Highlights on Recent Progresses in Quantitative Games

J.-F. Raskin
Université Libre de Bruxelles, U.L.B.

based on joint works with V. Bruyère (UMons), K. Chatterjee (IST Austria), L. Doyen (ENS Cachan), E. Filiot (ULB), T. Henzinger (IST Austria), A. Rabinovitch (U Tel Aviv), M. Randour (U Mons) and Y. Velner (U Tel Aviv)
Games for Synthesis (of Reactive Systems)

- support the design process with automatic synthesis

Sys is constructed by an algorithm
- Sys is \textbf{correct} by construction
- Underlying theory: \textit{2-player zero-sum games}
- Env is \textit{adversarial} (worst-case assumption)

\textbf{Winning strategy} = \textbf{Correct Sys}
Preliminaries:
2-player Zero-sum Games on Graphs
2-player Zero-sum Games on Graphs

Finite directed graph
Player 1 vertices
Player 2 vertices
How to play?

A token is moved along vertices

Infinite number of rounds:

- in each round: the player that owns the vertex moves the token to an adjacent vertex

Outcome=infinte path
2-player Zero-sum Games on Graphs
2-player Zero-sum Games on Graphs
2-player Zero-sum Games on Graphs
2-player Zero-sum Games on Graphs

1 ➞ 2 ➞ 1 ➞ 4
2-player Zero-sum Games on Graphs

1 → 2 → 1 → 4 → 5
2-player Zero-sum Games on Graphs

1 → 2 → 1 → 4 → 5 → 3
2-player Zero-sum Games on Graphs

1 → 2 → 1 → 4 → 5 → 3 → 4
2-player Zero-sum Games on Graphs
2-player Zero-sum Games on Graphs

1 → 2 → 1 → 4 → 5 → 3 → 4 → 5 → 4 → ...
2-player Zero-sum Games on Graphs

Who is winning? winning conditions

Win_1 \subseteq V^\omega: Set of good outcomes (paths) for Player 1

Win_2 = V^\omega \setminus Win_1 (zero sum)

Examples of winning conditions:

• Win_1 = \{ \pi \mid \pi \ visits \ Good \} 
  Reachability winning condition

• Win_1 = \{ \pi \mid \pi \ visits \ Good \ infinitely \ often \} 
  Büchi winning condition
Strategies

Unfolding of the game graph
Strategies

Unfolding of the game graph
Strategies

Unfolding of the game graph
Strategy for Player 1 =
One choice in each node of Player 1 in tree unfolding
Strategies

Strategy for Player 1 =
One choice in each node of Player I in tree unfolding

\[ \lambda_1 : V^*.V_1 \rightarrow \text{edge} \]
Strategy for Player 1 = One choice in each node of Player I in tree unfolding

\[ \lambda_1 : V^* \cdot V_1 \rightarrow \text{edge} \]

Strategy is **winning** (for Player 1), if all branches of the resulting tree winning paths
Types of strategies

(\text{Player 1}) \text{ strategy:}

\( \lambda_1 : V^* \rightarrow \text{edge.} \)

\( \Sigma_1 = \text{set of strategies of Pl.1} \)

\text{Finite-memory strategy:}

\( \lambda_{1,f} : V^* \rightarrow \text{edge but regular (Moore machine)} \)

\( \Sigma_{1,f} = \text{set of finite memory strategies of Player 1} \)

\text{Memoryless strategy:}

\( \lambda_{1,m} : V_1 \rightarrow \text{edge.} \)

\( \Sigma_{1,m} = \text{set of memoryless strategies of Player 1} \)

\text{Randomized strategy:}

\( \lambda_{1,m} : V^* \rightarrow \text{Dist(edge).} \)

\( \Sigma_{1,m} = \text{set of randomized strategies of Player 1} \)
Decision Problem - Determinacy

Given:

• a game graph $G$
• a winning condition $\text{Win}_1 \subseteq V^\omega$

• decide if Player 1 has a winning strategy

• determinacy:
  either Player 1 has a winning for $\text{Win}_1$
  or Player 2 has a winning strategy for $\text{Win}_2 = V^\omega \setminus \text{Win}_1$

It is true for a large class of objectives, e.g. $\omega$-regular objectives
2-player Zero-sum Games on Graphs

One token is placed on initial vertex

We play an infinite number of rounds:

- In each round: the player that owns the vertex moves the token to adjacent vertex

Outcome = infinite path

Synthesis of reactive systems:
Player I = system and Player II = environment
For embedded systems, we need quantities!
Plan

