Real-time Model Checking

— Priced timed automata —

Nicolas MARKEY

Lav. Spécification & Vérification
CNRS & ENS Cachan – France

March 3, 2010

Time is not always sufficient

Timed automata are (rather) well understood – Can we go further?

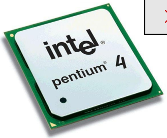
Time is not always sufficient

Timed automata are (rather) well understood – Can we go further?

Compute $D\times(C\times(A+B))+(A+B)+(C\times D)$ using two processors:

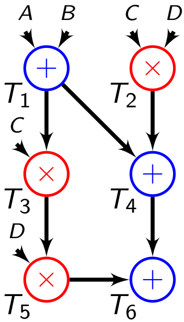
P_1 (fast):

	time
$+$	2 picosec.
$\times$	3 picosec.



P_2 (slow):

	time
$+$	5 picosec.
$\times$	7 picosec.

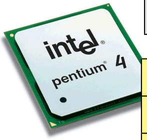


Time is not always sufficient

Timed automata are (rather) well understood – Can we go further?

Compute $D\times(C\times(A+B))+(A+B)+(C\times D)$ using two processors:

P_1 (fast):



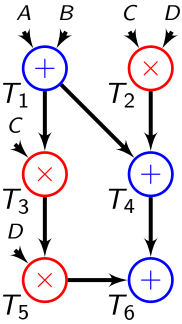
time	
$+$	2 picosec.
$\times$	3 picosec.

energy	
idle	10 W
in use	90 W

P_2 (slow):

time	
$+$	5 picosec.
$\times$	7 picosec.

energy	
idle	20 W
in use	30 W



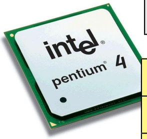
Time is not always sufficient

Timed automata are (rather) well understood – Can we go further?

Compute $D\times(C\times(A+B))+(A+B)+(C\times D)$ using two processors:

P_1 (fast):

time	
$+$	2 picosec.
$\times$	3 picosec.

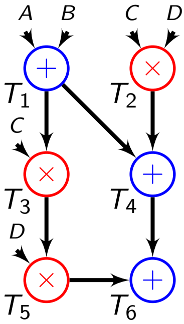


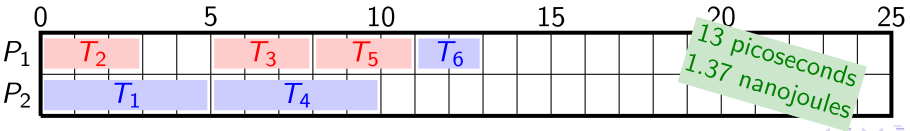
energy	
idle	10 W
in use	90 W

P_2 (slow):

time	
$+$	5 picosec.
$\times$	7 picosec.

energy	
idle	20 W
in use	30 W





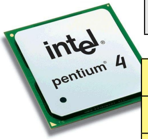
Time is not always sufficient

Timed automata are (rather) well understood – Can we go further?

Compute $D\times(C\times(A+B))+(A+B)+(C\times D)$ using two processors:

P_1 (fast):

	time
$+$	2 picosec.
$\times$	3 picosec.

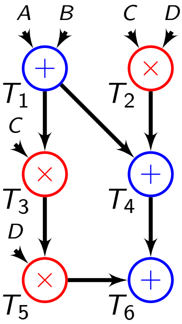


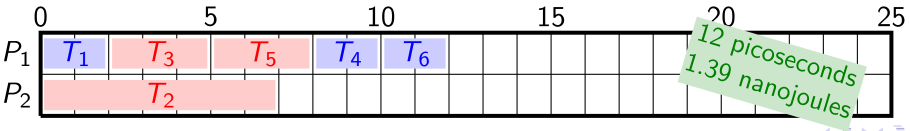
energy	
idle	10 W
in use	90 W

P_2 (slow):

	time
$+$	5 picosec.
$\times$	7 picosec.

energy	
idle	20 W
in use	30 W





Time is not always sufficient

Timed automata are (rather) well understood – Can we go further?

Compute $D\times(C\times(A+B))+(A+B)+(C\times D)$ using two processors:

P_1 (fast):

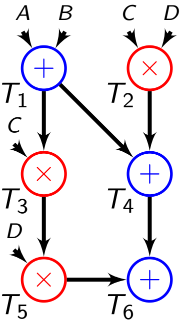
time	
$+$	2 picosec.
$\times$	3 picosec.

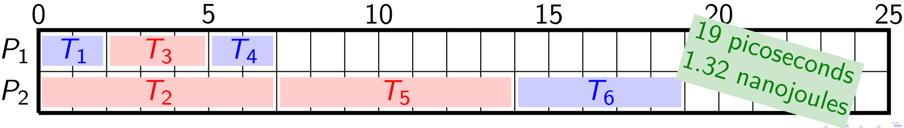
energy	
idle	10 W
in use	90 W

P_2 (slow):

time	
$+$	5 picosec.
$\times$	7 picosec.

energy	
idle	20 W
in use	30 W

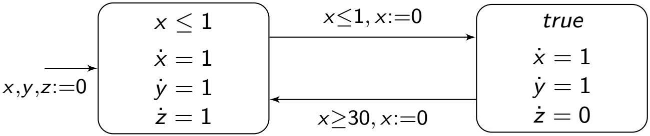




Time is not always sufficient

- hybrid automata: timed automata augmented with variables whose derivative is not constant.

$\leadsto$ examples: leaking gas burner, water-level monitor, ...



Theorem

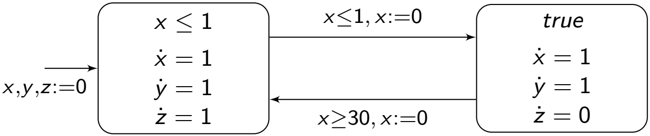
Reachability is undecidable (even for timed automata with one stopwatch).

Refs: [1] Henzinger, Kopke, Puri, Varaiya. *What's Decidable about Hybrid Automata?* (1995).

Time is not always sufficient

- hybrid automata: timed automata augmented with variables whose derivative is not constant.

$\leadsto$ examples: leaking gas burner, water-level monitor, ...



- timed automata with observers: similar to hybrid automata, but the behavior only depends on clock variables.

Refs: [1] Alur, La Torre, Pappas. *Optimal Paths in Weighted Timed Automata* (2001).
[2] Behrmann *et al.* *Minimum-cost reachability for priced timed automata* (2001).

Outline of the talk

1. Introduction
2. Timed automata with observers
3. Resource-optimization problems
   - Optimal reachabililty
   - Weighted temporal logics
   - Optimal strategies
4. Resource-management problems
5. Conclusions and perspectives

Outline of the talk

1 Introduction

2 Timed automata with observers

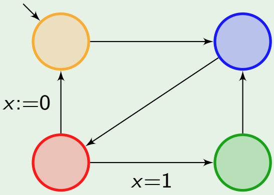
3 Resource-optimization problems
- Optimal reachabililty
- Weighted temporal logics
- Optimal strategies

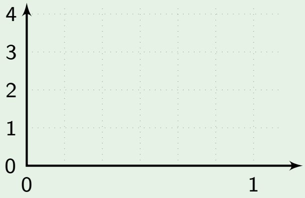
4 Resource-management problems

5 Conclusions and perspectives

Timed automata with (linear) observers

Example

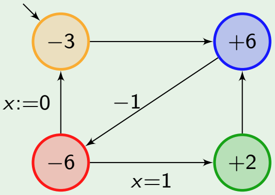


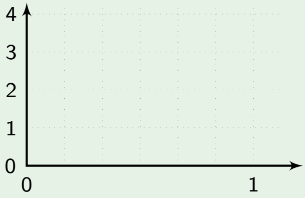


Refs: [1] Alur, La Torre, Pappas. *Optimal Paths in Weighted Timed Automata* (2001).
[2] Behrmann *et al. Minimum-cost reachability for priced timed automata* (2001).

Timed automata with (linear) observers

Example



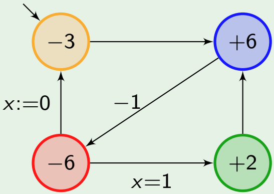


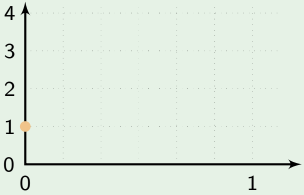
Refs: [1] Alur, La Torre, Pappas. *Optimal Paths in Weighted Timed Automata* (2001).

[2] Behrmann *et al. Minimum-cost reachability for priced timed automata* (2001).

Timed automata with (linear) observers

Example



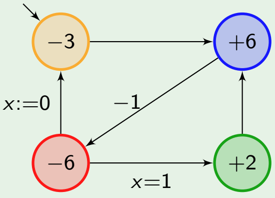


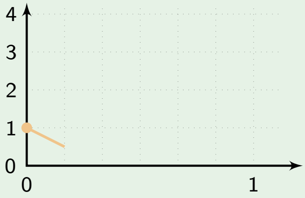
Refs: [1] Alur, La Torre, Pappas. *Optimal Paths in Weighted Timed Automata* (2001).

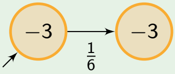
[2] Behrmann *et al. Minimum-cost reachability for priced timed automata* (2001).

Timed automata with (linear) observers

Example



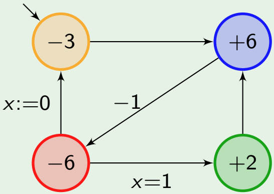


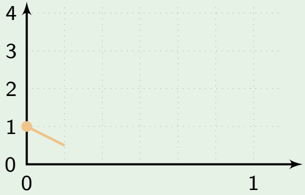


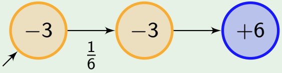
Refs: [1] Alur, La Torre, Pappas. *Optimal Paths in Weighted Timed Automata* (2001).
[2] Behrmann *et al. Minimum-cost reachability for priced timed automata* (2001).

Timed automata with (linear) observers

Example



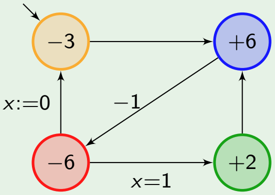


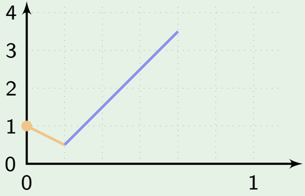


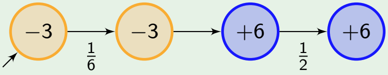
Refs: [1] Alur, La Torre, Pappas. *Optimal Paths in Weighted Timed Automata* (2001).
[2] Behrmann *et al. Minimum-cost reachability for priced timed automata* (2001).

