Planning a Journey in an Uncertain Environment: The Stochastic Shortest Path Problem Revisited

Mickael Randour (LSV - CNRS & ENS Cachan)
Jean-François Raskin (ULB)   Ocan Sankur (ULB)

11.12.2014

Laboratoire d'Informatique Fondamentale de Marseille

The talk in one slide

Strategy synthesis for Markov Decision Processes (MDPs)

Finding **good** controllers for systems interacting with a *stochastic* environment.

The talk in one slide

Strategy synthesis for Markov Decision Processes (MDPs)

Finding **good** controllers for systems interacting with a *stochastic* environment.

- Good? Performance evaluated through *payoff functions*.
- Usual problem is to optimize the *expected performance* or the *probability of achieving a given performance level*.
- Not sufficient for many practical applications.
  - ▷ Several extensions, more expressive but also more complex...

The talk in one slide

Strategy synthesis for Markov Decision Processes (MDPs)

Finding **good** controllers for systems interacting with a *stochastic* environment.

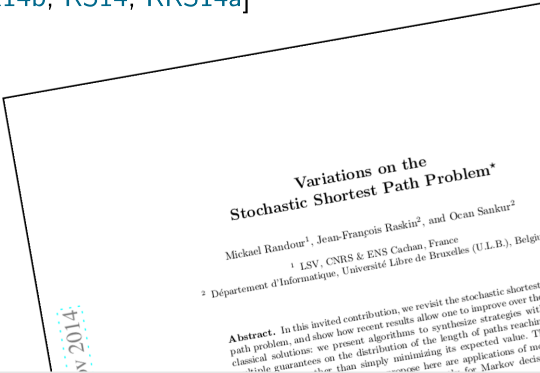
- Good? Performance evaluated through *payoff functions*.
- Usual problem is to optimize the *expected performance* or the *probability of achieving a given performance level*.
- Not sufficient for many practical applications.
  - ▷ Several extensions, more expressive but also more complex...

Aim of this survey talk

Give a flavor of classical questions and extensions, illustrated on the stochastic shortest path (SSP).

Advertisement

Invited lecture in VMCAI'15 [RRS15]
Full paper available on arXiv: abs/1411.0835
Based on recent work [BFRR14b, RS14, RRS14a]



Variations on the
Stochastic Shortest Path Problem*

Mickael Randour[1], Jean-François Raskin[2], and Ocan Sankur[2]

[1] LSV, CNRS & ENS Cachan, France
[2] Département d'Informatique, Université Libre de Bruxelles (U.L.B.), Belgi

Abstract. In this invited contribution, we revisit the stochastic shortest
path problem, and show how recent results allow one to improve over the
classical solutions: we present algorithms to synthesize strategies wit
guarantees on the distribution of the length of paths reachi
than simply minimizing its expected value.

Context

- PhD from UMONS (Belgium), 2014.
  - ▷ Supervised by V. Bruyère (UMONS) and J.-F. Raskin (ULB).
  - ▷ Title: *Synthesis in Multi-Criteria Quantitative Games* (available on my website).
- Talk partly based on research pursued during my thesis.

Context

- PhD from UMONS (Belgium), 2014.
  - ▷ Supervised by V. Bruyère (UMONS) and J.-F. Raskin (ULB).
  - ▷ Title: *Synthesis in Multi-Criteria Quantitative Games* (available on my website).
- Talk partly based on research pursued during my thesis.

General context important to understand the **motivation** behind the questions we study.

Multi-criteria quantitative synthesis

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

Multi-criteria quantitative synthesis

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

- Model of the (discrete) interaction?
  - Antagonistic environment: 2-player game on graph.
  - **Stochastic environment: MDP**.

