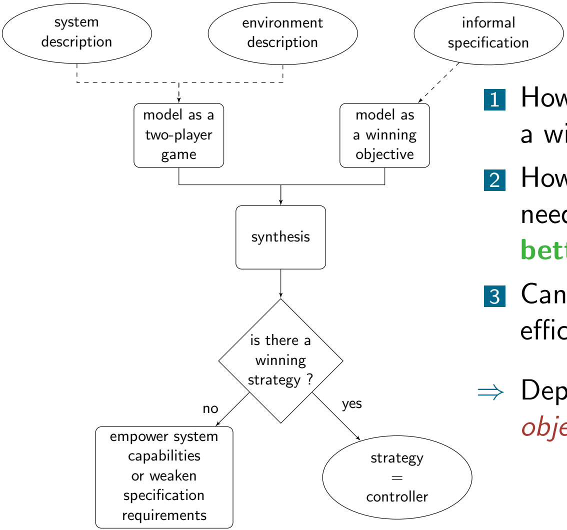
# Games with Window Quantitative Objectives

Mickael Randour (LSV - CNRS & ENS Cachan)

*Based on joint work with Krishnendu Chatterjee (IST Austria), Laurent Doyen (LSV - CNRS & ENS Cachan) and Jean-François Raskin (ULB).*

25.02.2015 - *Frontiers of Formal Methods 2015*

# General context: strategy synthesis in quantitative games

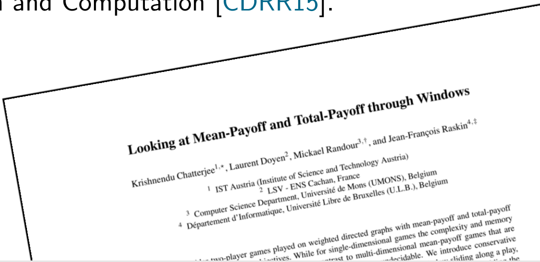



1. How complex is it to decide if a winning strategy exists?
2. How complex such a strategy needs to be? **Simpler is better**.
3. Can we synthesize one efficiently?

⇒ Depends on the *winning objective*.

# Aim of this talk

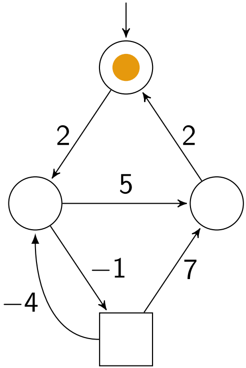
- **New family of quantitative objectives**, based on mean-payoff (MP) and total-payoff (TP).
- Convince you of its *advantages* and *usefulness*.
- No technical stuff but feel free to check the full paper!
  - ▷ arXiv [CDRR13a]: abs/1302.4248
  - ▷ Conference version in ATVA'13 [CDRR13b], full version to appear in Information and Computation [CDRR15].


**Looking at Mean-Payoff and Total-Payoff through Windows**

Krishnendu Chatterjee[1,*], Laurent Doyen[2], Mickael Randour[3,†], and Jean-François Raskin[4,‡]

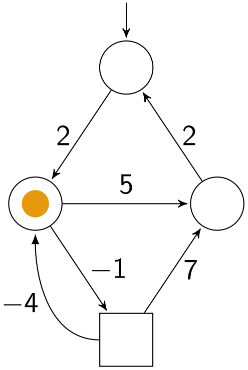
[1] IST Austria (Institute of Science and Technology Austria)
[2] LSV - ENS Cachan, France
[3] Computer Science Department, Université de Mons (UMONS), Belgium
[4] Département d'Informatique, Université Libre de Bruxelles (U.L.B.), Belgium

# Classical MP and TP games

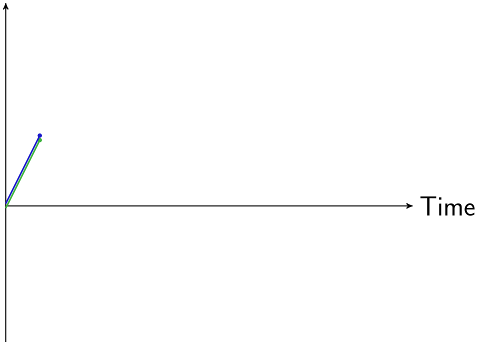



- $\text{TP}(\pi) = \liminf_{n\to\infty} \sum_{i=0}^{i=n-1} w(s_i, s_{i+1})$
- $\text{MP}(\pi) = \liminf_{n\to\infty} \frac{1}{n}\text{TP}(\pi(n))$