Timed automata with (linear) observers

Example





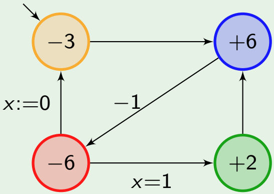


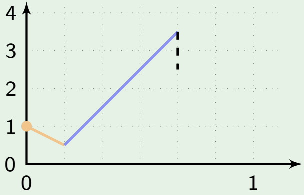
Refs: [1] Alur, La Torre, Pappas. *Optimal Paths in Weighted Timed Automata* (2001).

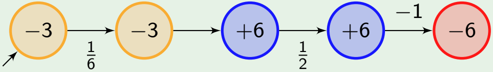
[2] Behrmann *et al. Minimum-cost reachability for priced timed automata* (2001).

Timed automata with (linear) observers

Example





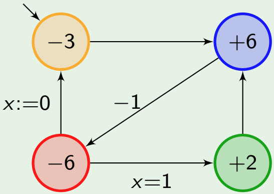


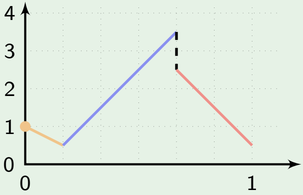
Refs: [1] Alur, La Torre, Pappas. *Optimal Paths in Weighted Timed Automata* (2001).

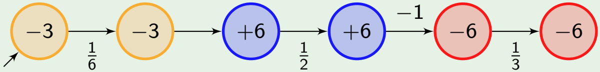
[2] Behrmann *et al.* *Minimum-cost reachability for priced timed automata* (2001).

Timed automata with (linear) observers

Example





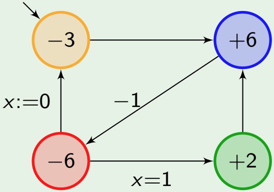


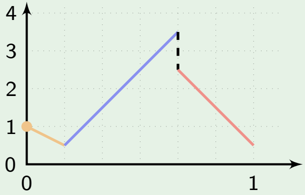
Refs: [1] Alur, La Torre, Pappas. *Optimal Paths in Weighted Timed Automata* (2001).

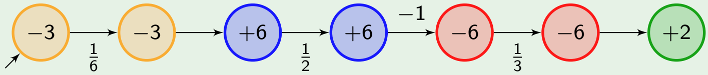
[2] Behrmann *et al.* *Minimum-cost reachability for priced timed automata* (2001).

Timed automata with (linear) observers

Example





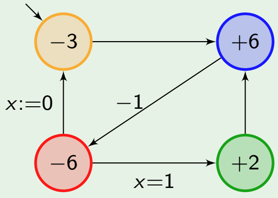


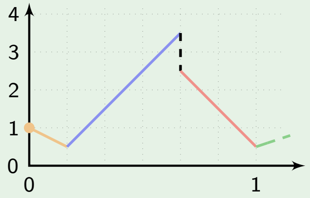
Refs: [1] Alur, La Torre, Pappas. *Optimal Paths in Weighted Timed Automata* (2001).

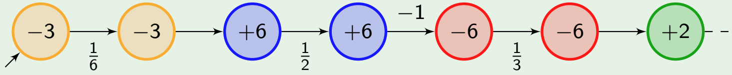
[2] Behrmann *et al.* *Minimum-cost reachability for priced timed automata* (2001).

Timed automata with (linear) observers

Example







Refs: [1] Alur, La Torre, Pappas. *Optimal Paths in Weighted Timed Automata* (2001).
[2] Behrmann *et al. Minimum-cost reachability for priced timed automata* (2001).

Outline of the talk

1 Introduction

2 Timed automata with observers

3 Resource-optimization problems
- Optimal reachabililty
- Weighted temporal logics
- Optimal strategies

4 Resource-management problems

5 Conclusions and perspectives

Outline of the talk

1 Introduction

2 Timed automata with observers

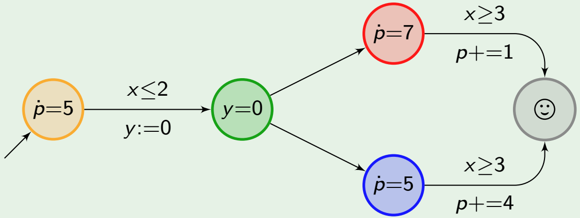
3 Resource-optimization problems
- Optimal reachabililty
- Weighted temporal logics
- Optimal strategies

4 Resource-management problems

5 Conclusions and perspectives

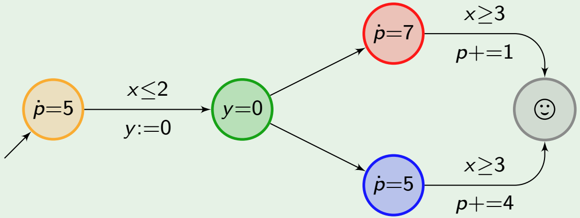
Optimal reachability

Example



Optimal reachability

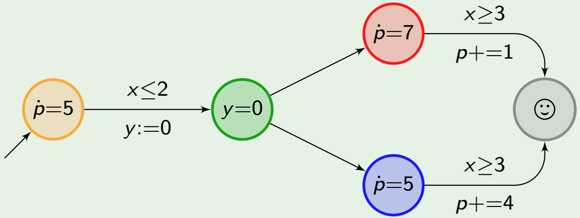
Example



Minimal cost for reaching ☺:

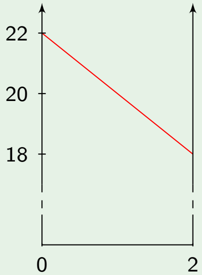
Optimal reachability

Example



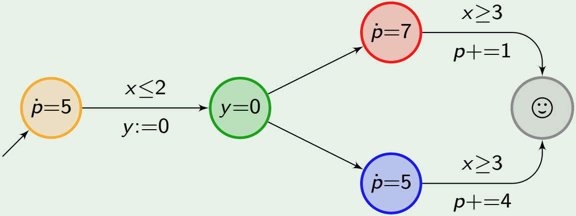
Minimal cost for reaching ☺:

$$5t + 7(3-t) + 1$$



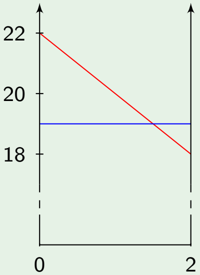
Optimal reachability

Example



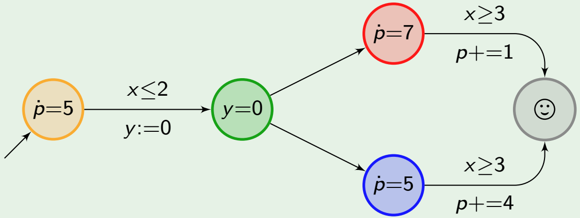
Minimal cost for reaching ☺:

$$5t + 7(3 - t) + 1$$
$$5t + 5(3 - t) + 4$$



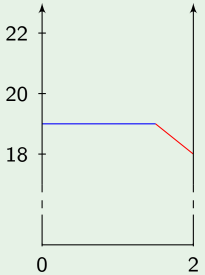
Optimal reachability

Example



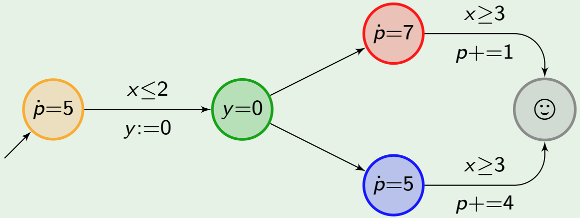
Minimal cost for reaching ☺:

$$\min \begin{pmatrix} 5t + 7(3-t) + 1 \\ 5t + 5(3-t) + 4 \end{pmatrix}$$



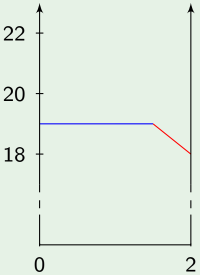
Optimal reachability

Example



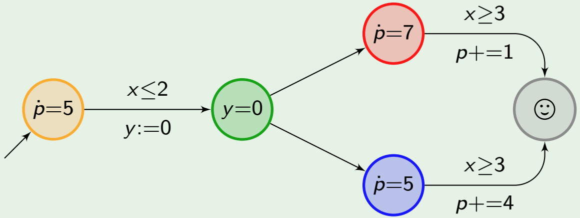
Minimal cost for reaching ☺:

$$\inf_{0 \le t \le 2} \min \begin{pmatrix} 5t + 7(3-t) + 1 \\ 5t + 5(3-t) + 4 \end{pmatrix}$$



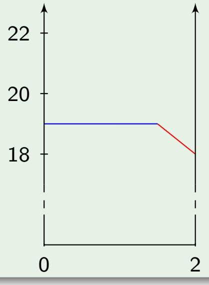
Optimal reachability

Example



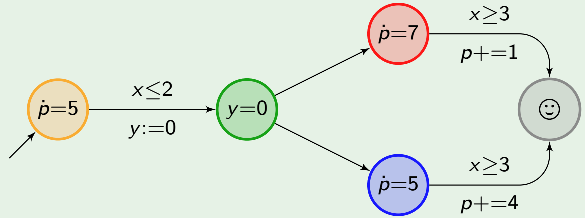
Minimal cost for reaching ☺:

$$\inf_{0\leq t\leq 2} \min \begin{pmatrix} 5t+7(3-t)+1 \\ 5t+5(3-t)+4 \end{pmatrix} = 18$$



Optimal reachability

Example

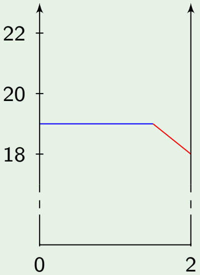


Minimal cost for reaching ☺:

$$\inf_{0\leq t\leq 2} \min \begin{pmatrix} 5t + 7(3-t) + 1 \\ 5t + 5(3-t) + 4 \end{pmatrix} = 18$$

The optimal *schedule* consists in

- waiting 2 time units in ◯;
- going through ◯.



Optimal reachability

Theorem

Optimal reachability in priced timed automata is PSPACE-complete.

Refs: [1] Alur, La Torre, Pappas. *Optimal Paths in Weighted Timed Automata* (2001).
[2] Behrmann *et al. Minimum-cost reachability for priced timed automata* (2001).
[3] Bouyer, Brihaye, Bruyère, Raskin. *On the Optimal Reachability Problem on Weighted Timed Automata* (2006).

