Expectations or Guarantees? I Want It All!
A Crossroad between Games and MDPs

V. Bruyère (UMONS)  E. Filiot (ULB)
M. Randour (UMONS-ULB)  J.-F. Raskin (ULB)

Grenoble - 05.04.2014

SR 2014 - 2nd International Workshop on Strategic Reasoning
The talk in two slides (1/2)

- Verification and synthesis:
  - a reactive system to control,
  - an interacting environment,
  - a specification to enforce.

- Focus on quantitative properties.
The talk in two slides (1/2)

- Verification and synthesis:
  - a reactive **system** to *control*,
  - an *interacting environment*,
  - a **specification** to *enforce*.

- Focus on *quantitative properties*.

- Several ways to look at the interactions, and in particular, *the nature of the environment*. 
The talk in two slides (2/2)

- **Games**
  - antagonistic adversary
  - guarantees on *worst-case*

- **MDPs**
  - stochastic adversary
  - optimize *expected value*
The talk in two slides (2/2)

- **Games**
  - antagonistic adversary
  - guarantees on *worst-case*

- **MDPs**
  - stochastic adversary
  - optimize *expected value*

**BWC synthesis**
- ensure *both*
The talk in two slides (2/2)

Games
→ antagonistic adversary
→ guarantees on worst-case

MDPs
→ stochastic adversary
→ optimize expected value

BWC synthesis
→ ensure both

Mean-Payoff

Shortest Path

Studied value functions
Advertisement

Featured in STACS'14 [BFRR14]
Full paper available on arXiv: abs/1309.5439
1. Context

2. BWC Synthesis

3. Mean-Payoff

4. Shortest Path

5. Conclusion
1. Context

2. BWC Synthesis

3. Mean-Payoff

4. Shortest Path

5. Conclusion
Quantitative games on graphs

- Graph $G = (S, E, w)$ with $w : E \to \mathbb{Z}$

- Two-player game $G = (G, S_1, S_2)$
  - $\mathcal{P}_1$ states = ○
  - $\mathcal{P}_2$ states = □

- Plays have values
  - $f : \text{Plays}(G) \to \mathbb{R} \cup \{-\infty, \infty\}$

- Players follow strategies
  - $\lambda_i : \text{Prefs}_i(G) \to \mathcal{D}(S)$
  - Finite memory $\Rightarrow$ stochastic output Moore machine $\mathcal{M}(\lambda_i) = (\text{Mem}, m_0, \alpha_u, \alpha_n)$
Quantitative games on graphs

- Graph $G = (S, E, w)$ with $w : E \to \mathbb{Z}$
- Two-player game $G = (G, S_1, S_2)$
  - $P_1$ states = $igcirc$
  - $P_2$ states = □
- Plays have values
  - $f : \text{Plays}(G) \to \mathbb{R} \cup \{-\infty, \infty\}$
- Players follow strategies
  - $\lambda_i : \text{Prefs}_i(G) \to D(S)$
  - Finite memory $\Rightarrow$ stochastic output Moore machine $M(\lambda_i) = (\text{Mem}, m_0, \alpha_u, \alpha_n)$
Quantitative games on graphs

- **Graph** $G = (S, E, w)$ with $w : E \to \mathbb{Z}$

- **Two-player game** $G = (G, S_1, S_2)$
  - $\mathcal{P}_1$ states = □
  - $\mathcal{P}_2$ states = □

- Plays have values
  - $f : \text{Plays}(G) \to \mathbb{R} \cup \{-\infty, \infty\}$

- Players follow strategies
  - $\lambda_i : \text{Prefs}_i(G) \to \mathcal{D}(S)$
  - Finite memory $\Rightarrow$ stochastic output Moore machine $\mathcal{M}(\lambda_i) = (\text{Mem}, m_0, \alpha_u, \alpha_n)$
Quantitative games on graphs

