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

UMONS Université de Mons
ULB Université Libre de Bruxelles

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*.

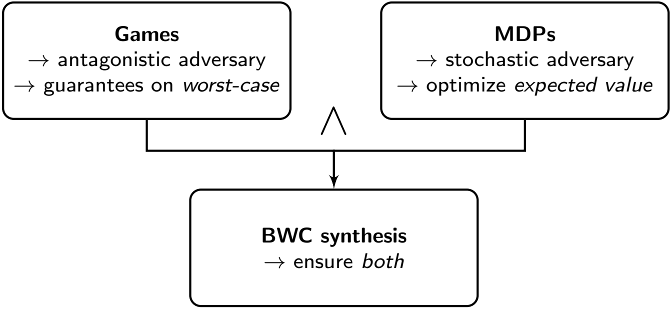
- Several ways to look at the interactions, and in particular, *the nature of the environment*.

The talk in two slides (2/2)

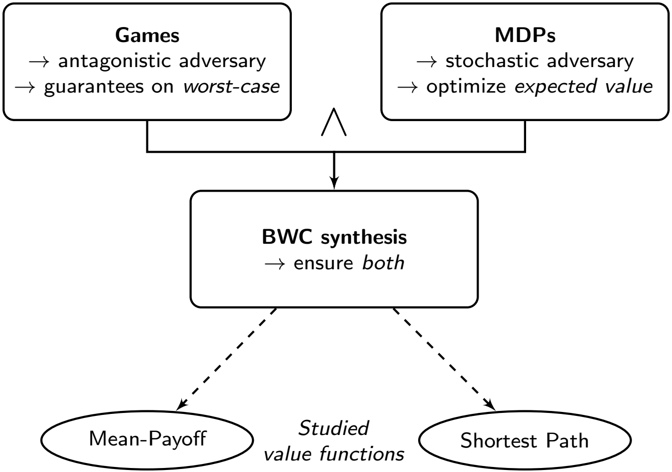
Games
$\rightarrow$ antagonistic adversary
$\rightarrow$ guarantees on *worst-case*

MDPs
$\rightarrow$ stochastic adversary
$\rightarrow$ optimize *expected value*

The talk in two slides (2/2)

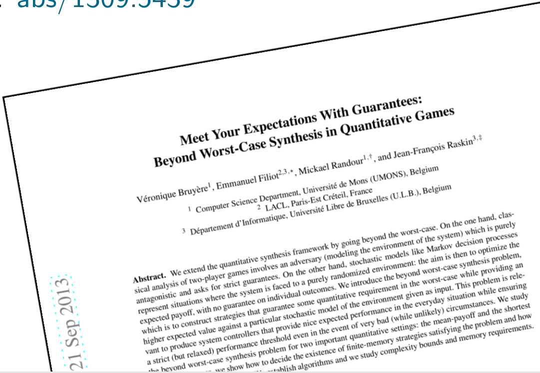


The talk in two slides (2/2)



Advertisement

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


21 Sep 2013

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

Véronique Bruyère[1], Emmanuel Filiot[2,3,⋆], Mickael Randour[1,†], and Jean-François Raskin[3,‡]

[1] Computer Science Department, Université de Mons (UMONS), Belgium
[2] LACL, Paris-Est Créteil, France
[3] Département d'Informatique, Université Libre de Bruxelles (U.L.B.), Belgium

Abstract. We extend the quantitative synthesis framework by going beyond the worst-case. On the one hand, classical analysis of two-player games involves an adversary (modeling the environment of the system) which is purely antagonistic and asks for strict guarantees. On the other hand, stochastic models like Markov decision processes represent situations where the system is faced to a purely randomized environment: the aim is then to optimize the expected payoff, with no guarantee on individual outcomes. We introduce the beyond worst-case synthesis problem, which is to construct strategies that guarantee some quantitative requirement in the worst-case while providing an higher expected value against a particular stochastic model of the environment given as input. This problem is relevant to produce system controllers that provide nice expected performance in the everyday situation while ensuring a strict (but relaxed) performance threshold even in the event of very bad (while unlikely) circumstances. We study the beyond worst-case synthesis problem for two important quantitative settings: the mean-payoff and the shortest ... we show how to decide the existence of finite-memory strategies satisfying the problem and how ... establish algorithms and we study complexity bounds and memory requirements.

1 Context

2 BWC Synthesis

3 Mean-Payoff

4 Shortest Path

5 Conclusion

1 Context

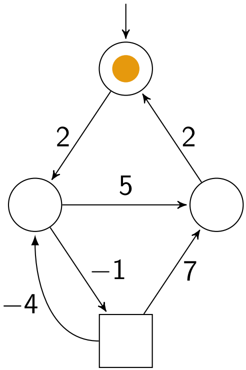
2 BWC Synthesis

3 Mean-Payoff

4 Shortest Path

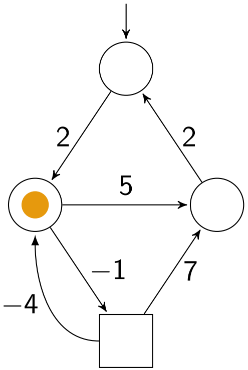
5 Conclusion

Quantitative games on graphs



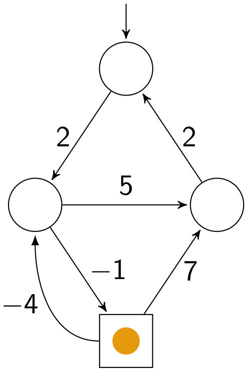
- Graph $\mathcal{G} = (S, E, w)$ with $w\colon E \to \mathbb{Z}$
- Two-player *game* $G = (\mathcal{G}, S_1, S_2)$
  - ▷ $\mathcal{P}_1$ states = ○
  - ▷ $\mathcal{P}_2$ states = □
- Plays have values
  - ▷ $f\colon \text{Plays}(\mathcal{G}) \to \mathbb{R} \cup \{-\infty, \infty\}$
- Players follow *strategies*
  - ▷ $\lambda_i\colon \text{Prefs}_i(G) \to \mathcal{D}(S)$
  - ▷ Finite memory ⇒ stochastic output Moore machine $\mathcal{M}(\lambda_i) = (\text{Mem}, \text{m}_0, \alpha_\text{u}, \alpha_\text{n})$

Quantitative games on graphs



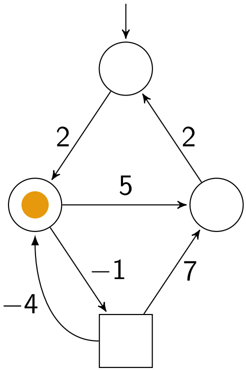
- Graph $\mathcal{G} = (S, E, w)$ with $w \colon E \to \mathbb{Z}$
- Two-player *game* $G = (\mathcal{G}, S_1, S_2)$
  - ▷ $\mathcal{P}_1$ states = ○
  - ▷ $\mathcal{P}_2$ states = □
- Plays have values
  - ▷ $f \colon \mathrm{Plays}(\mathcal{G}) \to \mathbb{R} \cup \{-\infty, \infty\}$
- Players follow *strategies*
  - ▷ $\lambda_i \colon \mathrm{Prefs}_i(G) \to \mathcal{D}(S)$
  - ▷ Finite memory ⇒ stochastic output Moore machine $\mathcal{M}(\lambda_i) = (\mathrm{Mem}, \mathrm{m}_0, \alpha_{\mathrm{u}}, \alpha_{\mathrm{n}})$

