The Multiple Dimensions of Mean-Payoff Games

Laurent Doyen
CNRS & LSV, ENS Paris-Saclay

RP 2017
Basics about mean-payoff games

- Algorithms & Complexity
- Strategy complexity – Memory

Focus

- Equivalent game forms
- Techniques for memoryless proofs
Mean-Payoff
Mean-Payoff
Mean-Payoff

Diagram:

\[ R_a \]

\[ R_b \]

\[ V \]

\[ q_a \]

\[ q_b \]

\[ (2,0) \]

\[ (0,2) \]
Mean-Payoff

\[ R_a \]

\[ R_b \]

\[ V \]

\( q_a \)

\( q_b \)

\((2,0)\)

\((0,0)\)

\((0,2)\)

\( q_a \)

\( q_0 \)

\( q_b \)
Mean-Payoff

Switching policy to get average power (1,1)?
Mean-Payoff

Switching policy to get average power \((1,1)\) ?

\[
\frac{n \cdot (2,0) + n \cdot (0,2) + 2 \cdot (0,0)}{2n + 2} = \frac{(2n, 2n)}{2n + 2} \rightarrow (1, 1)
\]
Mean-payoff value = limit-average of the visited weights

\[
\operatorname{MP}(q_0q_1 \ldots) = \lim_{n \to \infty} \frac{1}{n} \sum_{i=0}^{n-1} \text{weight}_k(q_i)
\]

\[1 \leq k \leq d\]
Mean-Payoff

Switching policy

- Infinite memory: \((1,1)\) vanishing frequency in \(q_0\)

\[
\text{weight : } Q \rightarrow \mathbb{Z}^d
\]

\[
\text{MP}(q_0q_1 \ldots) = \lim_{n \to \infty} \frac{1}{n} \cdot \sum_{i=0}^{n-1} \text{weight}_k(q_i)
\]

\(1 \leq k \leq d\)
Mean-Payoff

\[ MP = \lim_{{n \to \infty}} \frac{1}{n} \cdot \sum_{{i=0}}^{n-1} \text{weight}_k(q_i) \]
Mean-Payoff

$$\text{MP} = \lim_{n \to \infty} \frac{1}{n} \cdot \sum_{i=0}^{n-1} \text{weight}_k(q_i)$$

limit?

$$\overline{\text{MP}} = \limsup_{n \to \infty} \frac{1}{n} \cdot \sum_{i=0}^{n-1} \text{weight}_k(q_i)$$

$$\underline{\text{MP}} = \liminf_{n \to \infty} \frac{1}{n} \cdot \sum_{i=0}^{n-1} \text{weight}_k(q_i)$$

Mean-payoff is prefix-independent
Mean-Payoff

Switching policy

- Infinite memory: $(1,1)$ for liminf
Mean-Payoff

Switching policy

- Infinite memory: \((1,1)\) for \(\text{liminf}\) & \((2,2)\) for \(\text{limsup}\)
Games
Two-player games

\[ G = (Q, E) \]

\[ Q = Q_\circ \cup Q_\nabla \]

\[ E \subseteq Q \times Q \]
Two-player games

- Turn-based
- Infinite duration

\[ G = (Q, E) \]

\[ \begin{cases} Q = Q_\odot \cup Q_\square \\ E \subseteq Q \times Q \end{cases} \]
Two-player games

- Turn-based
- Infinite duration

Play: $a, d, a, b, e, f, g, d, a, c, \ldots$
Two-player games

Player 1 (maximizer)  Player 2 (minimizer)

