Meet Your Expectations With Guarantees: Beyond Worst-Case Synthesis in Quantitative Games

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

IST Austria - 17.06.2014
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 \textit{worst-case}

MDPs
→ stochastic adversary
→ optimize \textit{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

Studied value functions

Shortest Path
Advertisement

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

2. BWC Synthesis

3. Mean-Payoff

4. Shortest Path

5. Conclusion
Quantitative games on graphs

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

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

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

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

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

- Graph $G = (S, E, w)$ with $w : E \rightarrow \mathbb{Z}$
- Two-player game $G = (\mathcal{G}, S_1, S_2)$
  - $\mathcal{P}_1$ states = $\bigcirc$
  - $\mathcal{P}_2$ states = $\blacksquare$

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

- Players follow strategies
  - $\lambda_i : \text{Prefs}_i(G) \rightarrow \mathcal{D}(S)$
  - Finite memory $\Rightarrow$ stochastic output Moore machine $\mathcal{M}(\lambda_i) = (\text{Mem}, m_0, \alpha_u, \alpha_n)$
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 = $\bigcirc$
  - $P_2$ states = $\blacksquare$
- 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 $\mathcal{M}(\lambda_i) = (\text{Mem}, m_0, \alpha_u, \alpha_n)$
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 =دائرة
  - $\mathcal{P}_2$ states = مستطيل
- 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 $\mathcal{M}(\lambda_i) = (\text{Mem}, m_0, \alpha_u, \alpha_n)$

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 = ○
  - stochastic states = □

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

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

- MC $M = (G, \delta)$ with $\delta: S \to D(S)$
- MC = MDP + strategy of $\mathcal{P}_1$
  = game + both strategies
  $\triangleright$ $M = P[\lambda_1] = G[\lambda_1, \lambda_2]$
- Event $A \subseteq \text{Plays}(G)$
  $\triangleright$ probability $\mathbb{P}^M_{\text{init}}(A)$
- Measurable $f: \text{Plays}(G) \to \mathbb{R} \cup \{-\infty, \infty\}$
  $\triangleright$ expected value $\mathbb{E}^M_{\text{init}}(f)$
Classical interpretations

- **System** trying to ensure a specification $= P_1$
  - whatever the actions of its **environment**
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 $\mu \in \mathbb{Q}$
    - $\exists \lambda_1 \in \Lambda_1, \forall \lambda_2 \in \Lambda_2, \forall \pi \in \text{Outs}_G(s_{\text{init}}, \lambda_1, \lambda_2), f(\pi) \geq \mu$
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 \( \mu \in \mathbb{Q} \)
    
    - \( \exists \lambda_1 \in \Lambda_1, \forall \lambda_2 \in \Lambda_2, \forall \pi \in \text{Out}_{G}(s_{\text{init}}, \lambda_1, \lambda_2), f(\pi) \geq \mu \)

  ▶ **fully stochastic**
    
    - MDP, expected value threshold problem for \( \nu \in \mathbb{Q} \)
    
    - \( \exists \lambda_1 \in \Lambda_1, \mathbb{E}_{s_{\text{init}}}^{P[\lambda_1]}(f) \geq \nu \)
1 Context

2 BWC Synthesis

3 Mean-Payoff

4 Shortest Path

5 Conclusion
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.
  - $E = 33$, $WC = 71 > 60$.

- Optimal worst-case strategy: bicycle.
  - $E = WC = 45 < 60$. 

Diagram:

- Home to station: train with probability 2/10, back home with probability 1/10.
- Station to traffic: delay with probability 1/10, depart light with probability 35/10, wait with probability 4/10, depart medium with probability 20/10, wait with probability 20/10, depart heavy with probability 1/10.
- Traffic to work: bicycle with probability 45/10, car with probability 1/10, back home with probability 1/10.
- Work to waiting room: wait with probability 7/10, depart medium with probability 30/10, wait with probability 4/10, depart heavy with probability 1/10.
- Waiting room to home: delay with probability 1/10, wait with probability 4/10.
Example: going to work

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

- Optimal worst-case strategy: bicycle.
  - $E = WC = 45 < 60$.