- Graph $\mathcal{G} = (S, E, w)$ with $w : E \to \mathbb{Z}$

- Two-player game $G = (\mathcal{G}, S_1, S_2)$
  - $\mathcal{P}_1$ states =
  - $\mathcal{P}_2$ states =

- Plays have values
  - $f : \text{Plays}(\mathcal{G}) \to \mathbb{R} \cup \{-\infty, \infty\}$

- Players follow strategies
  - $\lambda_i : \text{Prefs}_i(G) \to \mathcal{D}(S)$
  - Finite memory $\Rightarrow$ stochastic output Moore machine $\mathcal{M}(\lambda_i) = (\text{Mem}, m_0, \alpha_u, \alpha_n)$
Quantitative games on graphs

- Graph $G = (S, E, w)$ with $w : E \to \mathbb{Z}$
- Two-player game $G = (G, S_1, S_2)$
  - $\mathcal{P}_1$ states = ○
  - $\mathcal{P}_2$ states = □
- Plays have values
  - $f : \text{Plays}(G) \to \mathbb{R} \cup \{-\infty, \infty\}$
- Players follow strategies
  - $\lambda_i : \text{Prefs}_i(G) \to \mathcal{D}(S)$
  - Finite memory $\Rightarrow$ stochastic output Moore machine $\mathcal{M}(\lambda_i) = (\text{Mem}, m_0, \alpha_u, \alpha_n)$
Quantitative games on graphs

- Graph $G = (S, E, w)$ with $w : E \to \mathbb{Z}$
- Two-player game $G = (G, S_1, S_2)$
  - $P_1$ states = ◯
  - $P_2$ states = □
- Plays have values
  - $f : \text{Plays}(G) \to \mathbb{R} \cup \{-\infty, \infty\}$
- Players follow strategies
  - $\lambda_i : \text{Prefs}_i(G) \to D(S)$
  - Finite memory $\Rightarrow$ stochastic output Moore machine $M(\lambda_i) = (\text{Mem}, m_0, \alpha_u, \alpha_n)$
Quantitative games on graphs

Graph $G = (S, E, w)$ with $w : E \rightarrow \mathbb{Z}$

Two-player game $G = (G, S_1, S_2)$
  - $P_1$ states = 
  - $P_2$ states =

Plays have values
  - $f : \text{Plays}(G) \rightarrow \mathbb{R} \cup \{-\infty, \infty\}$

Players follow strategies
  - $\lambda_i : \text{Prefs}_i(G) \rightarrow \mathcal{D}(S)$
  - Finite memory $\Rightarrow$ stochastic output Moore machine $M(\lambda_i) = (\text{Mem}, m_0, \alpha_u, \alpha_n)$

Then, $(2, 5, 2)\omega$
Markov decision processes

- MDP $P = (\mathcal{G}, S_1, S_\Delta, \Delta)$ with $\Delta : S_\Delta \rightarrow \mathcal{D}(S)$
  - $\mathcal{P}_1$ states = $igcirc$
  - stochastic states = $lacksquare$

- MDP = game + strategy of $\mathcal{P}_2$
  - $P = G[\lambda_2]$
Markov chains

- MC $M = (G, \delta)$ with $\delta: S \rightarrow \mathcal{D}(S)$
- MC = MDP + strategy of $P_1$
  = game + both strategies
  $\triangleright$ $M = P[\lambda_1] = G[\lambda_1, \lambda_2]$
Markov chains

- **MC** $M = (G, \delta)$ with $\delta: S \rightarrow D(S)$
- **MC** = MDP + strategy of $P_1$
  - = game + both strategies
    - $M = P[\lambda_1] = G[\lambda_1, \lambda_2]$
- Event $A \subseteq \text{Plays}(G)$
  - $\text{probability } \mathbb{P}^M\text{_{init}}(A)$
- Measurable $f: \text{Plays}(G) \rightarrow \mathbb{R} \cup \{-\infty, \infty\}$
  - $\text{expected value } \mathbb{E}^M\text{_{init}}(f)$
Classical interpretations