- Turn-based Play
- Infinite duration

Play:  a, d, a, b, e, f, g, d, a, c, ...
Two-player games

- Turn-based
- Infinite duration

Play: a, d, a, b, e, f, g, d, a, c, ...
Two-player games

- Turn-based
- Infinite duration

Play: $a, d, a, b, e, f, g, d, a, c, \ldots$
Two-player games

- Turn-based
- Infinite duration

Play:  a, d, a, b, e, f, g, d, a, c, …
Two-player games

- Turn-based
- Infinite duration

Play: $a, d, a, b, e, f, g, d, a, c, \ldots$
Two-player games

- Turn-based
- Infinite duration

Play: $a, d, a, b, e, f, g, d, a, c, \ldots$
Two-player games

- Turn-based
- Player 1 (maximizer)
- Player 2 (minimizer)
- Infinite duration

Play: a, d, a, b, e, f, g, d, a, c, ...
Two-player games

- Turn-based
- Infinite duration

Play: \(a, d, a, b, e, f, g, d, a, c, \ldots\)
Two-player games

• Turn-based

Play: $a, d, a, b, e, f, g, d, a, c, \ldots$
Two-player games

- Turn-based
- Infinite duration

Play: $a, d, a, b, e, f, g, d, a, c, \ldots$
Two-player games

- Turn-based
- Infinite duration

Strategies = recipe to extend the play prefix

Player 1: $\sigma : Q^* \cdot Q_\circ \rightarrow Q$

Player 2: $\pi : Q^* \cdot Q_\square \rightarrow Q$
Two-player games

• Turn-based

Player 1 (maximizer)

Player 2 (minimizer)

• Turn-based
• Infinite duration

Strategies = recipe to extend the play prefix

Player 1: \( \sigma : Q^* \cdot Q_\circ \rightarrow Q \)

Player 2: \( \pi : Q^* \cdot Q_\square \rightarrow Q \)

outcome of two strategies is a play
Mean-payoff games
Mean-payoff games

Mean-payoff game:
positive and negative weights
(encoded in binary)
\( w : E \rightarrow \mathbb{Z} \)

Decision problem:

Decide if there exists a player-1 strategy to ensure mean-payoff value \( \geq 0 \)

\( \exists \sigma : \forall \pi : \text{MP}(\text{outcome}_{q, \sigma, \pi}) \geq 0 \)

Value problem:

\[ \sup_{\sigma} \inf_{\pi} \text{MP}(\text{outcome}_{q, \sigma, \pi}) \]
Mean-payoff games

Key ingredients:
- identify memory requirement: infinite vs. finite vs. memoryless
- solve 1-player games (i.e., graphs)

Key arguments for memoryless proof:
- backward induction
- shuffle of plays
- nested memoryless objectives
Reduction to Reachability Games
Reduction to Reachability Games

Reachability objective: positive cycles ($v \geq 0$)
Reduction to Reachability Games

Reachability objective: positive cycles ($v \geq 0$)
Reduction to Reachability Games

Reachability objective: positive cycles ($v \geq 0$)
Reduction to Reachability Games

Reachability objective: positive cycles ($v \geq 0$)

If player 1 wins → only positive cycles are formed → mean-payoff value $\geq 0$

If player 2 wins → only negative cycles are formed → mean-payoff value $< 0$

(Note: limsup vs. liminf does not matter)
Reduction to Reachability Games

Reachability objective: positive cycles \((v \geq 0)\)

Mean-payoff game \(\Leftrightarrow\) Ensuring positive cycles

Memoryless strategy transfers to finite-memory mean-payoff winning strategy
Strategy Synthesis

Memoryless mean-payoff winning strategy?
Memoryless mean-payoff winning strategy?
Strategy Synthesis

Memoryless mean-payoff winning strategy?

Progress measure: minimum initial credit to stay always positive
Memoryless mean-payoff winning strategy?

Progress measure: minimum initial credit to stay always positive
Strategy Synthesis

Memoryless mean-payoff winning strategy?

Progress measure: minimum initial credit to stay always positive \( \geq 0 \)
Memoryless mean-payoff winning strategy?

Progress measure: minimum initial credit to stay always positive
Memoryless mean-payoff winning strategy?

Choose successor to stay above minimum credit

\[ \mu : Q \rightarrow \mathbb{N} \] minimum credit such that

\[ \exists \sigma \cdot \forall \pi \cdot \forall n : \mu(q_0) + \sum_{i=0}^{n} w(q_i, q_{i+1}) \geq 0 \]

Progress measure: minimum initial credit to stay always positive

In \( q \in Q_0 \) choose \( q' \) such that \( \mu(q) + w(q, q') \geq \mu(q') \)
Memoryless means winning strategy

"Energy Game"
(stay always positive, for some initial credit)

Progress measure: minimum initial credit to stay always positive

In $q \in Q_0$ choose $q'$ such that $\mu(q) + w(q, q') \geq \mu(q')$
Memoryless proofs

Key arguments for memoryless proof:

• backward induction
• shuffle of plays
• nested memoryless objectives
Energy Games

\[ \mu = \min_{n \in \mathbb{N}} \sum_{i=0}^{n-1} w_i \]

Energy: min-value of the prefix.
(if positive cycle; otherwise \(\infty\))

Mean-payoff: average-value of the cycle.

\[ \text{MP} = \lim_{n \to \infty} \frac{1}{n} \sum_{i=0}^{n-1} w_i \]

\[ \geq 0 \]

NP \( \cap \) coNP

\( q_1 \)
\( q_2 \)
\( q_3 \)
\( q_4 \)
Winning strategy?
Winning strategy?

