Looking at Mean-Payoff and Total-Payoff through Windows

K. Chatterjee (IST Austria)  L. Doyen (ENS Cachan)
M. Randour (UMONS)        J.-F. Raskin (ULB)

12.04.2013

Cassting kick-off meeting
Aim of this talk

1. Overview of the situation for (multi) MP and TP games
   - No P algorithm known in one dimension
   - In multi dimensions, MP is coNP-complete
   - First contribution: **TP is undecidable in multi dimensions**
Aim of this talk

1. Overview of the situation for (multi) MP and TP games
   - No P algorithm known in one dimension
   - In multi dimensions, MP is coNP-complete
   - First contribution: TP is undecidable in multi dimensions

2. Introduction of window objectives
   - Conservative approximation of MP/TP
   - Break the complexity barriers
   - Algorithms, complexity and memory requirements
   - Several flavors of the objective
1. Mean-Payoff and Total-Payoff Games
2. Total-Payoff Games in Multi Dimensions
3. Window Objectives
4. One-Dimension Fixed Window Problem
5. Multi-Dimension Bounded Window Problem
6. Conclusion
1. Mean-Payoff and Total-Payoff Games

2. Total-Payoff Games in Multi Dimensions

3. Window Objectives

4. One-Dimension Fixed Window Problem

5. Multi-Dimension Bounded Window Problem

6. Conclusion
Turn-based games

- $G = (S_1, S_2, E)$
- $S = S_1 \cup S_2, S_1 \cap S_2 = \emptyset, E \subseteq S \times S$
- $\mathcal{P}_1$ states = $\bigcirc$
- $\mathcal{P}_2$ states = $\square$
- Plays, prefixes, **pure** strategies.
Integer $k$-dim. payoff function

- $G = (S_1, S_2, E, k, w)$
- $w : E \rightarrow \mathbb{Z}^k$
- Play $\pi = s_0s_1s_2 \ldots$
- **Total-payoff**

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

- **Mean-payoff**

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

- **TP (MP) threshold problem**
  Given \( v \in \mathbb{Q}^k \) and \( s_{\text{init}} \in S \),
  \[ \exists \lambda_1 \in \Lambda_1 \text{ s.t. } \forall \lambda_2 \in \Lambda_2, \]
  \[ \text{TP}(\text{Outcome}_G(s_{\text{init}}, \lambda_1, \lambda_2)) \geq v \]
Known results

<table>
<thead>
<tr>
<th>MP/TP</th>
<th>Multi TP</th>
<th>Window MP</th>
<th>One-Dim. Fixed</th>
<th>Multi-Dim. Bounded</th>
<th>Conclusion</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th></th>
<th>one-dimension</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>complexity</td>
<td>(P_1) mem.</td>
<td>(P_2) mem.</td>
<td>complexity</td>
<td>(P_1) mem.</td>
</tr>
<tr>
<td><strong>MP / MP</strong></td>
<td>NP (\cap) coNP</td>
<td>mem-less</td>
<td></td>
<td>coNP-c. (\cap) NP (\cap) coNP</td>
<td>infinite</td>
</tr>
<tr>
<td><strong>TP / TP</strong></td>
<td>NP (\cap) coNP</td>
<td>mem-less</td>
<td></td>
<td>??</td>
<td>??</td>
</tr>
</tbody>
</table>

▷ See [EM79, Jur98, ZP96, GS09, CDHR10, VR11]
▷ No known polynomial time algorithm for one-dimension
▷ No result on multi-dimension total-payoff
1. Mean-Payoff and Total-Payoff Games

2. Total-Payoff Games in Multi Dimensions

3. Window Objectives

4. One-Dimension Fixed Window Problem

5. Multi-Dimension Bounded Window Problem

6. Conclusion
Multi-dimension TP games are undecidable

Theorem

The threshold problem for infimum and supremum total-payoff objectives is **undecidable** in multi-dimension games, for five dimensions.
Multi-dimension TP games are undecidable

**Theorem**

The threshold problem for infimum and supremum total-payoff objectives is undecidable in multi-dimension games, for five dimensions.