★ Weighted game graphs:
  ★ Mean-payoff and Energy games
  ★ From one dimension to \( k \) dimensions
  ★ Window mean-payoff objectives
Example of Classical Quantitative Objectives:
Mean-Payoff Games
Energy Games
Mean-Payoff Games [EM79]

defined on weighted directed graphs
Mean-Payoff Games \[EM79]\]

Edges are labelled with rewards

\[(1,4) (4,5) (5,4) \ldots (4,5) (5,4) \ldots = \text{play}
\]

\[= \lim_{n \to +\infty} \inf/\sup \sum_{i=1}^{n} r_i \]

\[= \text{MP}((1,4) (4,5) (5,4) \ldots (4,5) (5,4) \ldots) = 1\]

\[\text{Win} = \{ \text{play} \mid \text{MP}(\text{play}) \geq c\}\]

Note: not \(\omega\)-regular.
Mean-Payoff Games [EM79]

Edges are labelled with rewards:

\[(1,4) (4,5) (5,4) \ldots (4,5) (5,4) \ldots = \text{play}\]

\[4 \quad 3 \quad -1 \quad 3 \quad -1 \quad \ldots\]

\(\text{MP} = \lim \sup_{n \to +\infty} \frac{\sum_{i=1}^{n} r_i}{n} = 1\)

\[= \text{MP}(\text{system}, \text{environment})\]

\(\text{MP}\): positions of maximizer = system

\(\text{MP}\): positions of minimizer = environment

Note: not \(\omega\)-regular.
Mean-Payoff Games \[EM79\]

Lim Sup - Lim Inf do not define the same set of plays.
Energy Games [BFMLS08-CdAHS03]

Edges are labelled with rewards

\[(1,4) (4,5) (5,4) \ldots (4,5) (5,4) \ldots = \text{play}\]

\[4 \quad 3 \quad -1 \quad 3 \quad -1 \quad \ldots = \inf_{n \to +\infty} \sum_{i=1\ldots n} r_i\]

\[= E((1,4) (4,5) (5,4) \ldots (4,5) (5,4) \ldots)=0\]

\[\exists c \in \mathbb{Z} \text{ such that Pl. 1 enforces } \text{Win}(c)=\{ \text{play} | E(\text{play}) \geq c \}\]

Note: not \(\omega\)-regular.
Theorem [EM79,Jur98,ZP97,GZ09,BFMLS08]

(i) In mean-payoff, and energy games, the two players can play optimally with memoryless strategies

(ii) The winner can be decided in $\text{NP} \cap \text{coNP}$

Rem: Those results hold no matter if $\lim\sup/\lim\inf$ is used in the definition
MP/Energy Games

**Theorem [EM79,Jur98,ZP97,GZ09,BFMLS08]**

(i) In mean-payoff, and energy games, the two players can play optimally with memoryless strategies

(ii) The winner can be decided in $\text{NP} \cap \text{coNP}$

Rem: Those results hold no matter if $\lim \sup$/$\lim \inf$ is used in the definition

Open question: are MP/E games solvable in PTime?
Interlude:
Game trees and
Zermelo’s Theorem
Tree arenas

A branch
=a play in the game tree
Tree arenas
Zermelo's theorem states that at the root of the tree either Player 1 has a strategy to force a green leaf, or Player 2 has a strategy to force a red leaf. = determinacy
Proof by *induction* on the depth of the tree.

Each state of the tree can be labelled (backwardly):

- in **green** if Player 1 can force green leaves from there, and
- in **red** if Player 2 can force red leaves from there.
A tree based algorithm for MP/Energy games
FCU for solving MPG

**FCU**: unfold the weighted graph up to a first repetition of vertex:
- a leaf is **winning for Pl. 1** if the cycle has a non-negative sum
- a leaf is **winning for Pl. 2** if the cycle has a negative sum

⇒ By Zermelo's theorem:
either Pl. 1 can force **non-negative cycles**
or Pl. 2 can force **negative cycles**
Unfolding - an example

Unfolding of A starting in 1

A

Unfolding of A starting in 1

w(cycle) = 1

w(cycle) = -1

w(cycle) = 1
Unfolding - an example

In this example, Player 1 can force non-negative cycles!
Transfer of strategies

**Lemma [strategy transfer]** Winning strategies in the FCU can be transferred into winning strategies in the MP/EG game:

- If Player 1 can force green leaves in the unfolding of A then Player 1 has a winning strategy in A for the $\text{MP} \geq 0$ objective and in the EG;

- If Player 2 can force red leaves in the unfolding of A then Player 2 has a winning strategy in A for the $\text{MP} < 0$ objective and in the EG.