Follow the minimum initial credit!
Multi-dimension games
Multi-dimension games

• Energy: initial credit to stay always above (0,0)
• Mean-payoff: $\text{MP}(w_1) \geq 0 \land \text{MP}(w_2) \geq 0$

Multiple resources $w : Q \rightarrow \mathbb{Z}^d$
Multi-dimension games

- Energy: initial credit to stay always above (0,0)
- Mean-payoff: $\text{MP}(w_1) \geq 0 \land \text{MP}(w_2) \geq 0$

Multiple resources $w : Q \to \mathbb{Z}^d$

- same ?
- same as positive cycles?
Multi-dimension games

- Energy: initial credit to stay always above (0,0)
- Mean-payoff: $\text{MP}(w_1) \geq 0 \land \text{MP}(w_2) \geq 0$

If player 1 can ensure positive simple cycles, then energy and mean-payoff are satisfied.

**Not the converse!**
Multi-dimension games

If player 1 has initial credit to stay always positive (Energy) then finite-memory strategies are sufficient
Multi-dimension games

If player 1 has initial credit to stay always positive (Energy) then finite-memory strategies are sufficient

Let $\sigma_1$ be winning

On each branch

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

$(\mathbb{N}^d, \leq)$ is well-quasi ordered

$wqo +$ Koenig’s lemma
Multi-energy games

If player 1 has initial credit to stay always positive (Energy) then finite-memory strategies are sufficient.

For player 2?
For player 2, **memoryless** strategies are sufficient

- induction on player-2 states
- if \( \exists \) initial credit against all memoryless strategies, then \( \exists \) initial credit against all arbitrary strategies.
Multi-energy games

For player 2, memoryless strategies are sufficient

- induction on player-2 states
- if $\exists$ initial credit against all memoryless strategies, then $\exists$ initial credit against all arbitrary strategies.

\begin{align*}
\text{Play is a shuffle of left-game play and right-game play} \\
\text{Energy is sum of them}
\end{align*}
Multi-energy games

For player 2, memoryless strategies are sufficient

- induction on player-2 states
- if $\exists$ initial credit against all memoryless strategies, then $\exists$ initial credit against all arbitrary strategies.

In general, we need

$$\min(\text{val}(\rho_L), \text{val}(\rho_R)) \leq \text{val}(\text{shuffle}(\rho_L, \rho_R))$$
Memoryless proofs

Key arguments for memoryless proof:

• backward induction
• shuffle of plays
• nested memoryless objectives
Multi-energy games

If player 1 has initial credit to stay always positive (Energy) then finite-memory strategies are sufficient.

For player 2, memoryless strategies are sufficient. coNP?
Multi-energy games

If player 1 has initial credit to stay always positive (Energy) then finite-memory strategies are sufficient.

For player 2, memoryless strategies are sufficient.

coNP?

- guess a memoryless strategy \( \pi \) for Player 2
- Construct \( G_\pi \)
- check in polynomial time that \( G_\pi \) contains no cycle with nonnegative effect in all dimensions
Multi-weighted energy games

Detection of nonnegative cycles $\Rightarrow$ polynomial-time

- Flow constraints using LP
- Divide and conquer algorithm
Multi-weighted energy games

Detection of nonnegative cycles $\Rightarrow$ polynomial-time

- Flow constraints using LP

\[ x_1 + x_2 = x_3 \]
\[ \sum_i x_i \cdot w_i \geq 0 \]
Multi-weighted energy games

Detection of nonnegative cycles ⇒ polynomial-time

- Flow constraints using LP

\[ x_1 + x_2 = x_3 \]
\[ \sum_i x_i \cdot w_i = 0 \]

Not connected!
Multi-weighted energy games

Detection of nonnegative cycles $\Rightarrow$ polynomial-time

- Flow constraints using LP
- Divide and conquer algorithm

Mark the edges that belong to some (pseudo) solution.