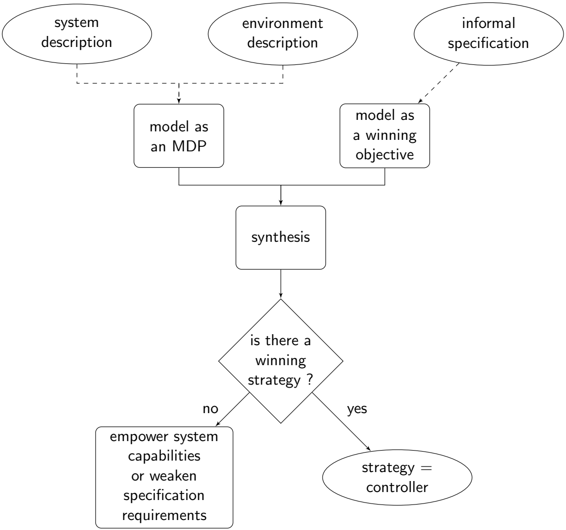
Multi-criteria quantitative synthesis

- Verification and synthesis:
  - a reactive **system** to *control*,
  - an *interacting* **environment**,
  - a **specification** to *enforce*.
- Model of the (discrete) interaction?
  - Antagonistic environment: 2-player game on graph.
  - **Stochastic environment: MDP**.
- **Quantitative** specifications. Examples:
  - Reach a state s before x time units $\rightsquigarrow$ shortest path.
  - Minimize the average response-time $\rightsquigarrow$ mean-payoff.

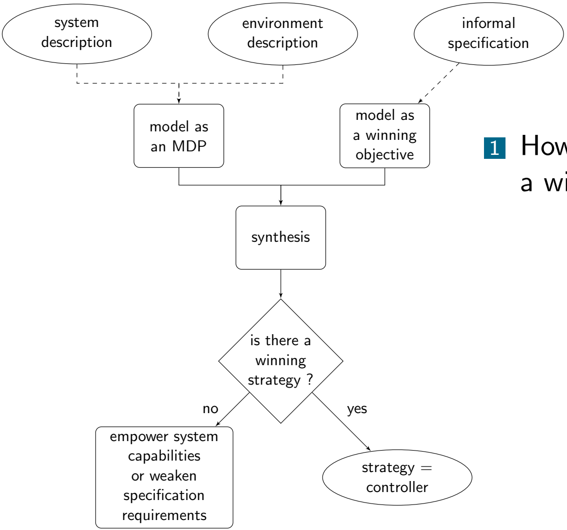
Multi-criteria quantitative synthesis

- Verification and synthesis:
  - a reactive **system** to *control*,
  - an *interacting* **environment**,
  - a **specification** to *enforce*.
- Model of the (discrete) interaction?
  - Antagonistic environment: 2-player game on graph.
  - **Stochastic environment: MDP**.
- **Quantitative** specifications. Examples:
  - Reach a state s before x time units $\rightsquigarrow$ shortest path.
  - Minimize the average response-time $\rightsquigarrow$ mean-payoff.
- Focus on **multi-criteria quantitative models**
  - to reason about *trade-offs* and *interplays*.

Strategy (policy) synthesis for MDPs

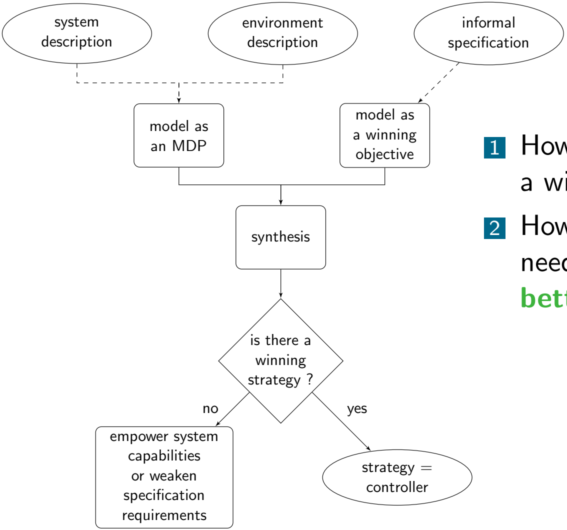


Strategy (policy) synthesis for MDPs



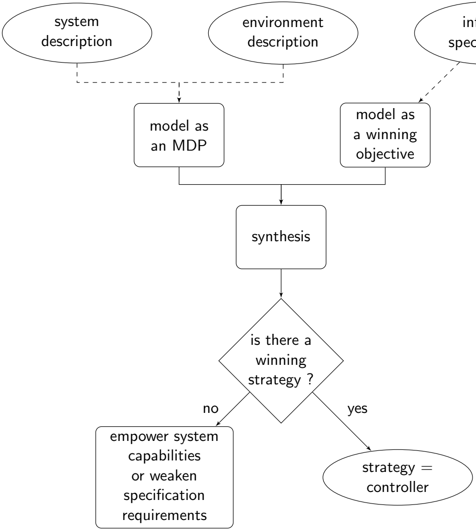
1. How complex is it to decide if a winning strategy exists?

