Energy games

Nicolas Markey (CNRS)
IRISA, team SUMO

Based on joint works with
Patricia Bouyer  Uli Fahrenberg  Piotr Hofman
Kim G. Larsen  Simon Laursen  Mickael Randour
    Jiri Srba  Martin Zimmermann

June 29, 2017
Model checking and synthesis

system:

property

always $3 \leq h \leq 12$

model-checking algorithm

yes/no
Model checking and synthesis

system:

property

always $3 \leq h \leq 12$

synthesis algorithm
(Average-)energy objectives: an example

Pressure-tank case study \[\text{CJL}^{+09}\]

(Average-)Energy objectives: an example

Pressure-tank case study [CJL⁺09]

Objectives:
- keep water level within given bounds
- minimize average level

(Average-)energy objectives: an example

Pressure-tank case study [CJL+09]

Objectives:
- keep water level within given bounds
- minimize average level

(Average-)energy objectives: an example

Pressure-tank case study [CJL⁺09]

Objectives:
- keep water level within given bounds
- minimize average level

(Average-)energy objectives: an example

Pressure-tank case study [CJL⁺09]

Objectives:

- keep water level within given bounds
- minimize average level

Games on weighted graphs

Example

- States: \( S = S^0 \uplus S^1 \)
- Weighted transitions: \( T \subseteq S \times \mathbb{Z} \times S \)
- Run: sequence of consecutive transitions:
- Strategy: transition to take depending on state/history:
  - \( \sigma \): always go to (from)
  - \( \sigma' \): alternate between (from)
Games on weighted graphs

Example

- states: \( S = S_\bullet \cup S_\square \)
- weighted transitions: \( T \subseteq S \times \mathbb{Z} \times S \)
Games on weighted graphs

Example

- states: $S = S_\bullet \cup S_\square$
- weighted transitions: $T \subseteq S \times \mathbb{Z} \times S$
- run: sequence of consecutive transitions:

```
-2 1 -4 5 2
```
Games on weighted graphs

Example

- **states**: \( S = S_\bullet \cup S_\square \)
- **weighted transitions**: \( T \subseteq S \times \mathbb{Z} \times S \)
- **run**: sequence of consecutive transitions:
- **strategy**: transition to take depending on state/history:
Games on weighted graphs

Example

- **states**: $S = S_0 \cup S_1$
- **weighted transitions**: $T \subseteq S \times \mathbb{Z} \times S$
- **run**: sequence of consecutive transitions:
  
  ![Diagram](image)

- **strategy**: transition to take depending on state/history:
  
  $\sigma_0$: always go to $\square$ (from $\bigcirc$)
Games on weighted graphs

Example

- states: $S = S_\bullet \uplus S_\square$
- weighted transitions: $T \subseteq S \times \mathbb{Z} \times S$
- run: sequence of consecutive transitions:

- strategy: transition to take depending on state/history:
  - $\sigma_\bullet$: always go to $\square$ (from $\bullet$)
  - $\sigma'_\bullet$: alternate between $\bullet$ and $\square$ (from $\bullet$)
Quantitative objectives

Decision problems

- **shortest path**: is it possible to go from source to target with accumulated weight less than a given threshold?


Quantitative objectives

Decision problems

- **shortest path:**
  is it possible to go from source to target with accumulated weight less than a given threshold?

  \[ \sim \text{ in polynomial time for 1-player games} \]
  (e.g. Bellman-Ford algorithm)

  \[ \sim \text{ in pseudo-polynomial time for 2-player games} \quad \text{[BGHM17]} \]
  \[ \text{in } \mathsf{NP} \cap \mathsf{coNP}, \text{ PTIME-hard} \]
  \[ \text{PTIME-complete with nonnegative weights} \quad \text{[KBB}^{+}08\text{]} \]


Quantitative objectives

Decision problems

- **mean-payoff objective**: from a given source, is it possible to make the average weight below a given threshold (in the long run)?
Quantitative objectives

Decision problems

- **mean-payoff objective**: from a given source, is it possible to make the average weight below a given threshold (in the long run)?

