Games with window parity objectives

Véronique Bruyère\textsuperscript{1}  Quentin Hautem\textsuperscript{1}  Mickael Randour\textsuperscript{2}
Jean-François Raskin\textsuperscript{2}

University of Mons\textsuperscript{1}, Université Libre de Bruxelles\textsuperscript{2}

April 2016
1 Synthesis

2 Games on graphs

3 Objectives

4 Window Mean-Payoff

5 Window Parity

6 Conclusion
Synthesis via Game Theory

Environment → System → Properties

Model with a game → Model with winning conditions

SYNTHESIS

Winning strategy?

Yes

Strategy = controller

No

Empower system or weaken specification requirements
<table>
<thead>
<tr>
<th></th>
<th>Synthesis</th>
<th>Games on graphs</th>
<th>Objectives</th>
<th>Window Mean-Payoff</th>
<th>Window Parity</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Synthesis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Games on graphs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Objectives</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Window Mean-Payoff</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Window Parity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Conclusion</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Games on graphs

- Model the antagonistic interaction between the system (○) and the environment (□).

- Vertices and edges.
Model the antagonistic interaction between the system (◯) and the environment (□).

A play starts in an initial vertex: imagine a token in the current vertex.
Games on graphs

- Model the antagonistic interaction between the system (○) and the environment (□).

- The player who owns the current vertex decides where the token goes (Turn-based).
- Players follow strategies: functions that associate to each history of the game a vertex.
Games on graphs

- Model the antagonistic interaction between the system (○) and the environment (□).

- The player who owns the current vertex decides where the token goes (Turn-based).

- Players follow strategies: functions that associate to each history of the game a vertex.
Games on graphs

- Model the antagonistic interaction between the system (○) and the environment (□).

- The player who owns the current vertex decides where the token goes (Turn-based).

- Players follow strategies: functions that associate to each history of the game a vertex.
Games on graphs

- Model the antagonistic interaction between the system (◯) and the environment (□).

- The player who owns the current vertex decides where the token goes (Turn-based).

- Players follow strategies: functions that associate to each history of the game a vertex.
Games on graphs

- Model the antagonistic interaction between the system (○) and the environment (□).

The player who owns the current vertex decides where the token goes (Turn-based).

Players follow strategies: functions that associate to each history of the game a vertex.
Games on graphs

- Model the antagonistic interaction between the system (○) and the environment (□).

Plays are infinite. They are said **winning** for a player if they satisfy his **winning condition**, otherwise they are said losing for this player.

→ Ex: vertex $v_2$ has to be visited infinitely often visited.
Games on graphs

- Model the antagonistic interaction between the system (○) and the environment (□).

- Plays are infinite. They are said winning for a player if they satisfy his winning condition, otherwise they are said losing for this player.

→ Ex: vertex $v_2$ has to be visited infinitely often visited.
Games on graphs

- Model the antagonistic interaction between the system ($\bigcirc$) and the environment ($\square$).

- Plays are infinite. They are said winning for a player if they satisfy his winning condition, otherwise they are said losing for this player.

  $\rightarrow$ Ex: vertex $v_2$ has to be visited infinitely often visited.
Games on graphs

- Model the antagonistic interaction between the system (◯) and the environment (□).

- Plays are infinite. They are said winning for a player if they satisfy his winning condition, otherwise they are said losing for this player.

→ Ex: vertex $v_2$ has to be visited infinitely often visited.
Games on graphs

- Model the antagonistic interaction between the system (□) and the environment (○).

- Plays are infinite. They are said *winning* for a player if they satisfy his *winning condition*, otherwise they are said losing for this player.

→ Ex: vertex $v_2$ has to be visited infinitely often visited.
Questions

Given a game structure $G$, an objective $\Omega$ and an initial vertex $v_0$,

- Does one player have a winning strategy from the initial vertex?
- If yes, can we decide which one?
- What is the complexity of the decision problem?
- How much memory is needed for a winning strategy?
<table>
<thead>
<tr>
<th></th>
<th>Synthesis</th>
<th>Games on graphs</th>
<th>Objectives</th>
<th>Window Mean-Payoff</th>
<th>Window Parity</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Objectives

Focus on two objectives.

- Let $w : E \rightarrow \mathbb{Z}$ be a weight function.
  Mean-Payoff (MP) : $\lim \liminf / \limsup$ of the average weight $\geq \nu^1$.