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

Formal definition

Given a game $G = (\mathcal{G}, S_1, S_2)$, with $\mathcal{G} = (S, E, w)$ its underlying graph, an initial state $s_{\text{init}} \in S$, a finite-memory stochastic model $\lambda_{\text{stoch}}^{\text{2stoch}} \in \Lambda_{\text{2stoch}}^{\text{finite}}$ of the adversary, represented by a stochastic Moore machine, a measurable value function $f : \text{Plays}(\mathcal{G}) \rightarrow \mathbb{R} \cup \{-\infty, \infty\}$, and two rational thresholds $\mu, \nu \in \mathbb{Q}$, the beyond worst-case (BWC) problem asks to decide if $\mathcal{P}_1$ has a finite-memory strategy $\lambda_1 \in \Lambda_1^{\text{finite}}$ 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)

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

(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 $\lambda_{\text{stoch}}^2 \in \Lambda_2^F$ 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 $\mathcal{P}_1$ 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)

and

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

(2)

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

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**
   - without worst-case guarantee
   - without good expectation

[WL99, FKR95]
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**
   - without worst-case guarantee
   - without good expectation

3. **Trade-off between expectation and variance**
   - statistical measure of the stability of the performance
   - no strict guarantee on individual outcomes
1. Context

2. BWC Synthesis

3. Mean-Payoff

4. Shortest Path

5. Conclusion
Mean-payoff value function

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

Sample play \( \pi = 2, -1, -4, 5, (2, 2, 5)^\omega \)

- \( \text{MP}(\pi) = 3 \)
- long-run average weight \( \sim \) prefix-independent
Mean-payoff value function

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

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

<table>
<thead>
<tr>
<th>worst-case complexity</th>
<th>expected value memory</th>
<th>BWC memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP ( \cap ) coNP</td>
<td>P</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
Inside a WEC
Inside a WEC

Game interpretation

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

Game interpretation

▷ Worst-case threshold is $\mu = 0$

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

MDP interpretation

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

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

BWC problem: what kind of thresholds ($\mu = 0, \nu$) 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 arbitrarily close to the optimal expectation while ensuring the worst-case
Combined strategy

Outcomes of the form

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

\[ WC > 0 \quad \mathbb{E} = ?? \]
Combined strategy

Outcomes of the form

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

K steps

L steps

WC > 0

What we want

\[ K, L \to \infty \]

\[ \mathbb{E} = \nu^* = 2 \]
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

\[ K, L \to \infty \]

\[ L = \text{linear}(K) \]

\[ \mathbb{P}(\text{----}) \to 0 \text{ exp. fast!} \quad [\text{Tra09, GO02}] \]

\[ \mathbb{E} = \nu^* = 2 \]
The ideal case: wrap-up

The combined strategy works in any subgame such that

1. it constitutes an EC in the MDP,
2. all states are worst-case winning in the subgame.

Such winning ECs (WECs) are the crux of BWC strategies in arbitrary games.

But to explain that, let’s first zoom out and consider the big picture.
Zooming out

Arbitrary game, with ideal case as a subgame. We assume all states are worst-case winning.

- BWC strategies must avoid WC losing states at all times: an antagonistic adversary can force WC losing outcomes from there (due to prefix-independence)
- Some preprocessing can be done and in the remaining game, $\mathcal{P}_1$ has a memoryless WC winning strategy from all states
An EC of the MDP $P = G[\lambda_{2}^{\text{stoch}}]$ is a subgraph in which $P_1$ can ensure to stay despite stochastic states [dA97], i.e., a set $U \subseteq S$ s.t.

(i) $(U, E \cap (U \times U))$ is strongly connected,

(ii) $\forall s \in U \cap S_\Delta$, $\text{Supp}(\Delta(s)) \subseteq U$, i.e., in stochastic states, all outgoing edges stay in $U$.
End-components: what they are

An EC of the MDP $P = G[\lambda_2^{stoch}]$ is a subgraph in which $P_1$ can ensure to stay despite stochastic states \cite{dA97}, i.e., a set $\mathcal{U} \subseteq S$ s.t.

(i) $(\mathcal{U}, E \cap (\mathcal{U} \times \mathcal{U}))$ is strongly connected,

(ii) $\forall s \in \mathcal{U} \cap S_\Delta$, $\text{Supp}(\Delta(s)) \subseteq \mathcal{U}$, i.e., in stochastic states, all outgoing edges stay in $\mathcal{U}$.