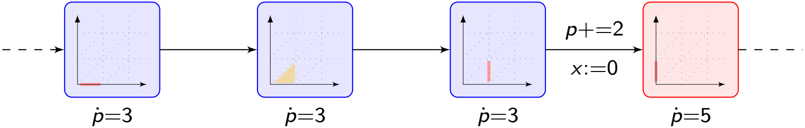
Optimal reachability

Theorem

Optimal reachability in priced timed automata is PSPACE-complete.

Proof.

- The region abstraction is not fine enough:



Refs: [1] Alur, La Torre, Pappas. *Optimal Paths in Weighted Timed Automata* (2001).
[2] Behrmann *et al.* *Minimum-cost reachability for priced timed automata* (2001).
[3] Bouyer, Brihaye, Bruyère, Raskin. *On the Optimal Reachability Problem on Weighted Timed Automata* (2006).

Optimal reachability

Theorem

Optimal reachability in priced timed automata is PSPACE-complete.

Proof.

- The idea is: "take transitions *close to integer dates*";

Refs: [1] Alur, La Torre, Pappas. *Optimal Paths in Weighted Timed Automata* (2001).
[2] Behrmann *et al.* *Minimum-cost reachability for priced timed automata* (2001).
[3] Bouyer, Brihaye, Bruyère, Raskin. *On the Optimal Reachability Problem on Weighted Timed Automata* (2006).

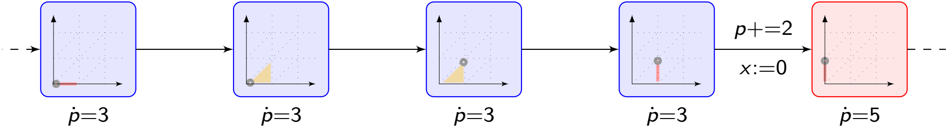
Optimal reachability

Theorem

Optimal reachability in priced timed automata is PSPACE-complete.

Proof.

- The idea is: "take transitions *close to integer dates*";
- Corner-point abstraction: only consider *corners* of regions:



Refs: [1] Alur, La Torre, Pappas. *Optimal Paths in Weighted Timed Automata* (2001).

[2] Behrmann *et al.* *Minimum-cost reachability for priced timed automata* (2001).

[3] Bouyer, Brihaye, Bruyère, Raskin. *On the Optimal Reachability Problem on Weighted Timed Automata* (2006).

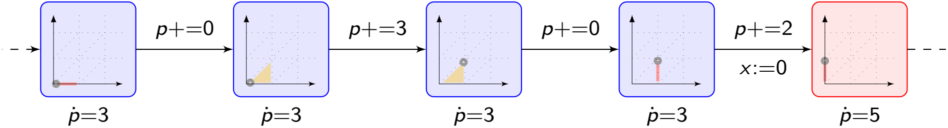
Optimal reachability

Theorem

Optimal reachability in priced timed automata is PSPACE-complete.

Proof.

- The idea is: "take transitions *close to integer dates*";
- Corner-point abstraction: only consider *corners* of regions:



Refs: [1] Alur, La Torre, Pappas. *Optimal Paths in Weighted Timed Automata* (2001).
[2] Behrmann *et al.* *Minimum-cost reachability for priced timed automata* (2001).
[3] Bouyer, Brihaye, Bruyère, Raskin. *On the Optimal Reachability Problem on Weighted Timed Automata* (2006).

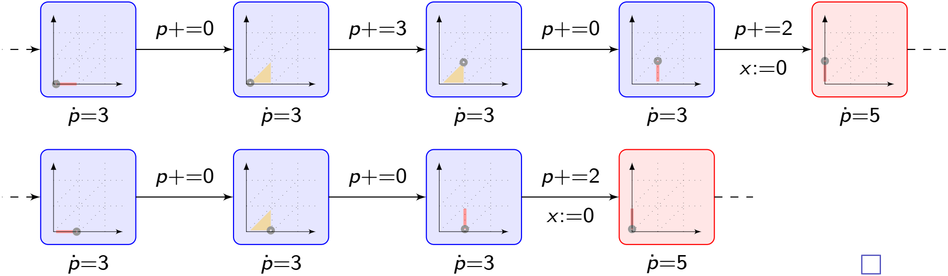
Optimal reachability

Theorem

Optimal reachability in priced timed automata is PSPACE-complete.

Proof.

- The idea is: "take transitions *close to integer dates*";
- Corner-point abstraction: only consider *corners* of regions:



□

Refs: [1] Alur, La Torre, Pappas. *Optimal Paths in Weighted Timed Automata* (2001).
[2] Behrmann *et al. Minimum-cost reachability for priced timed automata* (2001).
[3] Bouyer, Brihaye, Bruyère, Raskin. *On the Optimal Reachability Problem on Weighted Timed Automata* (2006).

Outline of the talk

1. Introduction
2. Timed automata with observers
3. Resource-optimization problems
   - Optimal reachabililty
   - Weighted temporal logics
   - Optimal strategies
4. Resource-management problems
5. Conclusions and perspectives

Weighted temporal logic

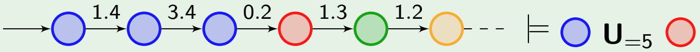
Example

Decorate temporal modalities with constraints on cost:

Weighted temporal logic

Example

Decorate temporal modalities with constraints on cost:



Weighted temporal logic

Example

Decorate temporal modalities with constraints on cost:

Weighted temporal logic

Example

Decorate temporal modalities with constraints on cost:

Example

- $\mathbf{G}(\texttt{failure} \Rightarrow \mathbf{F}_{\leq 250}\, \texttt{repaired})$

Weighted temporal logic

Example

Decorate temporal modalities with constraints on cost:

Example

- $\mathbf{G}(\texttt{failure} \Rightarrow \mathbf{F}_{\leq 250}\, \texttt{repaired})$
- $\mathbf{A}\,\mathbf{G}(\texttt{failure} \Rightarrow \mathbf{E}\,\mathbf{F}_{\texttt{time}\leq 5}(\texttt{repair} \wedge \mathbf{A}\,\mathbf{F}_{\texttt{cost}\leq 150}\, \texttt{running}))$

Undecidability results

Theorem

WMTL model-checking is undecidable.

Refs: [1] Bouyer, M. *Costs are Expensive!* (2007).

Undecidability results

Theorem

WMTL model-checking is undecidable.

Proof.

- encoding of a two-counter machine;

Undecidability results

Theorem

WMTL model-checking is undecidable.

Proof.

- encoding of a two-counter machine;
- Holds even for one clock and one cost variable.

□

Refs: [1] Bouyer, M. *Costs are Expensive!* (2007).

Undecidability results

Theorem

WMTL model-checking is undecidable.

Proof.

- encoding of a two-counter machine;
- Holds even for one clock and one cost variable.

□

Theorem

WCTL model-checking is undecidable.

Refs: [1] Bouyer, Brihaye, M. *Improved Undecidability Results on Weighted Timed Automata* (2006).
[2] Brihaye, Bruyère, Raskin. *Model-Checking for Weighted Timed Automata* (2004).

Undecidability results

Theorem

WMTL model-checking is undecidable.

Proof.

- encoding of a two-counter machine;
- Holds even for one clock and one cost variable.

□

Theorem

WCTL model-checking is undecidable.

Proof.

- encoding of a two-counter machine;

Refs: [1] Bouyer, Brihaye, M. *Improved Undecidability Results on Weighted Timed Automata* (2006).
[2] Brihaye, Bruyère, Raskin. *Model-Checking for Weighted Timed Automata* (2004).

Undecidability results

Theorem

WMTL model-checking is undecidable.

Proof.

- encoding of a two-counter machine;
- Holds even for one clock and one cost variable.

□

Theorem

WCTL model-checking is undecidable.

Proof.

- encoding of a two-counter machine;
- requires three clocks.

□

Refs: [1] Bouyer, Brihaye, M. *Improved Undecidability Results on Weighted Timed Automata* (2006).
[2] Brihaye, Bruyère, Raskin. *Model-Checking for Weighted Timed Automata* (2004).

Decidable subcases

Theorem

WCTL model-checking is PSPACE-complete on 1-clock weighted timed automata.

Refs: [1] Bouyer, Larsen, M. *Model-Checking One-Clock Priced Timed Automata* (2007).

Decidable subcases

Theorem

WCTL model-checking is PSPACE-complete on 1-clock weighted timed automata.

Proof.

- region-based algorithm;

Refs: [1] Bouyer, Larsen, M. *Model-Checking One-Clock Priced Timed Automata* (2007).

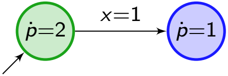
Decidable subcases

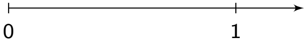
Theorem

WCTL model-checking is PSPACE-complete on 1-clock weighted timed automata.

Proof.

- region-based algorithm;
- but region are not fine enough:





$\mathbf{E}\,\mathbf{F}_{\leq 1}$

Refs: [1] Bouyer, Larsen, M. *Model-Checking One-Clock Priced Timed Automata* (2007).

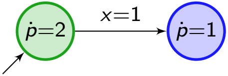
Decidable subcases

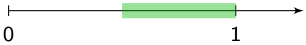
Theorem

WCTL model-checking is PSPACE-complete on 1-clock weighted timed automata.

Proof.

- region-based algorithm;
- but region are not fine enough:





$\mathbf{E\,F}_{\leq 1}$

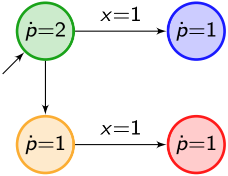
Decidable subcases

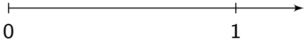
Theorem

WCTL model-checking is PSPACE-complete on 1-clock weighted timed automata.

Proof.

- region-based algorithm;
- but region are not fine enough:





$$\mathbf{E}[\neg(\mathbf{E}\,\mathbf{F}_{\leq 1} \bigcirc)\ \mathbf{U}_{\geq 1} \bigcirc]$$

Refs: [1] Bouyer, Larsen, M. *Model-Checking One-Clock Priced Timed Automata* (2007).

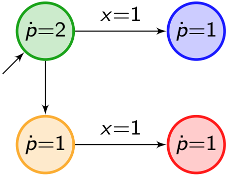
Decidable subcases

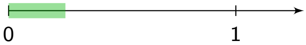
Theorem

WCTL model-checking is PSPACE-complete on 1-clock weighted timed automata.

Proof.