Strategy (policy) synthesis for MDPs



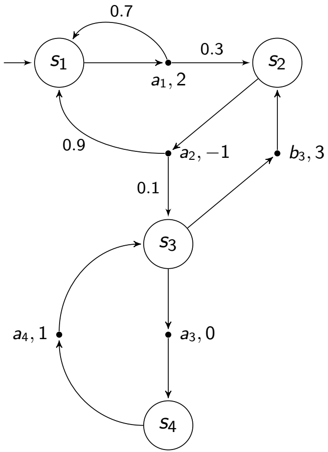
1. How complex is it to decide if a winning strategy exists?
2. How complex such a strategy needs to be? **Simpler is better**.

Strategy (policy) synthesis for MDPs



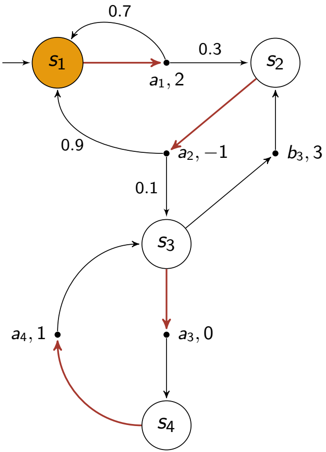
1. How complex is it to decide if a winning strategy exists?
2. How complex such a strategy needs to be? **Simpler is better**.
3. Can we synthesize one efficiently?

Markov decision processes



- MDP $D = (S, s_{\text{init}}, A, \delta, w)$
  - ▷ finite sets of states S and actions A
  - ▷ probabilistic transition $\delta \colon S \times A \to \mathcal{D}(S)$
  - ▷ weight function $w \colon A \to \mathbb{Z}$
- **Run** (or play): $\rho = s_1 a_1 \dots a_{n-1} s_n \dots$ such that $\delta(s_i, a_i, s_{i+1}) > 0$ for all $i \geq 1$
  - ▷ set of runs $\mathcal{R}(D)$
  - ▷ set of histories (finite runs) $\mathcal{H}(D)$
- **Strategy** $\sigma \colon \mathcal{H}(D) \to \mathcal{D}(A)$
  - ▷ $\forall\, h$ ending in s, $\text{Supp}(\sigma(h)) \in A(s)$