$\triangleright$ ECs: $\mathcal{E} = \{U_1\}$
End-components: what they are

An **EC** of the MDP $P = G[\lambda_2^{\text{stoch}}]$ is a subgraph in which $\mathcal{P}_1$ can ensure to stay despite stochastic states [dA97], i.e., a set $U \subseteq S$ s.t.

(i) $(U, E \cap (U \times U))$ is strongly connected,

(ii) $\forall s \in U \cap S_\Delta$, $\text{Supp}(\Delta(s)) \subseteq U$, i.e., in stochastic states, all outgoing edges stay in $U$.

▷ **ECs:** $\mathcal{E} = \{U_1, U_2\}$
End-components: what they are

An **EC** of the MDP $P = G[\lambda_2^{stoch}]$ is a subgraph in which $\mathcal{P}_1$ can ensure to stay despite stochastic states [dA97], i.e., a set $U \subseteq S$ s.t.

(i) $(U, E \cap (U \times U))$ is strongly connected,

(ii) $\forall s \in U \cap S_\Delta$, $\text{Supp}(\Delta(s)) \subseteq U$, i.e., in stochastic states, all outgoing edges stay in $U$.

▷ **ECs:** $\mathcal{E} = \{U_1, U_2, U_3\}$
End-components: what they are

An **EC** of the MDP $P = G[\lambda^\text{stoch}_2]$ is a subgraph in which $P_1$ can ensure to stay despite stochastic states [dA97], i.e., a set $U \subseteq S$ s.t.

(i) $(U, E \cap (U \times U))$ is strongly connected,

(ii) $\forall s \in U \cap S_{\Delta}, \text{Supp}(\Delta(s)) \subseteq U$, i.e., in stochastic states, all outgoing edges stay in $U$.

▷ **ECs:** $\mathcal{E} = \{U_1, U_2, U_3, \{s_5, s_6\}, \{s_6, s_7\}, \{s_1, s_3, s_4, s_5\}\}$
An EC of the MDP $P = G[\lambda_2^{\text{stoch}}]$ is a subgraph in which $P_1$ can ensure to stay despite stochastic states [dA97], i.e., a set $U \subseteq S$ s.t.

1. $(U, E \cap (U \times U))$ is strongly connected,
2. $\forall s \in U \cap S_\Delta$, $\text{Supp}(\Delta(s)) \subseteq U$, i.e., in stochastic states, all outgoing edges stay in $U$.

$\triangleright$ ECs: $\mathcal{E} = \{U_1, U_2, U_3, \{s_5, s_6\}, \{s_6, s_7\}, \{s_1, s_3, s_4, s_5\}\}$
End-components: why we care

Lemma (Long-run appearance of ECs \([\text{CY95, dA97}]\))

Let \(\lambda_1 \in \Lambda_1(P)\) be an arbitrary strategy of \(P_1\). Then, we have that

\[
\mathbb{P}_{P[\lambda_1]}(\{\pi \in \text{Outs}_{P[\lambda_1]}(s_{\text{init}}) \mid \inf(\pi) \in \mathcal{E}\}) = 1.
\]

- By prefix-independence, only long-run behavior matters
- The expectation on \(P[\lambda_1]\) depends uniquely on ECs
How to satisfy the BWC problem?

- *Expected value requirement*: reach ECs with the highest achievable expectations and stay in them
  - The optimal expected value is the same everywhere inside the EC [FV97], cf. ideal case
How to satisfy the BWC problem?

- **Expected value requirement**: reach ECs with the highest achievable expectations and stay in them
  - The optimal expected value is the same everywhere inside the EC [FV97], cf. ideal case

- **Worst-case requirement**: some ECs may need to be eventually avoided because risky!
  - The “ideal cases” are ECs but not all ECs are ideal cases...
  - Need to classify the ECs
Classification of ECs

\( \exists \lambda_1 \in \Lambda_1(G \downarrow U), \ \forall \lambda_2 \in \Lambda_2(G \downarrow U), \ \forall s \in U, \ \forall \pi \in \text{Outs}_{(G \downarrow U)}(s, \lambda_1, \lambda_2), \ MP(\pi) > 0 \)
Classification of ECs