Quantitative games on graphs



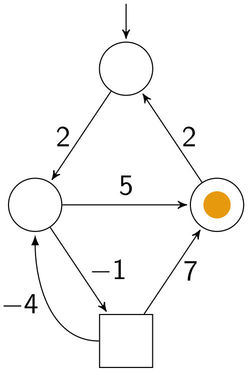
- Graph $\mathcal{G} = (S, E, w)$ with $w\colon E \to \mathbb{Z}$
- Two-player *game* $G = (\mathcal{G}, S_1, S_2)$
  - ▷ $\mathcal{P}_1$ states = ◯
  - ▷ $\mathcal{P}_2$ states = □
- Plays have values
  - ▷ $f\colon \text{Plays}(\mathcal{G}) \to \mathbb{R} \cup \{-\infty, \infty\}$
- Players follow *strategies*
  - ▷ $\lambda_i\colon \text{Prefs}_i(G) \to \mathcal{D}(S)$
  - ▷ Finite memory ⇒ stochastic output Moore machine $\mathcal{M}(\lambda_i) = (\text{Mem}, \text{m}_0, \alpha_\text{u}, \alpha_\text{n})$

Quantitative games on graphs



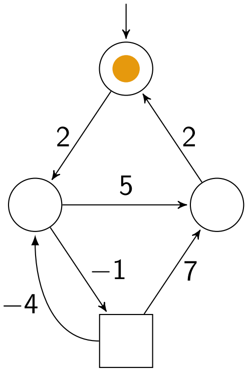
- Graph $\mathcal{G} = (S, E, w)$ with $w \colon E \to \mathbb{Z}$
- Two-player *game* $G = (\mathcal{G}, S_1, S_2)$
  - ▷ $\mathcal{P}_1$ states = ◯
  - ▷ $\mathcal{P}_2$ states = □
- Plays have values
  - ▷ $f \colon \text{Plays}(\mathcal{G}) \to \mathbb{R} \cup \{-\infty, \infty\}$
- Players follow *strategies*
  - ▷ $\lambda_i \colon \text{Prefs}_i(G) \to \mathcal{D}(S)$
  - ▷ Finite memory ⇒ stochastic output Moore machine $\mathcal{M}(\lambda_i) = (\text{Mem}, \text{m}_0, \alpha_\text{u}, \alpha_\text{n})$

Quantitative games on graphs



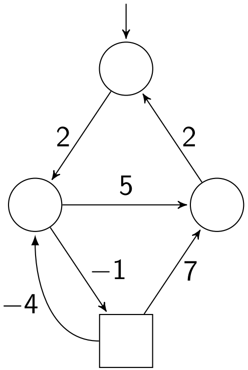
- Graph $\mathcal{G} = (S, E, w)$ with $w \colon E \to \mathbb{Z}$
- Two-player *game* $G = (\mathcal{G}, S_1, S_2)$
  - ▷ $\mathcal{P}_1$ states = ○
  - ▷ $\mathcal{P}_2$ states = □
- Plays have values
  - ▷ $f \colon \text{Plays}(\mathcal{G}) \to \mathbb{R} \cup \{-\infty, \infty\}$
- Players follow *strategies*
  - ▷ $\lambda_i \colon \text{Prefs}_i(G) \to \mathcal{D}(S)$
  - ▷ Finite memory ⇒ stochastic output Moore machine $\mathcal{M}(\lambda_i) = (\text{Mem}, \text{m}_0, \alpha_\text{u}, \alpha_\text{n})$

Quantitative games on graphs



- Graph $\mathcal{G} = (S, E, w)$ with $w \colon E \to \mathbb{Z}$
- Two-player *game* $G = (\mathcal{G}, S_1, S_2)$
  - ▷ $\mathcal{P}_1$ states = ◯
  - ▷ $\mathcal{P}_2$ states = □
- Plays have values
  - ▷ $f \colon \text{Plays}(\mathcal{G}) \to \mathbb{R} \cup \{-\infty, \infty\}$
- Players follow *strategies*
  - ▷ $\lambda_i \colon \text{Prefs}_i(G) \to \mathcal{D}(S)$
  - ▷ Finite memory ⇒ stochastic output Moore machine $\mathcal{M}(\lambda_i) = (\text{Mem}, \text{m}_0, \alpha_\text{u}, \alpha_\text{n})$

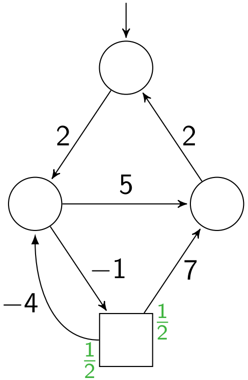
Quantitative games on graphs



Then, $(2, 5, 2)^\omega$

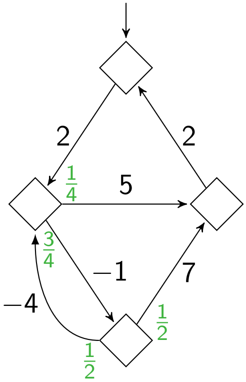
- Graph $\mathcal{G} = (S, E, w)$ with $w \colon E \to \mathbb{Z}$
- Two-player *game* $G = (\mathcal{G}, S_1, S_2)$
  - ▷ $\mathcal{P}_1$ states = ○
  - ▷ $\mathcal{P}_2$ states = □
- Plays have values
  - ▷ $f \colon \text{Plays}(\mathcal{G}) \to \mathbb{R} \cup \{-\infty, \infty\}$
- Players follow *strategies*
  - ▷ $\lambda_i \colon \text{Prefs}_i(G) \to \mathcal{D}(S)$
  - ▷ Finite memory ⇒ stochastic output Moore machine $\mathcal{M}(\lambda_i) = (\text{Mem}, \text{m}_0, \alpha_\text{u}, \alpha_\text{n})$

Markov decision processes



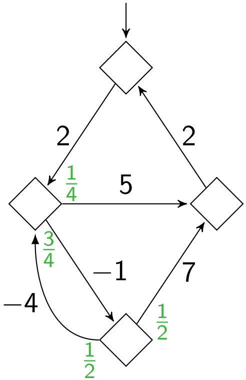
- MDP $P = (\mathcal{G}, S_1, S_\Delta, \Delta)$ with $\Delta\colon S_\Delta \to \mathcal{D}(S)$
  - ▷ $\mathcal{P}_1$ states = ◯
  - ▷ stochastic states = □
- MDP = game + strategy of $\mathcal{P}_2$
  - ▷ $P = G[\lambda_2]$

Markov chains



- MC $M = (\mathcal{G}, \delta)$ with $\delta\colon S \to \mathcal{D}(S)$
- MC = MDP + strategy of $\mathcal{P}_1$
  = game + both strategies
  - ▷ $M = P[\lambda_1] = G[\lambda_1, \lambda_2]$