- region-based algorithm;
- but region are not fine enough:





$\mathbf{E}[\neg(\mathbf{E}\,\mathbf{F}_{\leq 1} \bigcirc)\ \mathbf{U}_{\geq 1} \bigcirc]$

Refs: [1] Bouyer, Larsen, M. *Model-Checking One-Clock Priced Timed Automata* (2007).

Decidable subcases

Theorem

WCTL model-checking is PSPACE-complete on 1-clock weighted timed automata.

Proof.

- region-based algorithm;
- but region are not fine enough:
- Refine regions: granularity $1/M^{|\varphi|}$ is sufficient.

□

Refs: [1] Bouyer, Larsen, M. *Model-Checking One-Clock Priced Timed Automata* (2007).

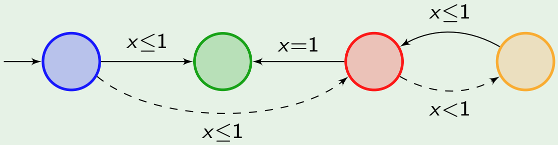
Outline of the talk

1. Introduction
2. Timed automata with observers
3. Resource-optimization problems
   - Optimal reachabililty
   - Weighted temporal logics
   - Optimal strategies
4. Resource-management problems
5. Conclusions and perspectives

Weighted timed games

Example

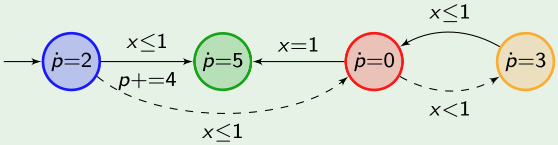
Timed games can also be extended with weights:



Weighted timed games

Example

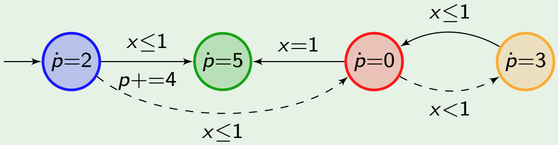
Timed games can also be extended with weights:



Weighted timed games

Example

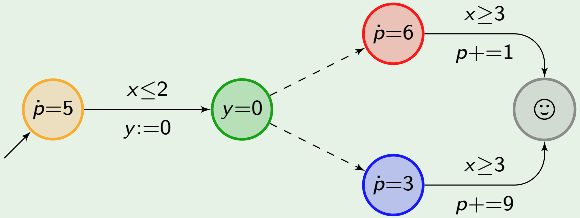
Timed games can also be extended with weights:



- A strategy for a player indicates which (action or delay) transition to play;
- A strategy is winning if all its outcomes are.

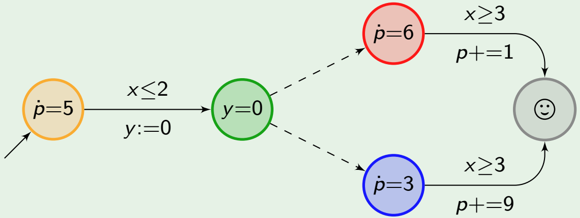
Optimal winning strategy

Example



Optimal winning strategy

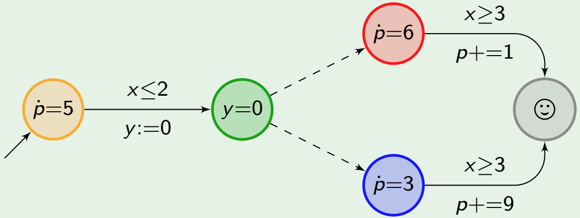
Example



Minimal cost for reaching ☺:

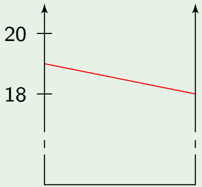
Optimal winning strategy

Example



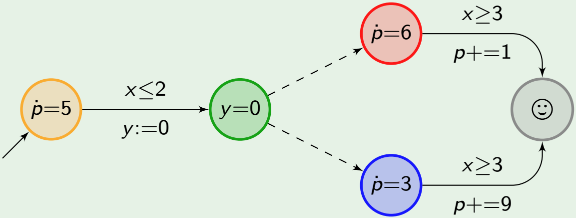
Minimal cost for reaching ☺:

$$5t + 6(3 - t) + 1$$



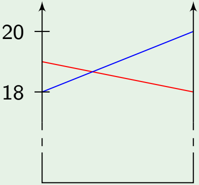
Optimal winning strategy

Example



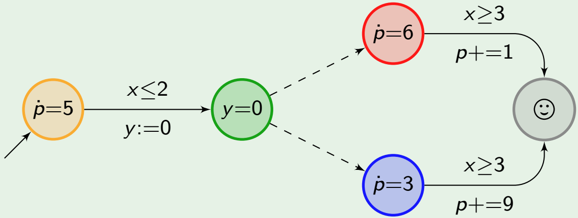
Minimal cost for reaching ☺:

$$5t + 6(3-t) + 1$$
$$5t + 3(3-t) + 9$$



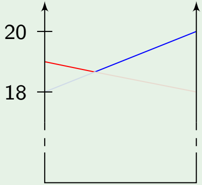
Optimal winning strategy

Example



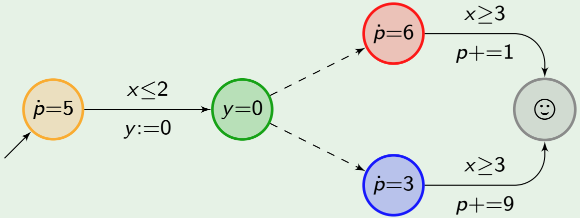
Minimal cost for reaching ☺:

$$\max \begin{pmatrix} 5t + 6(3-t) + 1 \\ 5t + 3(3-t) + 9 \end{pmatrix}$$



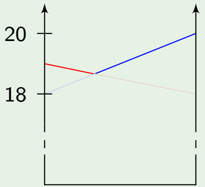
Optimal winning strategy

Example



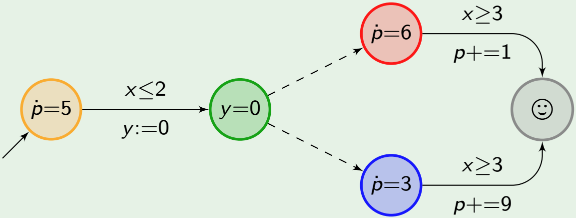
Minimal cost for reaching ☺:

$$\inf_{0 \le t \le 2} \max \begin{pmatrix} 5t + 6(3-t) + 1 \\ 5t + 3(3-t) + 9 \end{pmatrix}$$



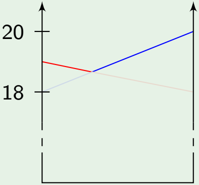
Optimal winning strategy

Example



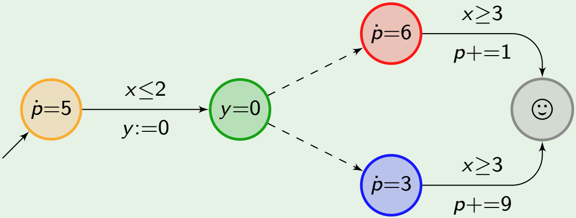
Minimal cost for reaching ☺:

$$\inf_{0\le t\le 2} \max \begin{pmatrix} 5t + 6(3-t) + 1 \\ 5t + 3(3-t) + 9 \end{pmatrix} = 56/3$$



Optimal winning strategy

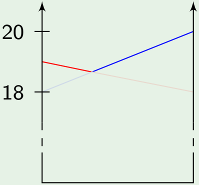
Example



Minimal cost for reaching ☺:

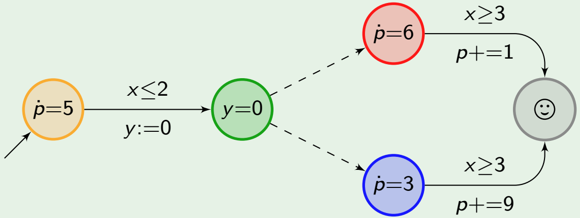
$$\inf_{0 \leq t \leq 2} \max \begin{pmatrix} 5t + 6(3-t) + 1 \\ 5t + 3(3-t) + 9 \end{pmatrix} = 56/3$$

which is achieved with $t = 1/3$



Optimal winning strategy

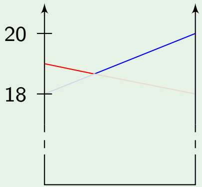
Example



Minimal cost for reaching ☺:

$$\inf_{0\leq t\leq 2} \max \begin{pmatrix} 5t + 6(3-t) + 1 \\ 5t + 3(3-t) + 9 \end{pmatrix} = 56/3$$

which is achieved with $t = 1/3$



Corollary

Regions are not sufficient for solving priced timed games.

Computing optimal winning strategies is undecidable

Theorem

Computing optimal strategies in priced timed games is undecidable.

Refs: [1] Bouyer, Brihaye, M. *Improved Undecidability Results on Weighted Timed Automata* (2006).

Computing optimal winning strategies is undecidable

Theorem

Computing optimal strategies in priced timed games is undecidable.

Proof.

The proof relies on simple modules that will allow encoding a two-counter machine:

Computing optimal winning strategies is undecidable

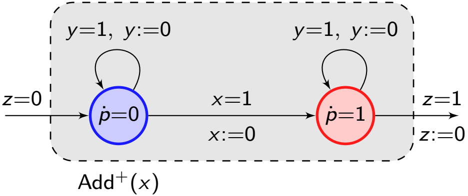
Theorem

Computing optimal strategies in priced timed games is undecidable.

Proof.

The proof relies on simple modules that will allow encoding a two-counter machine:

- Adding the value of clock x to the cost:



Refs: [1] Bouyer, Brihaye, M. *Improved Undecidability Results on Weighted Timed Automata* (2006).

Computing optimal winning strategies is undecidable

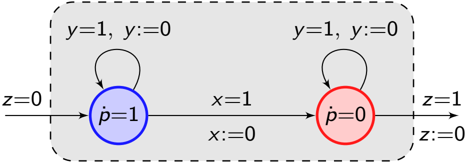
Theorem

Computing optimal strategies in priced timed games is undecidable.

Proof.

The proof relies on simple modules that will allow encoding a two-counter machine:

- Adding the value of clock x to the cost:
- Adding $1 - x$ to the cost:



$\mathrm{Add}^{-}(x)$

Refs: [1] Bouyer, Brihaye, M. *Improved Undecidability Results on Weighted Timed Automata* (2006).