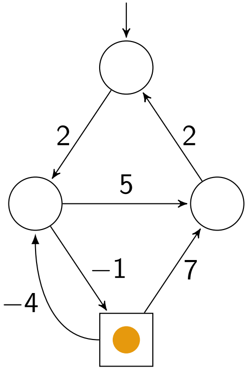
# Classical MP and TP games



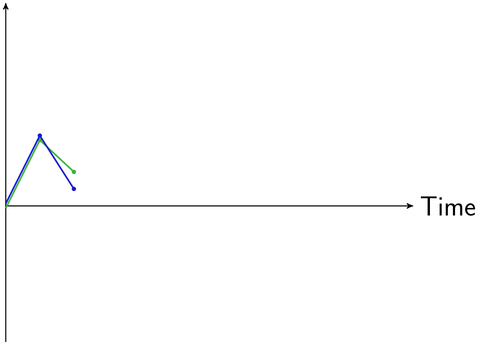

- TP($\pi$) $= \liminf_{n\to\infty} \sum_{i=0}^{i=n-1} w(s_i, s_{i+1})$
- MP($\pi$) $= \liminf_{n\to\infty} \frac{1}{n} \mathrm{TP}(\pi(n))$



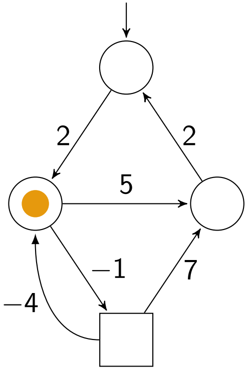
# Classical MP and TP games



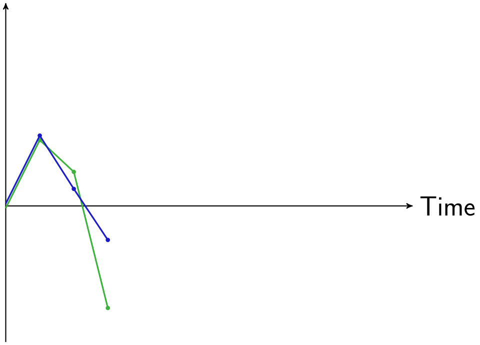

- $\text{TP}(\pi) = \liminf_{n\to\infty} \sum_{i=0}^{i=n-1} w(s_i, s_{i+1})$
- $\text{MP}(\pi) = \liminf_{n\to\infty} \frac{1}{n} \text{TP}(\pi(n))$



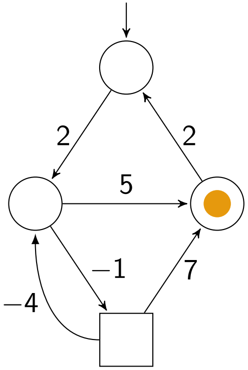
# Classical MP and TP games



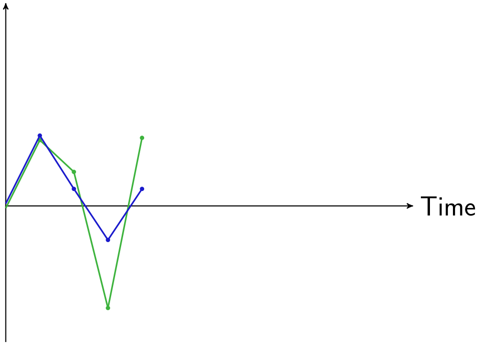

- $\text{TP}(\pi) = \liminf_{n\to\infty} \sum_{i=0}^{i=n-1} w(s_i, s_{i+1})$
- $\text{MP}(\pi) = \liminf_{n\to\infty} \frac{1}{n}\text{TP}(\pi(n))$



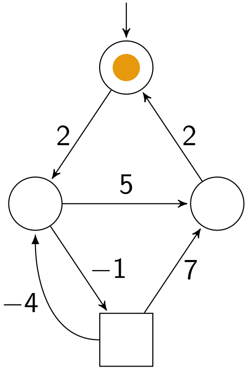
# Classical MP and TP games



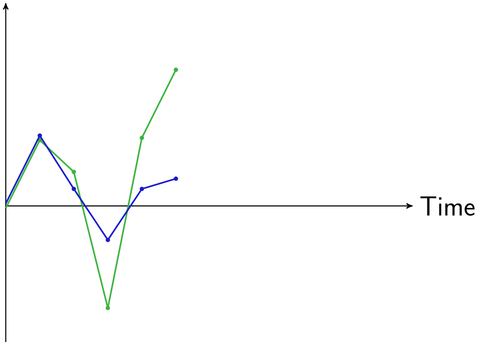

- $\text{TP}(\pi) = \liminf_{n \to \infty} \sum_{i=0}^{i=n-1} w(s_i, s_{i+1})$
- $\text{MP}(\pi) = \liminf_{n \to \infty} \frac{1}{n} \text{TP}(\pi(n))$