**Corollaries:**

1) $\text{MPG} \approx \text{EG}$ MPG $A \cdot \geq v$ and $\text{EG} A - v$ are equivalent!
2) MPG and EG are determined
Player 1 ensures MP value 1 in $A$ iff Player 1 wins EG $A^{-1}$. 

$\text{c}_0 = 6$
Unfolding - an example

Unfolding of A starting in \( 1 \)

Player 1 can force non-negative cycles!

Initial credit \( n.W \) is always sufficient!
A Pseudo-Polynomial Time Algorithm for EG
**EG and safety games**

\[ \text{SAFE}_i = \text{set of } (s, c) \in S_1 \times \mathbb{N} \cup S_2 \times \mathbb{N} \text{ s.t. from } (s, c), \]
Player 1 can maintain energy level non-negative for \( i \) steps

What are the controllable predecessors of \( \text{SAFE}_i \)?

\( \text{SAFE}_i \) is an upward closed set
We define $\preceq$ as $(s,c) \preceq (s',c')$ iff $s=s'$ and $c \leq c'$

- CPRE($X$) transforms $\preceq$-upward closed sets into $\preceq$-upward closed sets

CPRE$_1(X)$ where $X \subseteq (S_1 \times \mathbb{N} \cup S_2 \times \mathbb{N})$ is the set

\[
\{ (s_1,c) \in S_1 \times \mathbb{N} \mid \exists (s_1,w,s') \in E : (s',c+w) \in X \} \\
\cup \{ (s_2,c) \in S_2 \times \mathbb{N} \mid \forall (s_2,w,s') \in E : (s',c+w) \in X \}
\]
CPRE\(_1(X)\)

\[
\text{CPRE}_1(X) = \{ (s_1,c) \in S_1 \times \mathbb{N} \cup S_2 \times \mathbb{N} \mid \exists (s_1,w,s') \in E : (s',c+w) \in X \} \\
\cup \{ (s_2,c) \in S_2 \times \mathbb{N} \mid \forall (s_2,w,s') \in E : (s',c+w) \in X \}
\]

SAFE\(_0 = \{ (s_1,0),(s_2,0),(s_3,0),(s_4,0) \} \)
SAFE\(_1 = \{ (s_1,0),(s_2,2),(s_3,0),(s_4,1) \} \)
SAFE\(_2 = \{ (s_1,0),(s_2,2),(s_3,0),(s_4,2) \} \)
SAFE\(_3 = \{ (s_1,0),(s_2,2),(s_3,0),(s_4,3) \} \)
SAFE\(_4 = \{ (s_1,0),(s_2,2),(s_3,0),(s_4,4) \} \)
SAFE\(_5 = \{ (s_1,0),(s_2,2),(s_3,0),(s_4,5) \} \)
SAFE\(_6 = \{ (s_1,1),(s_2,2),(s_3,0),(s_4,6) \} \)
SAFE\(_7 = \{ (s_1,2),(s_2,2),(s_3,0),(s_4,7) \} \)
SAFE\(_8 = \{ (s_1,3),(s_2,2),(s_3,0),(s_4,8) \} \)
SAFE\(_9 = \{ (s_1,3),(s_2,2),(s_3,0),(s_4,9) \} \)
...
SAFE\(_k = \{ (s_1,3),(s_2,2),(s_3,0),(s_4,k) \} \) no stabilisation!
CPRE$_1[\mathbf{C}](X)$ to force termination

- Above energy requirement $\mathbf{C} \in \mathbb{N}$, we consider the game as **lost**! (**conservative approximation**)

- Let $\mathbf{C} \in \mathbb{N}$, define $U(\mathbf{C}) = \mathbb{P}((S_1 \cup S_2) \times \{0 \ldots \mathbf{C}\})$

- CPRE$_1[\mathbf{C}](X)$ where $X \in U(\mathbf{C})$ is the set

\[
\{ (s_1, c) \in S_1 \times \{0 \ldots \mathbf{C}\} \mid \exists (s_1, w, s') \in E : (s', c+w) \in X \} \\
\cup \{ (s_2, c) \in S_2 \times \{0 \ldots \mathbf{C}\} \mid \forall (s_2, w, s') \in E : (s', c+w) \in X \}
\]
CPRE\(_1\)[C](X) - properties

CPRE\(^*[C](X)\) is monotone over \(\mathcal{P}((S_1 \cup S_2) \times \{0\ldots C\})\)