▷ Reduction from the halting problem for 2CMs [Min61]
Two-counter machines

- Finite set of instructions
- Two counters $C_1$ and $C_2$ taking values $(v_1, v_2) \in \mathbb{N}^2$
- Instructions:
  - Increment
    
    $C_i \, \uparrow \uparrow$
  - Decrement
    
    $C_i \, \downarrow \downarrow$
  - Zero test and branch accordingly
    
    If $C_i == 0$ do this else do that

- W.l.o.g. if the machine stops, it stops with both counters to zero
Encoding a 2CM in a 5-dim. TP game

- TP objective (inf or sup) of threshold (0, 0, 0, 0, 0)
- $P_1$ must simulate faithfully
- $P_2$ retaliates if $P_1$ cheats
- At the end, $P_1$ wins the TP game iff the 2CM stops

Key idea: after $m$ steps, the TP has value $(v_1, -v_1, v_2, -v_2, -m)$ iff the 2CM counters have value $(v_1, v_2)$
Instructions

- **Increment** $C_1$

  $$(1, -1, 0, 0, -1)$$

- **Decrement** $C_1$

  $$(-1, 1, 0, 0, -1)$$
Instructions

- Checking counter $C_1$ is non-negative

If $P_1$ cheats, he is doomed!
Otherwise, $P_2$ has no interest in retaliating.
Instructions

- Checking a zero test on $C_1$

  ▶ If $P_1$ cheats, he is doomed!
  ▶ Otherwise, $P_2$ has no interest in retaliating.
Halting

- If the 2CM halts (with counters to zero w.l.o.g.)

$\xrightarrow{} (0, 0, 0, 0, 1)$

▶ Thanks to the fifth dim., $P_1$ wins only if the machine halts.
The case is closed

<table>
<thead>
<tr>
<th></th>
<th>one-dimension</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>complexity</td>
<td>(P_1) mem.</td>
<td>(P_2) mem.</td>
<td></td>
<td></td>
<td>(P_1) mem.</td>
<td>(P_2) mem.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MP / MP</td>
<td>NP (\cap) coNP</td>
<td>mem-less</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TP / (\overline{TP})</td>
<td>NP (\cap) coNP</td>
<td>mem-less</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Looking at MP and TP through Windows

Chatterjee, Doyen, Randour, Raskin
1. Mean-Payoff and Total-Payoff Games

2. Total-Payoff Games in Multi Dimensions

3. Window Objectives

4. One-Dimension Fixed Window Problem

5. Multi-Dimension Bounded Window Problem

6. Conclusion
Motivations

- Classical MP and TP objectives have some drawbacks
  - Complexity issues
  - Infimum vs. supremum
  - Describe what happens *at the limit*: no guarantee about a time frame
Motivations

- Classical MP and TP objectives have some drawbacks
  - Complexity issues
  - Infimum vs. supremum
  - Describe what happens at the limit: no guarantee about a time frame

- Window objectives consider what happens inside a finite window sliding along a play
  - Conservative approximation of MP/TP
  - Intuition: local deviations from the threshold must be compensated in a parametrized # of steps
  - Variety of results and algorithms
Illustration: WMP, threshold zero, maximal window = 4
Illustration: WMP, threshold zero, maximal window $= 4$
Illustration: WMP, threshold zero, maximal window = 4
Illustration: WMP, threshold zero, maximal window $= 4$
Illustration: WMP, threshold zero, maximal window $= 4$
Illustration: WMP, threshold zero, maximal window $= 4$
Illustration: WMP, threshold zero, maximal window = 4
Illustration: WMP, threshold zero, maximal window = 4
Illustration: WMP, threshold zero, maximal window = 4
Illustration: WMP, threshold zero, maximal window = 4
Illustration: WMP, threshold zero, maximal window = 4
Illustration: WMP, threshold zero, maximal window = 4
Illustration: WMP, threshold zero, maximal window = 4
Illustration: WMP, threshold zero, maximal window = 4
Definitions

- Given \( l_{\text{max}} \in \mathbb{N}_0 \), good window \( GW(l_{\text{max}}) \) asks for a positive sum in at most \( l_{\text{max}} \) steps (one window, from the first state).