Markov chains



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

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 $= \mathcal{P}_1$
  - ▷ whatever the actions of its **environment**
- The environment can be seen as
  - ▷ *antagonistic*
    - two-player game, *worst-case* threshold problem for $\mu \in \mathbb{Q}$
    - $\exists ?\, \lambda_1 \in \Lambda_1,\ \forall\, \lambda_2 \in \Lambda_2,\ \forall\, \pi \in \mathsf{Outs}_G(s_{\mathsf{init}}, \lambda_1, \lambda_2),\ f(\pi) \geq \mu$

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 $\mu \in \mathbb{Q}$
    - $\exists ?\, \lambda_1 \in \Lambda_1, \forall\, \lambda_2 \in \Lambda_2, \forall\, \pi \in \mathsf{Outs}_G(s_{\mathsf{init}}, \lambda_1, \lambda_2),\ f(\pi) \geq \mu$
  - ▷ *fully stochastic*
    - MDP, *expected value* threshold problem for $\nu \in \mathbb{Q}$
    - $\exists ?\, \lambda_1 \in \Lambda_1,\ \mathbb{E}^{P[\lambda_1]}_{s_{\mathsf{init}}}(f) \geq \nu$

1 Context

2 BWC Synthesis

3 Mean-Payoff

4 Shortest Path

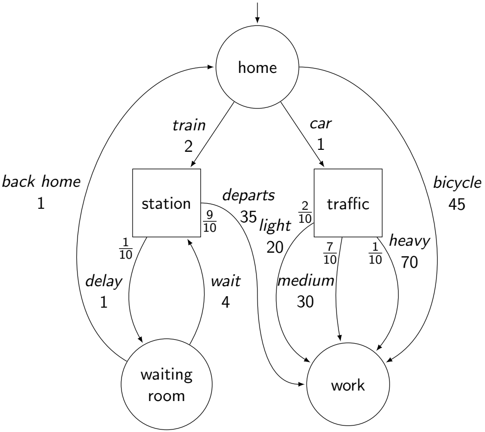
5 Conclusion

What if you want both?

In practice, we want both

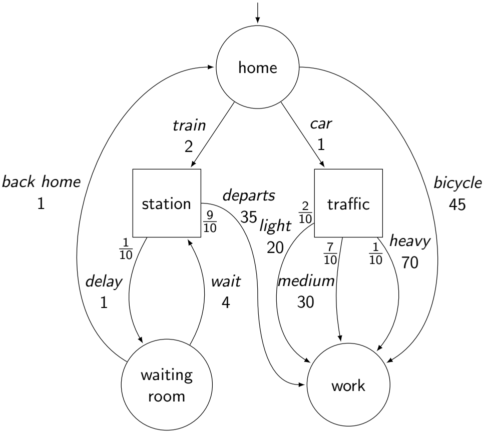
1. nice expected performance in the everyday situation,
2. strict (but relaxed) performance guarantees even in the event of very bad circumstances.

Example: going to work



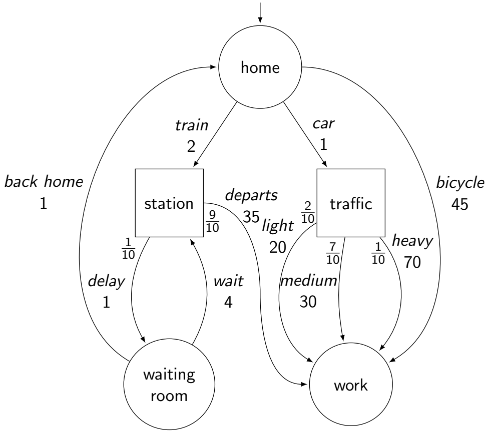
- Weights = minutes
- Goal: *minimize our expected time* to reach "work"
- **But**, important meeting in one hour! Requires *strict guarantees* on the worst-case reaching time.

Example: going to work



- Optimal expectation strategy: take the car.
  - $\mathbb{E} = 33$, $\text{WC} = 71 > 60$.
- Optimal worst-case strategy: bicycle.
  - $\mathbb{E} = \text{WC} = 45 < 60$.

Example: going to work



- ▷ Optimal expectation strategy: take the car.
  - $\mathbb{E} = 33$, $\text{WC} = 71 > 60$.
- ▷ Optimal worst-case strategy: bicycle.
  - $\mathbb{E} = \text{WC} = 45 < 60$.
- ▷ **Sample BWC strategy**: try train up to 3 delays then switch to bicycle.
  - $\mathbb{E} \approx 37.56$, $\text{WC} = 59 < 60$.
  - Optimal $\mathbb{E}$ 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 $\lambda_2^{\text{stoch}} \in \Lambda_2^F$ of the adversary, represented by a stochastic Moore machine, a measurable value function $f\colon \mathsf{Plays}(\mathcal{G}) \rightarrow \mathbb{R} \cup \{-\infty, \infty\}$, 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 $\lambda_1 \in \Lambda_1^F$ such that

$$\begin{cases} \forall\, \lambda_2 \in \Lambda_2,\ \forall\, \pi \in \mathsf{Outs}_G(s_{\text{init}}, \lambda_1, \lambda_2),\ f(\pi) > \mu & (1) \\ \mathbb{E}_{s_{\text{init}}}^{G[\lambda_1, \lambda_2^{\text{stoch}}]}(f) > \nu & (2) \end{cases}$$

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

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 $s_{\mathsf{init}} \in S$, a finite-memory stochastic model $\lambda_2^{\mathsf{stoch}} \in \Lambda_2^F$ of the adversary, represented by a stochastic Moore machine, a measurable value function $f\colon \mathsf{Plays}(\mathcal{G}) \to \mathbb{R} \cup \{-\infty, \infty\}$, 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

$$\begin{cases} \forall\, \lambda_2 \in \Lambda_2, \forall\, \pi \in \mathsf{Outs}_G(s_{\mathsf{init}}, \lambda_1, \lambda_2),\ f(\pi) > \mu & (1) \\ \mathbb{E}_{s_{\mathsf{init}}}^{G[\lambda_1, \lambda_2^{\mathsf{stoch}}]}(f) > \nu & (2) \end{cases}$$

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

1 Context

2 BWC Synthesis

3 Mean-Payoff

4 Shortest Path

5 Conclusion

Mean-payoff value function

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

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

Mean-payoff value function

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

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

	worst-case	expected value	**BWC**
complexity	NP ∩ coNP	P	**NP ∩ coNP**
memory	memoryless	memoryless	**pseudo-polynomial**

▷ [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

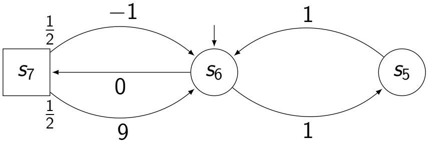
- ▷ Classical worst-case and expected value results and algorithms as *nuts and bolts*
- ▷ *Screw them together* in an adequate way

Three key ideas

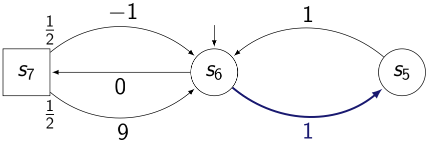
1. To characterize the expected value, look at *end-components* (ECs)
2. *Winning ECs* vs. *losing ECs*: the latter must be avoided to preserve the worst-case requirement!
3. *Inside a WEC*, we have an interesting way to play...