- $U \in \mathcal{W}$, the winning ECs, if $\mathcal{P}_1$ can win in $G \upharpoonright U$, from all states:

  $\exists \lambda_1 \in \Lambda_1(G \upharpoonright U), \forall \lambda_2 \in \Lambda_2(G \upharpoonright U), \forall s \in U, \forall \pi \in \text{Outs}_{(G \upharpoonright U)}(s, \lambda_1, \lambda_2), \text{MP}(\pi) > 0$

- $\mathcal{W} = \{U_1, U_3, \{s_5, s_6\}\}$

- $U_2$ losing: from state $s_1$, $\mathcal{P}_2$ can force the outcome $\pi = (s_1s_3s_4)^\omega$ of $\text{MP}(\pi) = -1/3 < 0$
Winning ECs: usefulness

Lemma (Long-run appearance of winning ECs)

Let $\lambda_f^1 \in \Lambda^F_1$ be a finite-memory strategy of $P_1$ that satisfies the BWC problem for thresholds $(0, \nu) \in \mathbb{Q}^2$. Then, we have that

$$\mathbb{P}_{\pi_{\text{init}}}^{P[\lambda_f^1]} \left( \left\{ \pi \in \text{Outs}_{P[\lambda_f^1]}(s_{\text{init}}) \mid \text{Inf}(\pi) \in \mathcal{W} \right\} \right) = 1.$$
Winning ECs: usefulness

Lemma (Long-run appearance of winning ECs)

Let \( \lambda^f_1 \in \Lambda^F_1 \) be a finite-memory strategy of \( P_1 \) that satisfies the BWC problem for thresholds \((0, \nu) \in \mathbb{Q}^2\). Then, we have that

\[
P^P_{s_{\text{init}}} \left( \left\{ \pi \in \text{Out}_{P[\lambda^f_1]}(s_{\text{init}}) \mid \text{Inf}(\pi) \in \mathcal{W} \right\} \right) = 1.
\]

▶ A good finite-memory strategy for the BWC problem should maximize the expected value achievable through winning ECs.
Winning ECs: computation

- Deciding if an EC is winning or not is in $\text{NP} \cap \text{coNP}$ (worst-case threshold problem)
- $|E| \leq 2^{|S|} \sim$ exponential # of ECs
Winning ECs: computation

- Deciding if an EC is winning or not is in $\text{NP} \cap \text{coNP}$ (worst-case threshold problem)
- $|E| \leq 2^{|S|} \sim$ exponential # of ECs
- Considering the maximal ECs does not suffice! See $U_3 \subset U_2$
Winning ECs: computation

- Deciding if an EC is winning or not is in NP \cap \text{coNP} (worst-case threshold problem)
- $|\mathcal{E}| \leq 2^{|S|} \sim$ exponential \# of ECs
- Considering the maximal ECs does not suffice! See $U_3 \subset U_2$

But,

- possible to define a recursive algorithm computing the maximal winning ECs, such that $|U_w| \leq |S|$, in NP \cap \text{coNP}.
- Uses polynomial number of of calls to
  - max. EC decomp. of sub-MDPs (each in $O(|S|^2)$ [CH12]),
  - worst-case threshold problem (NP \cap \text{coNP}).
- Critical complexity gain for the algorithm solving the BWC problem!
A natural way towards WECs

So we know we should only use WECs and we know how to play $\varepsilon$-optimally inside a WEC. What remains to settle?
A natural way towards WECs

So we know we should only use WECs and we know how to play $\varepsilon$-optimally inside a WEC. What remains to settle?

- Determine **which** WECs to reach and **how**!
A natural way towards WECs

So we know we should only use WECs and we know how to play $\varepsilon$-optimally inside a WEC. What remains to settle?

- Determine **which** WECs to reach and **how**!