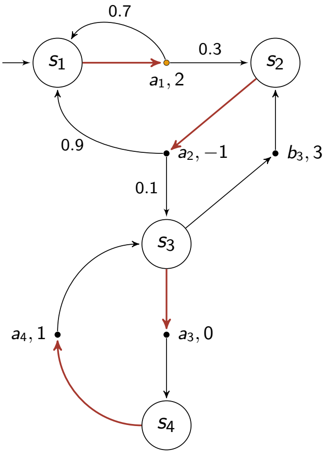
Markov decision processes



Sample *pure memoryless* strategy σ

Sample run $\rho = s_1$

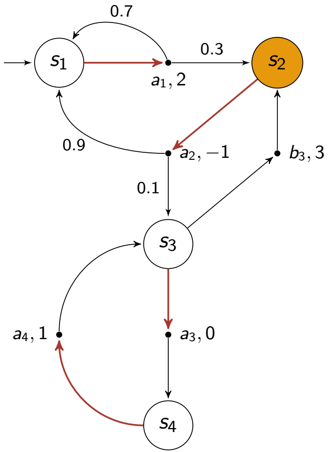
Markov decision processes



Sample *pure memoryless* strategy σ

Sample run $\rho = s_1 a_1$

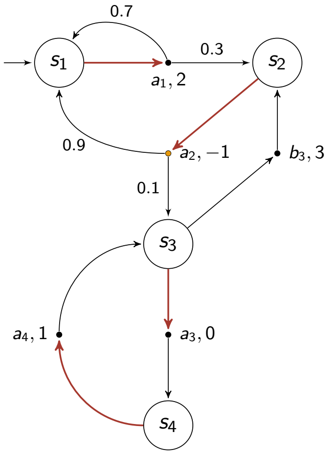
Markov decision processes



Sample *pure memoryless* strategy σ

Sample run $\rho = s_1 a_1 s_2$

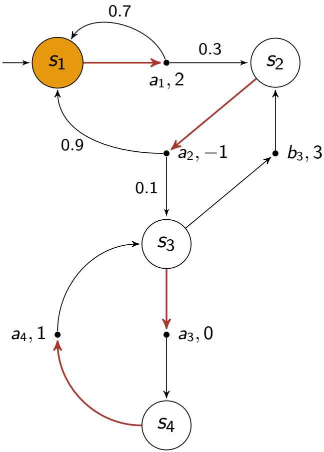
Markov decision processes



Sample *pure memoryless* strategy σ

Sample run $\rho = s_1 a_1 s_2 a_2$

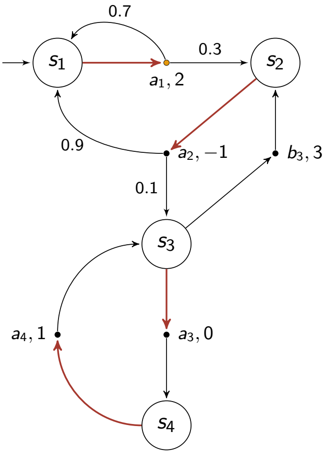
Markov decision processes



Sample *pure memoryless* strategy σ

Sample run $\rho = s_1 a_1 s_2 a_2 s_1$

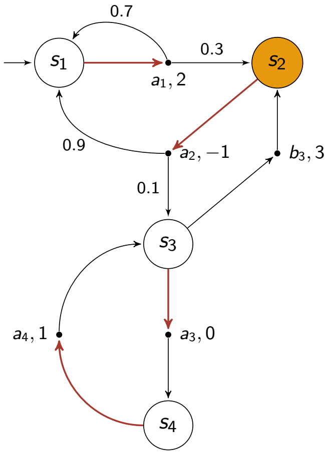
Markov decision processes



Sample *pure memoryless* strategy σ

Sample run $\rho = s_1 a_1 s_2 a_2 s_1 a_1$

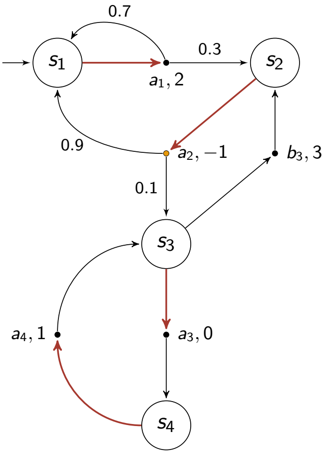
Markov decision processes



Sample *pure memoryless* strategy σ

Sample run $\rho = s_1 a_1 s_2 a_2 s_1 a_1 s_2$

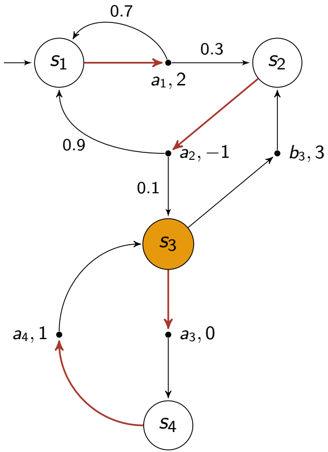
Markov decision processes



Sample *pure memoryless* strategy σ

Sample run $\rho = s_1 a_1 s_2 a_2 s_1 a_1 s_2 a_2$

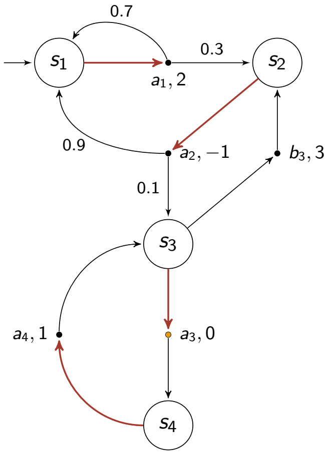
Markov decision processes



Sample *pure memoryless* strategy σ

Sample run $\rho = s_1 a_1 s_2 a_2 s_1 a_1 s_2 a_2 s_3$

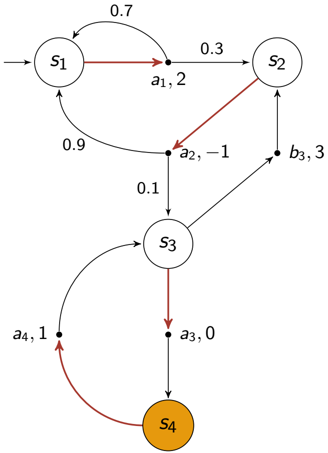
Markov decision processes



Sample *pure memoryless* strategy σ

Sample run $\rho = s_1 a_1 s_2 a_2 s_1 a_1 s_2 a_2 s_3 a_3$

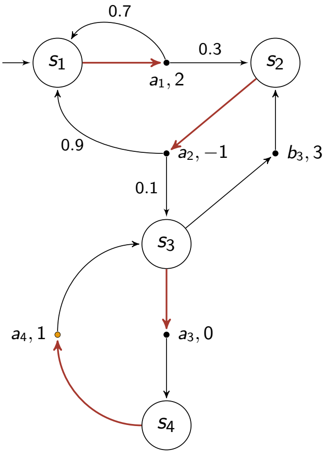
Markov decision processes



Sample *pure memoryless* strategy σ

Sample run $\rho = s_1 a_1 s_2 a_2 s_1 a_1 s_2 a_2 s_3 a_3 s_4$