$\Longrightarrow$ **Let's focus on an ideal case**

Inside a WEC



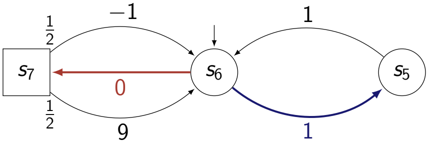
Inside a WEC



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



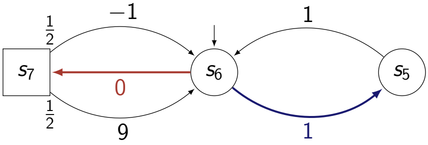
Game interpretation

- ▷ 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

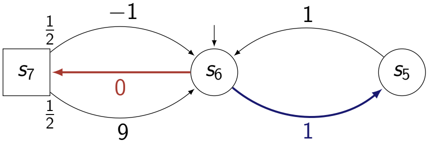
- ▷ Memoryless optimal expected value strategy $\lambda_1^{e} \in \Lambda_1^{PM}(P)$ achieves $\nu^* = 2$

A cornerstone of our approach



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

A cornerstone of our approach



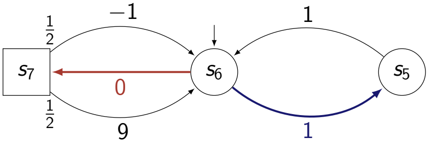
BWC problem: **what kind of thresholds $(\mu = 0, \nu)$ can we achieve?**

Key result

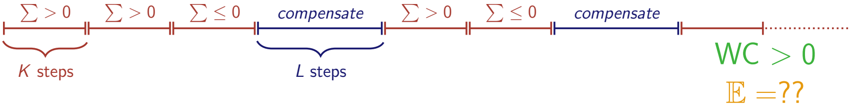
For all $\varepsilon > 0$, there exists a finite-memory strategy of $\mathcal{P}_1$ that satisfies the BWC problem for the thresholds pair $(0, \nu^* - \varepsilon)$.

- ▷ We can be **arbitrarily close to the optimal expectation** while ensuring the worst-case

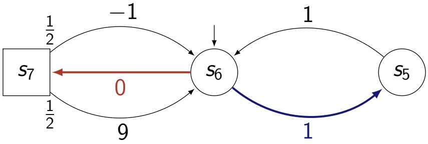
Combined strategy



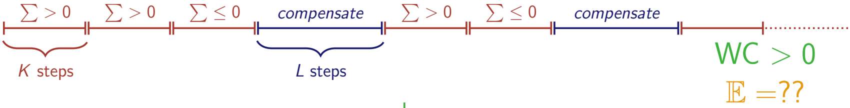
Outcomes of the form



Combined strategy



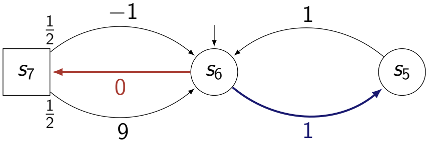
Outcomes of the form



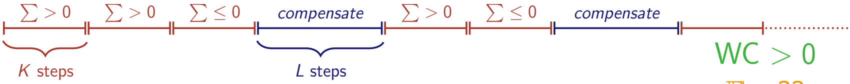
What we want

$K, L \to \infty$

Combined strategy



Outcomes of the form



$\mathrm{WC} > 0$

$\mathbb{E} = ??$

What we want

$K, L \to \infty$

$L = \mathrm{linear}(K)$

$\mathbb{P}(\longmapsto) \to 0$ exp. fast! [Tra09, GO02]

$\mathbb{E} = \nu^* = 2$

The ideal case: wrap-up

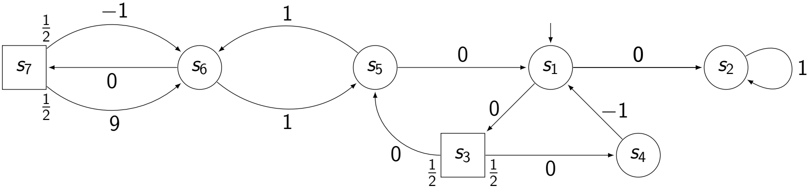
The combined strategy works in any subgame such that

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.

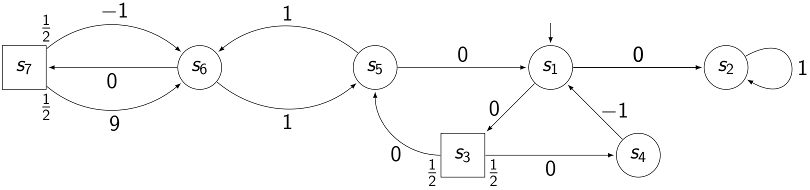
Zooming out



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

End-components: what they are

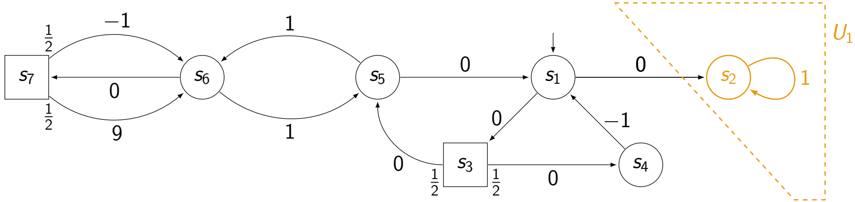


An **EC** of the MDP $P = G[\lambda_2^{\text{stoch}}]$ is a subgraph in which $\mathcal{P}_1$ can ensure to stay despite stochastic states [dA97], i.e., a set $U \subseteq S$ s.t.

(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.

End-components: what they are



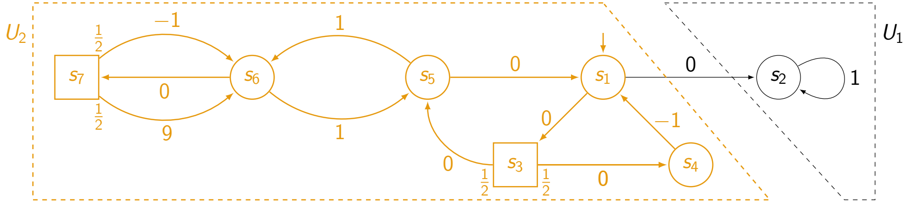
An **EC** of the MDP $P = G[\lambda_2^{\mathsf{stoch}}]$ is a subgraph in which $\mathcal{P}_1$ can ensure to stay despite stochastic states [dA97], i.e., a set $U \subseteq S$ s.t.

