On the reduction of energy consumption

Patricia Bouyer

LSV – CNRS & ENS de Cachan – France

Oxford University Computing Laboratory – UK
On the reduction of energy consumption
— A timed automaton approach —

Patricia Bouyer

LSV – CNRS & ENS de Cachan – France
Oxford University Computing Laboratory – UK
Outline

1. Introduction

2. Timed automata with costs

3. Optimal timed games

4. Conclusion
A starting example
Natural questions

- Can I reach Pontivy from Oxford?
Natural questions

- Can I reach Pontivy from Oxford?
- What is the **minimal time** to reach Pontivy from Oxford?
Natural questions

- Can I reach Pontivy from Oxford?
- What is the **minimal time** to reach Pontivy from Oxford?
- What is the **minimal fuel consumption** to reach Pontivy from Oxford?
Natural questions

- Can I reach Pontivy from Oxford?
- What is the minimal time to reach Pontivy from Oxford?
- What is the minimal fuel consumption to reach Pontivy from Oxford?
- What if there is an unexpected event?
A first model of the system
Can I reach Pontivy from Oxford?

This is a reachability question in a finite graph: Yes, I can!
A second model of the system
How long will that take?

It is a reachability (and optimization) question in a timed automaton: at least 350mn = 5h50mn!
Timed automata

Theorem

[AD90]

The reachability problem is decidable (and PSPACE-complete) for timed automata.
Timed automata

**Theorem [AD90]**

The reachability problem is decidable (and PSPACE-complete) for timed automata.
Timed automata

Theorem [AD90]
The reachability problem is decidable (and PSPACE-complete) for timed automata.
The region abstraction

......→ | ......→

......→ time elapsing

- - → reset to 0
Time-optimal reachability

Theorem [CY92]
The time-optimal reachability problem is decidable (and PSPACE-complete) for timed automata.
A third model of the system
How much fuel will I use?

It is a *quantitative* (optimization) problem in a *priced timed automaton*: at least 68 anti-planet units!
Outline

1. Introduction

2. Timed automata with costs

3. Optimal timed games

4. Conclusion
2001: the space odyssey
2001: the space odyssey of weighted timed automata
2001: the space odyssey of weighted/priced timed automata
A simple example

\[
\begin{align*}
\ell_0 & \xrightarrow{d\text{cost} \over dt = 5} \ell_1 \quad x \leq 2, c, y := 0 \\
\ell_1 & \xrightarrow{d\text{cost} \over dt = 10} \ell_2 \quad (y = 0) \\
\ell_1 & \xrightarrow{d\text{cost} \over dt = 1} \ell_3 \\
\ell_2 & \xrightarrow{x = 2, c, \text{cost} = +1} \ell_3 \\
\ell_2 & \xrightarrow{x = 2, c, \text{cost} = +7} \text{smiley} \\
\ell_3 & \xrightarrow{u} \ell_2 \\
\ell_3 & \xrightarrow{u} \ell_1
\end{align*}
\]

Question:
What is the optimal cost for reaching \text{smiley}?

\[
\inf_{0 \leq t \leq 2} \min \left( 5t + 10(2-t) + 1, 5t + (2-t) + 7 \right) = 9
\]

Strategy:
Leave immediately \( \ell_0 \), go to \( \ell_3 \), and wait there \( 2 \) t.u.
A simple example

\[
\begin{align*}
\ell_0 \xrightarrow{d\text{cost}/dt = 5} \ell_1 & \quad \text{(} x \leq 2, c, y := 0 \text{)} \\
\ell_1 & \quad (y = 0) \\
\ell_2 \xrightarrow{d\text{cost}/dt = 10} & \quad x = 2, c, \text{cost} = +1 \\
\ell_3 \xrightarrow{d\text{cost}/dt = 1} & \quad x = 2, c, \text{cost} = +7
\end{align*}
\]

\[
\begin{align*}
(\ell_0, (0, 0)) & \overset{1.3}{\rightarrow} (\ell_0, (1.3, 1.3)) \overset{<}{\leftarrow} (\ell_1, (1.3, 0)) \xrightarrow{u} (\ell_3, (1.3, 0)) \overset{0.7}{\rightarrow} (\ell_3, (2, 0.7)) \overset{c}{\rightarrow} \smiley
\end{align*}
\]

\[
\text{cost : } 6.5 + 0 + 0 + 0.7 + 7 = 14.2
\]
A simple example

Question: what is the optimal cost for reaching 😊?
A simple example

\[
\begin{align*}
\ell_0 & \quad x \leq 2, c, y := 0 \quad \frac{dcost}{dt} = 5 \\
\ell_1 & \quad (y = 0) \quad \frac{dcost}{dt} = 10 \\
\ell_2 & \quad u \quad x = 2, c, \text{cost} = +1 \\
\ell_3 & \quad u \quad x = 2, c, \text{cost} = +7 \\
& \quad \text{smiley}
\end{align*}
\]