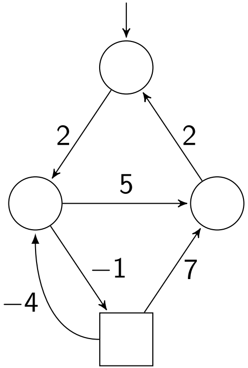
# Classical MP and TP games




- TP$(\pi) = \liminf_{n\to\infty} \sum_{i=0}^{i=n-1} w(s_i, s_{i+1})$
- MP$(\pi) = \liminf_{n\to\infty} \frac{1}{n} \text{TP}(\pi(n))$

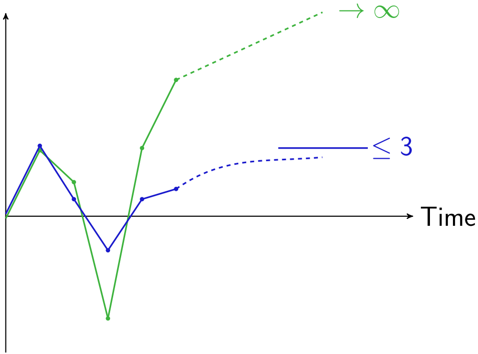


# Classical MP and TP games




Then, $(2, 5, 2)^\omega$

- $\text{TP}(\pi) = \liminf_{n\to\infty} \sum_{i=0}^{i=n-1} w(s_i, s_{i+1})$
- $\text{MP}(\pi) = \liminf_{n\to\infty} \frac{1}{n}\text{TP}(\pi(n))$



# What do we know?

| | one-dimension | | | $k$-dimension | | |
|---|---|---|---|---|---|---|
| | complexity | $\mathcal{P}_1$ mem. | $\mathcal{P}_2$ mem. | complexity | $\mathcal{P}_1$ mem. | $\mathcal{P}_2$ mem. |
| $\underline{\mathsf{MP}}$ / $\overline{\mathsf{MP}}$ | NP ∩ coNP | mem-less | | coNP-c. / NP ∩ coNP | infinite | mem-less |
| $\underline{\mathsf{TP}}$ / $\overline{\mathsf{TP}}$ | NP ∩ coNP | mem-less | | ?? | ?? | ?? |

▷ Long tradition of study. Non-exhaustive selection: [EM79, ZP96, Jur98, GZ04, GS09, CDHR10, VR11, CRR14, BFRR14]

# What about multi total-payoff?

| | one-dimension | | | $k$-dimension | | |
|---|---|---|---|---|---|---|
| | complexity | $\mathcal{P}_1$ mem. | $\mathcal{P}_2$ mem. | complexity | $\mathcal{P}_1$ mem. | $\mathcal{P}_2$ mem. |
| $\underline{\mathsf{MP}}$ / $\overline{\mathsf{MP}}$ | $\mathsf{NP} \cap \mathsf{coNP}$ | mem-less | | coNP-c. / $\mathsf{NP} \cap \mathsf{coNP}$ | infinite | mem-less |
| $\underline{\mathsf{TP}}$ / $\overline{\mathsf{TP}}$ | $\mathsf{NP} \cap \mathsf{coNP}$ | mem-less | | ?? | ?? | ?? |

- ▷ TP and MP look very similar in one-dimension
  - TP $\sim$ refinement of MP $= 0$
- ▷ Is it still true in multi-dimension?

# What about multi total-payoff?

| | one-dimension | | | $k$-dimension | | |
|---|---|---|---|---|---|---|
| | complexity | $\mathcal{P}_1$ mem. | $\mathcal{P}_2$ mem. | complexity | $\mathcal{P}_1$ mem. | $\mathcal{P}_2$ mem. |
| $\underline{\text{MP}}$ / $\overline{\text{MP}}$ | NP $\cap$ coNP | mem-less | | coNP-c. / NP $\cap$ coNP | infinite | mem-less |
| $\underline{\text{TP}}$ / $\overline{\text{TP}}$ | NP $\cap$ coNP | mem-less | | **Undec.** | - | - |

▷ **Unfortunately, no!**

*It would be nice to have...*

a **decidable** objective with the same flavor (some sort of approx.)

# Is the complexity barrier breakable?

| | one-dimension | | | $k$-dimension | | |
|---|---|---|---|---|---|---|
| | complexity | $\mathcal{P}_1$ mem. | $\mathcal{P}_2$ mem. | complexity | $\mathcal{P}_1$ mem. | $\mathcal{P}_2$ mem. |
| $\underline{\text{MP}}$ / $\overline{\text{MP}}$ | NP ∩ coNP | mem-less | | coNP-c. / NP ∩ coNP | infinite | mem-less |
| $\underline{\text{TP}}$ / $\overline{\text{TP}}$ | NP ∩ coNP | mem-less | | **Undec.** | - | - |

- ▷ P membership for the one-dimension case is a long-standing open problem!