- so it has a greatest fixed point, noted CPRE\(^*[C]\)
- computed iteratively from \(T = \mathcal{P}((S_1 \cup S_2) \times \{0\ldots C\})\)
- convergence is ensured now as \(\mathcal{P}((S_1 \cup S_2) \times \{0\ldots C\})\) is finite
- The greatest fixpoint computed in \(O(|V|.|E|.W)\), where \(W\) is the largest weight in absolute value in arena \(A\): the complexity is pseudo-polynomial
\textbf{CPRE}_1[\mathbf{C}](X)

\[
\begin{align*}
\text{SAFE}_0 &= \uparrow \{(s_1,0),(s_2,0),(s_3,0),(s_4,0)\} \\
\text{SAFE}_1 &= \uparrow \{(s_1,0),(s_2,2),(s_3,0),(s_4,1)\} \\
\text{SAFE}_2 &= \uparrow \{(s_1,0),(s_2,2),(s_3,0),(s_4,2)\} \\
\text{SAFE}_3 &= \uparrow \{(s_1,0),(s_2,2),(s_3,0),(s_4,3)\} \\
\text{SAFE}_4 &= \uparrow \{(s_1,3),(s_2,2),(s_3,0)\} \\
\text{SAFE}_5 &= \uparrow \{(s_1,3),(s_2,2),(s_3,0)\} = \text{SAFE}_\infty
\end{align*}
\]

\text{Stabilisation}!

\text{Greatest fixpoint}
Correctness

Theorem [correctness] \( \forall C \in \mathbb{N}, \forall (s,c) \in \mathcal{C} \text{PRE}_1^* [C], \) Player 1 wins EG from \( s \).
Completeness

- If $C = n \cdot W$ then the algorithm is complete.
- By FCU, a larger credit is never necessary.
- This fixpoint improves on the complexity obtained by Zwick and Paterson in [ZP96].
Quantitative Objectives:
Multi-dimension Extensions
Multi-dimension Extensions

As before but with vectors of weights instead of just one weight
Multi-dim. Energy Games
Lemma. Finite memory strategies are sufficient for Player 1 to win in MEGs.
Lemma. Finite memory strategies are sufficient for Player 1 to win in MEGs.

Proof. First, \((\mathbb{N}^k, \leq)\) is a well-quasi ordered set, i.e.:

1. \(\leq\) is a partial order (so a pre-order)

2. for all infinite sequences of elements \(m_0 \, m_1 \, m_2 \, \ldots \, m_n \, \ldots\) in \((\mathbb{N}^k)^\omega\),

   there exists \(i<j\) such that \(m_i \leq m_j\)
Lemma. Finite memory strategies are sufficient for Player 1 to win in MEGs.

Proof. First, \((\mathbb{N}^k, \leq)\) is a well-quasi ordered set.

Let \(\lambda_1\) be winning
Lemma. Finite memory strategies are sufficient for Player 1 to win in MEGs.

Proof. First, \((\mathbb{N}^k, \leq)\) is a well-quasi ordered set.
**Lemma.** Finite memory strategies are sufficient for Player 1 to win in MEGs.

**Proof.** First, \( (\mathbb{N}^k, \leq) \) is a well-quasi ordered set.

Let \( \lambda_1 \) be winning. On each branch with \( L_1 \leq L_2 \), stop and play as from \( L_1 \)!

Then \( \lambda'_1 \) is winning and finite memory.

\[ \text{wqo+Koenig’s lemma} \]
Lemma. Finite memory strategies are sufficient for Player 1 to win in MEGs.

Proof. First, \((\mathbb{N}^k, \leq)\) is a well-quasi ordered set.

Then \(\lambda'_1\) is winning and finite memory

Finite tree = winning strategy:

1. play according to the choices made in tree
2. in leaf, go to ancestor with lower or equal energy
Finite memory $\rightarrow$ Exponential memory

Then $\lambda'_1$ is winning and finite memory

1. Exponential memory is sufficient.
   - Use extensions of technics à la Rackoff (Petri nets) - refinements of [BJK10]

2. Exponential memory is needed

3. Leads to symbolic and incremental algorithms

\[ s_1 \rightarrow s_{1,L} \rightarrow \ldots \rightarrow s_{K} \rightarrow s_{K,L} \rightarrow t_{1,L} \rightarrow \ldots \rightarrow t_{K,L} \]

wqo+Koenig’s lemma
Finite memory → Exponential memory

① Exponential memory is sufficient.

- Use extensions of technics à la Rackoff (Petri nets) - refinements of [BJK10]

② Exponential memory is needed

③ Leads to symbolic and incremental algorithms
Exponential memory is sufficient.