- **System** trying to ensure a specification $= \mathcal{P}_1$
  - whatever the actions of its **environment**
Classical interpretations

- **System** trying to ensure a specification $= P_1$
  - whatever the actions of its environment
- The environment can be seen as
  - antagonistic
    - two-player game, worst-case threshold problem for $\mu \in \mathbb{Q}$
    - $\exists \lambda_1 \in \Lambda_1, \forall \lambda_2 \in \Lambda_2, \forall \pi \in \text{Outs}_G(s_{\text{init}}, \lambda_1, \lambda_2), f(\pi) \geq \mu$
Classical interpretations

- **System** trying to ensure a specification $= \mathcal{P}_1$
  - whatever the actions of its **environment**

- The environment can be seen as
  - *antagonistic*
    - two-player game, *worst-case* threshold problem for $\mu \in \mathbb{Q}$
    - $\exists \lambda_1 \in \Lambda_1, \forall \lambda_2 \in \Lambda_2, \forall \pi \in \text{Out}_{G}(s_{\text{init}}, \lambda_1, \lambda_2), f(\pi) \geq \mu$
  - *fully stochastic*
    - MDP, *expected value* threshold problem for $\nu \in \mathbb{Q}$
    - $\exists \lambda_1 \in \Lambda_1, \mathbb{E}_{s_{\text{init}}}^{P[\lambda_1]}(f) \geq \nu$
<table>
<thead>
<tr>
<th></th>
<th>Context</th>
<th>BWC Synthesis</th>
<th>Mean-Payoff</th>
<th>Shortest Path</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Context</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td><strong>BWC Synthesis</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td><strong>Mean-Payoff</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td><strong>Shortest Path</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td><strong>Conclusion</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
What if you want both?

In practice, we want both

1. nice expected performance in the everyday situation,
2. strict (but relaxed) performance guarantees even in the event of very bad circumstances.
Example: going to work

- **Weights** = minutes
- **Goal**: minimize our expected time to reach “work”
- **But**, important meeting in one hour! Requires strict guarantees on the worst-case reaching time.
Example: going to work

- **Optimal expectation strategy:** take the car.
  - $\mathbb{E} = 33$, WC $= 71 > 60$.

- **Optimal worst-case strategy:** bicycle.
  - $\mathbb{E} = WC = 45 < 60$.  

Diagram:

- Home to station: 1
- Train: 2
- Car: 1
- Back home: 1
- Station to traffic: 9/10
- Wait: 4
- Depart: 35
- Light: 20
- Medium: 30
- Heavy: 70
- Traffic to work: 7/10
- Work to waiting room: 1/10
- Waiting room to home: 45
- Bicycle: 45
- Back home to work: 1

Diagram shows transitions and delays with probabilities.
Example: going to work

- Optimal expectation strategy: take the car.
  - $\mathbb{E} = 33$, $WC = 71 > 60$.

- Optimal worst-case strategy: bicycle.
  - $\mathbb{E} = WC = 45 < 60$.

- Sample BWC strategy: try train up to 3 delays then switch to bicycle.
  - $\mathbb{E} \approx 37.56$, $WC = 59 < 60$.
  - Optimal $\mathbb{E}$ under WC constraint
  - Uses finite memory
Beyond worst-case synthesis

Formal definition

Given a game $G = (G, S_1, S_2)$, with $G = (S, E, w)$ its underlying graph, an initial state $s_{\text{init}} \in S$, a finite-memory stochastic model $\lambda_{\text{stoch}}^2 \in \Lambda_2^F$ of the adversary, represented by a stochastic Moore machine, a measurable value function $f : \text{Plays}(G) \to \mathbb{R} \cup \{-\infty, \infty\}$, and two rational thresholds $\mu, \nu \in \mathbb{Q}$, the beyond worst-case (BWC) problem asks to decide if $P_1$ has a finite-memory strategy $\lambda_1 \in \Lambda_1^F$ such that

$$\forall \lambda_2 \in \Lambda_2, \forall \pi \in \text{Outs}_G(s_{\text{init}}, \lambda_1, \lambda_2), f(\pi) > \mu$$