  - in polynomial time for 1-player games (e.g. Karp algorithm [Kar78])
  - in pseudo-polynomial time for 2-player games in $\mathbf{NP} \cap \mathbf{coNP}$, PTIME-hard [ZP96]


Energy objectives

Energy level

- energy level: $EL(\pi_{\leq n}) = \sum_{i \leq n} w(s_i \rightarrow s_{i+1})$ [aka. total payoff]
Energy objectives

Energy level

- energy level: \( EL(\pi_{\leq n}) = \sum_{i \leq n} w(s_i \rightarrow s_{i+1}) \) [aka. total payoff]

Quantitative objectives

- **shortest path**: minimize energy level when reaching a target
- **mean-payoff**: minimize ratio (energy level/path length) in the long run
Energy objectives

Energy level

- energy level: $EL(\pi_{\leq n}) = \sum_{i \leq n} w(s_i \rightarrow s_{i+1})$ [aka. total payoff]

Quantitative objectives

- **shortest path**: minimize energy level when reaching a target
- **mean-payoff**: minimize ratio (energy level/path length) in the long run

- **energy objectives**: maintain energy level within some bounds
  - above lower bound $L$
  - between bounds $L$ and $U$
Solving lower-bounded energy games

<table>
<thead>
<tr>
<th>1-player case [BFLMS08]</th>
</tr>
</thead>
<tbody>
<tr>
<td>- aim: maintain energy level above $L$ ( \leadsto ) maximize energy level</td>
</tr>
<tr>
<td>- Bellman-Ford-like algorithm to compute maximal remaining energy after $k$ steps</td>
</tr>
</tbody>
</table>

Solving lower-bounded energy games

1-player case [BFLMS08]

- aim: maintain energy level above $L \sim$ maximize energy level
- Bellman-Ford-like algorithm to compute maximal remaining energy after $k$ steps

$C_0 = 5$

Solving lower-bounded energy games

1-player case [BFLMS08]

- aim: maintain energy level above $L \leadsto$ maximize energy level
- Bellman-Ford-like algorithm to compute maximal remaining energy after $k$ steps

$\begin{align*}
c_0 &= 5 \\
-2 &\rightarrow -3 \\
4 &\rightarrow 3 \\
9 &\rightarrow 3
\end{align*}$

---

Solving lower-bounded energy games

1-player case [BFLMS08]

- aim: maintain energy level above $L \sim$ maximize energy level
- Bellman-Ford-like algorithm to compute maximal remaining energy after $k$ steps

$C_0 = 5$

\[
\begin{array}{c|cccc}
0 & 5 & -\infty & -\infty & -\infty \\
1 & 5 & 3 & -\infty & -\infty \\
\end{array}
\]
Solving lower-bounded energy games

1-player case [BFLMS08]

- aim: maintain energy level above $L \sim$ maximize energy level
- Bellman-Ford-like algorithm to compute maximal remaining energy after $k$ steps

![Diagram](image)

ifferential Equation

\[c_0 = 5 \]

\[
\begin{array}{l}
0 & 5 & -\infty & -\infty & -\infty \\
1 & 5 & 3 & -\infty & -\infty \\
2 & 5 & 3 & 7 & 1 \\
\end{array}
\]

Solving lower-bounded energy games

1-player case [BFLMS08]

- aim: maintain energy level above \( L \sim \) maximize energy level
- Bellman-Ford-like algorithm to compute maximal remaining energy after \( k \) steps

![Diagram with states and transitions]

<table>
<thead>
<tr>
<th>State</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5</td>
<td>-( \infty )</td>
<td>-( \infty )</td>
<td>-( \infty )</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>3</td>
<td>-( \infty )</td>
<td>-( \infty )</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>3</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>4</td>
<td>10</td>
<td>1</td>
</tr>
</tbody>
</table>

Solving lower-bounded energy games

1-player case [BFLMS08]

- **aim**: maintain energy level above $L \sim$ maximize energy level
- Bellman-Ford-like algorithm to compute maximal remaining energy after $k$ steps

![Graph and table]