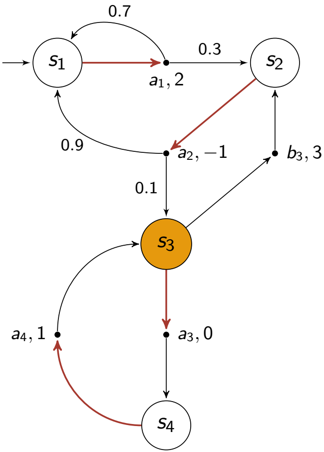
Markov decision processes



Sample *pure memoryless* strategy σ

Sample run $\rho = s_1 a_1 s_2 a_2 s_1 a_1 s_2 a_2 s_3 a_3 s_4 a_4$

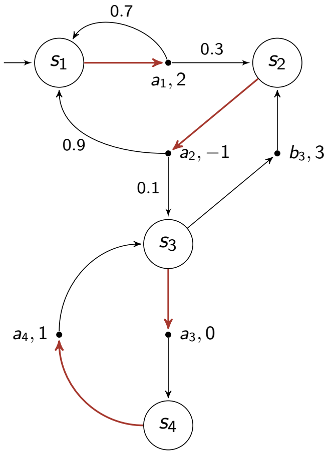
Markov decision processes



Sample *pure memoryless* strategy σ

Sample run $\rho = s_1 a_1 s_2 a_2 s_1 a_1 s_2 a_2 (s_3 a_3 s_4 a_4)^{\omega}$

Markov decision processes

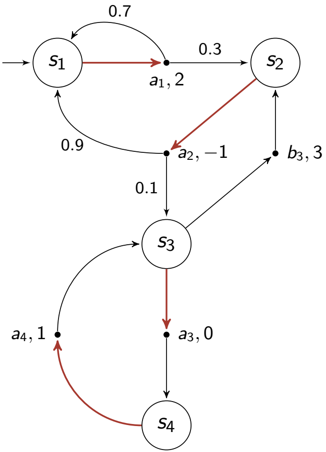


Sample *pure memoryless* strategy σ

Sample run $\rho = s_1 a_1 s_2 a_2 s_1 a_1 s_2 a_2 (s_3 a_3 s_4 a_4)^\omega$

Other possible run $\rho' = s_1 a_1 s_2 a_2 (s_3 a_3 s_4 a_4)^\omega$

Markov decision processes



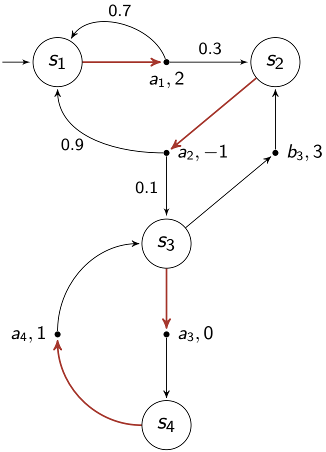
Sample *pure memoryless* strategy σ

Sample run $\rho = s_1 a_1 s_2 a_2 s_1 a_1 s_2 a_2 (s_3 a_3 s_4 a_4)^\omega$

Other possible run $\rho' = s_1 a_1 s_2 a_2 (s_3 a_3 s_4 a_4)^\omega$

- Strategies may use
  - finite or infinite **memory**
  - **randomness**
- **Payoff functions** map runs to numerical values
  - truncated sum up to $T = \{s_3\}$: $\mathsf{TS}^T(\rho) = 2$, $\mathsf{TS}^T(\rho') = 1$
  - mean-payoff: $\underline{\mathsf{MP}}(\rho) = \underline{\mathsf{MP}}(\rho') = 1/2$
  - many more

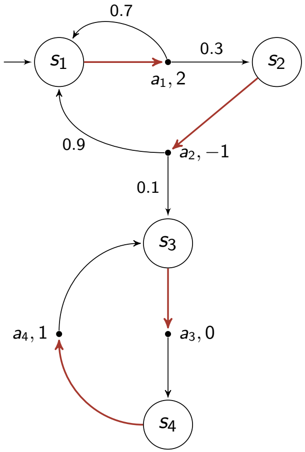
Markov chains



Once strategy σ fixed, fully stochastic process

$\rightsquigarrow$ **Markov chain** (**MC**)

Markov chains

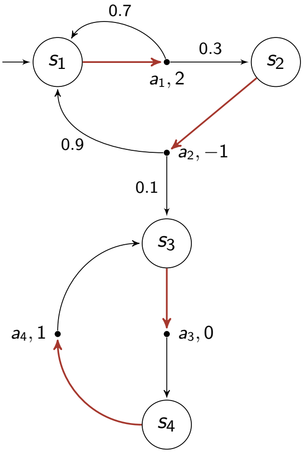


Once strategy σ fixed, fully stochastic process

$\rightsquigarrow$ **Markov chain** (**MC**)

State space = product of the MDP and the memory of σ

Markov chains



Once strategy σ fixed, fully stochastic process

$\rightsquigarrow$ **Markov chain** (**MC**)

State space = product of the MDP and the memory of σ