(1)

$$E_{s_{\text{init}}}^{G[\lambda_1, \lambda_{\text{stoch}}^2]}(f) > \nu$$

(2)

and the BWC synthesis problem asks to synthesize such a strategy if one exists.
Beyond worst-case synthesis

Formal definition

Given a game $G = (G, S_1, S_2)$, with $G = (S, E, w)$ its underlying graph, an initial state $s_{\text{init}} \in S$, a finite-memory stochastic model $\lambda^{\text{stoch}}_2 \in \Lambda^F_2$ of the adversary, represented by a stochastic Moore machine, a measurable value function $f : \text{Plays}(G) \rightarrow \mathbb{R} \cup \{-\infty, \infty\}$, and two rational thresholds $\mu, \nu \in \mathbb{Q}$, the **beyond worst-case (BWC) problem** asks to decide if $P_1$ has a finite-memory strategy $\lambda_1 \in \Lambda^F_1$ such that

\[
\forall \lambda_2 \in \Lambda_2, \forall \pi \in \text{Outs}_G(s_{\text{init}}, \lambda_1, \lambda_2), f(\pi) > \mu \tag{1}
\]

\[
\mathbb{E}^{G[\lambda_1, \lambda^{\text{stoch}}_2]}_{s_{\text{init}}}(f) > \nu \tag{2}
\]

and the **BWC synthesis problem** asks to synthesize such a strategy if one exists.

Notice the highlighted parts!
Related work

**Common philosophy:** avoiding outlier outcomes

1. Our strategies are *strongly risk averse*
   - avoid risk at all costs and optimize among safe strategies
Related work

**Common philosophy:** avoiding outlier outcomes

1. Our strategies are *strongly risk averse*
   - avoid risk at all costs and optimize among safe strategies

2. Other notions of risk ensure low probability of risked behavior [WL99, FKR95]
   - without worst-case guarantee
   - without good expectation
Related work

**Common philosophy:** avoiding outlier outcomes

1. **Our strategies are strongly risk averse**
   - avoid risk at all costs and optimize among safe strategies

2. **Other notions of risk ensure low probability of risked behavior** [WL99, FKR95]
   - without worst-case guarantee
   - without good expectation

3. **Trade-off between expectation and variance** [BCFK13, MT11]
   - statistical measure of the stability of the performance
   - no strict guarantee on individual outcomes
<table>
<thead>
<tr>
<th></th>
<th>Context</th>
<th>BWC Synthesis</th>
<th>Mean-Payoff</th>
<th>Shortest Path</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Context</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>BWC Synthesis</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Mean-Payoff</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Shortest Path</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Conclusion</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Mean-payoff value function

- $\text{MP}(\pi) = \lim_{n \to \infty} \inf \left[ \frac{1}{n} \cdot \sum_{i=0}^{i=n-1} w((s_i, s_{i+1})) \right]$

- Sample play $\pi = 2, -1, -4, 5, (2, 2, 5)^\omega$
  - $\text{MP}(\pi) = 3$
  - long-run average weight $\sim$ *prefix-independent*
Mean-payoff value function

- $\text{MP}(\pi) = \liminf_{n \to \infty} \left[ \frac{1}{n} \cdot \sum_{i=0}^{i=n-1} w((s_i, s_{i+1})) \right]$

- Sample play $\pi = 2, -1, -4, 5, (2, 2, 5)^\omega$
  - $\text{MP}(\pi) = 3$
  - long-run average weight $\sim$ prefix-independent