- Use extensions of technics à la Rackoff (Petri nets) - refinements of [BJK10]

Exponential memory is needed

Leads to symbolic and incremental algorithms

Finite memory \(\rightarrow\) Exponential memory

Lemma 3. There exists a family of multi energy games

\[ G_i^k \]

\[ \text{for any initial credit, } P_1 \text{ needs exponential memory to win.} \]

The idea is the following: if \( P_1 \) do not remember the exact choices of \( P_2 \) which require an exponential size Moore machine, there will exist some sequence of choices of \( P_2 \) such that \( P_1 \) cannot counteract a decrease in energy. Thus, by playing this sequence long enough, \( P_2 \) can force \( P_1 \) to lose whatever his initial credit is.

Fig. 2. Family of games requiring exponential memory:

\[ \uparrow \]

\[ \downarrow \]

\[ \text{for } i, j, K \]

Corollary 1. Exponential memory is both sufficient and, in general, necessary for finite-memory winning on multi mean-payoff parity games. Synthesizing a winning strategy (if one exists) can be done in time exponential in the size of the game.

Proof. Thanks to Theorem 4, we have equivalence between finite-memory winning for multi energy and multi mean-payoff games. The rest follows from straightforward application of Theorem 4 and Lemma 3.

4 Randomness as a substitute for finite-memory

Throughout previous sections, we have been interested in pure finite-memory strategies for MEPGs and MMPPGs, as they are of importance for practical applications. As introduced in Section 4, another widely studied class of strategies is the class of randomized strategies. In this section, we answer a fundamental question regarding the nature of strategies: "When and how pure finite-memory can be traded for randomized memorylessness?" Specifically, we study on which kind of games \( P_1 \) can replace a pure finite-memory winning strategy by an equally powerful, yet conceptually simpler, randomized memoryless one and...
Finite memory $\rightarrow$ Exponential memory

1. Exponential memory is sufficient.
   - Use extensions of technics à la Rackoff (Petri nets) - refinements of [BJK10]

2. Exponential memory is needed

3. Leads to symbolic and incremental algorithms

Then $\lambda_1'$ is winning and finite memory.

Diagram:

wqo+Koenig’s lemma
Finite memory vs. exponential memory

Exponential memory is sufficient.

\[ \text{• Use extensions of technics à la Rackoff (Petri nets) - refinements of [BJK10]} \]

Leads to symbolic and incremental algorithms

Finite memory \rightarrow Exponential memory

Lemma 3.

There exists a family of multi energy games $G_i$. Let $S_1, S_2, s_{\text{init}}, E, K$ s.t.

- for any initial credit, $P_1$ needs exponential memory to win.

The idea is the following: if $P_1$ does not remember the exact choices of $P_2$, which require an exponential size Moore machine, there will exist some sequence of choices of $P_2$ such that $P_1$ cannot counteract a decrease in energy. Thus, by playing this sequence long enough, $P_2$ can force $P_1$ to lose, given his initial credit.

Fig. 2.

Family of games requiring exponential memory:

\[ i, g, j, k, n, s, t, \ldots, \]

Corollary 1.

Exponential memory is both sufficient and, in general, necessary for finite-memory winning on multi mean-payoff games. Synthesizing a winning strategy (if one exists) can be done in time exponential in the size of the game.

Proof.

Thanks to [sz] Theorem u, we have equivalence between finite-memory winning for multi energy and multi mean-payoff games. The rest follows from straightforward application of Theorem sz Theorem t and Lemma u.

Randomness as a substitute for finite-memory

Throughout previous section, we have been interested in pure finite-memory strategies for MEPGs and MMPPGs, as they are of importance for practical applications. As introduced in Section t, another widely studied class of strategies is the class of randomized strategies. In this section, we answer a fundamental question regarding the nature of strategies: "When and how pure finite-memory can be traded for randomized memorylessness?" Specifically, we study on which kind of games $P_1$ can replace a pure finite-memory winning strategy by an equally powerful, yet conceptually simpler, randomized memoryless one and.

Then $\lambda_1'$ is winning and finite memory...
Multi energy games are **coNP** complete, **exponential memory** strategies may be necessary for Player 1, **memoryless strategies** are sufficient for Player 2
Multi-dim. Mean-Payoff Games
MMPGs - Infinite Memory

To play optimally MMPGs, infinite memory is necessary

- (2, 2) for Lim Sup MP
- (1, 1) is achievable for Lim Inf MP
- None of the two is achievable with finite memory
MMPGs - Infinite Memory

\[ (2,0) \quad (0,0) \quad (0,2) \]

\[
\begin{align*}
q_a &\quad (0,0) &\quad q_b \\
\uparrow & & \uparrow \\
\text{time} & & \text{time}
\end{align*}
\]

Both \( \limsup \) = 2

Both \( \liminf \) = 1
Lemma. If for all states \( v \in V_1 \cup V_2 \), for all \( i, 1 \leq i \leq k \), Player 1 has a winning strategy for winning the mean-payoff sup. for dimension \( i \), then for all states \( v \in V_1 \cup V_2 \), Player I has a winning strategy from \( v \) for the conjunction of all \( k \) mean-payoff objectives.

