \varnothing)} \cdots = +\infty \text{ (\checkmark not reached)}
\end{align*}
\]
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

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

\[
x < 1, x := 0, 0
\]

\[
x > 0, x := 0, 0
\]

\[
x \leq 2, \sigma_1 \\
\lfloor x \leq 2 \rfloor
\]

\[
x \leq 2, \sigma_2 \\
\lfloor x \leq 2 \rfloor
\]

\[
x > 1, 1
\]

\[
x \geq 1, x := 0, 0
\]

\[
x \geq 1, x := 0, 0
\]

\[
x \leq 1, 1
\]

\[
x \leq 1, \sigma_3 \\
\lfloor x \leq 2 \rfloor
\]

\[
x \geq 1, \sigma_4 \\
\lfloor x \geq 1 \rfloor
\]

\[
x \geq 1, \sigma_5 \\
\lfloor x \geq 1 \rfloor
\]

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

Goal of player $\bigcirc$: reach $\checkmark$ and minimize the cost
Goal of player $\Box$: avoid $\checkmark$ or, if not possible, maximize the cost

Main object of interest:
\[
\overline{\text{Val}}(\ell, v) = \inf_{\sigma_\bigcirc \in \text{Strat}_\bigcirc} \sup_{\sigma_\Box \in \text{Strat}_\Box} \text{Wt}(\text{Play}((\ell, v), \sigma_\bigcirc, \sigma_\Box)) \in \mathbb{R} \cup \{-\infty, +\infty\}
\]
\[
\text{Val}(\ell, v) = \sup_{\sigma_\Box \in \text{Strat}_\Box} \inf_{\sigma_\bigcirc \in \text{Strat}_\bigcirc} \text{Wt}(\text{Play}((\ell, v), \sigma_\bigcirc, \sigma_\Box)) \in \mathbb{R} \cup \{-\infty, +\infty\}
\]
What can players guarantee as a payoff? and design good strategies
State of the art

$F_{\leq K} \checkmark$: $\exists$ a strategy in the PTG (priced timed game) for player $\bigcirc$ reaching $\checkmark$ with a cost $\leq K$?
State of the art

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

- 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

\[ F_{\leq K} \supseteq : \exists \text{ a strategy in the PTG (priced timed game) for player } \bigcirc \text{ reaching } \checkmark \text{ with a cost } \leq K? \]

- 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

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

- **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

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

- **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


- **One-clock PTGs with non-negative costs**: exponential algorithm  
State of the art

\[ F_{\leq K} \checkmark : \exists \text{ a strategy in the PTG (priced timed game) for player } \bigcirc \text{ reaching } \checkmark \text{ with a cost } \leq K? \]

- 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

This talk: **PTGs with negative costs**
More complex when negative costs

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

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

- Memory complexity
  - Player $\bigcirc$ needs memory, even in the untimed setting

- Player $\Box$ may require infinite memory
Known results with negative costs [Brihaye, Geeraerts, Krishna, Manasa, Monmege, and Trivedi, 2014]

- $F_{\leq K}$ \checkmark undecidable for 2 or more clocks

Proof by reduction of 2-counter machines.
Known results with negative costs [Brihaye, Geeraerts, Krishna, Manasa, Monmege, and Trivedi, 2014]

- $F_{\leq K}$ undecidable for 2 or more clocks
  Proof by reduction of 2-counter machines.
- Pseudo-polynomial algorithm for One-clock Bi-valued PTG

**Assumption:** rates of locations $\{p^-, p^+\}$ included in $\{0, +d, -d\}$ ($d \in \mathbb{N}$) (no assumption on costs of transitions)
Known results with negative costs [Brihaye, Geeraerts, Krishna, Manasa, Monmege, and Trivedi, 2014]

- $F_{\leq K}$ undecidable for 2 or more clocks

Proof by reduction of 2-counter machines.

- Pseudo-polynomial algorithm for One-clock Bi-valued PTG

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

Method: Corner point abstraction.
Solving min-cost reachability games [Brihaye, Geeraerts, Haddad, and Monmege, 2015]
Solving min-cost reachability games [Brihaye, Geeraerts, Haddad, and Monmege, 2015]
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
- for SPTGs: compute value functions $\overline{\text{Val}}(\ell, x)$.
SPTGs with arbitrary weights

\[ \ell_6 -12 \xrightarrow{1} -2 \xrightarrow{\ell_1} \ell_4 3 \xrightarrow{\ell_4} \text{✓} \]

\[ \ell_5 8 \xrightarrow{-14} \ell_2 \xrightarrow{2} -16 \xrightarrow{\ell_7} -7 \]

\[ \ell_3 4 \xrightarrow{6} \]
SPTGs with arbitrary weights

\[
\text{Val}(\ell_4, x) = 3(1 - x) - 7 = -3x - 4
\]
SPTGs with arbitrary weights

Val(\(\ell_4, x\)) = -3x - 4, \hspace{1cm} Val(\(\ell_7, x\)) = -16(1 - x)
SPTGs with arbitrary weights

\[
\begin{align*}
\text{Val}(\ell_4, x) &= -3x - 4, \\
\text{Val}(\ell_7, x) &= -16(1 - x), \\
\text{Val}(\ell_3, x) &= \inf_{0 \leq t \leq 1 - x} [4t + \min(-3(x + t) - 4, 6 - 16(1 - (x + t)))] = \min(-3x - 4, 16x - 10)
\end{align*}
\]
SPTGs with arbitrary weights
Recursive elimination of states