<table>
<thead>
<tr>
<th>worst-case complexity</th>
<th>expected value memory</th>
<th>BWC memory complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{NP} \cap \text{coNP}$</td>
<td>memoryless</td>
<td>$\text{NP} \cap \text{coNP}$</td>
</tr>
<tr>
<td>$\text{P}$</td>
<td>memoryless</td>
<td>pseudo-polynomial</td>
</tr>
</tbody>
</table>

- [LL69, EM79, ZP96, Jur98, GS09, Put94, FV97]
- Additional modeling power for free!
Philosophy of the algorithm

- Classical worst-case and expected value results and algorithms as *nuts and bolts*
- *Screw them together* in an adequate way
Philosophy of the algorithm

- Classical worst-case and expected value results and algorithms as *nuts and bolts*
- *Screw them together* in an adequate way

Three key ideas

1. To characterize the expected value, look at *end-components* (ECs)
2. *Winning ECs vs. losing ECs*: the latter must be avoided to preserve the worst-case requirement!
3. *Inside a WEC*, we have an interesting way to play...
Philosophy of the algorithm

- Classical worst-case and expected value results and algorithms as *nuts and bolts*
- *Screw them together* in an adequate way

**Three key ideas**

1. To characterize the expected value, look at *end-components* (ECs)
2. *Winning ECs vs. losing ECs*: the latter must be avoided to preserve the worst-case requirement!
3. *Inside a WEC, we have an interesting way to play...*

⇒ Let’s focus on an ideal case
An ideal situation
An ideal situation

Game interpretation

- Worst-case threshold is $\mu = 0$
- **All** states are winning: memoryless optimal worst-case strategy $\lambda_{wc}^1 \in \Lambda_{1}^{PM}(G)$, ensuring $\mu^* = 1 > 0$
An ideal situation

![Game diagram](image)

**Game interpretation**

- Worst-case threshold is $\mu = 0$
- All states are winning: memoryless optimal worst-case strategy $\lambda_{wc}^1 \in \Lambda_{PM}^1(G)$, ensuring $\mu^* = 1 > 0$

**MDP interpretation**

- Memoryless optimal expected value strategy $\lambda_{e}^1 \in \Lambda_{PM}^1(P)$ achieves $\nu^* = 2$
A cornerstone of our approach

BWC problem: what kind of thresholds \((0, \nu)\) can we achieve?
A cornerstone of our approach

BWC problem: what kind of thresholds \((0, \nu)\) can we achieve?

Key result

For all \(\varepsilon > 0\), there exists a finite-memory strategy of \(\mathcal{P}_1\) that satisfies the BWC problem for the thresholds pair \((0, \nu^* - \varepsilon)\).

We can be \textit{arbitrarily close to the optimal expectation} while ensuring the worst-case!
Combined strategy

Outcomes of the form

\[ \sum > 0 \quad \sum > 0 \quad \sum \leq 0 \quad \text{compensate} \quad \sum > 0 \quad \sum \leq 0 \quad \text{compensate} \]

\[ WC > 0 \]

\[ \mathbb{E} = ?? \]
Combined strategy

Outcomes of the form

\[ \sum > 0 \quad \sum > 0 \quad \sum \leq 0 \quad \text{compensate} \quad \sum > 0 \quad \sum \leq 0 \quad \text{compensate} \]

What we want

\[ K, L \to \infty \]

\[ \mathbb{E} = \nu^* = 2 \]
Combined strategy: crux of the proof

Precise reasoning on convergence rates using involved techniques

- When $K$ grows, $L$ needs to grow linearly to ensure WC
Combined strategy: crux of the proof

Precise reasoning on convergence rates using involved techniques

- When $K$ grows, $L$ needs to grow linearly to ensure WC
- When $K$ grows, $P(\overline{1'}) \to 0$ and it decreases exponentially fast
  - application of Chernoff bounds and Hoeffding’s inequality for Markov chains [Tra09, GO02]
Combined strategy: crux of the proof