Intuition: play each of the \( k \) winning strategies one after the other for longer and longer time intervals.
Consider the following algorithm:

1. Compute $W_i$ = set of states where Pl. I wins the 1 dim. game defined by dim. i
2. Let $W$ be the intersection of all $W_i$’s
3. Remove states that are not in $W$

Repeat until no states are removed

Let $Win$ be the states that survived this process
Consider the following algorithm:

1. Compute $W_i$: set of states where Pl. I wins the 1 dim. game defined by dim. $i$

2. Let $W$ be the intersection of all $W_i$’s

3. Remove states that are not in $W$

Repeat until no states are removed

Let $W_{\text{in}}$ be the states that survived this process

Intersection
Consider the following algorithm:

1. Compute $W_i =$ set of states where Pl. I wins the 1 dim. game defined by dim. $i$

2. Let $W$ be the intersection of all $W_i$’s

3. Remove states that are not in $W$

Repeat until no states are removed

Let $Win$ be the states that survived this process
Lemma. From all states in Win, Player 1 has a winning strategy for each dimension. From all states that are not in Win, Player II has a winning strategy for at least one dimension.

Theorem [RV11]. From all states in Win, Player 1 has a winning strategy for all the dimensions (for Lim Sup).

Corollary. Deciding MMPGs with Lim-sup is in NP∩coNP.
MMPGs

Theorem [CDHR10, VR11]

(i) Multi mean-payoff games \( \text{lim-sup} \) are in \( \text{NP} \cap \text{coNP} \),
    infinite memory strategies may be necessary for Player 1,
    memoryless strategies are sufficient for Player 2

(ii) Multi mean-payoff games \( \text{lim-inf} \) are \( \text{coNP complete} \),
    infinite memory strategies may be necessary for Player 1,
    memoryless strategies are sufficient for Player 2
## Results for Extensions

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MP</strong></td>
<td>Memoryless</td>
<td>Memoryless</td>
<td>NP∩coNP</td>
</tr>
<tr>
<td><strong>MMPG - Sup</strong></td>
<td>Infinite</td>
<td>Memoryless</td>
<td>NP∩coNP</td>
</tr>
<tr>
<td><strong>MMPG - Inf</strong></td>
<td>Infinite</td>
<td>Memoryless</td>
<td>coNP-C</td>
</tr>
<tr>
<td><strong>MMPG - Mix</strong></td>
<td>Infinite</td>
<td>Memoryless</td>
<td>coNP-C</td>
</tr>
<tr>
<td><strong>MEG</strong></td>
<td>Exponential</td>
<td>Memoryless</td>
<td>coNP-C</td>
</tr>
</tbody>
</table>
Additional negative results on MP/EG

- **Theorem**: MP and EG games with imperfect information are undecidable.

  Aldric Degorre, Laurent Doyen, Raffaella Gentilini, Jean-François Raskin, Szymon Torunczyk: *Energy and Mean-Payoff Games with Imperfect Information*. CSL 2010: 260-274

- **Theorem**: Games with objectives given as Boolean combinations of MP are undecidable.

  Yaron Velner: *Robust Multidimensional Mean-Payoff Games are Undecidable*. FoSSaCS 2015: 312-327
Variations on MP: Window Objectives
Window Objectives

- Space for new definitions as classical objectives have **drawbacks:**
  1) complexity of MP is open
  2) MMP is sensitive to lim inf. vs. lim sup.
  3) MP with imperfect information are undecidable
  4) Boolean combinations of MP constraints are undecidable

- **Window objectives:** look at the payoff through a *local finite window sliding* over the play

- conservative approximations of MP

- ensure good properties within a **bounded time horizon**
Window Objectives - Definitions

Idea: look at payoffs through a local finite window
⇒ mean should be above zero within window size
**Window Objectives - Definitions**

**Idea:** look at payoffs through a local finite window

$\Rightarrow$ mean should be above zero *within* window size
Idea: look at payoffs through a local finite window
⇒ mean should be above zero within window size
Idea: look at payoffs through a local finite window
\[\Rightarrow \text{mean should be above zero within window size}\]
**Window Objectives - Definitions**