**Question:** what is the optimal cost for reaching ☺?

\[
5t + 10(2 - t) + 1 \quad , \quad 5t + (2 - t) + 7
\]
A simple example

Question: what is the optimal cost for reaching ☺?

\[
\min \left( 5t + 10(2 - t) + 1 , \ 5t + (2 - t) + 7 \right)
\]
A simple example

**Question:** what is the optimal cost for reaching 😊?

\[
\inf_{0 \leq t \leq 2} \min \left( 5t + 10(2 - t) + 1, 5t + (2 - t) + 7 \right) = 9
\]
A simple example

Question: what is the optimal cost for reaching 😊?

$$\inf_{0 \leq t \leq 2} \min (5t + 10(2 - t) + 1, 5t + (2 - t) + 7) = 9$$

→ strategy: leave immediately $\ell_0$, go to $\ell_3$, and wait there 2 t.u.
The idea “go through corners” extends in the general case.

**Theorem**

Optimal reachability is decidable in timed automata.
The idea “go through corners” extends in the general case.

**Theorem**

Optimal reachability is decidable in timed automata. It is PSPACE-complete.
The region abstraction is not fine enough

- - - - - time elapsing

- - - reset to 0
The corner-point abstraction
Mean-Cost Optimization

Question: How to minimize $\lim_{n \to +\infty} \frac{\text{accumulated cost}(\pi_n)}{\text{accumulated reward}(\pi_n)}$?
An example

Two machines $M_1(D = 3, P = 3, G = 4, p = 5, g = 3)$ and $M_2(D = 6, P = 3, G = 2, p = 5, g = 2)$. An operator $O(4)$. 

Schedule with ratio 1.455

Schedule with ratio 1.478
Mean-cost optimization

Theorem [BBL04]

The mean-cost optimization problem is decidable (and PSPACE-complete) for priced timed automata.

☞ The corner-point abstraction is sound and complete.
Uppaal Cora

A branch of Uppaal for cost optimal reachability
Outline

1. Introduction

2. Timed automata with costs

3. Optimal timed games

4. Conclusion
What if an unexpected event happens?
What if an unexpected event happens?

Optimal timed games
What if an unexpected event happens?

Flight cancelled!

modelled as timed games
A simple example of timed games

\[
\begin{align*}
\ell_0 & \quad \text{cost}\frac{d}{dt} = 5 \\
\ell_1 & \quad (y=0) \\
\ell_2 & \quad \text{cost}\frac{d}{dt} = 10 \\
\ell_3 & \quad \text{cost}\frac{d}{dt} = 1 \\
\ell_{\infty} & \quad x=2, c, \text{cost}\frac{d}{dt} = 1 \\
\end{align*}
\]
A simple example of timed games

\[ d\text{cost} \frac{dt}{dt} = 5 \]

\[ x \leq 2, c, y:=0 \]

\[ d\text{cost} \frac{dt}{dt} = 10 \]

\[ x = 2, c, \text{cost}=+1 \]

\[ d\text{cost} \frac{dt}{dt} = 1 \]

\[ x = 2, c, \text{cost}=+7 \]
Decidability of timed games

Theorem [AMPS98] [HK99] safety and reachability control in timed automata are decidable and EXPTIME-complete. (the attractor is computable...) classical regions are sufficient for solving such problems.
Decidability of timed games

Theorem [AMPS98] [HK99]
Safety and reachability control in timed automata are decidable and EXPTIME-complete.
Decidability of timed games

**Theorem [AMPS98] [HK99]**

Safety and reachability control in timed automata are decidable and EXPTIME-complete.

(the attractor is computable...)
Decidability of timed games

**Theorem** [AMPS98] [HK99]

Safety and reachability control in timed automata are decidable and EXPTIME-complete.

(the attractor is computable...)

classical regions are sufficient for solving such problems
A forward on-the-fly algorithm for solving reachability timed games implemented as a branch of Uppaal
Uppaal Tiga

A forward on-the-fly algorithm for solving reachability timed games implemented as a branch of Uppaal
Optimal timed games

Back to the simple example

\begin{align*}
\ell_0 \quad & \quad x \leq 2, c, y := 0 \\
\ell_1 \quad & \quad (y = 0) \\
\ell_2 \quad & \quad x = 2, c, \text{cost} = +1 \\
\ell_3 \quad & \quad x = 2, c, \text{cost} = +7 \\
\end{align*}

- \( \frac{d\text{cost}}{dt} = 5 \) on \( \ell_0 \)
- \( \frac{d\text{cost}}{dt} = 10 \) on \( \ell_2 \)
- \( \frac{d\text{cost}}{dt} = 1 \) on \( \ell_3 \)

\text{Question: what is the optimal cost we can ensure in state}\ \ell_0? \Rightarrow \inf_{0 \leq t \leq 2} \max (5t + 10(2-t) + 1, 5t + (2-t) + 7) = 14 + \frac{1}{3}

\text{Strategy: wait in } \ell_0, \text{ and when } t = 4/3, \text{ go to } \ell_1.