Precise reasoning on convergence rates using involved techniques

- When $K$ grows, $L$ needs to grow linearly to ensure WC
- When $K$ grows, $\mathbb{P}(\overline{\mathcal{I}}) \to 0$ and it decreases exponentially fast
  - application of Chernoff bounds and Hoeffding's inequality for Markov chains [Tra09, GO02]
- Overall we are good: WC > 0 and $\mathbb{E} > \nu^* - \varepsilon$ for sufficiently large $K, L$. 
1. Context

2. BWC Synthesis

3. Mean-Payoff

4. Shortest Path

5. Conclusion
Shortest path - truncated sum

- Assume strictly positive integer weights, \( w : E \to \mathbb{N}_0 \)
- Let \( T \subseteq S \) be a target set that \( P_1 \) wants to reach with a path of bounded value (cf. introductory example)
  - inequalities are reversed, \( \nu < \mu \)
- \( TS_T(\pi = s_0s_1s_2\ldots) = \sum_{i=0}^{n-1} w((s_i, s_{i+1})) \), with \( n \) the first index such that \( s_n \in T \), and \( TS_T(\pi) = \infty \) if \( \forall n, s_n \notin T \)
Shortest path - truncated sum

- Assume strictly positive integer weights, \( w : E \to \mathbb{N}_0 \)
- Let \( T \subseteq S \) be a target set that \( \mathcal{P}_1 \) wants to reach with a path of bounded value (cf. introductory example)
  - inequalities are reversed, \( \nu < \mu \)
- \( TS_T(\pi = s_0s_1s_2 \ldots) = \sum_{i=0}^{n-1} w((s_i, s_{i+1})) \), with \( n \) the first index such that \( s_n \in T \), and \( TS_T(\pi) = \infty \) if \( \forall n, s_n \notin T \)

<table>
<thead>
<tr>
<th>worst-case complexity</th>
<th>expected value memory</th>
<th>BWC</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathcal{P} )</td>
<td>( \mathcal{P} )</td>
<td>pseudo-poly. / NP-hard</td>
</tr>
<tr>
<td>memoryless</td>
<td>memoryless</td>
<td>pseudo-poly.</td>
</tr>
</tbody>
</table>

- [BT91, dA99]
- Problem inherently harder than worst-case and expectation.
- NP-hardness by \( K^{th} \) largest subset problem [JK78, GJ79]
Key difference with MP case

**Useful observation**

The set of all worst-case winning strategies for the shortest path can be represented through a *finite game*.

**Sequential approach** solving the BWC problem:

1. represent all WC winning strategies,
2. optimize the expected value within those strategies.
Pseudo-polynomial algorithm: sketch

1. Start from $G = (G, S_1, S_2)$, $G = (S, E, w)$, $T = \{s_3\}$, $\mathcal{M}(\lambda_2^{stoch})$, $\mu = 8$, and $\nu \in \mathbb{Q}$
Pseudo-polynomial algorithm: sketch

1. Start from $G = (G, S_1, S_2)$, $G = (S, E, w)$, $T = \{s_3\}$, $\mathcal{M}(\lambda_2^{\text{stoch}})$, $\mu = 8$, and $\nu \in \mathbb{Q}$

2. Build $G'$ by unfolding $G$, tracking the current sum up to the worst-case threshold $\mu$, and integrating it in the states of $G'$. 
Pseudo-polynomial algorithm: sketch
Pseudo-polynomial algorithm: sketch

Here, $\nu^* = \frac{9}{2}$.