Computing optimal winning strategies is undecidable

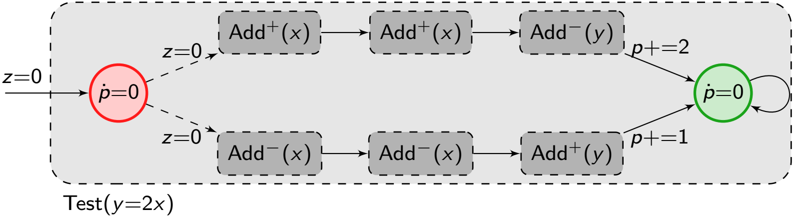
Theorem

Computing optimal strategies in priced timed games is undecidable.

Proof.

The proof relies on simple modules that will allow encoding a two-counter machine:

- Checking that $y = 2x$:



Refs: [1] Bouyer, Brihaye, M. *Improved Undecidability Results on Weighted Timed Automata* (2006).

Computing optimal winning strategies is undecidable

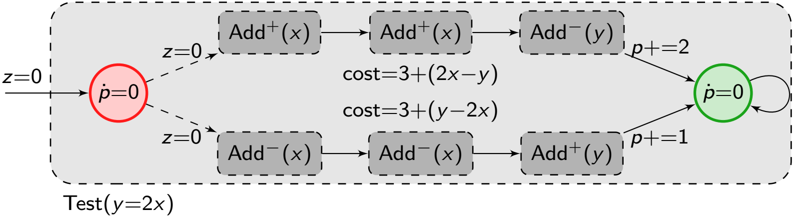
Theorem

Computing optimal strategies in priced timed games is undecidable.

Proof.

The proof relies on simple modules that will allow encoding a two-counter machine:

- Checking that $y = 2x$:



Refs: [1] Bouyer, Brihaye, M. *Improved Undecidability Results on Weighted Timed Automata* (2006).

Computing optimal winning strategies is undecidable

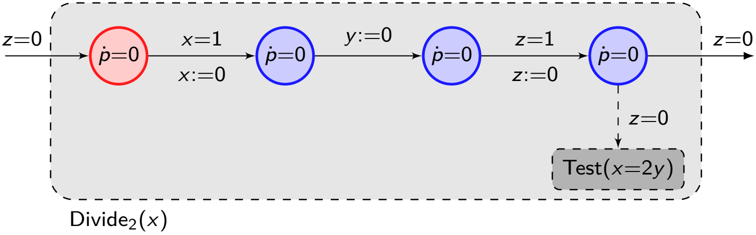
Theorem

Computing optimal strategies in priced timed games is undecidable.

Proof.

The proof relies on simple modules that will allow encoding a two-counter machine:

- Checking that $y = 2x$:
- Dividing clock x by 2:



Refs: [1] Bouyer, Brihaye, M. *Improved Undecidability Results on Weighted Timed Automata* (2006).

Computing optimal winning strategies is undecidable

Theorem

Computing optimal strategies in priced timed games is undecidable.

Proof.

The proof relies on simple modules that will allow encoding a two-counter machine:

- encode counter c_1 as $x_1 = 2^{-c_1}$ and counter c_2 as $x_2 = 3^{-c_1}$;
- by cleverly juggling with clocks, we can achieve this encoding with three clocks.

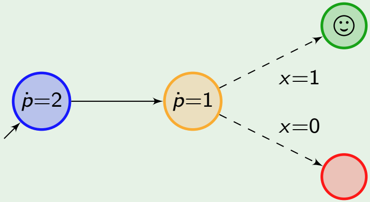
□

Refs: [1] Bouyer, Brihaye, M. *Improved Undecidability Results on Weighted Timed Automata* (2006).

Turn-based 1-clock priced timed games are decidable

Example

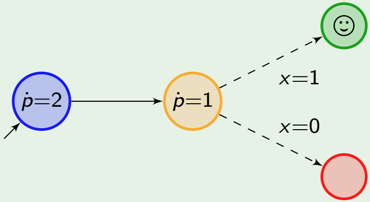
- Optimal strategies do not always exist:



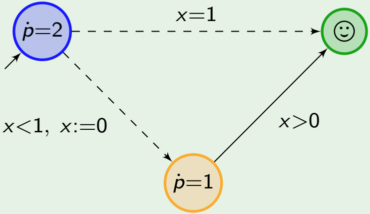
Turn-based 1-clock priced timed games are decidable

Example

- Optimal strategies do not always exist:



- Optimal strategies may require memory:



Turn-based 1-clock priced timed games are decidable

Theorem

Turn-based 1-clock priced timed games always admit ε-optimal winning strategies, and such strategies can be computed.

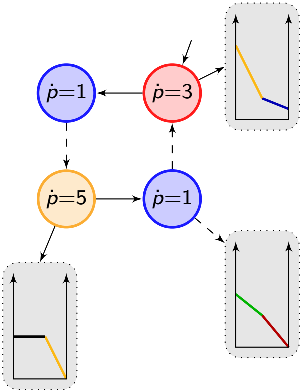
Refs: [1] Bouyer, Cassez, Fleury, Larsen. *Optimal Strategies in Priced Timed Game Automata* (2004).
[2] Bouyer, Larsen, M., Rasmussen. *Almost Optimal Strategies in One-Clock Priced Timed Automata* (2006).

Turn-based 1-clock priced timed games are decidable

Theorem

Turn-based 1-clock priced timed games always admit ε-optimal winning strategies, and such strategies can be computed.

Proof.



Refs: [1] Bouyer, Cassez, Fleury, Larsen. *Optimal Strategies in Priced Timed Game Automata* (2004).

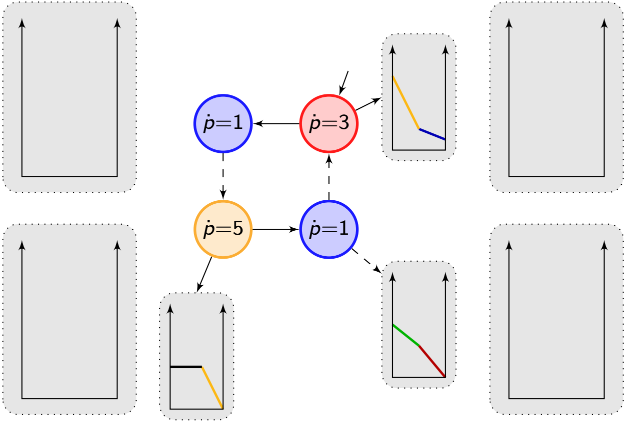
[2] Bouyer, Larsen, M., Rasmussen. *Almost Optimal Strategies in One-Clock Priced Timed Automata* (2006).

Turn-based 1-clock priced timed games are decidable

Theorem

Turn-based 1-clock priced timed games always admit ε-optimal winning strategies, and such strategies can be computed.

Proof.



Refs: [1] Bouyer, Cassez, Fleury, Larsen. *Optimal Strategies in Priced Timed Game Automata* (2004).

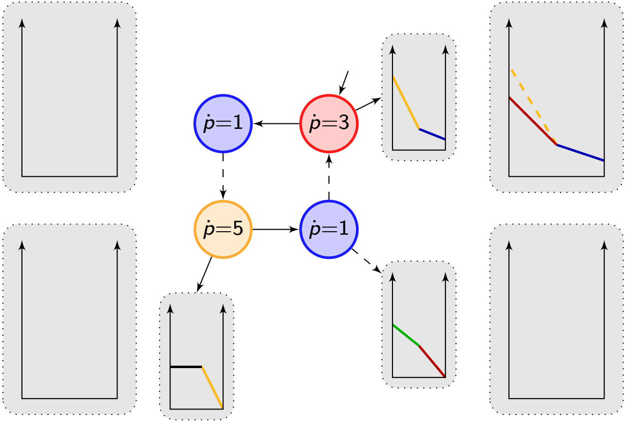
[2] Bouyer, Larsen, M., Rasmussen. *Almost Optimal Strategies in One-Clock Priced Timed Automata* (2006).

Turn-based 1-clock priced timed games are decidable

Theorem

Turn-based 1-clock priced timed games always admit ε-optimal winning strategies, and such strategies can be computed.

Proof.



Refs: [1] Bouyer, Cassez, Fleury, Larsen. *Optimal Strategies in Priced Timed Game Automata* (2004).

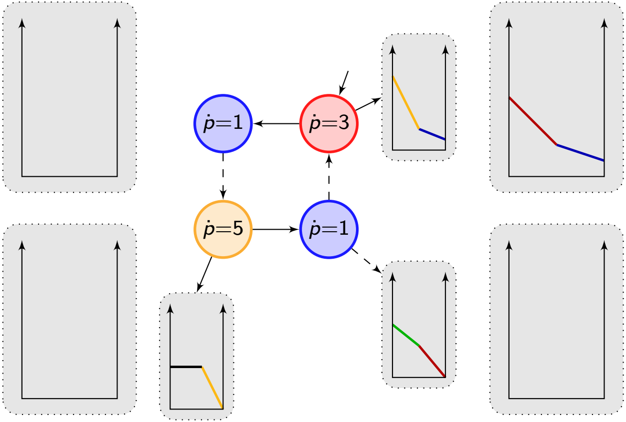
[2] Bouyer, Larsen, M., Rasmussen. *Almost Optimal Strategies in One-Clock Priced Timed Automata* (2006).

Turn-based 1-clock priced timed games are decidable

Theorem

Turn-based 1-clock priced timed games always admit ε-optimal winning strategies, and such strategies can be computed.

Proof.



Refs: [1] Bouyer, Cassez, Fleury, Larsen. *Optimal Strategies in Priced Timed Game Automata* (2004).

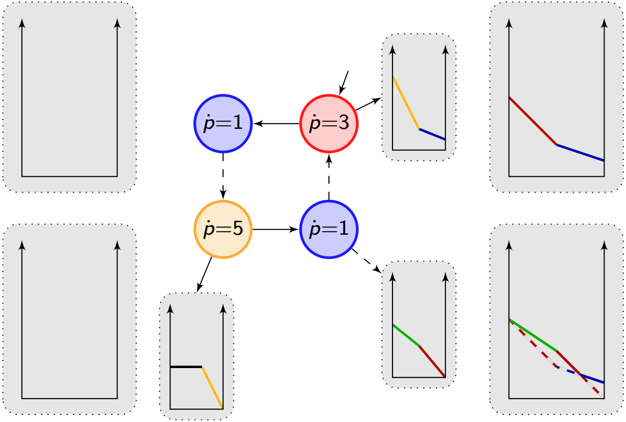
[2] Bouyer, Larsen, M., Rasmussen. *Almost Optimal Strategies in One-Clock Priced Timed Automata* (2006).