- Let $p : V \rightarrow \{0, \ldots, k\}$ be a parity function.
  Parity : minimum priority seen infinitely often is even.

---

$^1$We can assume w.l.o.g. that $\nu = 0$
Example (1/2)

Player $\bigcirc$ has a winning \textit{memoryless} strategy to ensure a $\text{MP} \geq 0$.
Example (2/2)

Player $\bigcirc$ has a winning **memoryless** strategy to ensure the Parity objective.
**Known results** [Jur98, ZP96]

<table>
<thead>
<tr>
<th>Complexity</th>
<th>MP</th>
<th>Parity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complexity</td>
<td>NP ∩ coNP</td>
<td>memoryless</td>
</tr>
<tr>
<td>P1 memory</td>
<td>memoryless</td>
<td></td>
</tr>
<tr>
<td>P2 memory</td>
<td>memoryless</td>
<td></td>
</tr>
</tbody>
</table>
Known results [Jur98, ZP96]

<table>
<thead>
<tr>
<th></th>
<th>MP</th>
<th>Parity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complexity</td>
<td>NP ∩ coNP</td>
<td>memoryless</td>
</tr>
<tr>
<td>P1 memory</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P2 memory</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Open question: Is there a polynomial algorithm to solve these games?
Known results [Jur98, ZP96]

<table>
<thead>
<tr>
<th>Complexity</th>
<th>MP</th>
<th>Parity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complexity</td>
<td>MP</td>
<td>Parity</td>
</tr>
<tr>
<td>P1 memory</td>
<td>MP</td>
<td>memoryless</td>
</tr>
<tr>
<td>P2 memory</td>
<td>MP</td>
<td>memoryless</td>
</tr>
</tbody>
</table>

Open question: Is there a polynomial algorithm to solve these games?

MP and Parity objectives deal with limit behavior.

\[\sim\] No explicit bound.
<table>
<thead>
<tr>
<th></th>
<th>Synthesis</th>
<th>Games on graphs</th>
<th>Objectives</th>
<th>Window Mean-Payoff</th>
<th>Window Parity</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Synthesis</td>
<td>Games on graphs</td>
<td>Objectives</td>
<td>Window Mean-Payoff</td>
<td>Window Parity</td>
<td>Conclusion</td>
</tr>
<tr>
<td>2</td>
<td>Games on graphs</td>
<td>Objectives</td>
<td>Window Mean-Payoff</td>
<td>Window Parity</td>
<td>Conclusion</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Objectives</td>
<td>Window Mean-Payoff</td>
<td>Window Parity</td>
<td>Conclusion</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Window Mean-Payoff</td>
<td>Window Parity</td>
<td>Conclusion</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Window Parity</td>
<td>Conclusion</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Conclusion</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Fixed Window Mean-Payoff

- Idea: average weight over a local **bounded** window sliding along the play

**Definition**

Given a threshold $\nu^2 \in \mathbb{Q}$ and a window size $\lambda \in \mathbb{N}\setminus\{0\}$,

$$\text{DirFWMP}(\lambda, \nu) = \{ \rho \in \text{Plays}(G) \mid \forall k \geq 0, \exists l \in \{1, \ldots, \lambda\}, MP(\rho[k,k+l]) \geq \nu \},$$

$$\text{FWMP}(\lambda, \nu) = \{ \rho \in \text{Plays}(G) \mid \exists i \in \mathbb{N}, \rho[i...] \in \text{DirFWMP}(\lambda, \nu) \}.$$  

\[^2\text{W.l.o.g we can assume that } \nu = 0\]
$TP(\rho \leq i)$
$TP(\rho \leq i)$

$\lambda = 3$
\[ TP(\rho \leq i) \]

\[ \lambda = 3 \]
$TP(\rho_{\leq i})$

$\lambda = 3$
$TP(\rho \leq i)$

$\lambda = 3$

\[\vdots\]
\[ TP(\rho \leq i) \]

\[ \lambda = 3 \]
\[ TP(\rho_{\leq i}) \]

\[ \lambda = 3 \]
\[ TP(\rho \leq i) \]

\[ \lambda = 3 \]

\[ TP(\rho \leq i) \]

\[ \lambda = 3 \]

...
$TP(\rho \leq i)$

$\lambda = 3$
Example

$\text{DirFWMP}(\lambda, 0)$ with $\lambda = 2$. 