- **Direct Fixed Window**: \( DFW(l_{\text{max}}) \equiv \Box GW(l_{\text{max}}) \)

- **Fixed Window**: \( FW(l_{\text{max}}) \equiv \Diamond DFW(l_{\text{max}}) \)

- **Direct Bounded Window**: \( DBW \equiv \exists l_{\text{max}}, DFW(l_{\text{max}}) \)

- **Bounded Window**: \( BW \equiv \Diamond DBW \equiv \exists l_{\text{max}}, FW(l_{\text{max}}) \)
Definitions

- Given $l_{\text{max}} \in \mathbb{N}_0$, *good window* $\text{GW}(l_{\text{max}})$ asks for a positive sum in at most $l_{\text{max}}$ steps (one window, from the first state)

- *Direct Fixed Window*: $\text{DFW}(l_{\text{max}}) \equiv \square \text{GW}(l_{\text{max}})$

- *Fixed Window*: $\text{FW}(l_{\text{max}}) \equiv \Diamond \text{DFW}(l_{\text{max}})$

- *Direct Bounded Window*: $\text{DBW} \equiv \exists l_{\text{max}}, \text{DFW}(l_{\text{max}})$

- *Bounded Window*: $\text{BW} \equiv \Diamond \text{DBW} \equiv \exists l_{\text{max}}, \text{FW}(l_{\text{max}})$

- A window *closes* when the sum becomes positive
- A window is *open* if not yet closed
Examples

\[ \text{FW}(2) \] is satisfied, \text{DBW} is not, MP is satisfied.
Examples

\[ \textbf{FW}(2) \text{ is satisfied, DBW is not, MP is satisfied.} \]

\[ \textbf{MP is satisfied but none of the window objectives is.} \]
Conservative approximation of MP ? (one-dim.)

The following are true

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

$\text{BW} \Leftarrow \text{MP} > 0$
## Results overview

<table>
<thead>
<tr>
<th></th>
<th>one-dimension</th>
<th>k-dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>complexity</td>
<td>$P_1$ mem.</td>
</tr>
<tr>
<td><strong>MP / MP</strong></td>
<td>$NP \cap \text{coNP}$</td>
<td>mem-less</td>
</tr>
<tr>
<td><strong>TP / TP</strong></td>
<td>$NP \cap \text{coNP}$</td>
<td>mem-less</td>
</tr>
<tr>
<td><strong>WMP: fixed polynomial window</strong></td>
<td>$P$-c.</td>
<td>mem. req. $\leq \text{linear}(</td>
</tr>
<tr>
<td><strong>WMP: fixed arbitrary window</strong></td>
<td>$P(</td>
<td>S</td>
</tr>
<tr>
<td><strong>WMP: bounded window problem</strong></td>
<td>$NP \cap \text{coNP}$</td>
<td>mem-less</td>
</tr>
</tbody>
</table>

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

For one-dim. games with poly. windows, we are in $P$. 

Looking at MP and TP through Windows

Chatterjee, Doyen, Randour, Raskin

21 / 33
## Results overview

<table>
<thead>
<tr>
<th>MP/TP</th>
<th>Multi TP</th>
<th>Window MP</th>
<th>One-Dim. Fixed</th>
<th>Multi-Dim. Bounded</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th><strong>One-Dimension</strong></th>
<th></th>
<th><strong>K-Dimension</strong></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Complexity</td>
<td>$P_1$ mem.</td>
<td>$P_2$ mem.</td>
<td>Complexity</td>
<td>$P_1$ mem.</td>
</tr>
<tr>
<td><strong>MP / $\overline{MP}$</strong></td>
<td>$NP \cap \text{coNP}$</td>
<td>mem-less</td>
<td></td>
<td>$NP \cap \text{coNP}$</td>
<td>mem-less</td>
</tr>
<tr>
<td><strong>TP / $\overline{TP}$</strong></td>
<td>$NP \cap \text{coNP}$</td>
<td>mem-less</td>
<td></td>
<td></td>
<td>undec.</td>
</tr>
<tr>
<td><strong>WMP: fixed polynomial window</strong></td>
<td>$P$-c.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>WMP: fixed arbitrary window</strong></td>
<td>$P(</td>
<td>S</td>
<td>, V, l_{max})$</td>
<td>mem. req. $\leq \text{linear}(</td>
<td>S</td>
</tr>
<tr>
<td><strong>WMP: bounded window problem</strong></td>
<td>$NP \cap \text{coNP}$</td>
<td>mem-less</td>
<td>infinite</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- $|S|$ the # of states, $V$ the length of the binary encoding of weights, and $l_{\text{max}}$ the window size.
- For one-dim. games with poly. windows, we are in $P$.
- No time to discuss everything. **Focus.**
1 Mean-Payoff and Total-Payoff Games

2 Total-Payoff Games in Multi Dimensions

3 Window Objectives

4 One-Dimension Fixed Window Problem

5 Multi-Dimension Bounded Window Problem

6 Conclusion
High level sketch: top-down approach

- \( \text{FW}(l_{\text{max}}) \equiv \Diamond \text{DFW}(l_{\text{max}}) \)