\text{How to automatically compute such optimal costs?}

\text{How to synthesize optimal strategies (if one exists?)}
Back to the simple example

\[ \frac{d\text{cost}}{dt} = 5 \]

\[ (y=0) \]

\[ x \leq 2, c, y := 0 \]

\[ \frac{d\text{cost}}{dt} = 10 \]

\[ x = 2, c, \text{cost} = +1 \]

\[ \frac{d\text{cost}}{dt} = 1 \]

\[ x = 2, c, \text{cost} = +7 \]

Question: what is the optimal cost we can ensure in state \( \ell_0 \)?

\[ \inf_{0 \leq t \leq 2} \max \left( 5t + 10(2-t) + 1, 5t + (2-t) + 7 \right) = 14 + \frac{1}{3} \]

Strategy: wait in \( \ell_0 \), and when \( t = \frac{4}{3} \), go to \( \ell_1 \)

How to automatically compute such optimal costs?

How to synthesize optimal strategies (if one exists)?
Back to the simple example

Question: what is the optimal cost we can ensure in state $\ell_0$?
Back to the simple example

Question: what is the optimal cost we can ensure in state $l_0$?

$$5t + 10(2 - t) + 1, \; 5t + (2 - t) + 7$$
Back to the simple example

**Question:** what is the optimal cost we can ensure in state $\ell_0$?

$$\max \left( 5t + 10(2 - t) + 1 \ , \ 5t + (2 - t) + 7 \right)$$
Back to the simple example

**Question:** what is the optimal cost we can ensure in state $\ell_0$?

$$\inf_{0 \leq t \leq 2} \max \left( 5t + 10(2 - t) + 1, \ 5t + (2 - t) + 7 \right) = 14 + \frac{1}{3}$$
Back to the simple example

Question: what is the optimal cost we can ensure in state \( \ell_0 \)?

\[
\inf_{0 \leq t \leq 2} \max (5t + 10(2 - t) + 1, 5t + (2 - t) + 7) = 14 + \frac{1}{3}
\]

→ strategy: wait in \( \ell_0 \), and when \( t = \frac{4}{3} \), go to \( \ell_1 \)
Back to the simple example

Question: what is the optimal cost we can ensure in state $\ell_0$?

$$\inf_{0 \leq t \leq 2} \max \left( 5t + 10(2 - t) + 1, 5t + (2 - t) + 7 \right) = 14 + \frac{1}{3}$$

→ strategy: wait in $\ell_0$, and when $t = \frac{4}{3}$, go to $\ell_1$

▶ How to automatically compute such optimal costs?
Back to the simple example

\[
\begin{align*}
\ell_0 & \xrightarrow{x \leq 2, c, y:=0} \ell_1 \\
\frac{d\text{cost}}{dt} &= 5 \\
\ell_1 & \xrightarrow{(y=0)} \ell_2 \\
\frac{d\text{cost}}{dt} &= 10 \\
\ell_2 & \xrightarrow{x=2, c, \text{cost}=-1} \ell_1 \\
\ell_3 & \xrightarrow{x=2, c, \text{cost}=-7} \ell_1 \\
\frac{d\text{cost}}{dt} &= 1
\end{align*}
\]

Question: what is the optimal cost we can ensure in state \( \ell_0 \)?

\[
\inf_{0 \leq t \leq 2} \max \left( 5t + 10(2 - t) + 1, \ 5t + (2 - t) + 7 \right) = 14 + \frac{1}{3}
\]

→ strategy: wait in \( \ell_0 \), and when \( t = \frac{4}{3} \), go to \( \ell_1 \)

- How to automatically compute such optimal costs?
- How to synthesize optimal strategies (if one exists)?
A fairly hot topic!

- optimal time is computable in timed games
A fairly hot topic!

- optimal time is computable in timed games
- case of acyclic games
A fairly hot topic!

- optimal time is computable in timed games
- case of acyclic games
- general case
  - complexity of $k$-step games
  - under a strongly non-zeno assumption, optimal cost is computable
A fairly hot topic!

- optimal time is computable in timed games

- case of acyclic games

- general case
  - complexity of $k$-step games
  - under a strongly non-zeno assumption, optimal cost is computable

- general case
  - structural properties of strategies (e.g. memory)
  - under a strongly non-zeno assumption, optimal cost is computable
A fairly hot topic!