Intuition from [Bouyer, Larsen, Markey, and Rasmussen, 2006b, Rutkowski, 2011]:

- Player \( \circ \) prefers to stay as long as possible in locations with minimal price: add a final location allowing him to stay until the end, and make the location urgent
Recursive elimination of states

Intuition from [Bouyer, Larsen, Markey, and Rasmussen, 2006b, Rutkowski, 2011]:

- Player ○ prefers to stay as long as possible in locations with **minimal price**: add a final location allowing him to stay until the end, and make the location urgent
- Player □ prefers to leave as soon as possible in locations with **minimal price**: make the location urgent
Recursive elimination of states

Intuition from [Bouyer, Larsen, Markey, and Rasmussen, 2006b, Rutkowski, 2011]:

- Player \( \bigcirc \) prefers to stay as long as possible in locations with **minimal price**: add a final location allowing him to stay until the end, and make the location urgent

- Player \( \Box \) prefers to leave as soon as possible in locations with **minimal price**: make the location urgent

Problem: intuition not always true... you may have to change decision!

Recursive algorithm + construction of the value functions from right \((x = 1)\) to left \((x = 0)\)
Recursive elimination of states

Intuition from [Bouyer, Larsen, Markey, and Rasmussen, 2006b, Rutkowski, 2011]:

- Player ♣ prefers to stay as long as possible in locations with minimal price: add a final location allowing him to stay until the end, and make the location urgent
- Player □ prefers to leave as soon as possible in locations with minimal price: make the location urgent

Problem: intuition not always true... you may have to change decision!

Recursive algorithm + construction of the value functions from right \((x = 1)\) to left \((x = 0)\)

Challenges with arbitrary weights:

- Proof of correctness does not generalise: initially two distinct proofs for ♣ and □
- Proof of termination does not generalise: difficult because of the double recursion...
Make a symmetric treatment of $\bigcirc$ and $\Box$

**Theorem**

*PTGs are determined ($\overline{\text{Val}} = \text{Val}$), and value functions are continuous (over regions).*

Determinacy follows from Gale-Stewart determinacy result...
Theorem

PTGs are determined ($\overline{Val} = \underline{Val}$), and value functions are continuous (over regions).

Determinacy follows from Gale-Stewart determinacy result...

Advantage: both players are dual...
Make a symmetric treatment of \(\bigcirc\) and \(\square\)

**Theorem**

PTGs are determined \((\overline{\text{Val}} = \text{Val})\), and value functions are continuous (over regions).

Determinacy follows from Gale-Stewart determinacy result...

Advantage: both players are dual...

**Theorem**

For every SPTG, all value functions are piecewise affine, with at most an exponential number of cutpoints (in number of locations).
Make a symmetric treatment of \( \bigcirc \) and \( \square \)

**Theorem**

PTGs are determined \((\overline{\text{Val}} = \overline{\text{Val}})\), and value functions are continuous (over regions).

Determinacy follows from Gale-Stewart determinacy result...

Advantage: both players are dual...

**Theorem**

For every SPTG, all value functions are piecewise affine, with at most an exponential number of cutpoints (in number of locations).

For general 1-clock PTGs?

- removing guards and invariants: previously used techniques work!
- removing resets: previously, bound the number of resets...
Bounding the number of resets needed is not possible

\[ x = 1, x := 0 \]

\[ x \leq 1 \]

\[ x = 1 \]
Bounding the number of resets needed is not possible

Player $\bigcirc$ can guarantee (i.e., ensure to be below) value $\varepsilon$ for all $\varepsilon > 0$...
Bounding the number of resets needed is not possible

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

… but cannot obtain 0: hence, no optimal strategy…
Bounding the number of resets needed is not possible

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

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

... moreover, to obtain $\varepsilon$, $\bigcirc$ needs to loop at least $W + \lceil 1 / \log \varepsilon \rceil$ times before reaching $\checkmark$!
Current solution: Reset-acyclic 1-clock PTGs

exponential time algorithm for reset-acyclic 1-clock PTGs with arbitrary weights
Summary and Future Work

Results

- Extension of iterative elimination for reset-acyclic 1-clock PTGs with arbitrary weights
- Study of the value function: determination, upper and lower bound, number of cutpoints...
Summary and Future Work

Results

- Extension of iterative elimination for reset-acyclic 1-clock PTGs with arbitrary weights
- Study of the value function: determination, upper and lower bound, number of cutpoints...

- Future work: final extension of the result for all 1-clock PTGs?
- Use the result for 1-clock to approximate/compute the value of general PTGs with adequate structural properties
- Implementation and test of different algorithms on real instances
Summary and Future Work

Results

- Extension of iterative elimination for reset-acyclic 1-clock PTGs with arbitrary weights
- Study of the value function: determination, upper and lower bound, number of cutpoints...

- Future work: final extension of the result for all 1-clock PTGs?
- Use the result for 1-clock to approximate/compute the value of general PTGs with adequate structural properties
- Implementation and test of different algorithms on real instances

Thank you for your attention


References II


