Looking at Mean-Payoff and Total-Payoff through Windows

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

Hanoi - 16.10.2013

ATVA 2013 - 11th International Symposium on Automated Technology for Verification and Analysis
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**
   ▶  No timing guarantee
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
   - No timing guarantee

2. Introduction of window objectives
   - Conservative approximation of MP/TP
   - Break the complexity barriers
   - Specifies timing requirements
   - Algorithms, complexity and memory requirements
   - Several flavors of the objective

Full details available on arXiv: abs/1302.4248
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
MP and TP games

\[ G = (S_1, S_2, E, w) \]
\[ S = S_1 \cup S_2, S_1 \cap S_2 = \emptyset, E \subseteq S \times S, \]
\[ w : E \rightarrow \mathbb{Z} \]
\[ \mathcal{P}_1 \text{ states } = \bigcirc \]
\[ \mathcal{P}_2 \text{ states } = \square \]

- Plays, prefixes, **pure** strategies.
MP and TP games

\[
\begin{align*}
\text{TP}(\pi) &= \liminf_{n \to \infty} \sum_{i=0}^{i=n-1} w(s_i, s_{i+1}) \\
\text{MP}(\pi) &= \liminf_{n \to \infty} \frac{1}{n} \text{TP}(\pi(n))
\end{align*}
\]

2

\[
\begin{array}{cccc}
\text{2} & \text{2} & \text{5} & \text{2} \\
\end{array}
\]

-4

7

Time

Looking at MP and TP through Windows

Chatterjee, Doyen, Randour, Raskin
MP and TP games

\[ \text{TP}(\pi) = \lim_{n \to \infty} \inf \sum_{i=0}^{i=n-1} w(s_i, s_{i+1}) \]

\[ \text{MP}(\pi) = \lim_{n \to \infty} \frac{1}{n} \text{TP}(\pi(n)) \]
MP and TP games

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

\[ \text{MP}(\pi) = \liminf_{n \to \infty} \frac{1}{n} \text{TP}(\pi(n)) \]
MP and TP games

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

\[ \text{MP}(\pi) = \liminf_{n \to \infty} \frac{1}{n} \text{TP}(\pi(n)) \]
MP and TP games

\[ \text{TP}(\pi) = \lim_{n \to \infty} \inf \sum_{i=0}^{i=n-1} w(s_i, s_{i+1}) \]

\[ \text{MP}(\pi) = \lim_{n \to \infty} \inf \frac{1}{n} \text{TP}(\pi(n)) \]
MP and TP games

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

\[
\text{MP}(\pi) = \liminf_{n \to \infty} \frac{1}{n} \text{TP}(\pi(n))
\]
MP and TP games

\[ \text{TP}(\pi) = \lim_{n \to \infty} \inf \sum_{i=0}^{i=n-1} w(s_i, s_{i+1}) \]

\[ \text{MP}(\pi) = \lim_{n \to \infty} \inf \frac{1}{n} \text{TP}(\pi(n)) \]

Then, \((2, 5, 2)^\omega\)
MP and TP games

TP (MP) threshold problem
Given $v \in \mathbb{Q}$ and $s_{init} \in S$,

$$\exists \lambda_1 \in \Lambda_1 \text{ s.t. } \forall \lambda_2 \in \Lambda_2,$$

$$\text{TP}(\text{Outcome}_G(s_{init}, \lambda_1, \lambda_2)) \geq v$$

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

<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 $\cap$ coNP</td>
<td>mem-less</td>
</tr>
<tr>
<td><strong>TP / TP</strong></td>
<td>NP $\cap$ coNP</td>
<td>mem-less</td>
</tr>
</tbody>
</table>

- Long tradition of study. Non-exhaustive selection:
  [EM79, ZP96, Jur98, GZ04, GS09, CDHR10, VR11, CRR12]
- $k$-dimension: weights = integer vectors
- *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
    $$C_i \leftarrow C_i + 1$$
  - Decrement
    $$C_i \leftarrow C_i - 1$$
  - Zero test and branch accordingly
    $$\text{If } C_i = 0 \text{ do this else do that}$$
- 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 $C_1$

- Decrement $C_1$
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.)

\[
\begin{array}{c}
(0,0,0,0,1) \\
\end{array}
\]

▷ 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>k-dimension</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<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><strong>MP</strong> / <strong>MP</strong></td>
<td>NP $\cap$ coNP</td>
<td>mem-less</td>
<td></td>
<td>coNP-c. / NP $\cap$ coNP</td>
<td>infinite</td>
<td>mem-less</td>
</tr>
<tr>
<td><strong>TP</strong> / <strong>TP</strong></td>
<td>NP $\cap$ coNP</td>
<td>mem-less</td>
<td></td>
<td>Undec.</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
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
    - P membership for the one-dim. case is a long-standing open problem
    - TP undecidable in $k$-dim.
  - Infimum vs. supremum
  - **no timing guarantee**: the “good behavior” occurs at the limit...
Window objectives: key idea

- **Window** of fixed size **sliding** along a play
  - 🔄 defines a local finite horizon

- Objective: see a **local** $MP \geq 0$ *before hitting the end* of the window
  - 🔄 needs to be verified at every step
Window objectives: key idea

- **Window** of fixed size **sliding** along a play
  \( \leadsto \) defines a local finite horizon

- Objective: see a **local** \( MP \geq 0 \) **before hitting the end** of the window
  \( \leadsto \) needs to be verified at every step

- 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 $= 4$
Multiple variants

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

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

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

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

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

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

- A window \textit{closes} when the sum becomes positive

- A window is \textit{open} if not yet closed
Examples

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

▷ **FW(2)** is satisfied, **DBW** is not, **MP** is satisfied.