- Key idea: define a **global strategy** that will go towards the highest valued WECs and avoid LECs
Global strategy via modified MDP

\[ WEC \ U_3 - E = 2 \]

\[ WEC \ U_2 - E = 3 \]

\[ LEC \ U_1 - E = 4 \]
Global strategy via modified MDP

Modify weights:

\[ \forall e = (s_1, s_2) \in E, \ w'(e) := \begin{cases} w(e) & \text{if } \exists U \in \mathcal{U}_w \text{ s.t. } \{s_1, s_2\} \subseteq U, \\ 0 & \text{otherwise}. \end{cases} \]
Global strategy via modified MDP

\[ \text{WEC } U_3 - \mathbb{E} = 2 \]

\[ \text{WEC } U_2 - \mathbb{E} = 3 \]

\[ \text{LEC } U_1 - \mathbb{E} = 0 \]

2 Memoryless optimal expectation strategy \( \lambda^e_1 \) on \( P' \)

\[ \text{the probability to be in a good WEC (here, } U_2 \text{) after } N \text{ steps tends to one when } N \to \infty \]
Global strategy via modified MDP

3 \( \lambda_1^{glb} \in \Lambda_1^{PF}(G) \):

(a) Play \( \lambda_1^e \in \Lambda_1^{PM}(G) \) for \( N \) steps.

(b) Let \( s \in S \) be the reached state.

   (b.1) If \( s \in U \in \mathcal{U}_w \), play corresponding \( \lambda_1^{cmb} \in \Lambda_1^{PF}(G) \) forever.

   (b.2) Else play \( \lambda_1^{wc} \in \Lambda_1^{PM}(G) \) forever.

\( \lambda_1^{wc} \) exists everywhere as WC losing states have been removed.

Parameter \( N \in \mathbb{N} \) can be chosen so that overall expectation is arbitrarily close to optimal in \( P' \), or equivalently, optimal for BWC strategies in \( P \).

Our algorithm computes this optimal value \( \nu^* \) and answers YES iff \( \nu^* > \nu \sim \) it is correct and complete.
BWC MP problem: bounds

- **Complexity**
  - problem in NP \(\cap\) coNP (P if MP games proved in P)
  - lower bound via reduction from MP games
BWC MP problem: bounds

- **Complexity**
  - problem in $\text{NP} \cap \text{coNP}$ ($P$ if MP games proved in $P$)
  - lower bound via reduction from MP games

- **Memory**
  - pseudo-polynomial upper bound via global strategy
  - matching lower bound via family $(G(X))_{X \in \mathbb{N}_0}$ requiring polynomial memory in $W = X + 5$ to satisfy the BWC problem for thresholds $(0, \nu \in \left]1, \frac{5}{4}\right[)$
    - need to use $(s_1, s_3)$ infinitely often for $\mathbb{E}$ but need pseudo-polynomial memory to counteract $-X$ for the WC requirement
1. Context

2. BWC Synthesis

3. Mean-Payoff

4. Shortest Path

5. Conclusion
Shortest path

- Strictly positive integer weights, $w : E \rightarrow \mathbb{N}_0$
- $\mathcal{P}_1$ wants to minimize its total cost up to target
  ▶ inequalities are reversed
Shortest path

- Strictly positive integer weights, $w : E \to \mathbb{N}_0$
- $\mathcal{P}_1$ wants to minimize its total cost up to target
  - inequalities are reversed

<table>
<thead>
<tr>
<th>worst-case complexity</th>
<th>expected value</th>
<th>BWC</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P$</td>
<td>$P$</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 $G = (G, S_1, S_2)$, $G = (S, E, w)$, $T = \{s_3\}$, $\mathcal{M}(\lambda_2^{\text{stoch}})$, $\mu = 8$, and $\nu \in \mathbb{Q}$
Pseudo-polynomial algorithm: sketch

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

Here, $\nu^* = \frac{9}{2}$.
Pseudo-polynomial algorithm: sketch
Pseudo-polynomial algorithm: sketch

Here, $\nu^* = \frac{9}{2}$.
Pseudo-polynomial algorithm: sketch

\[ s_1 \xrightarrow{1} s_2 \xleftarrow{\frac{1}{2}} s_3 \]

\[ s_3 \xrightarrow{5} s_1 \xrightarrow{1} s_2 \]