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

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

▷ ECs: $\mathcal{E} = \{U_1$

End-components: what they are



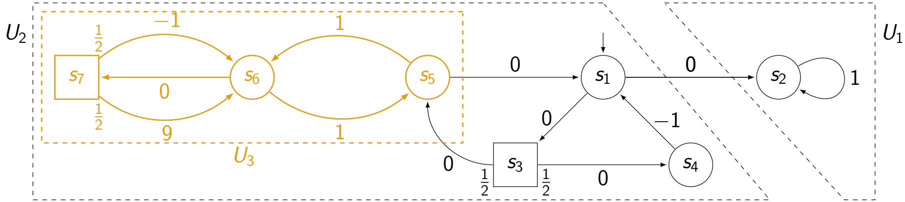
An **EC** of the MDP $P = G[\lambda_2^{\text{stoch}}]$ is a subgraph in which $\mathcal{P}_1$ can ensure to stay despite stochastic states [dA97], i.e., a set $U \subseteq S$ s.t.

(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



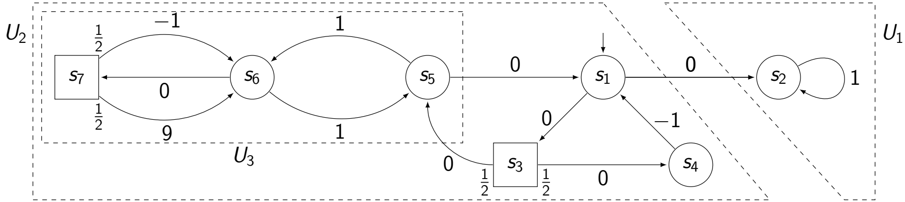
An **EC** of the MDP $P = G[\lambda_2^{\mathsf{stoch}}]$ is a subgraph in which $\mathcal{P}_1$ can ensure to stay despite stochastic states [dA97], i.e., a set $U \subseteq S$ s.t.

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

(ii) $\forall\, s \in U \cap S_\Delta$, $\mathsf{Supp}(\Delta(s)) \subseteq U$, 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



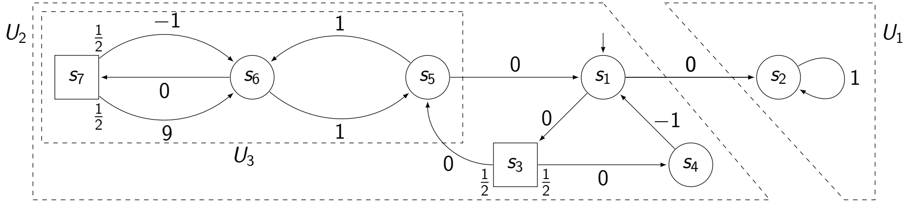
An **EC** of the MDP $P = G[\lambda_2^{\text{stoch}}]$ is a subgraph in which $\mathcal{P}_1$ can ensure to stay despite stochastic states [dA97], i.e., a set $U \subseteq S$ s.t.

(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: $\mathcal{E} = \{U_1, U_2, U_3, \{s_5, s_6\}, \{s_6, s_7\}, \{s_1, s_3, s_4, s_5\}\}$

End-components: what they are



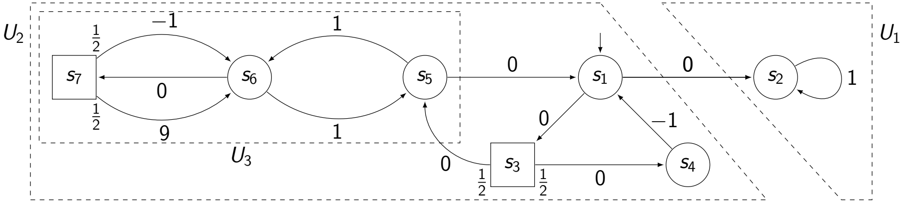
An **EC** of the MDP $P = G[\lambda_2^{\text{stoch}}]$ is a subgraph in which $\mathcal{P}_1$ can ensure to stay despite stochastic states [dA97], i.e., a set $U \subseteq S$ s.t.

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

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

▷ ECs: $\mathcal{E} = \{U_1, U_2, U_3, \{s_5, s_6\}, \{s_6, s_7\}, \{s_1, s_3, s_4, s_5\}\}$

End-components: why we care



Lemma (Long-run appearance of ECs [CY95, dA97])

Let $\lambda_1 \in \Lambda_1(P)$ *be an* **arbitrary strategy** *of* $\mathcal{P}_1$*. Then, we have that*

$$\mathbb{P}^{P[\lambda_1]}_{s_{\mathsf{init}}}\left(\{\pi \in \mathsf{Outs}_{P[\lambda_1]}(s_{\mathsf{init}}) \mid \mathsf{Inf}(\pi) \in \mathcal{E}\}\right) = 1.$$

- ▷ By prefix-independence, only long-run behavior matters
- ▷ **The expectation on $P[\lambda_1]$ depends uniquely on ECs**

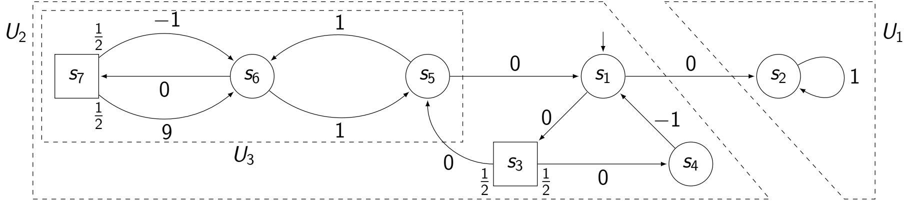
How to satisfy the BWC problem?

- *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

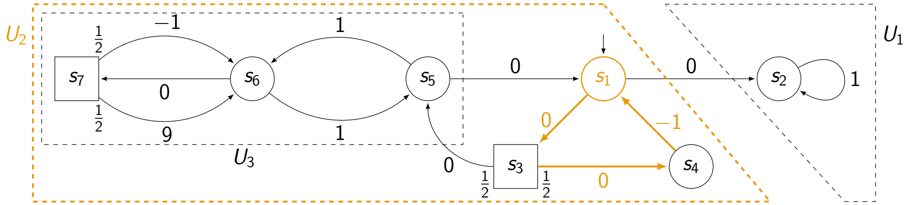
Classification of ECs



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

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

Classification of ECs



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

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

- $\mathcal{W} = \{U_1, U_3, \{s_5, s_6\}\}$
- U_2 **losing**: from state s_1, $\mathcal{P}_2$ can force the outcome $\pi = (s_1 s_3 s_4)^\omega$ of $\mathsf{MP}(\pi) = -1/3 < 0$

Winning ECs: usefulness

Lemma (Long-run appearance of winning ECs)

Let $\lambda_1^f \in \Lambda_1^F$ *be a* **finite-memory** *strategy of* $\mathcal{P}_1$ *that* **satisfies** *the BWC problem for thresholds* $(0, \nu) \in \mathbb{Q}^2$*. Then, we have that*

$$\mathbb{P}_{s_{\mathsf{init}}}^{P[\lambda_1^f]}\left(\left\{\pi \in \mathsf{Outs}_{P[\lambda_1^f]}(s_{\mathsf{init}}) \mid \mathsf{Inf}(\pi) \in \mathcal{W}\right\}\right) = 1.$$

Winning ECs: usefulness

Lemma (Long-run appearance of winning ECs)