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5</td>
<td>$-\infty$</td>
<td>$-\infty$</td>
<td>$-\infty$</td>
<td>$-\infty$</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>3</td>
<td>$-\infty$</td>
<td>$-\infty$</td>
<td>$-\infty$</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>3</td>
<td>7</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>4</td>
<td>10</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>4</td>
<td>10</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

Solving lower-bounded energy games

2-player case

- aim: maximize (resp. minimize) energy level
- both players have memoryless optimal strategies
- deciding the winner is in $\mathsf{NP} \cap \mathsf{coNP}$
- mean-payoff games are logspace-reducible to $L$-energy games
Solving interval-bounded energy games

Reduction to safety condition in pseudo-polynomial graph

Objective: keep energy level between \( L = 0 \) and \( U = 3 \)

\[
\begin{align*}
  c_0 &= 2 \\
  -1 &\quad -2 \\
\end{align*}
\]

\[
\begin{align*}
  2 &\quad 1 \\
  2 &\quad 1 \\
\end{align*}
\]
Solving interval-bounded energy games

Reduction to safety condition in pseudo-polynomial graph

Objective: keep energy level between $L = 0$ and $U = 3$

$c_0 = 2$

$c_0 = 2$

$1$

$2$

$3$

$< 0$

$> 3$

$2$

$1$

$0$
Solving interval-bounded energy games

Reduction to safety condition in pseudo-polynomial graph

Objective: keep energy level between $L = 0$ and $U = 3$

$c_0 = 2$

1-player interval-bounded energy games are PSPACE-complete.
2-player interval-bounded energy games are EXPTIME-complete.

Theorem ([BFLMS08,FJ13])

Energy parity games

Parity games

Objective of Player 1: make $M(\gamma)$ even for any outcome.

Theorem: Both players have memoryless optimal strategies. Deciding the winner is in \(NP \cap \text{coNP}\).
Energy parity games

Objective of Player 1: make $M(\gamma)$ even for any outcome.

Parity games

$M(\gamma) = \text{maximal value seen along } \gamma.$

Theorem

Both players have memoryless optimal strategies.

Deciding the winner is in $\text{NP} \cap \text{coNP}$. 
Energy parity games

Parity games

$M(\gamma) = \text{maximal value seen along } \gamma.$

Objective of Player 1:
make $M(\gamma)$ even for any outcome.

Theorem

Both players have memoryless optimal strategies.
Deciding the winner is in $\text{NP} \cap \text{coNP}$. 
Objective of Player 1: for any $\gamma$, make $M(\gamma)$ even and keep energy above $L$ all along $\gamma$.

Theorem ([CD12])

Player 2 has memoryless optimal strategies.
Player 1 has optimal strategies combining 2 memoryless strategies.

Deciding the winner is in $\text{NP} \cap \text{coNP}$. 
Objective of Player 1: for any $\gamma$

- make $M(\gamma)$ even
- keep energy above $L$ all along $\gamma$
Objective of Player 1: for any $\gamma$
- make $M(\gamma)$ even
- keep energy above $L$ all along $\gamma$

Theorem ([CD12])

Player 2 has memoryless optimal strategies.
Player 1 has optimal strategies combining 2 memoryless strategies.
Deciding the winner is in $\text{NP} \cap \text{coNP}$.

Average-energy games

Average energy is not mean-payoff

- **Mean payoff** = average of weight on transitions

\[ MP(\pi \leq n) = \limsup_{n \to \infty} \frac{1}{n} EL(\pi \leq n) \]
Average-energy games

Average energy is not mean-payoff

- **mean payoff** = average of weight on transitions
  \[ MP(\pi_{\leq n}) = \limsup_{n \to \infty} \frac{1}{n} EL(\pi_{\leq n}) \]

- **average energy** = average of accumulated weight
  \[ AE(\pi_{\leq n}) = \limsup_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} EL(\pi_{\leq i}) \]
Average-energy games

Average energy is not mean-payoff

- **mean payoff** = average of weight on transitions

  \[ MP(\pi_{\leq n}) = \limsup_{n \to \infty} \frac{1}{n} EL(\pi_{\leq n}) \]

- **average energy** = average of accumulated weight

  \[ AE(\pi_{\leq n}) = \limsup_{n \to \infty} \frac{1}{n} \sum_{i=1}^{n} EL(\pi_{\leq i}) \]
Average-energy games

**Theorem ([BMRL15])**

- 1-player AE games can be solved in PTIME (memoryless strategies are sufficient)

Average-energy games

Theorem ([BMRL15])

- 1-player AE games can be solved in PTIME (memoryless strategies are sufficient)
  - compute $\overline{AE}(C_{k,s})$ for all simple cycles of length $k$ on $s$;
  - minimize $EL(\rho_{s_0 \rightarrow s}) + \overline{AE}(C_s)$.