- Event $\mathcal{E} \subseteq \mathcal{R}(M)$
  - ▷ probability $\mathbb{P}_M(\mathcal{E})$
- Measurable $f \colon \mathcal{R}(M) \to \mathbb{R} \cup \{\infty\}$,
  - ▷ expected value $\mathbb{E}_M(f)$

Aim of this survey

Review and compare different types of quantitative specifications for MDPs

- ▷ w.r.t. the complexity of the decision problem
- ▷ w.r.t. the complexity of winning strategies

Recent extensions share a common philosophy: framework for the synthesis of strategies with *richer performance guarantees*

- ▷ our work deals with many different payoff functions

Aim of this survey

Review and compare different types of quantitative specifications for MDPs

- ▷ w.r.t. the complexity of the decision problem
- ▷ w.r.t. the complexity of winning strategies

Recent extensions share a common philosophy: framework for the synthesis of strategies with *richer performance guarantees*

- ▷ our work deals with many different payoff functions

Focus on the **shortest path problem** in this talk

- ▷ not the most involved technically
- ▷ natural applications

⇝ useful to understand the practical interest of each variant

+ brief mention of results for other payoffs

Stochastic shortest path

Shortest path problem for *weighted graphs*

Given state $s \in S$ and target set $T \subseteq S$, find a path from s to a state $t \in T$ that minimizes the sum of weights along edges.

- PTIME algorithms (Dijkstra, Bellman-Ford, etc) [CGR96]

Stochastic shortest path

Shortest path problem for *weighted graphs*

Given state $s \in S$ and target set $T \subseteq S$, find a path from s to a state $t \in T$ that minimizes the sum of weights along edges.