\begin{center}
\begin{tikzpicture}
  \node[draw] (v0) at (0, 0) {$v_0$};
  \node[draw] (v1) at (2, 0) {$v_1$};
  \draw[->] (v0) -- node[below] {0} (v1);
  \draw[->] (v1) -- node[above] {0} (v0);
  \draw[->] (v0) -- node[above] {−1} (v1);
  \draw[->] (v1) -- node[below right] {+1} (v0);
\end{tikzpicture}
\end{center}
Example

\[ \text{DirFWMP}(\lambda, 0) \text{ with } \lambda = 2. \]

Approximations

- \( \text{WMP} \geq 0 \Rightarrow \text{MP} \geq 0. \)
- \( \text{WMP} \geq 0 \Leftrightarrow \text{MP} > 0. \)
Bounded Window Mean-Payoff

**Definition**

Given a threshold $\nu \in \mathbb{Q}$,

\[
\text{DirBndWMP}(\nu) = \{ \rho \in \text{Plays}(G) \mid \exists \lambda \in \mathbb{N} \setminus \{0\}, \rho \in \text{DirFWMP}(\lambda, \nu) \}, \\
\text{BndWMP}(\nu) = \{ \rho \in \text{Plays}(G) \mid \exists \lambda \in \mathbb{N} \setminus \{0\}, \rho \in \text{FWMP}(\lambda, \nu) \}.
\]
Bounded Window Mean-Payoff

Definition

Given a threshold $\nu \in \mathbb{Q}$,

$$\text{DirBndWMP}(\nu) = \{ \rho \in \text{Plays}(G) \mid \exists \lambda \in \mathbb{N} \setminus \{0\}, \rho \in \text{DirFWMP}(\lambda, \nu) \},$$

$$\text{BndWMP}(\nu) = \{ \rho \in \text{Plays}(G) \mid \exists \lambda \in \mathbb{N} \setminus \{0\}, \rho \in \text{FWMP}(\lambda, \nu) \}.$$
Known results [CDRR15]

### One dimension

<table>
<thead>
<tr>
<th>Complexity</th>
<th>P1 memory</th>
<th>P2 memory</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WMP</strong>: fixed polynomial windows</td>
<td><strong>P-complete</strong></td>
<td>pseudo-polynomial</td>
</tr>
<tr>
<td><strong>WMP</strong>: fixed arbitrary windows</td>
<td>(P(</td>
<td>V</td>
</tr>
<tr>
<td><strong>WMP</strong>: bounded window problem</td>
<td>(\text{NP} \cap \text{coNP})</td>
<td>memoryless</td>
</tr>
</tbody>
</table>
### Known results [CDRR15]

#### Multiple dimensions

<table>
<thead>
<tr>
<th>Complexity</th>
<th>P1 memory</th>
<th>P2 memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>WMP : fixed polynomial windows</td>
<td>PSPACE-hard</td>
<td></td>
</tr>
<tr>
<td></td>
<td>EXPTIME-easy</td>
<td>exponential</td>
</tr>
<tr>
<td>WMP : fixed arbitrary windows</td>
<td>EXPTIME-complete</td>
<td>exponential</td>
</tr>
<tr>
<td>WMP : bounded window problem</td>
<td>NPR-hard</td>
<td>-</td>
</tr>
</tbody>
</table>
1 Synthesis

2 Games on graphs

3 Objectives

4 Window Mean-Payoff

5 Window Parity

6 Conclusion
Fixed Window Parity

Same idea!
Fixed Window Parity

Same idea!

\[
\begin{align*}
\cdots & \rho_k \cdots \\
\rho_k & \geq 0 \\
\rho_k+1 & \cdots \\
\cdots & \lambda \cdots
\end{align*}
\]
Fixed Window Parity

Same idea!

$$\min p(\rho_i) \text{ is even}$$

$$\rho_k \cdots \rho_{k+1} = \lambda$$
Fixed Window Parity

Same idea!

\[
\begin{align*}
\min p(\rho_i) \text{ is even}\\
\rho_k & \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \rho_{k+1}\\
\cdots & \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \cdots\\
\end{align*}
\]

\[= \lambda\]

