Simple Priced Timed Games are not that simple

From FSTTCS 2015

Lefaucheux Engel
Inria Rennes, France

Thomas Brihaye (UMons), Gilles Geeraerts (ULB), Axel Haddad (UMons), Benjamin Monmege (LIF Marseille)

September 7, 2016
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

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

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

\[
\begin{align*}
(\ell_1, 0) \xrightarrow{0.4} (\ell_4, 0.4) \xrightarrow{0.6} (\ell_5, 0)
\end{align*}
\]
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:
- $\infty$ if not reached
- total payoff up to otherwise

$\ell_1 = (0.4, 0.4) \xrightarrow{0.6} (\ell_5, 0) \xleftarrow{1.5} (\ell_4, 0) \xrightarrow{1.1} (\ell_5, 0) \xrightarrow{2} (\checkmark, 2)$
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

Cost of a play:
- \( \{ + \infty \text{ if not reached} \) total payoff up to otherwise

\[
\begin{align*}
(\ell_1, 0) & \xrightarrow{0.4} (\ell_4, 0.4) \xrightarrow{0.6} (\ell_5, 0) \xrightarrow{1.5} (\ell_4, 0) \xrightarrow{1.1} (\ell_5, 0) \xrightarrow{2} (\checkmark, 2) \\
0.4 + 1 & \quad -3 \times 0.6 + 1.5 - 3 \times 1.1 + 2 \times 2 + 2 = 3.8
\end{align*}
\]
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

Cost of a play:

\[\begin{cases} 
+\infty & \text{if not reached} \\
\text{total payoff up to otherwise} 
\end{cases}\]

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

\[0.4 + 1 - 3 \times 0.6 + 1.5 - 3 \times 1.1 + 2 \times 2 + 2 = 3.8\]

\[(\ell_1, 0) \xrightarrow{0.2, \nearrow} (\ell_2, 0) \xrightarrow{0.9, \rightarrow} (\ell_3, 0.9) \xrightarrow{0.2, \varnothing} (\ell_3, 0) \xrightarrow{0.9, \varnothing} (\ell_3, 0) \cdots \]

\[= +\infty (\checkmark \text{ not reached})\]
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

Cost of a play: 
\[
\left\{
\begin{array}{ll}
+\infty & \text{if } \checkmark \text{ not reached} \\
\text{total payoff up to } \checkmark & \text{otherwise}
\end{array}
\right.
\]
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:
\( \text{Val}(\ell, v) = \) minimal cost player ○ can guarantee
\( \text{Val}(\ell, v) = \) maximal cost player □ can guarantee

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

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

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

- 2-player PTGs: undecidable [Brihaye, Bruyère, and Raskin, 2005, Bouyer, Brihaye, and Markey, 2006a], even with only non-negative costs and 3 clocks

More complex with negative costs.
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 \( \leq K \)?

- 2-player PTGs: undecidable [Brihaye, Bruyère, and Raskin, 2005, Bouyer, Brihaye, and Markey, 2006a], even with only non-negative costs and 3 clocks

More complex with negative costs.

- \( F_{\leq K}^\checkmark \) undecidable for 2 or more clocks and pseudo-polynomial algorithm for One-clock Bi-valued PTG [Brihaye, Geeraerts, Krishna, Manasa, Monmege, and Trivedi, 2014]

One-clock Bi-valued PTG: important restriction on the allowed rates of locations
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 $\overline{\text{Val}}(\ell, x)$. 

Figure 1: Example of an SPTG, showing value functions and an optimal worst case bound for any algorithm solving PTGs was 2

(d) For the special case of PTGs with all rates being 1 (i.e., all states $r_1 = 9$, $r_2 = 3$, $r_3 = 6$, $r_4 = 9$, $r_5 = 0$, $c_{(3,1)} = 3$, $c_{(5,\perp)} = 5$

The best previous bound on $L(G)$ is a lower bound on the size of the explicit description of these general cases (e.g., such as those of UPPAAL, Trivedi [18] also observed that the region abstraction algorithm of

$T_{\ell, x}$, where $T_{\ell, x}$ was that algorithm as

$\overline{\text{Val}}(\ell, x)$. 

We shall refer to that algorithm as

$\overline{\text{Val}}(\ell, x)$.

This special case is also

$\overline{\text{Val}}(\ell, x)$.

Given the analysis of this algorithm to our algorithm.

Inspired by other previous techniques for 1-clock PTGs?
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

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 2
- Proof of termination does not generalise: difficult because of the double recursion...
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 \(\square\) 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 $\bigcirc$ 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 $\square$ 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 $\bigcirc$ and $\square$
- Proof of termination does not generalise: difficult because of the double recursion...
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.
Make a symmetric treatment of $\bigcirc$ and $\blacksquare$

**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...
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).
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).

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

Player\# 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 $\varepsilon$, \# needs to loop at least $W + \lceil 1/\log \varepsilon \rceil$ times before reaching $\square$.
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