Let $\lambda_1^f \in \Lambda_1^F$ *be a* **finite-memory** *strategy of* $\mathcal{P}_1$ *that* **satisfies** *the BWC problem for thresholds* $(0, \nu) \in \mathbb{Q}^2$. *Then, we have that*

$$\mathbb{P}^{P[\lambda_1^f]}_{s_{\mathsf{init}}}\left(\left\{\pi \in \mathsf{Outs}_{P[\lambda_1^f]}(s_{\mathsf{init}}) \mid \mathsf{Inf}(\pi) \in \mathcal{W}\right\}\right) = 1.$$

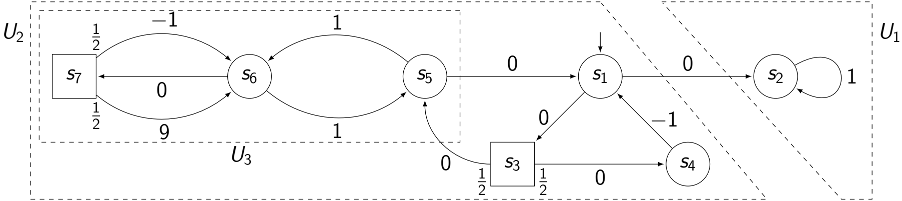
▷ A good finite-memory strategy for the BWC problem should *maximize* the expected value achievable through *winning* ECs

Winning ECs: computation

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

Winning ECs: computation

- ▷ Deciding if an EC is winning or not is in NP $\cap$ coNP (worst-case threshold problem)
- ▷ $|\mathcal{E}| \leq 2^{|S|} \rightsquigarrow$ 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 $\cap$ coNP (worst-case threshold problem)
- $|\mathcal{E}| \leq 2^{|S|} \rightsquigarrow$ 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 $|\mathcal{U}_{\mathrm{W}}| \leq |S|$, in NP $\cap$ coNP.
- Uses polynomial number of of calls to
  - max. EC decomp. of sub-MDPs (each in $\mathcal{O}(|S|^2)$ [CH12]),
  - worst-case threshold problem (NP $\cap$ 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 ε-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 ε-optimally inside a WEC. *What remains to settle?*

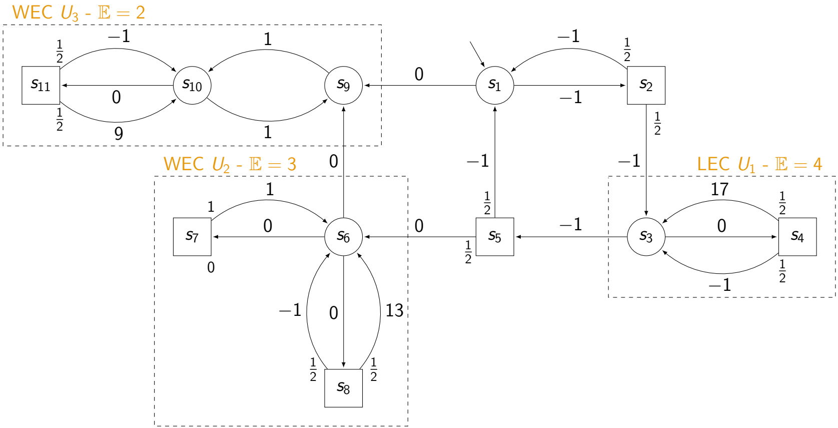
▷ Determine **which** WECs to reach and **how**!

A natural way towards WECs

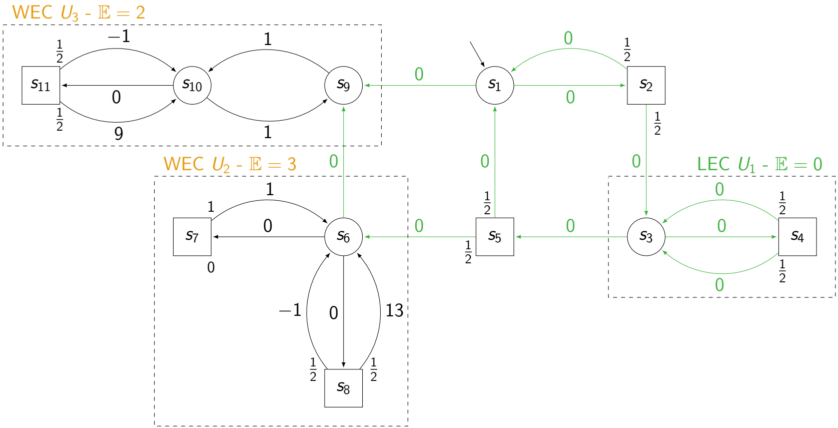
So we know we should only use WECs and we know how to play ε-optimally inside a WEC. *What remains to settle?*

- ▷ 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



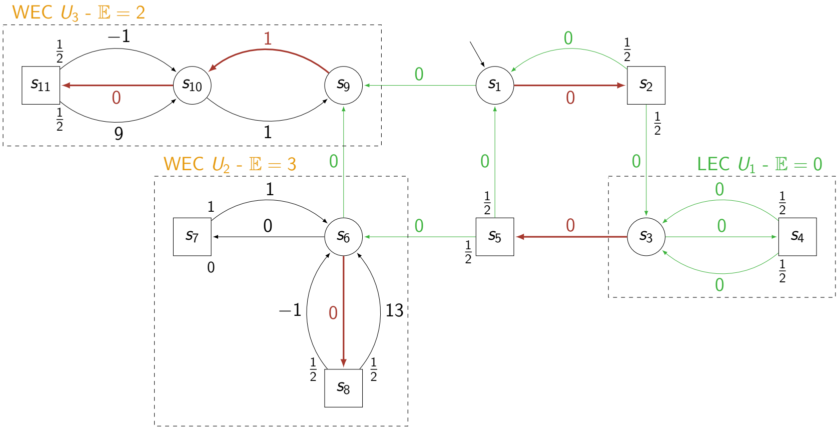
Global strategy via modified MDP



1. Modify weights:

$$\forall\, e = (s_1, s_2) \in E,\ w'(e) := \begin{cases} w(e) \text{ if } \exists\, U \in \mathcal{U}_{\mathrm{w}} \text{ s.t. } \{s_1, s_2\} \subseteq U, \\ 0 \text{ otherwise.} \end{cases}$$

Global strategy via modified MDP



2 Memoryless optimal expectation strategy λ_1^e on P'

- the probability to be in a good WEC (here, U_2) after N 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 $s \in U \in \mathcal{U}_{\mathrm{W}}$, play corresponding $\lambda_1^{cmb} \in \Lambda_1^{PF}(G)$ forever.
     - $(b.2)$ Else play $\lambda_1^{wc} \in \Lambda_1^{PM}(G)$ forever.