- Assume we can compute \( \text{DFW}(l_{\text{max}}) \),
- Compute attractor, declare winning and recurse on subgame.
High level sketch: top-down approach

- $FW(l_{\text{max}}) \equiv \Diamond DFW(l_{\text{max}})$

- Assume we can compute $DFW(l_{\text{max}})$,
- Compute attractor, declare winning and recurse on subgame.
High level sketch: top-down approach

- \( FW(l_{\text{max}}) \equiv \Diamond DFW(l_{\text{max}}) \)

▶ Assume we can compute \( DFW(l_{\text{max}}) \),
▶ Compute attractor, declare winning and recurse on subgame.
High level sketch: top-down approach

- \( FW(l_{\text{max}}) \equiv \Diamond DFW(l_{\text{max}}) \)

- Assume we can compute \( DFW(l_{\text{max}}) \).
- Compute attractor, declare winning and recurse on subgame.
High level sketch: top-down approach

- \( \text{FW}(l_{\text{max}}) \equiv \Diamond \text{DFW}(l_{\text{max}}) \)

▷ Assume we can compute \( \text{DFW}(l_{\text{max}}) \),
▷ Compute attractor, declare winning and recurse on subgame.
High level sketch: top-down approach

- \( \text{FW}(l_{\text{max}}) \equiv \diamond \text{DFW}(l_{\text{max}}) \)

- Assume we can compute \( \text{DFW}(l_{\text{max}}) \).
- Compute attractor, declare winning and recurse on subgame.
High level sketch: top-down approach

- $\text{DFW}(l_{\text{max}}) \equiv \Box \text{GW}(l_{\text{max}})$

- Assume we can compute $\text{GW}(l_{\text{max}})$,
- Compute the stable set s.t. $\mathcal{P}_1$ can satisfy it repeatedly.
High level sketch: top-down approach

- $\text{DFW}(l_{\text{max}}) \equiv \Box \text{GW}(l_{\text{max}})$

- Assume we can compute $\text{GW}(l_{\text{max}})$,
- Compute the stable set s.t. $\mathcal{P}_1$ can satisfy it repeatedly.
High level sketch: top-down approach

- \( \text{DFW}(l_{\text{max}}) \equiv \Box \text{GW}(l_{\text{max}}) \)

- Assume we can compute \( \text{GW}(l_{\text{max}}) \),
- Compute the stable set s.t. \( P_1 \) can satisfy it repeatedly.
High level sketch: top-down approach

- $\text{DFW}(l_{\text{max}}) \equiv \square \text{GW}(l_{\text{max}})$

- Assume we can compute $\text{GW}(l_{\text{max}})$,
- Compute the stable set s.t. $\mathcal{P}_1$ can satisfy it repeatedly.

\[ G \]

Subgame
High level sketch: top-down approach

- \( \text{DFW}(l_{\text{max}}) \equiv \Box \text{GW}(l_{\text{max}}) \)

- Assume we can compute \( \text{GW}(l_{\text{max}}) \),
- Compute the stable set s.t. \( P_1 \) can satisfy it repeatedly.

![Diagram with a stable set inside a graph]

Looking at MP and TP through Windows  
Chatterjee, Doyen, Randour, Raskin  
24 / 33
High level sketch: top-down approach

- $\text{GW}(l_{\text{max}})$

  - Simply compute the best sum achievable in at most $l_{\text{max}}$ steps and check if positive.
High level sketch: top-down approach

- $GW(l_{\text{max}})$

- Simply compute the best sum achievable in at most $l_{\text{max}}$ steps and check if positive.

- Finally,

---

**Theorem**

In two-player one-dimension games,
(a) the fixed arbitrary window MP problem is decidable in time polynomial in the size of the game and the window size,
(b) the fixed polynomial window MP problem is P-complete,
(c) both players require memory, and memory of size linear in the size of the game and the window size is sufficient.
1. Mean-Payoff and Total-Payoff Games

2. Total-Payoff Games in Multi Dimensions

3. Window Objectives

4. One-Dimension Fixed Window Problem

5. Multi-Dimension Bounded Window Problem

6. Conclusion
Approach

- We prove non-primitive recursive\(^1\) (NPR) hardness
- Reduction from the termination problem in reset nets (Petri nets with reset arcs) [Sch02]