**Idea:** look at payoffs through a local finite window

⇒ mean should be above zero within window size
Idea: look at payoffs through a local finite window
⇒ mean should be above zero within window size
Window Objectives - Definitions

Idea: look at payoffs through a local finite window $\Rightarrow$ mean should be above zero within window size
**Window Objectives - Definitions**

**Idea:** look at payoffs through a local finite window

$\implies$ mean should be above zero *within* window size

![Graph showing time vs. sum with a window of size n highlighted.](image-url)
Idea: look at payoffs through a local finite window

\[ \Rightarrow \text{mean should be above zero within window size} \]
Idea: look at payoffs through a local finite window \( \Rightarrow \) mean should be above zero within window size.
Idea: look at payoffs through a local finite window
\[ \implies \text{mean should be above zero within window size} \]

Window Objectives - Definitions

```
Idea: look at payoffs through a local finite window
\[ \implies \text{mean should be above zero within window size} \]
```
Example

- **Fixed window** is satisfied for size $\geq 2$

- **Direct fixed window** is not satisfied for any size!
Player 1 wins mean-payoff
... but he looses for every window objectives
Examples

For $n=3$: memory needed
For $n=4$: no memory needed
Relation with “classical” objectives
Relation between MP/TP and W.O.

If the answer to one of window mean-payoff problems is YES, then the answer to the mean-payoff threshold problem for threshold zero is also YES.
Relation between MP/TP and W.O.

If the answer to one of window mean-payoff problems is YES, then the answer to the mean-payoff threshold problem for threshold zero is also YES.
Relation between MP/TP and W.O.

If the answer to one of window mean-payoff problems is YES, then the answer to the mean-payoff threshold problem for threshold zero is also YES.
Relation between MP/TP and W.O.

If the answer to one of window mean-payoff problems is YES, then the answer to the mean-payoff threshold problem for threshold zero is also YES.
Relation between MP/TP and W.O.

If the answer to one of window mean-payoff problems is YES, then the answer to the mean-payoff threshold problem for threshold zero is also YES.
Relation between MP/TP and W.O.

If the answer to one of window mean-payoff problems is YES, then the answer to the mean-payoff threshold problem for threshold zero is also YES.
Relation between MP/TP and W.O.

If there exists $\varepsilon > 0$ s.t. the answer to the mean-payoff threshold problem for threshold $\varepsilon$ is YES, then the answer to the BW problem is also YES.
Relation between MP/TP and W.O.

If there exists $\varepsilon > 0$ s.t. the answer to the mean-payoff threshold problem for threshold $\varepsilon$ is YES, then the answer to the BW problem is also YES.

Pl. I can force $\varepsilon$-positive cycles

$\Rightarrow$

Pl. I can win the $\text{DFW}(n)$ objective for $n$ large enough

i.e. $n = (|S| - 1) \cdot (1 + |S| \cdot W)$

Proof uses cycle decomposition of outcomes
Theorem

1. If the answer to the one of window mean-payoff problems is YES, then the answer to the mean-payoff threshold problem for threshold zero is also YES.

2. If there exists $\varepsilon > 0$ s.t. the answer to the mean-payoff threshold problem for threshold $\varepsilon$ is YES, then the answer to the BW problem is also YES.

⇒ Window objectives can be seen as $\varepsilon$-approximations of mean-payoff for large enough windows
Main Results

<table>
<thead>
<tr>
<th>one-dimension</th>
<th>k-dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>complexity</td>
<td>$\mathcal{P}_1$ mem.</td>
</tr>
<tr>
<td>MP / MP</td>
<td>NP $\cap$ coNP</td>
</tr>
<tr>
<td>TP / $\overline{TP}$</td>
<td>NP $\cap$ coNP</td>
</tr>
<tr>
<td>WMP: fixed polynomial window</td>
<td>P-c. (Thm. 2)</td>
</tr>
<tr>
<td>WMP: fixed arbitrary window</td>
<td>P($</td>
</tr>
<tr>
<td>WMP: bounded window problem</td>
<td>NP $\cap$ coNP (Thm. 3)</td>
</tr>
</tbody>
</table>

+ imperfect information OK
Paul Hunter, Guillermo A. Pérez, Jean-François Raskin:
Looking at Mean-Payoff Through Foggy Windows.
ATVA 2015: 429-445

+ Boolean combinaisons OK
Véronique Bruyère, Quentin Hautem, Jean-François Raskin:
On the Complexity of Heterogeneous Multidimensional Games.
CONCUR 2016: 11:1-11:15
The Complexity of Multi-Mean-Payoff and Multi-Energy Games

Yaron Velner, Krishnendu Chatterjee, Laurent Doyen, Thomas A. Henzinger, Alexander Rabinovich, and Jean-François Raskin

1 The Blavatnik School of Computer Science, Tel Aviv University, Israel
2 IST Austria (Institute of Science and Technology Austria)
3 LSV, ENS Cachan & CNRS, France
4 Département d’Informatique, Université Libre de Bruxelles (U.L.B.)