*It would be nice to have...*

an approximation decidable in **polynomial time**

# Do we *really* want to play eternally?

| | one-dimension | | | $k$-dimension | | |
|---|---|---|---|---|---|---|
| | complexity | $\mathcal{P}_1$ mem. | $\mathcal{P}_2$ mem. | complexity | $\mathcal{P}_1$ mem. | $\mathcal{P}_2$ mem. |
| $\underline{\text{MP}}$ / $\overline{\text{MP}}$ | NP $\cap$ coNP | mem-less | | coNP-c. / NP $\cap$ coNP | infinite | mem-less |
| $\underline{\text{TP}}$ / $\overline{\text{TP}}$ | NP $\cap$ coNP | mem-less | | **Undec.** | - | - |

- MP and TP give **no timing guarantee**: the “good behavior” occurs at the limit. . .
- Sure, in one-dim., memoryless strategies suffice and provide bounds on cycles, but what if we are given an arbitrary play?

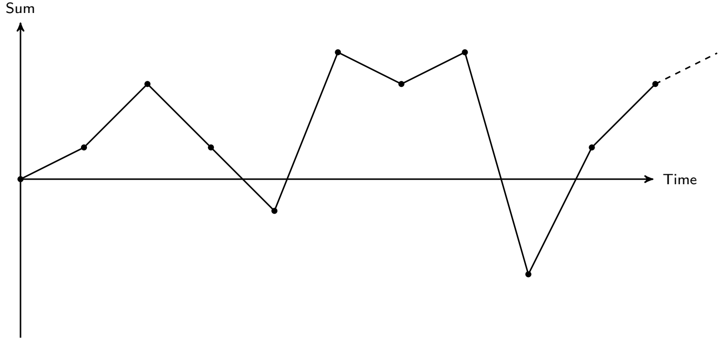
*It would be nice to have. . .*

a quantitative measure that **specifies timing requirements**

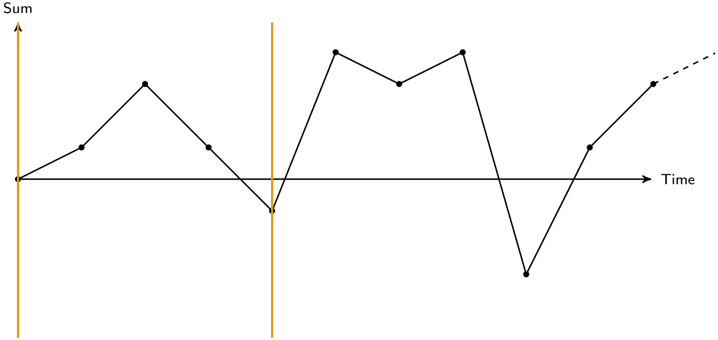
# Window objectives: key idea

- **Window** of fixed size **sliding** along a play
  $\rightsquigarrow$ defines a local finite horizon
- Objective: see a **local** $MP \geq 0$ *before hitting the end* of the window
  $\rightsquigarrow$ needs to be verified at *every* step

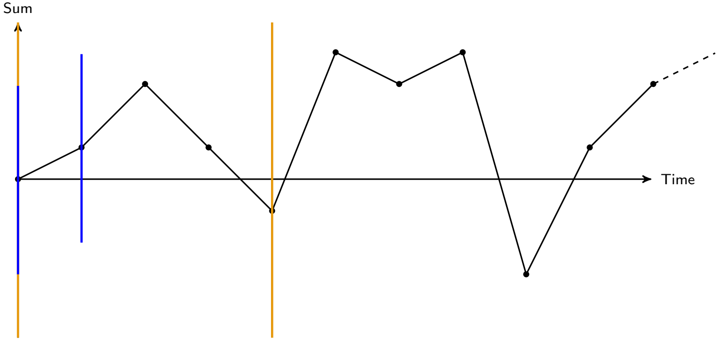
# Window MP, threshold zero, maximal window = 4



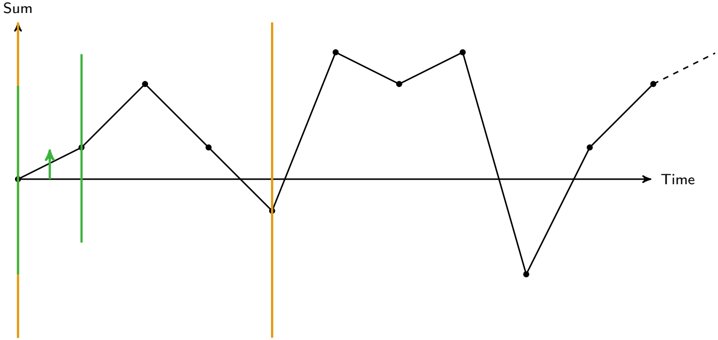
# Window MP, threshold zero, maximal window = 4