\(^1\)Cf. Ackermann function
Reset nets

- Classic Petri net (places, tokens, transitions) with added *reset arcs*

Transitions may empty a place from all its tokens
Reset nets

- Classic Petri net (places, tokens, transitions) with added reset arcs

- Transitions may empty a place from all its tokens

- Given an initial marking, the termination problem asks if there exists an infinite sequence of transitions that can be fired
From reset nets to **direct** bounded window games

- **Crux of the construction: encoding the markings**
  - We use one dimension for each place
  - If a place $p$ contains $m$ tokens, then there will be an open window on dimension $p$ with sum value $-m-1$
  - Hence **during a faithful simulation, all windows remain open** (you cannot consume tokens that do not exist)
From reset nets to **direct** bounded window games

- **Crux of the construction:** encoding the markings
  - We use one dimension for each place
  - If a place $p$ contains $m$ tokens, then there will be an open window on dimension $p$ with sum value $-m - 1$
  - Hence **during a faithful simulation, all windows remain open** (you cannot consume tokens that do not exist)

- $P_2$ simulates the net
- $P_1$ checks if he is faithful
- $P_1$ wants to win the direct bounded window MP obj.
  - only able to do so if $P_2$ cheats, i.e., if all runs terminate
The construction in a nutshell

- The initial marking open corresponding windows in all places
- $\mathcal{P}_2$ chooses transitions to fire, which consume tokens
- $\mathcal{P}_1$ can branch or continue (and apply reset, then output)
The construction in a nutshell

- If no infinite execution exists, at some point, $\mathcal{P}_2$ must choose a transition without the needed tokens on some place $p$.
- The window closes on dimension $p$.
- By branching $\mathcal{P}_1$ can close all other windows and ensure winning.
The construction in a nutshell

- If $\mathcal{P}_1$ branches while $\mathcal{P}_2$ is honest, one window stays open forever and he loses.
- The additional dimension ensures that $\mathcal{P}_1$ leaves the reset state.

Looking at MP and TP through Windows

Chatterjee, Doyen, Randour, Raskin

30 / 33
Extension to bounded window objective

- More involved construction

**Theorem**

In two-player multi-dimension games, the bounded window mean-payoff problem is non-primitive recursive hard.
1. Mean-Payoff and Total-Payoff Games

2. Total-Payoff Games in Multi Dimensions

3. Window Objectives

4. One-Dimension Fixed Window Problem

5. Multi-Dimension Bounded Window Problem

6. Conclusion
### A new family of objectives

<table>
<thead>
<tr>
<th></th>
<th>one-dimension</th>
<th>k-dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>complexity</td>
<td>complexity</td>
</tr>
<tr>
<td><strong>MP / MP</strong></td>
<td>$\text{NP} \cap \text{coNP}$</td>
<td>$\text{coNP-c.} / \text{NP} \cap \text{coNP}$</td>
</tr>
<tr>
<td><strong>TP / TP</strong></td>
<td>$\text{NP} \cap \text{coNP}$</td>
<td>infinite</td>
</tr>
<tr>
<td>WMP: fixed</td>
<td>$\mathbb{P}$-c.</td>
<td>$\text{PSPACE-h.}$</td>
</tr>
<tr>
<td>polynomial window</td>
<td>$\mathbb{P}(</td>
<td>S</td>
</tr>
<tr>
<td>WMP: fixed</td>
<td>$\mathbb{P}(</td>
<td>S</td>
</tr>
<tr>
<td>arbitrary window</td>
<td></td>
<td>exponential</td>
</tr>
<tr>
<td>WMP: bounded</td>
<td>$\text{NP} \cap \text{coNP}$</td>
<td>$\text{NPR-h.}$</td>
</tr>
<tr>
<td>window problem</td>
<td>mem-less</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>infinite</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>$\leq \text{linear}(</td>
<td>S</td>
</tr>
</tbody>
</table>

- Conservative approximation of MP/TP
- Provides timing guarantees
- Breaks the $\text{NP} \cap \text{coNP}$ barrier in one-dim. poly. window case
- Decidable approximation of TP in multi-dim. case
- **Open question**: is BW decidable in multi-dim. ?


M.L. Minsky.
Recursive unsolvability of Post’s problem of “tag” and other topics in theory of Turing machines.


P. Schnoebelen.
Verifying lossy channel systems has nonprimitive recursive complexity.

Y. Velner and A. Rabinovich.
Church synthesis problem for noisy input.

U. Zwick and M. Paterson.
The complexity of mean payoff games on graphs.