Turn-based 1-clock priced timed games are decidable

Theorem

Turn-based 1-clock priced timed games always admit ε-optimal winning strategies, and such strategies can be computed.

Proof.



Refs: [1] Bouyer, Cassez, Fleury, Larsen. *Optimal Strategies in Priced Timed Game Automata* (2004).

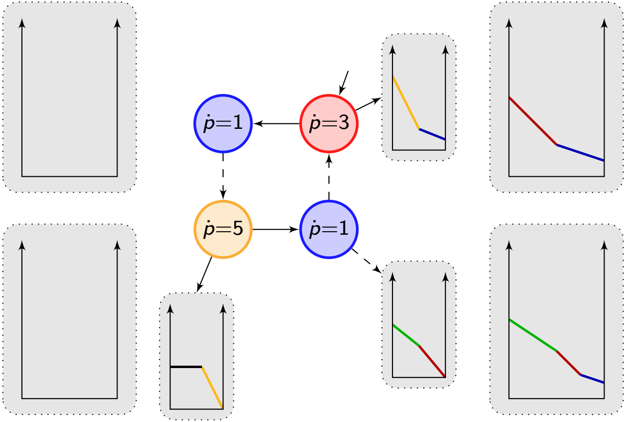
[2] Bouyer, Larsen, M., Rasmussen. *Almost Optimal Strategies in One-Clock Priced Timed Automata* (2006).

Turn-based 1-clock priced timed games are decidable

Theorem

Turn-based 1-clock priced timed games always admit ε-optimal winning strategies, and such strategies can be computed.

Proof.



Refs: [1] Bouyer, Cassez, Fleury, Larsen. *Optimal Strategies in Priced Timed Game Automata* (2004).

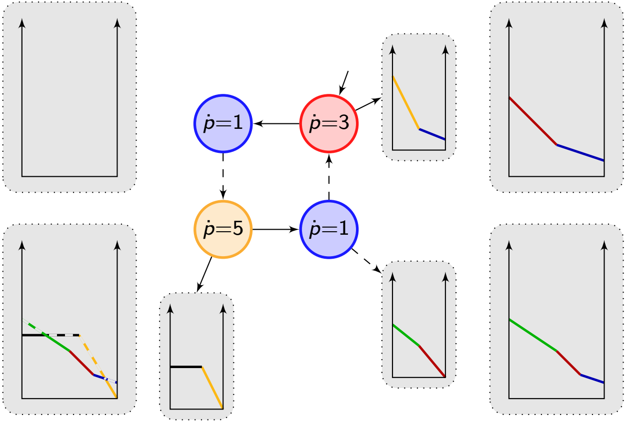
[2] Bouyer, Larsen, M., Rasmussen. *Almost Optimal Strategies in One-Clock Priced Timed Automata* (2006).

Turn-based 1-clock priced timed games are decidable

Theorem

Turn-based 1-clock priced timed games always admit ε-optimal winning strategies, and such strategies can be computed.

Proof.



Refs: [1] Bouyer, Cassez, Fleury, Larsen. *Optimal Strategies in Priced Timed Game Automata* (2004).

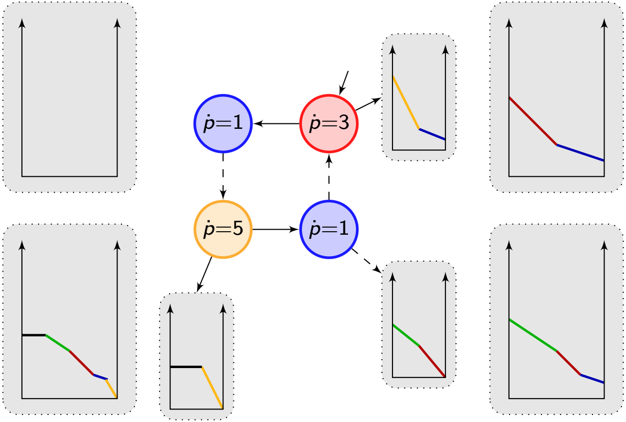
[2] Bouyer, Larsen, M., Rasmussen. *Almost Optimal Strategies in One-Clock Priced Timed Automata* (2006).

Turn-based 1-clock priced timed games are decidable

Theorem

Turn-based 1-clock priced timed games always admit ε-optimal winning strategies, and such strategies can be computed.

Proof.



Refs: [1] Bouyer, Cassez, Fleury, Larsen. *Optimal Strategies in Priced Timed Game Automata* (2004).

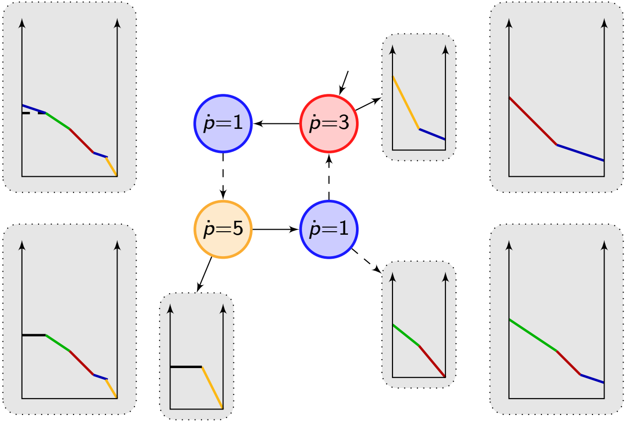
[2] Bouyer, Larsen, M., Rasmussen. *Almost Optimal Strategies in One-Clock Priced Timed Automata* (2006).

Turn-based 1-clock priced timed games are decidable

Theorem

Turn-based 1-clock priced timed games always admit ε-optimal winning strategies, and such strategies can be computed.

Proof.



Refs: [1] Bouyer, Cassez, Fleury, Larsen. *Optimal Strategies in Priced Timed Game Automata* (2004).

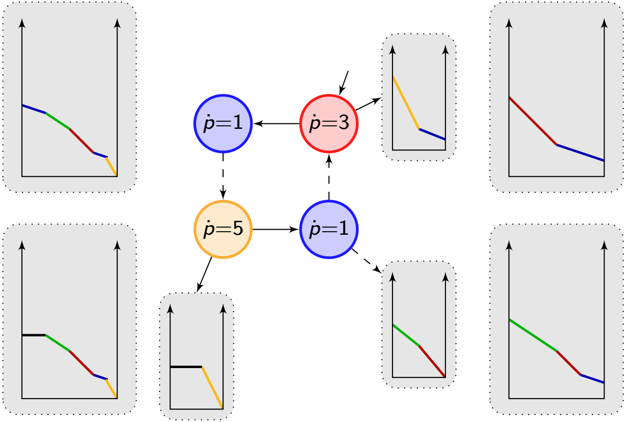
[2] Bouyer, Larsen, M., Rasmussen. *Almost Optimal Strategies in One-Clock Priced Timed Automata* (2006).

Turn-based 1-clock priced timed games are decidable

Theorem

Turn-based 1-clock priced timed games always admit ε-optimal winning strategies, and such strategies can be computed.

Proof.



Refs: [1] Bouyer, Cassez, Fleury, Larsen. *Optimal Strategies in Priced Timed Game Automata* (2004).

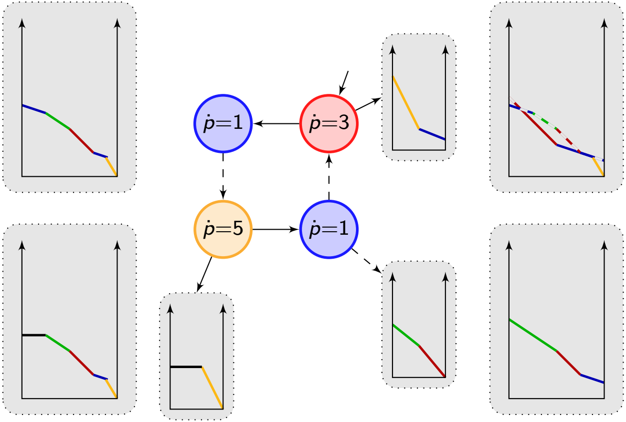
[2] Bouyer, Larsen, M., Rasmussen. *Almost Optimal Strategies in One-Clock Priced Timed Automata* (2006).

Turn-based 1-clock priced timed games are decidable

Theorem

Turn-based 1-clock priced timed games always admit ε-optimal winning strategies, and such strategies can be computed.

Proof.



Refs: [1] Bouyer, Cassez, Fleury, Larsen. *Optimal Strategies in Priced Timed Game Automata* (2004).

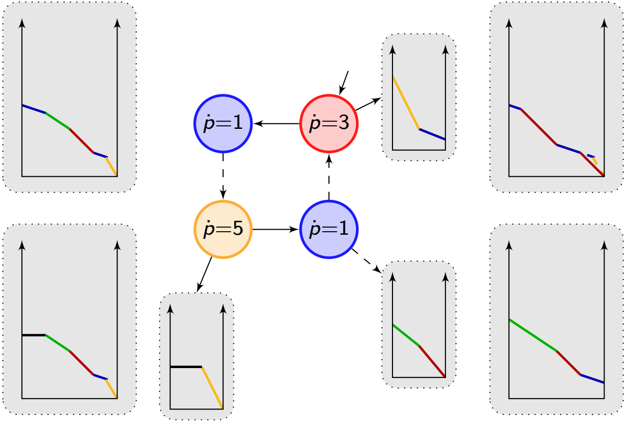
[2] Bouyer, Larsen, M., Rasmussen. *Almost Optimal Strategies in One-Clock Priced Timed Automata* (2006).

Turn-based 1-clock priced timed games are decidable

Theorem

Turn-based 1-clock priced timed games always admit ε-optimal winning strategies, and such strategies can be computed.

Proof.



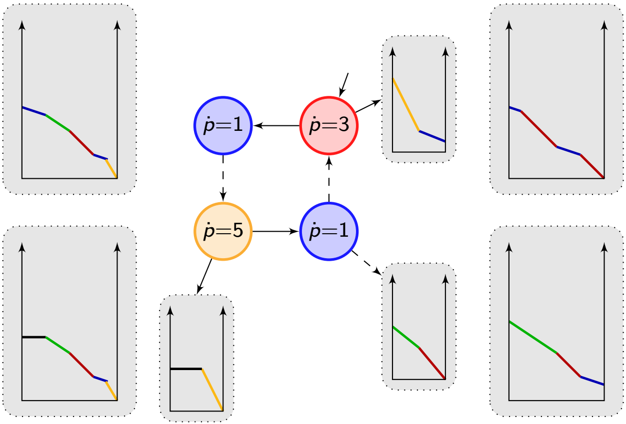
Refs: [1] Bouyer, Cassez, Fleury, Larsen. *Optimal Strategies in Priced Timed Game Automata* (2004).
[2] Bouyer, Larsen, M., Rasmussen. *Almost Optimal Strategies in One-Clock Priced Timed Automata* (2006).