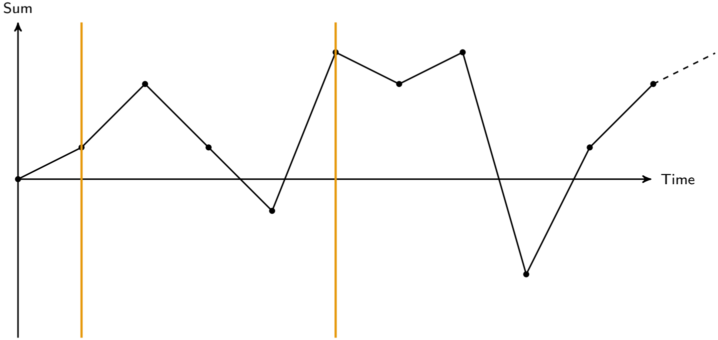
# Window MP, threshold zero, maximal window = 4



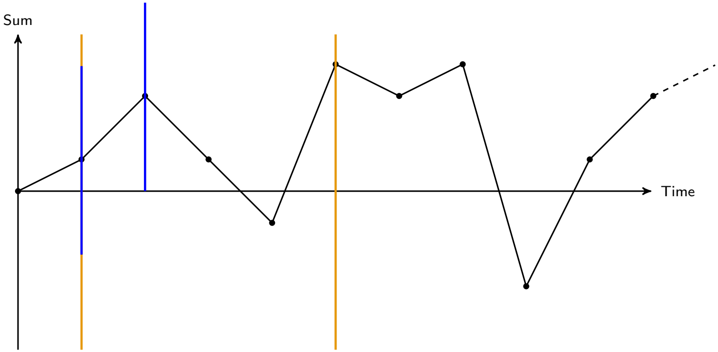
# Window MP, threshold zero, maximal window = 4



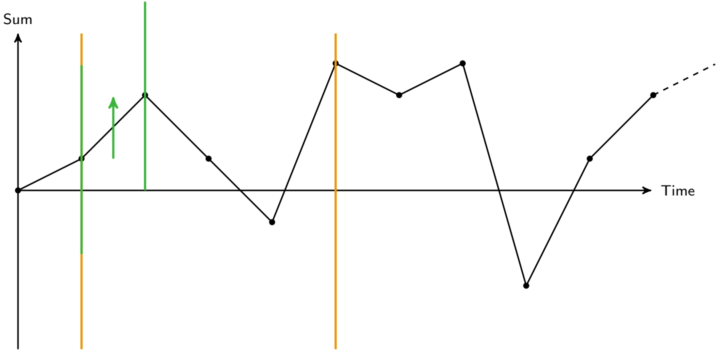
# Window MP, threshold zero, maximal window = 4



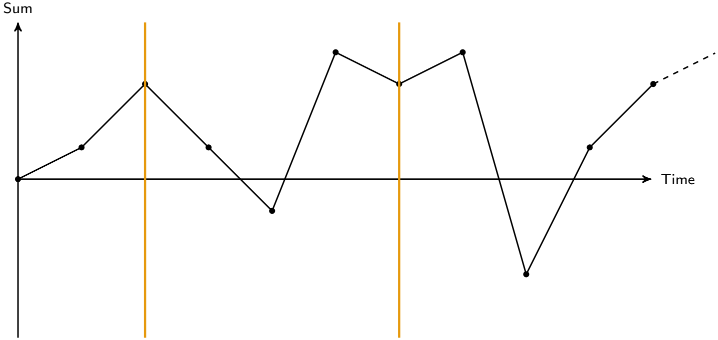
# Window MP, threshold zero, maximal window = 4



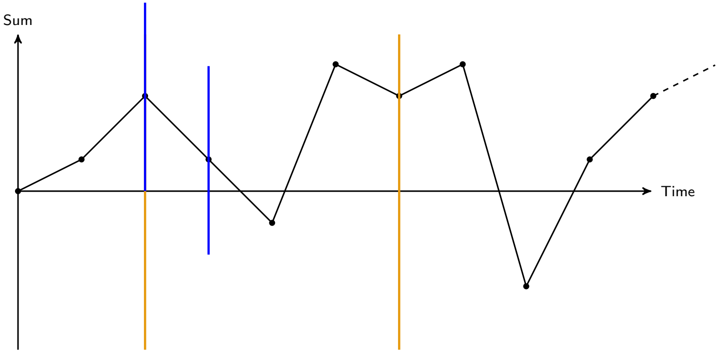
# Window MP, threshold zero, maximal window = 4



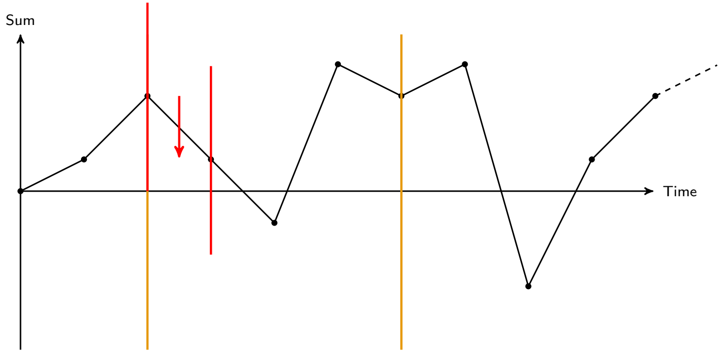
# Window MP, threshold zero, maximal window = 4