- general case
  - with five clocks, optimal cost is not computable!
  - with one clock and one stopwatch cost, optimal cost is computable
A fairly hot topic!

- general case
  - with five clocks, optimal cost is not computable!
  - with one clock and one stopwatch cost, optimal cost is computable

- general case
  - with three clocks, optimal cost is not computable
A fairly hot topic!

- general case
  - with five clocks, optimal cost is not computable!
  - with one clock and one stopwatch cost, optimal cost is computable
- general case
  - with three clocks, optimal cost is not computable
- the single-clock case
  - with one clock, optimal cost is computable
Why is that hard?

Given two clocks $x$ and $y$, we can check whether $y = 2x$. 
Optimal timed games

Why is that hard?

Given two clocks $x$ and $y$, we can check whether $y = 2x$

The cost is increased by $x_0$

The cost is increased by $1 - x_0$
Why is that hard?

Given two clocks $x$ and $y$, we can check whether $y = 2x$
Why is that hard?

Given two clocks $x$ and $y$, we can check whether $y = 2x$

- In the first branch, cost $= 2x_0 + (1 - y_0) + 2$

- In the second branch, cost $= 2(1 - x_0) + y_0 + 1$
Why is that hard?

Given two clocks $x$ and $y$, we can check whether $y = 2x$

- In the first branch, cost = $2x_0 + (1 - y_0) + 2$
- In the second branch, cost = $2(1 - x_0) + y_0 + 1$
Why is that hard?

Given two clocks $x$ and $y$, we can check whether $y = 2x$

- In $\bigcirc$, cost = $2x_0 + (1 - y_0) + 2$
- In $\bigcirc$, cost = $2(1 - x_0) + y_0 + 1$

- if $y_0 < 2x_0$, player 2 chooses the first branch: cost $> 3$
Why is that hard?

Given two clocks $x$ and $y$, we can check whether $y = 2x$

- In $\bigcirc$, cost = $2x_0 + (1 - y_0) + 2$
- In $\bigcirc$, cost = $2(1 - x_0) + y_0 + 1$

- If $y_0 < 2x_0$, player 2 chooses the first branch: cost > 3
  - if $y_0 > 2x_0$, player 2 chooses the second branch: cost > 3
Why is that hard?

Given two clocks $x$ and $y$, we can check whether $y = 2x$.

\[
\begin{align*}
z = 0 & \quad \text{Add}^+(x) \quad \text{Add}^+(x) \quad \text{Add}^-(y) \quad +2 \\
z = 0 & \quad \text{Add}^-(x) \quad \text{Add}^-(x) \quad \text{Add}^+(y) \quad +1
\end{align*}
\]

- If $y_0 < 2x_0$, player 2 chooses the first branch: cost $> 3$
- If $y_0 > 2x_0$, player 2 chooses the second branch: cost $> 3$
- If $y_0 = 2x_0$, in both branches, cost $= 3$

In $\text{Smiley}$, cost $= 2x_0 + (1 - y_0) + 2$

In $\text{Sad}$, cost $= 2(1 - x_0) + y_0 + 1$
Why is that hard?

Given two clocks $x$ and $y$, we can check whether $y = 2x$

$\begin{align*}
z &= 0 \\
\text{Add}^+ (x) &\rightarrow \text{Add}^+ (x) &\rightarrow \text{Add}^- (y) &\rightarrow +2 \\
\text{Add}^- (x) &\rightarrow \text{Add}^- (x) &\rightarrow \text{Add}^+ (y) &\rightarrow +1
\end{align*}$

- In $\smiley$, cost $= 2x_0 + (1 - y_0) + 2$
- In $\smiley$, cost $= 2(1 - x_0) + y_0 + 1$

- if $y_0 < 2x_0$, player 2 chooses the first branch: cost $> 3$
- if $y_0 > 2x_0$, player 2 chooses the second branch: cost $> 3$
- if $y_0 = 2x_0$, in both branches, cost $= 3$

- Player 1 has a winning strategy with cost $\leq 3$ iff $y_0 = 2x_0$
Outline

1. Introduction
2. Timed automata with costs
3. Optimal timed games
4. Conclusion
Conclusion

Priced timed automata, a model and framework to represent quantitative constraints on timed systems.
Conclusion

**Priced timed automata**, a model and framework to represent quantitative constraints on timed systems.

**Not mentioned here**

- all works on model-checking issues (extensions of CTL, LTL)
  - very few decidability results
Conclusion

Priced timed automata, a model and framework to represent quantitative constraints on timed systems.

Not mentioned here

- all works on model-checking issues (extensions of CTL, LTL)
  - very few decidability results

Further work

- approximate optimal timed games to circumvent undecidability results