Abstract. In mean-payoff games, the objective of the protagonist is to ensure that the limit average of an infinite sequence of numeric weights is nonnegative. In energy games, the objective is to ensure that the running sum of weights is always nonnegative. Multi-mean-payoff and multi-energy games replace individual weights by tuples, and the limit average (resp. running sum) of each coordinate must be (resp. remain) nonnegative. These games have applications in the synthesis of resource-bounded processes with multiple resources.

We prove the finite-memory determinacy of multi-energy games and show the inter-reducibility of multi-mean-payoff and multi-energy games for finite-memory strategies. We also improve the computational complexity for solving both classes of games with finite-memory strategies: while the previously best known upper bound was EXPSPACE, and no lower bound was known, we give an optimal coNP-complete bound. For memoryless strategies, we show that the problem of deciding the existence of a winning strategy for the protagonist is NP-complete. Finally, we present the first solution of multi-mean-payoff games with infinite-memory strategies. We show that multi-mean-payoff games with mean-payoff-sup objectives can be decided in NP ∩ coNP, whereas multi-mean-payoff games with mean-payoff-inf objectives are coNP-complete.

Keywords: Games on graphs; mean-payoff objectives; energy objectives; multi-dimensional objectives.

1 Introduction

Graph games and multi-objectives. Two-player games on graphs are central in many applications of computer science. For example, in the synthesis problem, implementations of reactive systems are obtained from winning strategies in games with a qualitative specification [22, 21, 1]. In these applications, the objective determines which player wins: natural models of computation are often designed to satisfy a winning objective.
Looking at Mean-Payoff and Total-Payoff through Windows

Krishnendu Chatterjee¹,*, Laurent Doyen², Mickael Randour³,†, and Jean-François Raskin⁴,‡

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

Abstract. We consider two-player games played on weighted directed graphs with mean-payoff and total-payoff objectives, which are two classical quantitative objectives. While for single dimensional objectives all results for mean-payoff and total-payoff coincide, we show that in contrast to multi-dimensional mean-payoff games that are known to be coNP-complete, multi-dimensional total-payoff games are undecidable. We introduce conservative approximations of these objectives, where the payoff is considered over a local finite window sliding along a play, instead of the whole play. For single dimension, we show that (i) if the window size is polynomial, then the problem can be solved in polynomial time, and (ii) the existence of a bounded window can be decided in NP ∩ coNP, and is at least as hard as solving mean-payoff games. For multiple dimensions, we show that (i) the problem with fixed window size is EXPTIME-complete, and (ii) there is no primitive-recursive algorithm to decide the existence of a bounded window.

1 Introduction

Mean-payoff and total-payoff games. Two-player mean-payoff and total-payoff games are played on finite weighted directed graphs (in which every edge has an integer weight) with two types of vertices: in player-1 vertices, player 1 chooses the successor vertex from the set of outgoing edges; in player-2 vertices, player 2 does likewise. The game results in an infinite path (called a play) through the graph. The mean-payoff (resp. total-payoff) value of a play is the long-run average (resp. sum) of the edge-weights along the

see on arXiv: http://arxiv.org/abs/1302.4248
and in

Faster algorithms for mean-payoff games

Formal Methods in System Design
April 2011, Volume 38, Issue 2, pp 97–118

L. Brim, J. Chaloupka, L. Doyen, R. Gentilini, J. F. Raskin

Article
First Online: 08 December 2010
DOI (Digital Object Identifier): 10.1007/s10703-010-0105-x

Cite this article as:

Conclusion

- **Quantitative games**: energy games (+generalizations by Larsen, Bouyer,...), mean-payoff games, ... is an active research area, part of a larger program...

- **From quality to quantity**: broad effort in order to lift boolean verification/synthesis to quantitative verification/synthesis, e.g.:
  - quantitative languages (Henzinger et al.) def. by **weighted automata**

- In this talk, we have shown:
  - MP games can be extended to multi-dim.
  - Space for alternative objectives such as **window mean-payoff objectives**

- but also (not in this talk), alternative solution concepts:
  - regret minimization versus winning strategy
  - combine worst-case and expectation (games+MDP)
  - non-zero sum extensions (Nash equilibria, admissibility, etc.)