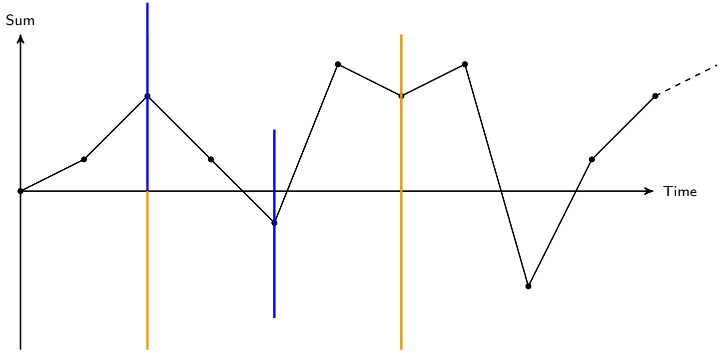
# Window MP, threshold zero, maximal window = 4



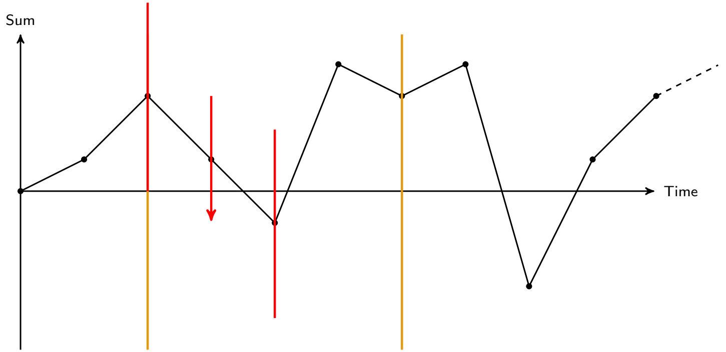
# Window MP, threshold zero, maximal window = 4



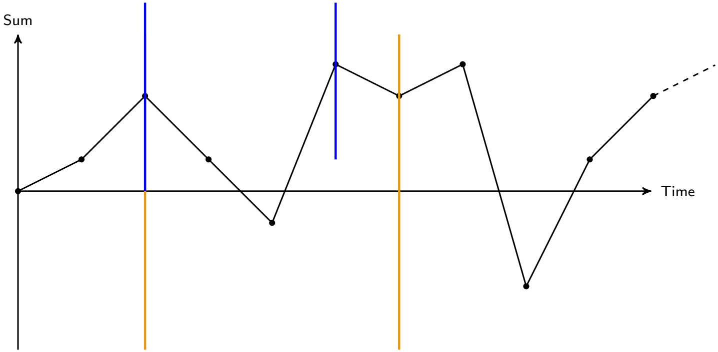
# Window MP, threshold zero, maximal window = 4



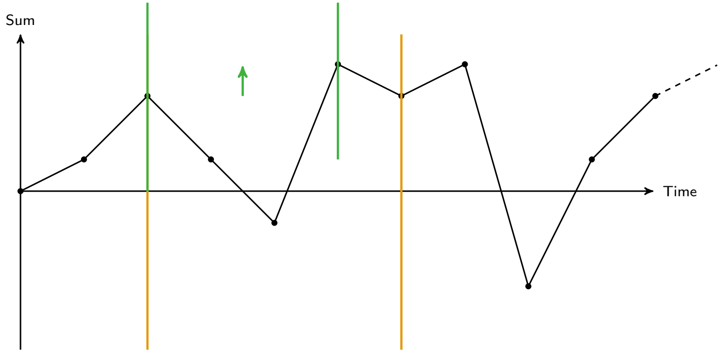
# Window MP, threshold zero, maximal window = 4



# Window MP, threshold zero, maximal window = 4



# Window MP, threshold zero, maximal window = 4



# Multiple variants

- Given $l_{\max} \in \mathbb{N}_0$, *good window* $\mathbf{GW}(l_{\max})$ asks for a positive sum in at most $l_{\max}$ steps (one window, from the first state)
- *Direct Fixed Window*: $\mathbf{DFW}(l_{\max}) \equiv \Box\mathbf{GW}(l_{\max})$
- *Fixed Window*: $\mathbf{FW}(l_{\max}) \equiv \Diamond\mathbf{DFW}(l_{\max})$
- *Direct Bounded Window*: $\mathbf{DBW} \equiv \exists\, l_{\max},\ \mathbf{DFW}(l_{\max})$
- *Bounded Window*: $\mathbf{BW} \equiv \Diamond\mathbf{DBW} \equiv \exists\, l_{\max},\ \mathbf{FW}(l_{\max})$