Turn-based 1-clock priced timed games are decidable

Theorem

Turn-based 1-clock priced timed games always admit ε-optimal winning strategies, and such strategies can be computed.

Proof.



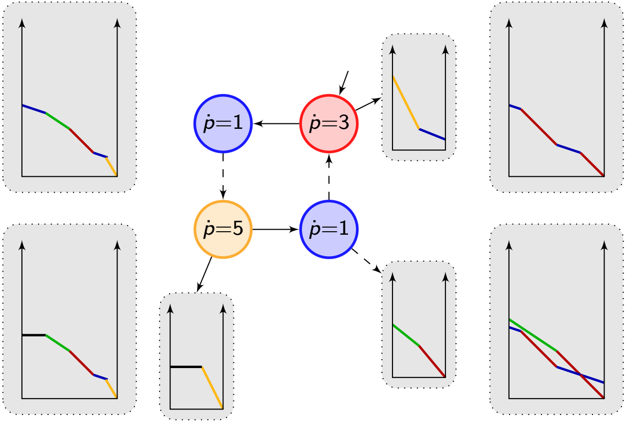
Refs: [1] Bouyer, Cassez, Fleury, Larsen. *Optimal Strategies in Priced Timed Game Automata* (2004).
[2] Bouyer, Larsen, M., Rasmussen. *Almost Optimal Strategies in One-Clock Priced Timed Automata* (2006).

Turn-based 1-clock priced timed games are decidable

Theorem

Turn-based 1-clock priced timed games always admit ε-optimal winning strategies, and such strategies can be computed.

Proof.



Refs: [1] Bouyer, Cassez, Fleury, Larsen. *Optimal Strategies in Priced Timed Game Automata* (2004).

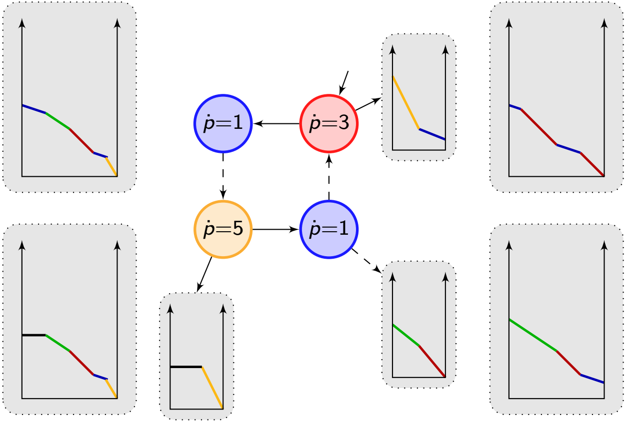
[2] Bouyer, Larsen, M., Rasmussen. *Almost Optimal Strategies in One-Clock Priced Timed Automata* (2006).

Turn-based 1-clock priced timed games are decidable

Theorem

Turn-based 1-clock priced timed games always admit ε-optimal winning strategies, and such strategies can be computed.

Proof.



Refs: [1] Bouyer, Cassez, Fleury, Larsen. *Optimal Strategies in Priced Timed Game Automata* (2004).

[2] Bouyer, Larsen, M., Rasmussen. *Almost Optimal Strategies in One-Clock Priced Timed Automata* (2006).

Turn-based 1-clock priced timed games are decidable

Theorem

Turn-based 1-clock priced timed games always admit ε-optimal winning strategies, and such strategies can be computed.

Proof.

- The procedure terminates;
- There is a positive granularity for with the region abstraction is correct;
- The optimal cost functions are piecewise affine, continuous, decreasing functions. Their slopes are rates of the automaton.

□

Refs: [1] Bouyer, Cassez, Fleury, Larsen. *Optimal Strategies in Priced Timed Game Automata* (2004).
[2] Bouyer, Larsen, M., Rasmussen. *Almost Optimal Strategies in One-Clock Priced Timed Automata* (2006).

Outline of the talk

1. Introduction

2. Timed automata with observers

3. Resource-optimization problems
   - Optimal reachabililty
   - Weighted temporal logics
   - Optimal strategies

4. Resource-management problems

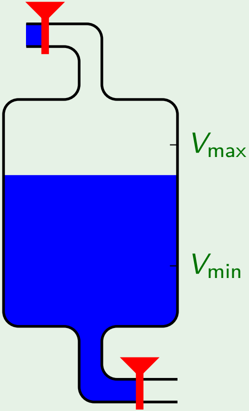
5. Conclusions and perspectives

Managing resources

Example

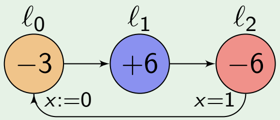
In some cases, resources can both be consumed and regained.

The aim is then to keep the level of resources within given bounds.



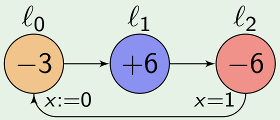
Managing resources

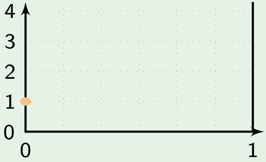
Example



Managing resources

Example



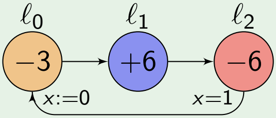


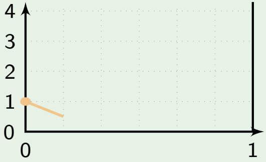
Three variants of the problem:

1. lower bound: the aim is to maintain the level of resources above a given bound.

Managing resources

Example



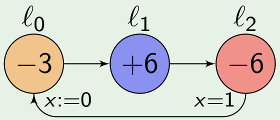


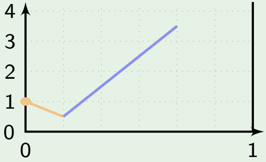
Three variants of the problem:

1. lower bound: the aim is to maintain the level of resources above a given bound.

Managing resources

Example



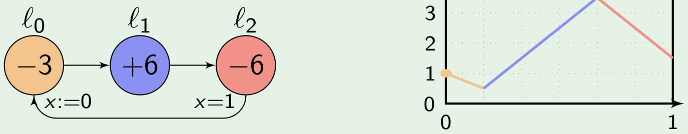


Three variants of the problem:

1. lower bound: the aim is to maintain the level of resources above a given bound.

Managing resources

Example

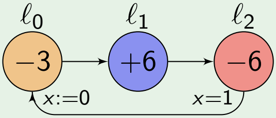


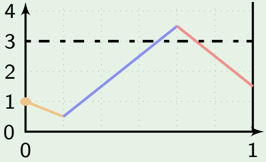
Three variants of the problem:

1. lower bound: the aim is to maintain the level of resources above a given bound.

Managing resources

Example



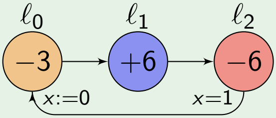


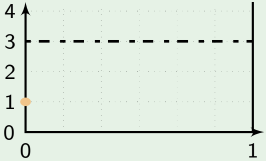
Three variants of the problem:

1. lower bound: the aim is to maintain the level of resources above a given bound.
2. interval: the aim is to keep the level of resources within an interval.

Managing resources

Example



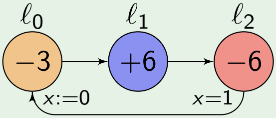


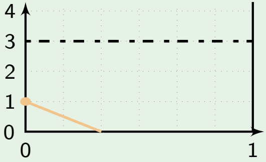
Three variants of the problem:

1. lower bound: the aim is to maintain the level of resources above a given bound.
2. interval: the aim is to keep the level of resources within an interval.

Managing resources

Example



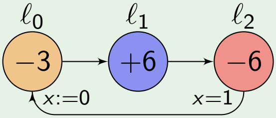


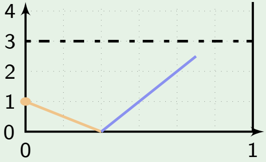
Three variants of the problem:

1. lower bound: the aim is to maintain the level of resources above a given bound.
2. interval: the aim is to keep the level of resources within an interval.

Managing resources

Example



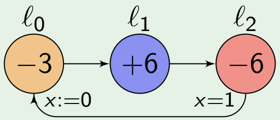


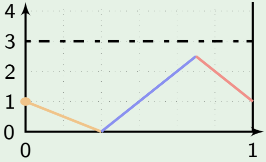
Three variants of the problem:

1. lower bound: the aim is to maintain the level of resources above a given bound.
2. interval: the aim is to keep the level of resources within an interval.

Managing resources

Example



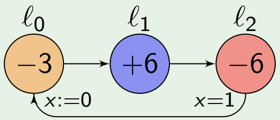


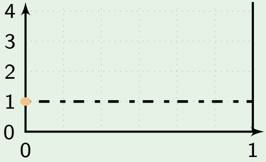
Three variants of the problem:

1. lower bound: the aim is to maintain the level of resources above a given bound.
2. interval: the aim is to keep the level of resources within an interval.

Managing resources

Example



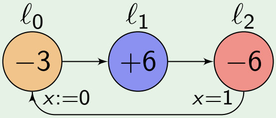


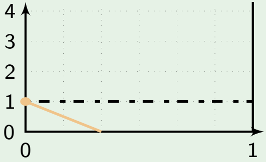
Three variants of the problem:

1. lower bound: the aim is to maintain the level of resources above a given bound.
2. interval: the aim is to keep the level of resources within an interval.
3. lower bound with finite capacity: the aim is to keep the level of resources above a given lower bound, but with a finite capacity.

Managing resources

Example



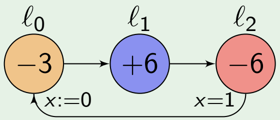


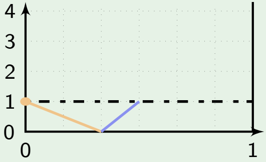
Three variants of the problem:

1. lower bound: the aim is to maintain the level of resources above a given bound.
2. interval: the aim is to keep the level of resources within an interval.
3. lower bound with finite capacity: the aim is to keep the level of resources above a given lower bound, but with a finite capacity.