Definition

Given a parity function \( p \) and a window size \( \lambda \in \mathbb{N} \setminus \{0\} \),

\[
\begin{align*}
\text{DirFWP}(\lambda, p) &= \left\{ \rho \in \text{Plays}(G) \mid \forall k \geq 0, \exists l \in \{1, \ldots, \lambda\}, \right. \\
&\quad \min\{p(\rho_i) \mid i \in \{k, \ldots, k + l\}\} \text{ is even} \right\}, \\
\text{FWP}(\lambda, p) &= \left\{ \rho \in \text{Plays}(G) \mid \exists i \in \mathbb{N}, \rho[i...] \in \text{DirFWP}(\lambda, p) \right\}.
\end{align*}
\]
Example

\[ \rho \notin \text{DirFWP}(\lambda = 2, p) \]

\[ \rho \in \text{DirFWP}(\lambda = 3, p) \]

[Diagram of a graph with nodes labeled v0, v1, v2, v3 and edges showing transitions between them, with numbers 3, 1, 2, 0 on the nodes.]

- DirFWP(\lambda, p) \Rightarrow \text{Parity}(p)
Bounded Window Parity

Definition

Given a parity function $p$,

$$\text{DirBndWP}(p) = \{ \rho \in \text{Plays}(G) \mid \exists \lambda \in \mathbb{N} \setminus \{0\}, \ \rho \in \text{DirFWP}(\lambda, p) \},$$

$$\text{BndWP}(p) = \{ \rho \in \text{Plays}(G) \mid \exists \lambda \in \mathbb{N} \setminus \{0\}, \ \rho \in \text{FWP}(\lambda, p) \}. $$
Bounded Window Parity

Definition

Given a parity function $p$,

$$\text{DirBndWP}(p) = \{ \rho \in \text{Plays}(G) \mid \exists \lambda \in \mathbb{N} \setminus \{0\}, \rho \in \text{DirFWP}(\lambda, p) \},$$

$$\text{BndWP}(p) = \{ \rho \in \text{Plays}(G) \mid \exists \lambda \in \mathbb{N} \setminus \{0\}, \rho \in \text{FWP}(\lambda, p) \}.$$

Losing for $\text{DirBndWP}(p)$ from $v_0$ but winning for $\text{Parity}(p)$. 
Interesting Fact (1/3)

Definition

Let $\rho$ be a play, $k$ be a position and $p$ be a parity function.

$$dist_k(\rho, p) = \begin{cases} 0 & \text{if } p(\rho_k) \text{ is even;} \\ \inf \{k - k' \mid k' > k, \ p(\rho_{k'}) \text{ is even and } p(\rho_{k'}) < p(\rho_k)\} & \text{if } p(\rho_k) \text{ is odd.} \end{cases}$$

Theorem

$$\text{DirBndWP}(p) = \{\rho \in \text{Plays}(G) \mid \exists \lambda \in \mathbb{N} \setminus \{0\}, \ \forall k \geq 0 \ dist_k(\rho, p) \leq \lambda\}.$$
Interesting Fact (1/3)

Definition

Let $\rho$ be a play, $k$ be a position and $p$ be a parity function.

\[
dist_k(\rho, p) = \begin{cases} 
0 & \text{if } p(\rho_k) \text{ is even;} \\
\inf \{k - k' \mid k' > k, \ p(\rho_{k'}) \text{ is even and } p(\rho_{k'}) < p(\rho_k)\} & \text{if } p(\rho_k) \text{ is odd.}
\end{cases}
\]

Theorem

\[
\text{DirBndWP}(p) = \{\rho \in \text{Plays}(G) \mid \exists \lambda \in \mathbb{N} \setminus \{0\}, \ \forall k \geq 0 \ dist_k(\rho, p) \leq \lambda\}.
\]

This set is studied in [CHH09] and it is showed to be in PTIME.
New definitions?! Parity-Response

\[ p(\rho_j) > p(\rho_{j'}) \text{ even and } j' \text{ min} \]
New definitions?! Parity-Response

Definition

Given a parity function $p$ and a bound $\lambda \in \mathbb{N} \setminus \{0\}$,

$$\text{DirPR}(\lambda, p) = \left\{ \rho \in \text{Plays}(G) \mid \forall j \geq 0, \ dist_j(\rho, p) \leq \lambda \right\},$$

$$\text{PR}(\lambda, p) = \left\{ \rho \in \text{Plays}(G) \mid \exists i \in \mathbb{N}, \rho[i...] \in \text{DirPR}(\lambda, p) \right\}.$$
Interesting Fact (2/3)

\[ \text{DirFWP}(\lambda, \rho) \subset \text{DirPR}(\lambda, \rho) \]

\[ \lambda = 2, \quad \rho \in \text{DirPR}(2, \rho) \text{ but } \rho \notin \text{DirFWP}(2, \rho). \]
### Interesting fact (3/3)