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

\[
\begin{align*}
\nu^* &= \frac{9}{2} \\

s_1, 0 &\rightarrow s_2, 1 \\
\frac{1}{2} &\rightarrow 1 \\
5 &\rightarrow 1 \\

s_3, 5 &\rightarrow s_2, 1 \\
1 &\rightarrow \frac{1}{2} \\
\frac{1}{2} &\rightarrow 1 \\

s_2, 1 &\rightarrow s_1, 2 \\
1 &\rightarrow 1 \\
5 &\rightarrow 1 \\

s_3, 2 &\rightarrow s_1, 4 \\
5 &\rightarrow 1 \\
1 &\rightarrow \frac{1}{2} \\
s_3, 7 &\rightarrow s_2, 3 \\
1 &\rightarrow 1 \\

s_3, 7 &\rightarrow s_2, 5 \\
5 &\rightarrow 1 \\
1 &\rightarrow \frac{1}{2} \\

s_3, T &\rightarrow s_3, 3 \\
1 &\rightarrow 1 \\
5 &\rightarrow \frac{1}{2} \\

\end{align*}
\]
Pseudo-polynomial algorithm: sketch

\[ \nu^* = \frac{9}{2} \]

\[ s_1, 0 \rightarrow s_2, 1 \rightarrow s_1, 2 \rightarrow s_2, 3 \rightarrow s_1, 4 \rightarrow s_2, 5 \rightarrow s_1, 6 \]

\[ s_3, 5 \rightarrow s_3, 2 \rightarrow s_3, 7 \rightarrow s_3, 4 \rightarrow s_3, T \rightarrow s_3, 6 \]
Pseudo-polynomial algorithm: sketch

Here, \( \nu^* = \frac{9}{2} \)
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

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

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

6. Compute memoryless optimal expectation strategy

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

Here, $\nu^* = 9/2$
Memory bounds

- Upper bound provided by synthesized strategy

- Lower bound given by family of games \( (G(\mu))_{\mu \in \{13+k \cdot 4 \mid k \in \mathbb{N}\}} \) requiring memory linear in \( \mu \)

\[ \sim \text{ play } (s_1, s_2) \text{ exactly } \left\lfloor \frac{\mu}{4} \right\rfloor \text{ times and then switch to } (s_1, s_3) \text{ to minimize expected value while ensuring the worst-case} \]
Complexity lower bound: NP-hardness

- Truly-polynomial algorithm very unlikely...
- Reduction from the $K^{th}$ largest subset problem
  - commonly thought to be outside NP as natural certificates are larger than polynomial [JK78, GJ79]
Complexity lower bound: NP-hardness

- Truly-polynomial algorithm very unlikely...
- Reduction from the $K^{th}$ largest subset problem
  - commonly thought to be outside NP as natural certificates are larger than polynomial [JK78, GJ79]

$K^{th}$ largest subset problem

Given a finite set $A$, a size function $h: A \rightarrow \mathbb{N}_0$ assigning strictly positive integer values to elements of $A$, and two naturals $K, L \in \mathbb{N}$, decide if there exist $K$ distinct subsets $C_i \subseteq A$, $1 \leq i \leq K$, such that $h(C_i) = \sum_{a \in C_i} h(a) \leq L$ for all $K$ subsets.

- Build a game composed of two gadgets
Random subset selection gadget

- Stochastically generates paths representing subsets of $A$: an element is selected in the subset if the upper edge is taken when leaving the corresponding state

- All subsets are equiprobable
Choice gadget

▷ $s_e$ leads to lower expected values but may be dangerous for the worst-case requirement

▷ $s_{wc}$ is always safe but induces an higher expected cost
There exist (non-trivial) values for thresholds and weights so that

(i) an optimal (i.e., minimizing the expectation while guaranteeing a given worst-case threshold) strategy for $P_1$ consists in choosing state $s_e$ only when the randomly generated subset $C \subseteq A$ satisfies $h(C) \leq L$;

(ii) this strategy satisfies the BWC problem if and only if there exist $K$ distinct subsets that verify this bound.
1. Context

2. BWC Synthesis

3. Mean-Payoff

4. Shortest Path

5. Conclusion
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 and M. Henzinger.
An $\mathcal{O}(n^2)$ time algorithm for alternating Büchi games.

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

C. Courcoubetis and M. Yannakakis.
The complexity of probabilistic verification.

L. de Alfaro.
Formal verification of probabilistic systems.

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.
References III

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.

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 ∩ 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.
References IV

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.