Managing resources

Example



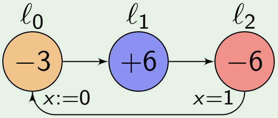


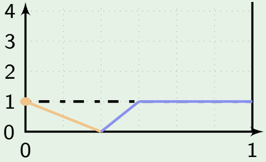
Three variants of the problem:

1. lower bound: the aim is to maintain the level of resources above a given bound.
2. interval: the aim is to keep the level of resources within an interval.
3. lower bound with finite capacity: the aim is to keep the level of resources above a given lower bound, but with a finite capacity.

Managing resources

Example





Three variants of the problem:

1. lower bound: the aim is to maintain the level of resources above a given bound.
2. interval: the aim is to keep the level of resources within an interval.
3. lower bound with finite capacity: the aim is to keep the level of resources above a given lower bound, but with a finite capacity.

Results in the untimed case

Theorem

In the untimed case, the following results hold:

	existential problem	***universal problem***	***games***
Lower bound	$\in$ PTIME	$\in$ PTIME	$\in$ UP $\cap$ coUP PTIME-*hard*
Lower bound, finite capacity	$\in$ PTIME	$\in$ PTIME	$\in$ NP PTIME-*hard*
Interval	$\in$ PSPACE NP-*hard*	$\in$ PTIME	EXPTIME-*c.*

Refs: [1] Bouyer, Fahrenberg, Larsen, M., Srba. *Infinite Runs in Weighted Timed Automata with Energy Constraints* (2008).

Results in the 1-clock case

Theorem

In the 1-clock case, the existence of an infinite run with resource level above a given lower bound is decidable in EXPTIME.

Refs: [1] Bouyer, Fahrenberg, Larsen, M. *Timed Automata with Observers under Energy Constraints* (2010).

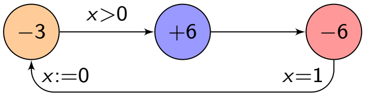
Results in the 1-clock case

Theorem

In the 1-clock case, the existence of an infinite run with resource level above a given lower bound is decidable in EXPTIME.

Proof.

- Corner-point abstraction:



Refs: [1] Bouyer, Fahrenberg, Larsen, M. *Timed Automata with Observers under Energy Constraints* (2010).

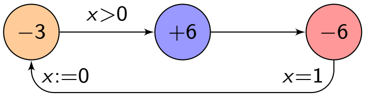
Results in the 1-clock case

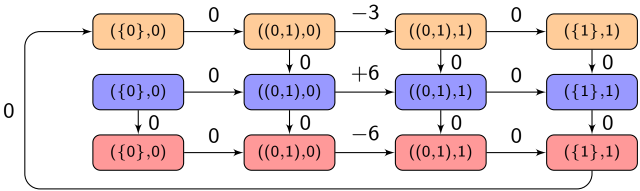
Theorem

In the 1-clock case, the existence of an infinite run with resource level above a given lower bound is decidable in EXPTIME.

Proof.

- Corner-point abstraction:





Refs: [1] Bouyer, Fahrenberg, Larsen, M. *Timed Automata with Observers under Energy Constraints* (2010).

Results in the 1-clock case

Theorem

In the 1-clock case, the existence of an infinite run with resource level above a given lower bound is decidable in EXPTIME.

Proof.

- Corner-point abstraction: Only correct if no discrete costs!

Refs: [1] Bouyer, Fahrenberg, Larsen, M. *Timed Automata with Observers under Energy Constraints* (2010).

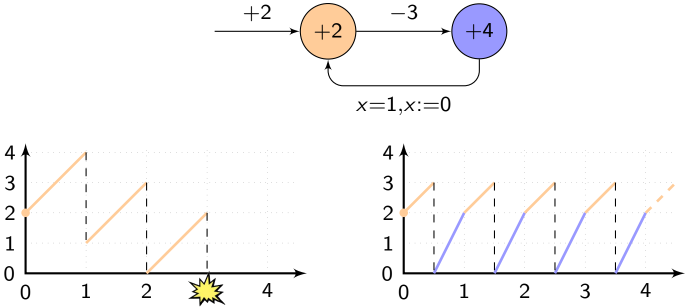
Results in the 1-clock case

Theorem

In the 1-clock case, the existence of an infinite run with resource level above a given lower bound is decidable in EXPTIME.

Proof.

- Corner-point abstraction: Only correct if no discrete costs!



Refs: [1] Bouyer, Fahrenberg, Larsen, M. *Timed Automata with Observers under Energy Constraints* (2010).

Results in the 1-clock case

Theorem

In the 1-clock case, the existence of an infinite run with resource level above a given lower bound is decidable in EXPTIME.

Proof.

- Corner-point abstraction: Only correct if no discrete costs!
- In the presence of discrete costs:

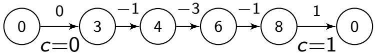
Results in the 1-clock case

Theorem

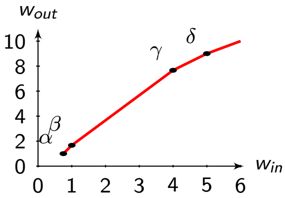
In the 1-clock case, the existence of an infinite run with resource level above a given lower bound is decidable in EXPTIME.

Proof.

- Corner-point abstraction: Only correct if no discrete costs!
- In the presence of discrete costs:



  - compute optimal final resource-level along a non-resetting path;



Refs: [1] Bouyer, Fahrenberg, Larsen, M. *Timed Automata with Observers under Energy Constraints* (2010).

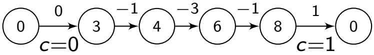
Results in the 1-clock case

Theorem

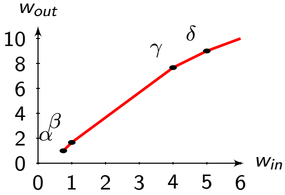
In the 1-clock case, the existence of an infinite run with resource level above a given lower bound is decidable in EXPTIME.

Proof.

- Corner-point abstraction: Only correct if no discrete costs!
- In the presence of discrete costs:



  - compute optimal final resource-level along a non-resetting path;
  - compose the resulting functions for general paths.



□

Refs: [1] Bouyer, Fahrenberg, Larsen, M. *Timed Automata with Observers under Energy Constraints* (2010).

Results in the 1-clock case

Theorem

In the 1-clock case, the existence of a strategy for maintaining the resource level within a given interval is undecidable.

Refs: [1] Bouyer, Fahrenberg, Larsen, M., Srba. *Infinite Runs in Weighted Timed Automata with Energy Constraints* (2008).

Results in the 1-clock case

Theorem

In the 1-clock case, the existence of a strategy for maintaining the resource level within a given interval is undecidable.

Proof.

- Encoding of a two-counter machine: both counters are stored in one cost, as $\ell = 5 - 2^{-c_1} \cdot 3^{-c_2}$.

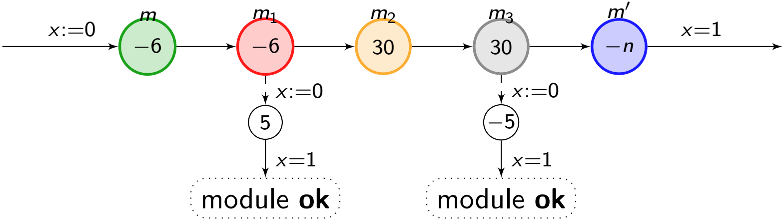
Results in the 1-clock case

Theorem

In the 1-clock case, the existence of a strategy for maintaining the resource level within a given interval is undecidable.

Proof.

- Encoding of a two-counter machine: both counters are stored in one cost, as $\ell = 5 - 2^{-c_1} \cdot 3^{-c_2}$.
- The following module is used to increment and decrement:



Refs: [1] Bouyer, Fahrenberg, Larsen, M., Srba. *Infinite Runs in Weighted Timed Automata with Energy Constraints* (2008).

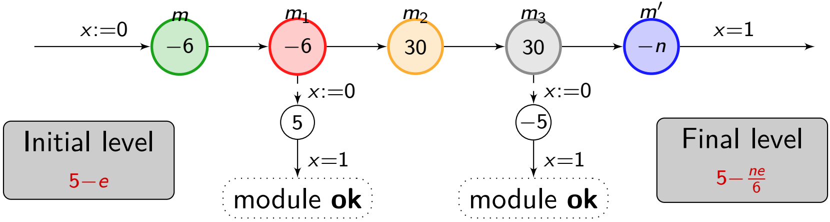
Results in the 1-clock case

Theorem

In the 1-clock case, the existence of a strategy for maintaining the resource level within a given interval is undecidable.

Proof.

- Encoding of a two-counter machine: both counters are stored in one cost, as $\ell = 5 - 2^{-c_1} \cdot 3^{-c_2}$.
- The following module is used to increment and decrement:



Refs: [1] Bouyer, Fahrenberg, Larsen, M., Srba. *Infinite Runs in Weighted Timed Automata with Energy Constraints* (2008).

Outline of the talk

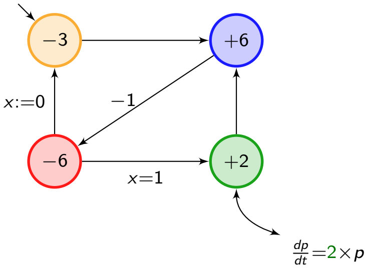
1. Introduction
2. Timed automata with observers
3. Resource-optimization problems
   - Optimal reachabililty
   - Weighted temporal logics
   - Optimal strategies
4. Resource-management problems
5. Conclusions and perspectives

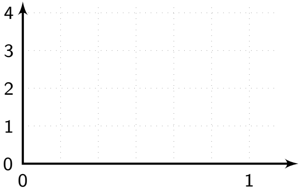
Conclusions and perspectives

- Weighted timed automata are a powerful formalism for modeling resources:
  - expressive enough for many applications;
  - several problems remain decidable;
  - some algorithms can be made symbolic and are implemented in Uppaal CORA.

Conclusions and perspectives

- Weighted timed automata are a powerful formalism for modeling resources:
  - expressive enough for many applications;
  - several problems remain decidable;
  - some algorithms can be made symbolic and are implemented in Uppaal CORA.
- Many open problems:
  - energy constraints for automata with several clocks;
  - timed automata with observers having richer dynamics.





Conclusions and perspectives

- Weighted timed automata are a powerful formalism for modeling resources:
  - expressive enough for many applications;
  - several problems remain decidable;
  - some algorithms can be made symbolic and are implemented in Uppaal CORA.
- Many open problems:
  - energy constraints for automata with several clocks;
  - timed automata with observers having richer dynamics.