Solve the connected subgraphs.
Multi-weighted energy games

Detection of nonnegative cycles ⇒ polynomial-time

- Flow constraints using LP
- Divide and conquer algorithm

Mark the edges that belong to some (pseudo) solution.

Solve the connected subgraphs.
Multi-dimension games

If player 1 has initial credit to stay always positive (Energy) then **finite-memory** strategies are sufficient

For player 2, **memoryless** strategies are sufficient

Equivalent with mean-payoff games (under finite-memory):

If player 1 wins → positive cycles are formed → mean-payoff value ≥ 0

Otherwise, for all finite-memory strategy of player 1 (with memory M), player 2 can repeat a negative cycle (in G x M)
## Multi-dimension games

<table>
<thead>
<tr>
<th>Player 1</th>
<th>Energy</th>
<th>MP - liminf</th>
<th>MP - limsup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finite memory</td>
<td></td>
<td>coNP-complete</td>
<td>Player 2 memoryless</td>
</tr>
<tr>
<td>Infinite memory</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- liminf MP - limsup
Multi-dimension games

<table>
<thead>
<tr>
<th>Player 1</th>
<th>Energy</th>
<th>MP - liminf</th>
<th>MP - limsup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finite memory</td>
<td>coNP-complete Player 2 memoryless</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Infinite memory</td>
<td>coNP-complete Player 2 memoryless</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Player 2 memoryless (shuffle argument)
- Graph problem in PTIME (LP argument)

$$\min(\text{MP}(\rho_L), \text{MP}(\rho_R)) \leq \text{MP}(\text{shuffle}(\rho_L, \rho_R))$$

\{ True for \text{MP} \quad False for \overline{\text{MP}} \}
Multi-mean-payoff games

The winning region $R$ of player 1 has the following characterization:

Player 1 wins $\bigwedge_i \overline{MP}(w_i) \geq 0$ from every state in $R$

if and only if player 1 wins each $\overline{MP}(w_i) \geq 0$ from every state in $R$

Proof idea: $\Box\Diamond(1 \land 2) \equiv \Box\Diamond 1 \land \Box\Diamond 2$ (without leaving $R$)
Multi-mean-payoff games

The winning region $R$ of player 1 has the following characterization:

Player 1 wins $\bigwedge_i \overline{MP}(w_i) \geq 0$ from every state in $R$

if and only if player 1 wins each $\overline{MP}(w_i) \geq 0$ from every state in $R$

Proof idea: $\Box \diamond (1 \land 2) \equiv \Box \diamond 1 \land \Box \diamond 2$

Losing for player 1 for single objective

Winning for player 2, with memoryless strategy

By induction, player 2 is memoryless in the subgame
Memoryless proofs

Key arguments for memoryless proof:
- backward induction
- shuffle of plays
- nested memoryless objectives
### Multi-dimension games

<table>
<thead>
<tr>
<th>Player 1</th>
<th>Energy</th>
<th>MP - liminf</th>
<th>MP - limsup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finite memory</td>
<td></td>
<td>coNP-complete</td>
<td>Player 2 memoryless</td>
</tr>
<tr>
<td>Infinite memory</td>
<td></td>
<td>coNP-complete</td>
<td>NP ∩ coNP Pl. 2 memoryless</td>
</tr>
</tbody>
</table>

**Finite memory**: Player 1 has limited memory, while Player 2 is memoryless.

**Infinite memory**: Both Player 1 and Player 2 have unlimited memory, with additional complexity in the game states.
Window games

Issues with mean-payoff

- limsup vs. liminf
- limit-behaviour, unbounded delay
- complexity
Window games

Issues with mean-payoff
- limsup vs. liminf
- limit-behaviour, unbounded delay
- complexity

Sliding window of size at most B
At every step, MP \geq 0 within the window
Window games

Window objective:
from some point on, at every step, MP ≥ 0 within window of B steps

prefix-independent

bounded delay

Implies the mean-payoff condition
Window games

Window objective:
from some point on, at every step, MP ≥ 0 within window of B steps

prefix-independent  bounded delay

Implies the mean-payoff condition

Complexity, Algorithm?
- like coBüchi objective $\diamond \square (\sum_{\leq B} \geq 0)$

$O(V^2 \cdot E \cdot B \cdot \log W)$

min-max cost (for $\leq B$ steps)

stable set (safety)

attractor & subgame iteration
Window objective:
from some point on, at every step, $MP \geq 0$ within window of $B$ steps

prefix-independent  bounded delay

Implies the mean-payoff condition

Complexity, Algorithm?