#### Definition ([CHH09])

Given a parity function $p$,

\[
\begin{align*}
\text{DirBndPR}(p) &= \{ \rho \in \text{Plays}(G) \mid \exists \lambda \in \mathbb{N} \setminus \{0\}, \rho \in \text{DirPR}(\lambda, p) \}, \\
\text{BndPR}(p) &= \{ \rho \in \text{Plays}(G) \mid \exists \lambda \in \mathbb{N} \setminus \{0\}, \rho \in \text{PR}(\lambda, p) \}.
\end{align*}
\]

#### Theorem

\[
\text{DirBndWP}(p) = \text{DirBndPR}(p) \quad \text{and} \quad \text{BndWP}(p) = \text{BndPR}(p).
\]
Wrap-up

- **Window parity (WP):**
  \[
  \min p(\rho_i) \text{ is even}
  \]

- **Parity-response (PR):**
  \[
  p(\rho_j) > p(\rho_{j'}) \text{ even and } j' \text{ min}
  \]
## Preliminary results

<table>
<thead>
<tr>
<th>Complexity</th>
<th>One-dimensional</th>
<th>Multi-dimensional</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Direct) Fixed WP</td>
<td>P-complete</td>
<td></td>
</tr>
<tr>
<td>(Direct) Bnd WP</td>
<td></td>
<td>EXPTIME-complete</td>
</tr>
<tr>
<td>(Direct) PR</td>
<td>PSPACE-easy</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P-hard</td>
<td></td>
</tr>
<tr>
<td>(Direct) Bnd PR</td>
<td>P-easy [CHH09]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P-hard</td>
<td></td>
</tr>
</tbody>
</table>
1 Synthesis

2 Games on graphs

3 Objectives

4 Window Mean-Payoff

5 Window Parity

6 Conclusion
<table>
<thead>
<tr>
<th>Complexity</th>
<th>One-dimensional</th>
<th>Multi-dimensional</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Direct) Fixed WP</td>
<td>P-complete</td>
<td></td>
</tr>
<tr>
<td>(Direct) Bnd WP</td>
<td></td>
<td>EXPTIME-complete</td>
</tr>
<tr>
<td>(Direct) PR</td>
<td>PSPACE-easy</td>
<td>P-hard</td>
</tr>
<tr>
<td>(Direct) Bnd PR</td>
<td>P-easy [CHH09]</td>
<td>P-hard</td>
</tr>
<tr>
<td>Complexity</td>
<td>One-dimensional</td>
<td>Multi-dimensional</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>(Direct) Fixed WP</td>
<td>P-complete</td>
<td></td>
</tr>
<tr>
<td>(Direct) Bnd WP</td>
<td>P-complete</td>
<td>EXPTIME-complete</td>
</tr>
<tr>
<td>(Direct) PR</td>
<td>PSPACE-easy P-hard</td>
<td></td>
</tr>
<tr>
<td>(Direct) Bnd PR</td>
<td>P-easy [CHH09] P-hard</td>
<td></td>
</tr>
</tbody>
</table>
- Multi-dimensional WMP objective.

<table>
<thead>
<tr>
<th></th>
<th>Complexity</th>
<th>P1 memory</th>
<th>P2 memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>WMP : fixed polynomial windows</td>
<td>PSPACE-hard</td>
<td></td>
<td>exponential</td>
</tr>
<tr>
<td></td>
<td>EXPTIME-easy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WMP : fixed arbitrary windows</td>
<td>EXPTIME-complete</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WMP : bounded window problem</td>
<td>NPR-hard</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Work in progress \( \sim \) Paper soon on ArXiV.
Work in progress ↷ Paper soon on ArXiv.

Thank you!
Krishnendu Chatterjee, Laurent Doyen, Mickael Randour, and Jean-François Raskin.
Looking at mean-payoff and total-payoff through windows.

K. Chatterjee, T.A. Henzinger, and F. Horn.
Finitary winning in omega-regular games.

Marcin Jurdzinski.
Deciding the winner in parity games is in UP n co-up.

Uri Zwick and Mike Paterson.
The complexity of mean payoff games on graphs.