$s_1$, 0

$s_2$, 1

$s_3$, 5

$s_1$, 1

$s_2$, 1

$s_3$, 2

$s_1$, 2

$s_3$, 2
Pseudo-polynomial algorithm: sketch
Pseudo-polynomial algorithm: sketch
Pseudo-polynomial algorithm: sketch
Pseudo-polynomial algorithm: sketch

Beyond Worst-Case Synthesis  Bruyère, Filiot, Randour, Raskin
Pseudo-polynomial algorithm: sketch
Pseudo-polynomial algorithm: sketch

3. Compute $R$, the attractor of $T$ with cost $< \mu = 8$
4. Consider $G_\mu = G' \downarrow R$
Pseudo-polynomial algorithm: sketch

3 Compute $R$, the attractor of $T$ with cost $< \mu = 8$
4 Consider $G_{\mu} = G' \upharpoonright R$
Pseudo-polynomial algorithm: sketch

5. Consider $P = G_\mu \otimes \mathcal{M}(\lambda_2^{\text{stoch}})$

6. Compute memoryless optimal expectation strategy

7. If $\nu^* < \nu$, answer YES, otherwise answer NO

Here, $\nu^* = 9/2$
<table>
<thead>
<tr>
<th></th>
<th>Context</th>
<th>BWC Synthesis</th>
<th>Mean-Payoff</th>
<th>Shortest Path</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Context</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>BWC Synthesis</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Mean-Payoff</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Shortest Path</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>
</tr>
</tbody>
</table>
In a nutshell

- BWC framework combines worst-case and expected value requirements
  - a natural wish in many practical applications
  - few existing theoretical support
In a nutshell

- BWC framework combines worst-case and expected value requirements
  - a natural wish in many practical applications
  - few existing theoretical support

- Mean-payoff: additional modeling power for no complexity cost (decision-wise)

- Shortest path: harder than the worst-case, pseudo-polynomial with NP-hardness result
In a nutshell

- BWC framework combines worst-case and expected value requirements
  - a natural wish in many practical applications
  - few existing theoretical support

- Mean-payoff: additional modeling power for no complexity cost (decision-wise)

- Shortest path: harder than the worst-case, pseudo-polynomial with NP-hardness result

- In both cases, pseudo-polynomial memory is both sufficient and necessary
  - but strategies have natural representations based on states of the game and simple integer counters
Beyond BWC synthesis?

Possible future works include

- study of other quantitative objectives,

- extension of our results to more general settings (multi-dimension [CDHR10, CRR12], decidable classes of games with imperfect information [DDG+10], etc),

- application of the BWC problem to various practical cases.
Beyond BWC synthesis?

Possible future works include

- study of other quantitative objectives,
- extension of our results to more general settings (multi-dimension [CDHR10, CRR12], decidable classes of games with imperfect information [DDG+10], etc),
- application of the BWC problem to various practical cases.

Thanks!
Do not hesitate to discuss with us!
References I

T. Brázdil, K. Chatterjee, V. Forejt, and A. Kucera.
Trading performance for stability in Markov decision processes.

V. Bruyère, E. Filiot, M. Randour, and J.-F. Raskin.
Meet your expectations with guarantees: beyond worst-case synthesis in quantitative games.

D.P. Bertsekas and J.N. Tsitsiklis.
An analysis of stochastic shortest path problems.

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.

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

L. de Alfaro.
Computing minimum and maximum reachability times in probabilistic systems.
Energy and mean-payoff games with imperfect information.

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

Percentile performance criteria for limiting average Markov decision processes.

J. Filar and K. Vrieze.
Competitive Markov decision processes.

M.R. Garey and D.S. Johnson.

P.W. Glynn and D. Ormoneit.
Hoeffding’s inequality for uniformly ergodic Markov chains.

Games through nested fixpoints.
References III

D.B. Johnson and S.D. Kashdan.
Lower bounds for selection in $X + Y$ and other multisets.

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

T.M. Liggett and S.A. Lippman.
Stochastic games with perfect information and time average payoff.

S. Mannor and J.N. Tsitsiklis.
Mean-variance optimization in Markov decision processes.

M.L. Puterman.
Markov decision processes: discrete stochastic dynamic programming.

M. Tracol.
Fast convergence to state-action frequency polytopes for MDPs.

C. Wu and Y. Lin.
Minimizing risk models in Markov decision processes with policies depending on target values.
U. Zwick and M. Paterson.
The complexity of mean payoff games on graphs.