# Multiple variants

- Given $l_{\max} \in \mathbb{N}_0$, *good window* $\mathbf{GW}(l_{\max})$ asks for a positive sum in at most $l_{\max}$ steps (one window, from the first state)
- *Direct Fixed Window*: $\mathbf{DFW}(l_{\max}) \equiv \Box \mathbf{GW}(l_{\max})$
- *Fixed Window*: $\mathbf{FW}(l_{\max}) \equiv \Diamond \mathbf{DFW}(l_{\max})$
- *Direct Bounded Window*: $\mathbf{DBW} \equiv \exists\, l_{\max},\, \mathbf{DFW}(l_{\max})$
- *Bounded Window*: $\mathbf{BW} \equiv \Diamond \mathbf{DBW} \equiv \exists\, l_{\max},\, \mathbf{FW}(l_{\max})$

**Conservative approximations** in one-dim.

$$\text{Any window obj.} \Rightarrow \mathbf{BW} \Rightarrow \text{MP} \geq 0$$
$$\mathbf{BW} \Leftarrow \text{MP} > 0$$

# Results overview

<table>
<tr><td rowspan="2"></td><td colspan="3">one-dimension</td><td colspan="3">$k$-dimension</td></tr>
<tr><td>complexity</td><td>$\mathcal{P}_1$ mem.</td><td>$\mathcal{P}_2$ mem.</td><td>complexity</td><td>$\mathcal{P}_1$ mem.</td><td>$\mathcal{P}_2$ mem.</td></tr>
<tr><td>$\underline{\text{MP}}$ / $\overline{\text{MP}}$</td><td>NP $\cap$ coNP</td><td colspan="2">mem-less</td><td>coNP-c. / NP $\cap$ coNP</td><td>infinite</td><td>mem-less</td></tr>
<tr><td>$\underline{\text{TP}}$ / $\overline{\text{TP}}$</td><td>NP $\cap$ coNP</td><td colspan="2">mem-less</td><td>undec.</td><td>-</td><td>-</td></tr>
<tr><td>WMP: fixed polynomial window</td><td>P-c.</td><td colspan="2" rowspan="2">mem. req.<br>$\leq$ linear$(|S| \cdot l_{\max})$</td><td>PSPACE-h.<br>EXP-easy</td><td colspan="2" rowspan="2">exponential</td></tr>
<tr><td>WMP: fixed arbitrary window</td><td>P$(|S|, V, l_{\max})$</td><td>EXP-c.</td></tr>
<tr><td>WMP: bounded window problem</td><td>NP $\cap$ coNP</td><td>mem-less</td><td>infinite</td><td>NPR-h.</td><td>-</td><td>-</td></tr>
</table>

- $|S|$ the # of states, $V$ the length of the binary encoding of weights, and $l_{\max}$ the window size.

# Results overview: advantages

| | one-dimension | | | $k$-dimension | | |
|---|---|---|---|---|---|---|
| | complexity | $\mathcal{P}_1$ mem. | $\mathcal{P}_2$ mem. | complexity | $\mathcal{P}_1$ mem. | $\mathcal{P}_2$ mem. |
| MP / $\overline{\text{MP}}$ | NP $\cap$ coNP | mem-less | | coNP-c. / NP $\cap$ coNP | infinite | mem-less |
| TP / $\overline{\text{TP}}$ | NP $\cap$ coNP | mem-less | | **undec.** | - | - |
| WMP: fixed polynomial window | **P-c.** | **mem. req.** $\leq$ **linear**$(\|S\| \cdot l_{\max})$ | | **PSPACE-h.** **EXP-easy** | **exponential** | |
| WMP: fixed arbitrary window | **P**$(\|S\|, V, l_{\max})$ | | | **EXP-c.** | | |
| WMP: bounded window problem | **NP $\cap$ coNP** | **mem-less** | **infinite** | **NPR-h.** | - | - |

- ▷ $|S|$ the # of states, $V$ the length of the binary encoding of weights, and $l_{\max}$ the window size.
- ▷ For one-dim. games with poly. windows, we are in **P**.
- ▷ For multi-dim. games with fixed windows, we are **decidable**.
- ▷ Window objectives provide **timing guarantees**.

# Taste of the proofs ingredients

- For those who like it technical, we use
  - ▷ 2CMs [Min61],
  - ▷ membership problem for APTMs [CKS81],
  - ▷ countdown games [JSL08] ,
  - ▷ generalized reachability [FH10],
  - ▷ reset nets [DFS98, Sch02, LNO+08],
  - ▷ ...
- *Open question*: is bounded window decidable in multi-dim. ?

Check the full version on arXiv! abs/1302.4248

**Thanks!**

Do not hesitate to discuss with us!

# References I


V. Bruyère, E. Filiot, M. Randour, and J.-F. Raskin.
Meet your expectations with guarantees: Beyond worst-case synthesis in quantitative games.
In Proc. of STACS, LIPIcs 25, pages 199–213. Schloss Dagstuhl - LZI, 2014.