▷ **MP** is satisfied but none of the window objectives is.
Conservative approximation of MP (one-dim.)

The following are true

Any window obj. ⇒ \( BW \) ⇒ MP ≥ 0

\( BW \) ⇐ MP > 0
# Results overview

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

$|S|$ the # of states, $V$ the length of the binary encoding of weights, and $l_{max}$ the window size.
## 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>$\mathcal{P}_1$ mem.</td>
</tr>
<tr>
<td><strong>MP / MP</strong></td>
<td>$\text{NP} \cap \text{coNP}$</td>
<td>mem-less</td>
</tr>
<tr>
<td><strong>TP / TP</strong></td>
<td>$\text{NP} \cap \text{coNP}$</td>
<td>mem-less</td>
</tr>
<tr>
<td><strong>WMP: fixed</strong></td>
<td>$\text{P-c.}$</td>
<td>mem. req.</td>
</tr>
<tr>
<td>polynomial window</td>
<td>$\text{P}(</td>
<td>S</td>
</tr>
<tr>
<td><strong>WMP: fixed</strong></td>
<td>$\text{NP} \cap \text{coNP}$</td>
<td>mem-less</td>
</tr>
<tr>
<td>arbitrary window</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>WMP: bounded</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>window problem</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- $|S|$ 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 $\text{P}$
- For multi-dim. games with fixed windows, we are **decidable**
- Window obj. provide **timing guarantees**
## 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 $\cap$ coNP</td>
<td>mem-less</td>
</tr>
<tr>
<td><strong>TP / TP</strong></td>
<td>NP $\cap$ coNP</td>
<td>mem-less</td>
</tr>
<tr>
<td><strong>WMP: fixed</strong></td>
<td>$P$-c.</td>
<td>mem. req.</td>
</tr>
<tr>
<td>polynomial window</td>
<td></td>
<td>$\leq \text{linear}(</td>
</tr>
<tr>
<td><strong>WMP: fixed</strong></td>
<td>$P(</td>
<td>S</td>
</tr>
<tr>
<td>arbitrary window</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>WMP: bounded</strong></td>
<td>NP $\cap$ coNP</td>
<td>infinite</td>
</tr>
<tr>
<td>window problem</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.
- 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

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

- 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

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

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

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

- 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

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

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

\[ G \]

\[ \text{GW}(l_{\text{max}}) \]
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.
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.
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

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

- 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
- $P_2$ chooses transitions to fire, which consume tokens
- $P_1$ can branch or continue (and apply reset, then output)
The construction in a nutshell

If no infinite execution exists, at some point, \( P_2 \) must choose a transition without the needed tokens on some place \( p \).

The window closes on dimension \( p \).

By branching \( P_1 \) 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
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>$P_1$ mem.</td>
</tr>
<tr>
<td>MP / MP</td>
<td>NP $\cap$ coNP</td>
<td>mem-less</td>
</tr>
<tr>
<td>TP / TP</td>
<td>NP $\cap$ coNP</td>
<td>mem-less</td>
</tr>
<tr>
<td>WMP: fixed polynomial window</td>
<td>$P$-c.</td>
<td>mem. req.</td>
</tr>
<tr>
<td>WMP: fixed arbitrary window</td>
<td>$P(</td>
<td>S</td>
</tr>
<tr>
<td>WMP: bounded window problem</td>
<td>NP $\cap$ coNP</td>
<td>mem-less</td>
</tr>
</tbody>
</table>

- Conservative approximation of MP/TP
- Provides timing guarantees
- Breaks the NP $\cap$ coNP barrier in one-dim. poly. window case
- Decidable approximation of TP in multi-dim. case
- Open question: is BW decidable in multi-dim. ?
Check the full version on arXiv!  

abs/1302.4248

Thanks!

Do not hesitate to discuss with us!
References I

Generalized mean-payoff and energy games.

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

Alternation.

K. Chatterjee, M. Randour, and J.-F. Raskin.
Strategy synthesis for multi-dimensional quantitative objectives.

C. Dufourd, A. Finkel, and P. Schnoebelen.
Reset nets between decidability and undecidability.

A. Ehrenfeucht and J. Mycielski.
Positional strategies for mean payoff games.

N. Fijalkow and F. Horn.
The surprising complexity of generalized reachability games.
References II

Games through nested fixpoints.

H. Gimbert and W. Zielonka.
When can you play positionally?

M. Jurdziński, J. Sproston, and F. Laroussinie.
Model checking probabilistic timed automata with one or two clocks.

M. Jurdziński.
Deciding the winner in parity games is in UP \cap co-UP.

Nets with tokens which carry data.

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.
References III

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

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