Average-energy games

Theorem ([BMRL15])

1-player AE games can be solved in PTIME (memoryless strategies are sufficient)

- compute $\overline{AE}(C_{k,s})$ for all simple cycles of length $k$ on $s$;
- minimize $EL(\rho_{s_0 \rightarrow s}) + \overline{AE}(C_s)$.

Average-energy games

**Theorem ([BMRLL15])**

- 1-player AE games can be solved in PTIME (memoryless strategies are sufficient)
  - compute $\overline{AE}(C_{k,s})$ for all simple cycles of length $k$ on $s$;
  - minimize $EL(\rho_{s_0 \rightarrow s}) + \overline{AE}(C_s)$.

- In 2-player AE games, both players have memoryless optimal strategies
- Deciding the winner is in $\text{NP} \cap \text{coNP}$
- Mean-payoff games are logspace-reducible to AE games

---

Energy- and average-energy constraints

Mixing average-energy and interval constraints

\[ a, 0, b, 0, c, 0, a, 1, b, 1, c, 1, a, 2, b, 2, c, 2, a, 3, b, 3, c, 3 \]

Theorem ([BMRLL15])

1-player AELU-games are in \( \text{EXPTIME} \), and \( \text{PSPACE} \)-hard.

2-player AELU-games are \( \text{EXPTIME} \)-complete.

Energy- and average-energy constraints

Mixing average-energy and interval constraints

Theorem ([BMRLL15])

1-player AELU-games are in EXPTIME, and PSPACE-hard.

2-player AELU-games are EXPTIME-complete.

Energy- and average-energy constraints

Mixing average-energy and interval constraints

Theorem ([BMRLL15]) 1-player AELU-games are in EXPTIME, and PSPACE-hard. 2-player AELU-games are EXPTIME-complete.

Energy- and average-energy constraints

Theorem ([BMRLL15])

1-player AELU-games are in EXPTIME, and PSPACE-hard.
2-player AELU-games are EXPTIME-complete.

Energy- and average-energy constraints

Mixing average-energy and lower-bound constraints

Theorem ([BHM+17]) 1-player AEL-games are in \(\text{EXPTIME}\), and NP-hard.

2-player AEL-games are in 2-EXPTIME, and EXPSPACE-hard.
Energy- and average-energy constraints

Mixing average-energy and lower-bound constraints

Bound peak height $U$:
- pseudo-polynomial for 1-player games
- 2-exponential for 2-player games

Theorem ([BHM+17])

1-player AEL-games are in $\text{EXPTIME}$, and $\text{NP}$-hard.

2-player AEL-games are in 2-$\text{EXPTIME}$, and $\text{EXPSPACE}$-hard.
Energy- and average-energy constraints

Mixing average-energy and lower-bound constraints

Bound peak height $U$:
- pseudo-polynomial for 1-player games
- 2-exponential for 2-player games

Theorem ([BHM+17])

1-player AEL-games are in EXPTIME, and NP-hard.
2-player AEL-games are in 2-EXPTIME, and EXPSPACE-hard.

Timed energy games

Timed automata: example of a computer mouse

- **left**
  - left_button?
  - left_click!
  - left_double_click!

- **idle**
  - left_button?
  - right_button?
  - right_click!
  - right_double_click!

- **right**
  - right_button?

Theorem ([AD94, AMP98]):

Reachability in timed automata is PSPACE-complete.
Reachability in timed games is EXPTIME-complete.
Timed energy games

Timed automata: example of a computer mouse

- **left**
  - left_button?
  - left_click!
  - left_double_click!