K. Chatterjee, L. Doyen, T.A. Henzinger, and J.-F. Raskin.
Generalized mean-payoff and energy games.
In Proc. of FSTTCS, LIPIcs 8, pages 505–516. Schloss Dagstuhl - LZI, 2010.

K. Chatterjee, L. Doyen, M. Randour, and J.-F. Raskin.
Looking at mean-payoff and total-payoff through windows.
CoRR, abs/1302.4248, 2013.

K. Chatterjee, L. Doyen, M. Randour, and J.-F. Raskin.
Looking at mean-payoff and total-payoff through windows.
In Proc. of ATVA, LNCS 8172, pages 118–132. Springer, 2013.

K. Chatterjee, L. Doyen, M. Randour, and J.-F. Raskin.
Looking at mean-payoff and total-payoff through windows.
Information and Computation, 2015.
To appear.

A.K. Chandra, D. Kozen, and L.J. Stockmeyer.
Alternation.
J. ACM, 28(1):114–133, 1981.

# References II


K. Chatterjee, M. Randour, and J.-F. Raskin.
Strategy synthesis for multi-dimensional quantitative objectives.
In Proc. of CONCUR, LNCS 7454, pages 115–131. Springer, 2012.

K. Chatterjee, M. Randour, and J.-F. Raskin.
Strategy synthesis for multi-dimensional quantitative objectives.
Acta Informatica, 51(3-4):129–163, 2014.

C. Dufourd, A. Finkel, and P. Schnoebelen.
Reset nets between decidability and undecidability.
In Proc. of ICALP, LNCS 1443, pages 103–115. Springer, 1998.

A. Ehrenfeucht and J. Mycielski.
Positional strategies for mean payoff games.
Int. Journal of Game Theory, 8(2):109–113, 1979.

N. Fijalkow and F. Horn.
The surprizing complexity of generalized reachability games.
CoRR, abs/1010.2420, 2010.

T. Gawlitza and H. Seidl.
Games through nested fixpoints.
In Proc. of CAV, LNCS 5643, pages 291–305. Springer, 2009.

H. Gimbert and W. Zielonka.
When can you play positionally?
In Proc. of MFCS, LNCS 3153, pages 686–697. Springer, 2004.

# References III


M. Jurdziński, J. Sproston, and F. Laroussinie.
Model checking probabilistic timed automata with one or two clocks.
Logical Methods in Computer Science, 4(3), 2008.

M. Jurdziński.
Deciding the winner in parity games is in UP ∩ co-UP.
Inf. Process. Lett., 68(3):119–124, 1998.

R. Lazic, T. Newcomb, J. Ouaknine, A.W. Roscoe, and J. Worrell.
Nets with tokens which carry data.
Fundam. Inform., 88(3):251–274, 2008.

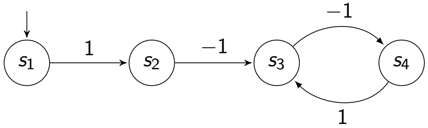
M.L. Minsky.
Recursive unsolvability of Post's problem of "tag" and other topics in theory of Turing machines.
The Annals of Mathematics, 74(3):437–455, 1961.

P. Schnoebelen.
Verifying lossy channel systems has nonprimitive recursive complexity.
Inf. Process. Lett., 83(5):251–261, 2002.

Y. Velner and A. Rabinovich.
Church synthesis problem for noisy input.
In Proc. of FOSSACS, LNCS 6604, pages 275–289. Springer, 2011.

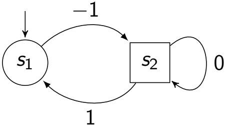
U. Zwick and M. Paterson.
The complexity of mean payoff games on graphs.
Theoretical Computer Science, 158:343–359, 1996.

## Example 1



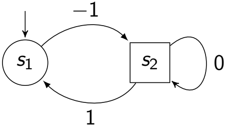

- MP is satisfied
  - ▷ the cycle is non-negative
- **FW**(2) is satisfied
  - ▷ thanks to prefix-independence
- **DBW** is not
  - ▷ the window opened in $s_2$ never closes

# Example 2




- MP is satisfied
  - ▷ all simple cycles are non-negative
- *but* none of the window objectives is
  - ▷ $\mathcal{P}_2$ can force opening windows and delay their closing for as long as he wants (but not forever due to prefix-independence)

# Example 2




- MP is satisfied
  - ▷ all simple cycles are non-negative
- *but* none of the window objectives is
  - ▷ $\mathcal{P}_2$ can force opening windows and delay their closing for as long as he wants (but not forever due to prefix-independence)

## **BW** vs. MP

- BW asks for timing guarantees which cannot be enforced here
- Observe that $\mathcal{P}_2$ **needs infinite memory**