• like coBüchi objective $\diamond \Box (\Sigma^{\leq B} \geq 0)$ $O(V^2 \cdot E \cdot B \cdot \log W)$

• multi-dimension: EXPTIME-complete
Hyperplane Separation

Multi-dimension mean-payoff (liminf): coNP-complete

Naive algorithm: exponential in number of states

Hyperplane separation: reduction to single-dimension mean-payoff games

\[ \lambda = (1, 3) \]
\[ w = (-1, 2) \]

\[ \lambda \cdot w^T = 1 \cdot (-1) + 3 \cdot 2 = 5 \]
Hyperplane Separation

Multi-dimension

\[ \vec{\lambda} = (1, 3) \]

Single dimension

\[ \vec{\lambda} \cdot w^T \]
Hyperplane Separation

\[(1, -3) \quad (-3, 1)\]

\[(-1, 3) \quad (3, -1)\]
Hyperplane Separation

Player 1 cannot ensure $\text{MP}_\lambda \geq 0$ for some $\lambda$

$\iff$

Player 1 loses the multi-dimension game
Hyperplane Separation

Player 1 wins the multi-dimension game

\[ \lambda \geq 0 \quad \forall \lambda \in (\mathbb{R}^+)^d \]

\[ \Leftrightarrow \]

Player 1 wins the multi-dimension game
Hyperplane Separation

$(1, -3)$  $(−3, 1)$

$(−1, 3)$  $(3, −1)$

Player 1 wins MP if $\lambda \geq 0$ for all $\lambda \in \mathbb{R}^+$

$\iff$

Player 1 wins the multi-dimension game
Hyperplane Separation

Player 1 wins the multi-dimension game

Player 1 wins $MP_\lambda \geq 0$ for all $\lambda \in (\mathbb{R}^+)^d$

$\iff$

Player 1 wins the multi-dimension game
Hyperplane Separation

Player 1 wins the multi-dimension game

Player 1 wins MP\(\lambda\) \(\geq 0\) for all \(\lambda \in (\mathbb{R}^+)^d\)

\(\Leftrightarrow\)

Player 1 wins the multi-dimension game
Hyperplane Separation

$(1, -3)$ \quad $(-3, 1)$

$(1, -3)$ \quad $(-3, 1)$

Player 1 wins MP, $\lambda \geq 0$ for all $\lambda \in \mathbb{R}^+$

$\Leftrightarrow$

Player 1 wins the multi-dimension game

Player 1 wins MP, $\lambda \geq 0$ for all $\lambda \in (\mathbb{R}^+)^d$
Hyperplane Separation

- Multi-dimension mean-payoff (liminf): coNP-complete
- Naive algorithm: exponential in number of states
- Hyperplane separation: reduction to single-dimension mean-payoff games

Player 1 wins MP$_\lambda$ $\geq$ 0 for all $\lambda \in \mathbb{R}^d$

$\iff$

Player 1 wins the multi-dimension game

In fact, it is sufficient for player 1 to win for all $\lambda \in \{0,..,(d \cdot n \cdot W)^{d+1}\}^d$

Fixpoint algorithm:
- remove states if losing for some $\lambda$
- remove attractor (for player 2) of losing states

Solving $O(n \cdot M^d)$ mean-payoff games in $O(n \cdot m \cdot M)$

$O(n^2 \cdot m \cdot M^{d+1})$
Conclusion

Multiple dimensions of mean-payoff games
  • Reachability game
  • Energy game
  • Cycle-forming game

Multi-dimension mean-payoff games

Memoryless proofs

Other directions: parity condition, stochasticity, imperfect information
• Energy/Mean-Payoff Games is joint work with Lubos Brim, Jakub Chaloupka, Raffaela Gentilini, Jean-Francois Raskin.

• Multi-dimension Games is joint work with Krishnendu Chatterjee, Jean-Francois Raskin, Alexander Rabinovich, Yaron Velner.

• Window games is joint work with Krishnendu Chatterjee, Michael Randour, Jean-Francois Raskin.

Other important contributions:


Thank you!

Questions?