- **idle**
  - left_button?
  - right_button?
  - right_click!
  - right_double_click!

- **right**
  - right_button?

Theorem ([AD94, AMP + 98])

Reachability in timed automata is PSPACE-complete.
Reachability in timed games is EXPSPACE-complete.
Timed energy games

Timed automata: example of a computer mouse

Theorem ([AD94, AM98])

Reachability in timed automata is PSPACE-complete.

Reachability in timed games is EXPTIME-complete.
Timed energy games

Timed automata: example of a computer mouse

left

idle

right

\( x \leq 300 \)

\( x = 300 \)

left_button?

right_button?

left_double_click!

right_double_click!

left_button?

right_button?

Theorem ([AD94, AMP+98])

Reachability in timed automata is PSPACE-complete.
Reachability in timed games is EXPTIME-complete.

Theorem ([BFLMS08, BFLM10])

For 1 player:
- Lower-bound problem for 1-clock timed automata is in \( \text{EXPTIME} \)
- Interval problem for 2-clock timed automata is undecidable

For 2 players:
- Interval problem for 1-clock timed automata is undecidable
Timed energy games

Weighted timed automata

\[ \ell_0 \xrightarrow{x:=0} \ell_1 \xrightarrow{x=1} \ell_2 \]

\[ x = 0 \]

\[ x = 1 \]

Theorem ([BFLMS08, BFLM10])

For 1 player:
- Lower-bound problem for 1-clock timed automata is in EXPTIME
- Interval problem for 2-clock timed automata is undecidable

For 2 players:
- Interval problem for 1-clock timed automata is undecidable
Timed energy games

Weighted timed automata

\[ \ell_0 - 3 \xrightarrow{x:=0} \ell_1 + 6 \xrightarrow{x=1} \ell_2 - 6 \]

Theorem ([BFLMS08, BFLM10])

- For 1 player:
  - Lower-bound problem for 1-clock timed automata is in \( EXPTIME \)
  - Interval problem for 2-clock timed automata is undecidable

- For 2 players:
  - Interval problem for 1-clock timed automata is undecidable
Timed energy games

Weighted timed automata

\( \ell_0 \xrightarrow{x:=0} \ell_1 \xrightarrow{x=1} \ell_2 \)

\[ x = 0 \]

\[ x = 1 \]

Theorem ([BFLMS08, BFLM10])

For 1 player:
- Lower-bound problem for 1-clock timed automata is in \( \text{EXPTIME} \)
- Interval problem for 2-clock timed automata is undecidable

For 2 players:
- Interval problem for 1-clock timed automata is undecidable
Timed energy games

Weighted timed automata

For 1 player:
- lower-bound problem for 1-clock timed automata is in \( \text{EXPTIME} \)
- interval problem for 2-clock timed automata is undecidable

For 2 players:
- interval problem for 1-clock timed automata is undecidable

\[
\ell_0 - 3 \xrightarrow{x:=0} \ell_1 + 6 \xrightarrow{x=1} \ell_2 - 6
\]

\[
x = 0 \quad \ell_0
\]

\[
x = 1 \quad \ell_2
\]

\[
\ell_1
\]

\[
\text{energy}
\]

\[
\text{time}
\]

\[
x = 1
\]

\[
x = 0
\]
Timed energy games

Weighted timed automata

\[
\ell_0 -3 \xrightarrow{x:=0} \ell_1 +6 \xrightarrow{x=1} \ell_2 -6
\]

Theorem ([BFLMS08,BFLM10])

- For 1 player:
  - lower-bound problem for 1-clock timed automata is in \text{EXPTIME}
  - interval problem for 2-clock timed automata is undecidable
- For 2 players:
  - interval problem for 1-clock timed automata is undecidable

Conclusion and future works

**Conclusion**

- **Weighted games**, in particular **energy games**, conveniently model resource-management problems;
- they are rather **well-understood**, but with significant open problems.
- **no real tools available**, only prototypes.

**Future works**

- extend to **stochastic strategies**, **stochastic games**
- **multiple-player** quantitative games
- combine **weighted timed games** **with imprecisions**