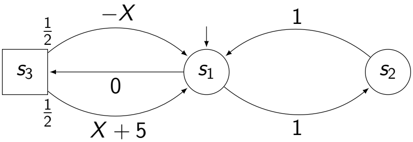
- ▷ λ_1^{wc} exists everywhere as WC losing states have been removed
- ▷ Parameter $N \in \mathbb{N}$ can be chosen so that overall expectation is arbitrarily close to optimal in P', or equivalently, optimal for BWC strategies in P
- ▷ Our algorithm computes this optimal value ν^* and answers $\textsc{Yes}$ iff $\nu^* > \nu \rightsquigarrow$ it is *correct* and *complete*

BWC MP problem: bounds

- *Complexity*
  - ▷ problem in NP ∩ coNP (P if MP games proved in P)
  - ▷ lower bound via reduction from MP games

BWC MP problem: bounds

- *Complexity*
  - ▷ problem in NP ∩ 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 $(0, \nu \in]1, 5/4[)$
    - ⇝ need to use (s_1, s_3) infinitely often for $\mathbb{E}$ but need pseudo-poly. 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, $w \colon E \to \mathbb{N}_0$
- $\mathcal{P}_1$ wants to minimize its total cost up to target
  - ▷ inequalities are reversed

Shortest path

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

	worst-case	expected value	**BWC**
complexity	P	P	**pseudo-poly. / NP-hard**
memory	memoryless	memoryless	**pseudo-poly.**

▷ [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

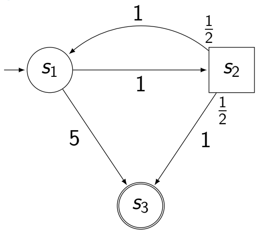
Useful observation

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:

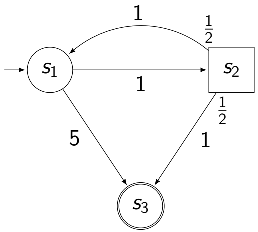
1. represent all WC winning strategies,
2. optimize the expected value within those strategies.

Pseudo-polynomial algorithm: sketch



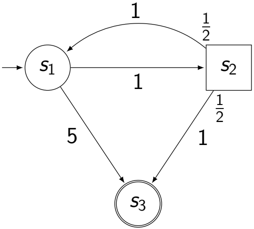
1. Start from $G = (\mathcal{G}, S_1, S_2)$, $\mathcal{G} = (S, E, w)$, $T = \{s_3\}$, $\mathcal{M}(\lambda_2^{\text{stoch}})$, $\mu = 8$, and $\nu \in \mathbb{Q}$

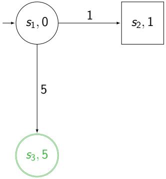
Pseudo-polynomial algorithm: sketch



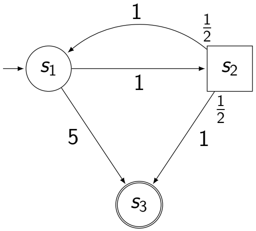
1. Start from $G = (\mathcal{G}, S_1, S_2)$, $\mathcal{G} = (S, E, w)$, $T = \{s_3\}$, $\mathcal{M}(\lambda_2^{\mathsf{stoch}})$, $\mu = 8$, and $\nu \in \mathbb{Q}$
2. Build G' by unfolding $\mathcal{G}$, tracking the current sum *up to the worst-case threshold* μ, and integrating it in the states of $\mathcal{G}'$.

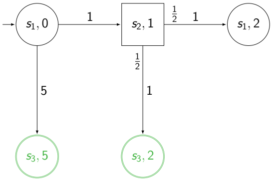
Pseudo-polynomial algorithm: sketch



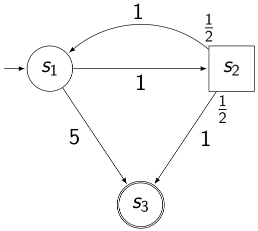


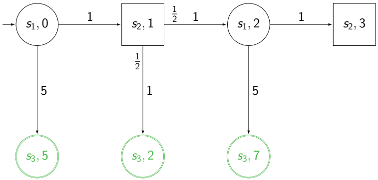
Pseudo-polynomial algorithm: sketch



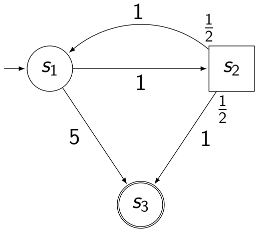


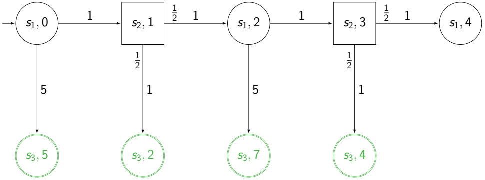
Pseudo-polynomial algorithm: sketch



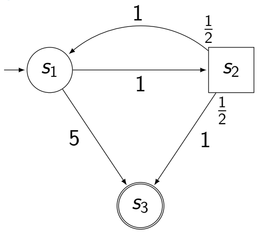


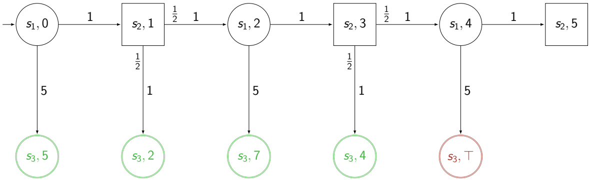
Pseudo-polynomial algorithm: sketch



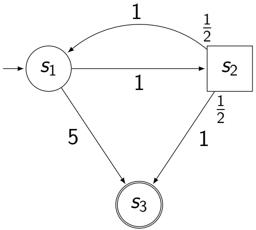


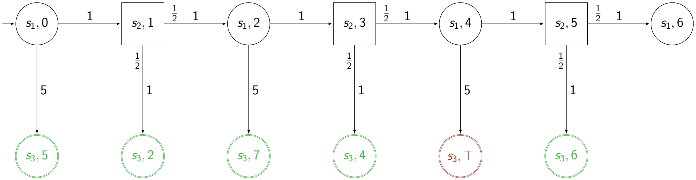
Pseudo-polynomial algorithm: sketch



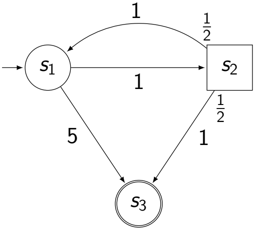


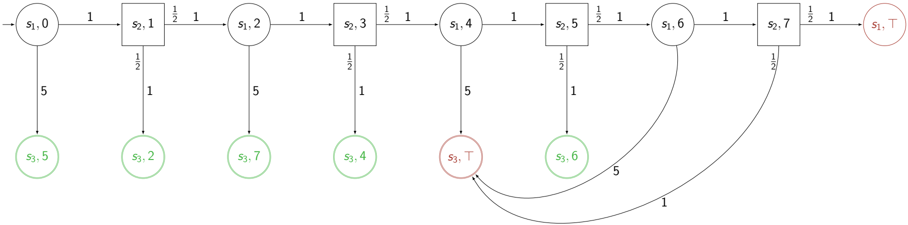
Pseudo-polynomial algorithm: sketch





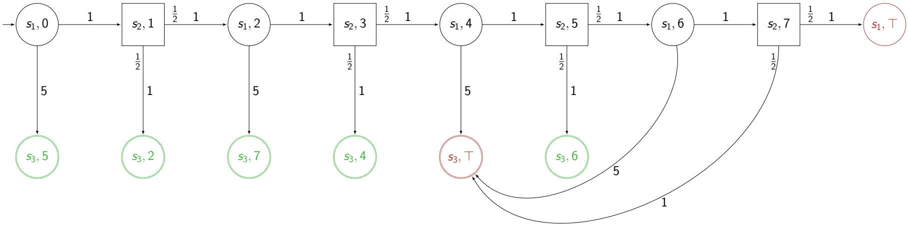
Pseudo-polynomial algorithm: sketch





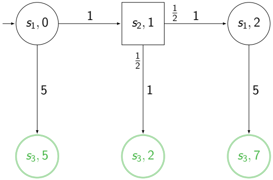
Pseudo-polynomial algorithm: sketch

3. Compute R, the attractor of T with cost $< \mu = 8$
4. Consider $G_\mu = G' \downharpoonright R$



Pseudo-polynomial algorithm: sketch

3. Compute R, the attractor of T with cost $< \mu = 8$
4. Consider $G_\mu = G' \downharpoonleft R$

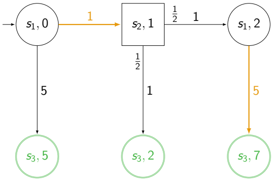


Pseudo-polynomial algorithm: sketch

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

6 Compute memoryless optimal expectation strategy

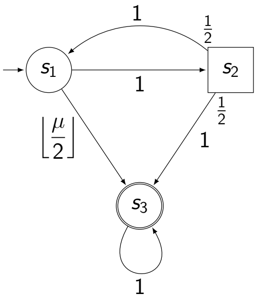
7 If $\nu^* < \nu$, answer YES, otherwise answer NO



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 μ
  - $\rightsquigarrow$ play (s_1, s_2) exactly $\left\lfloor \frac{\mu}{4} \right\rfloor$ times and then switch to (s_1, s_3) 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

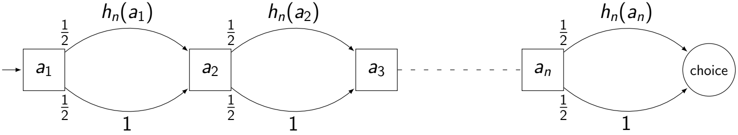
- Truly-polynomial algorithm very unlikely...
- Reduction from the K^{th} **largest subset problem**
  - ▷ commonly thought to be outside NP as natural certificates are larger than polynomial [JK78, GJ79]

K^{th} largest subset problem

Given a finite set A, a size function $h\colon A \rightarrow \mathbb{N}_0$ 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.

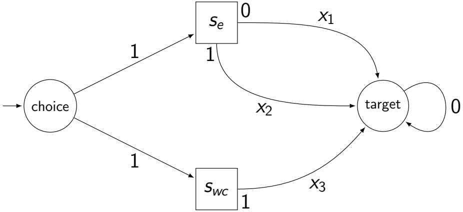
- Build a game composed of *two gadgets*

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

Crux of the reduction

There exist (non-trivial) values for thresholds and weights s.t.

(i) an optimal (i.e., minimizing the expectation while guaranteeing a given worst-case threshold) strategy for $\mathcal{P}_1$ 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.
In Proc. of LICS, pages 331–340. IEEE Computer Society, 2013.

V. Bruyère, E. Filiot, M. Randour, and J.-F. Raskin.
Meet your expectations with guarantees: beyond worst-case synthesis in quantitative games.
In Proc. of STACS, LIPIcs 25, pages 199–213. Schloss Dagstuhl - LZI, 2014.

D.P. Bertsekas and J.N. Tsitsiklis.
An analysis of stochastic shortest path problems.
Mathematics of Operations Research, 16:580–595, 1991.

K. Chatterjee, L. Doyen, T.A. Henzinger, and J.-F. Raskin.
Generalized mean-payoff and energy games.
In Proc. of FSTTCS, LIPIcs 8, pages 505–516. Schloss Dagstuhl - LZI, 2010.

K. Chatterjee, L. Doyen, M. Randour, and J.-F. Raskin.
Looking at mean-payoff and total-payoff through windows.
In Proc. of ATVA, LNCS 8172, pages 118–132. Springer, 2013.

K. Chatterjee and M. Henzinger.
An $\mathcal{O}(n^2)$ time algorithm for alternating Büchi games.
In Proc. of SODA, pages 1386–1399. SIAM, 2012.

K. Chatterjee, M. Randour, and J.-F. Raskin.
Strategy synthesis for multi-dimensional quantitative objectives.
In Proc. of CONCUR, LNCS 7454, pages 115–131. Springer, 2012.

References II

C. Courcoubetis and M. Yannakakis.
The complexity of probabilistic verification.
J. ACM, 42(4):857–907, 1995.

L. de Alfaro.
Formal verification of probabilistic systems.
PhD thesis, Stanford University, 1997.

L. de Alfaro.
Computing minimum and maximum reachability times in probabilistic systems.
In Proc. of CONCUR, LNCS 1664, pages 66–81. Springer, 1999.

A. Degorre, L. Doyen, R. Gentilini, J.-F. Raskin, and S. Torunczyk.
Energy and mean-payoff games with imperfect information.
In Proc. of CSL, LNCS 6247, pages 260–274. Springer, 2010.

A. Ehrenfeucht and J. Mycielski.
Positional strategies for mean payoff games.
Int. Journal of Game Theory, 8(2):109–113, 1979.

J.A. Filar, D. Krass, and K.W. Ross.
Percentile performance criteria for limiting average Markov decision processes.
Transactions on Automatic Control, pages 2–10, 1995.

J. Filar and K. Vrieze.
Competitive Markov decision processes.
Springer, 1997.

References III

M.R. Garey and D.S. Johnson.
Computers and intractability: a guide to the Theory of NP-Completeness.
Freeman New York, 1979.

P.W. Glynn and D. Ormoneit.
Hoeffding's inequality for uniformly ergodic Markov chains.
Statistics & Probability Letters, 56(2):143–146, 2002.

T. Gawlitza and H. Seidl.
Games through nested fixpoints.
In Proc. of CAV, LNCS 5643, pages 291–305. Springer, 2009.

D.B. Johnson and S.D. Kashdan.
Lower bounds for selection in $X + Y$ and other multisets.
Journal of the ACM, 25(4):556–570, 1978.

M. Jurdziński.
Deciding the winner in parity games is in UP $\cap$ co-UP.
Inf. Process. Lett., 68(3):119–124, 1998.

T.M. Liggett and S.A. Lippman.
Stochastic games with perfect information and time average payoff.
Siam Review, 11(4):604–607, 1969.

S. Mannor and J.N. Tsitsiklis.
Mean-variance optimization in Markov decision processes.
In Proc. of ICML, pages 177–184. Omnipress, 2011.

References IV

M.L. Puterman.
Markov decision processes: discrete stochastic dynamic programming.
John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1994.

M. Tracol.
Fast convergence to state-action frequency polytopes for MDPs.
Oper. Res. Lett., 37(2):123–126, 2009.

C. Wu and Y. Lin.
Minimizing risk models in Markov decision processes with policies depending on target values.
Journal of Mathematical Analysis and Applications, 231(1):47–67, 1999.

U. Zwick and M. Paterson.
The complexity of mean payoff games on graphs.
Theoretical Computer Science, 158:343–359, 1996.