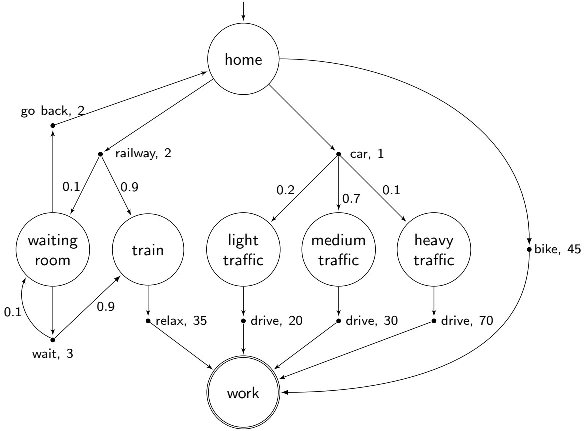
- PTIME algorithms (Dijkstra, Bellman-Ford, etc) [CGR96]

We focus on MDPs with **strictly positive weights** in this talk

- **Truncated sum** payoff function for $\rho = s_1 a_1 s_2 a_2 \ldots$ and target set T:

$$\mathsf{TS}^T(\rho) = \begin{cases} \sum_{j=1}^{n-1} w(a_j) \text{ if } s_n \text{ first visit of } T \\ \infty \text{ if } T \text{ is never reached} \end{cases}$$

Planning a journey in an uncertain environment



Each action takes time, target = work.

▷ What kind of strategies are we looking for when the environment is stochastic?

SSP-E: minimizing the expected length to target

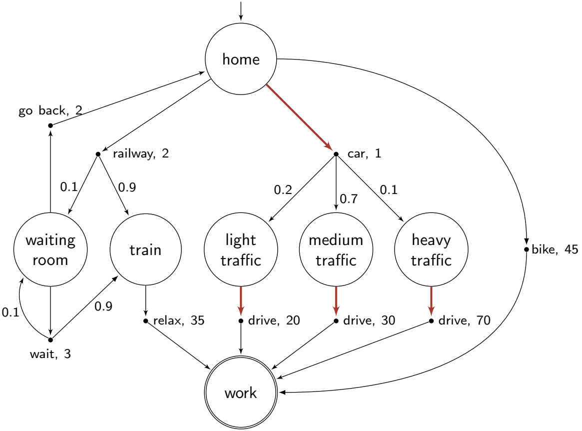
SSP-E problem

Given MDP $D = (S, s_{\mathsf{init}}, A, \delta, w)$, target set T and threshold $\ell \in \mathbb{N}$, decide if there exists σ such that $\mathbb{E}^{\sigma}_{D}(\mathsf{TS}^{T}) \leq \ell$.

Theorem [BT91]

The SSP-E problem can be decided in polynomial time. Optimal pure memoryless strategies always exist and can be constructed in polynomial time.

SSP-E: illustration



- ▷ Pure memoryless strategies suffice.
- ▷ Taking the **car** is optimal: $\mathbb{E}^{\sigma}_{D}(\mathsf{TS}^{T}) = 33$.

SSP-E: PTIME algorithm

1. Graph analysis (linear time)
   - s not connected to $T \Rightarrow \infty$ and remove
   - $s \in T \Rightarrow 0$
2. **Linear programming** (**LP**, polynomial time)

SSP-E: PTIME algorithm

1. Graph analysis (linear time)
   - ▷ s not connected to $T \Rightarrow \infty$ and remove
   - ▷ $s \in T \Rightarrow 0$
2. **Linear programming** (**LP**, polynomial time)

For each $s \in S \setminus T$, one variable x_s,

$$\max \sum_{s \in S \setminus T} x_s$$

under the constraints

$$x_s \leq w(a) + \sum_{s' \in S \setminus T} \delta(s, a, s') \cdot x_{s'} \qquad \text{for all } s \in S \setminus T, \text{ for all } a \in A(s).$$

SSP-E: PTIME algorithm

1. Graph analysis (linear time)
   - s not connected to $T \Rightarrow \infty$ and remove
   - $s \in T \Rightarrow 0$
2. **Linear programming** (**LP**, polynomial time)

Optimal solution $\mathbf{v}$

$\rightsquigarrow$ $\mathbf{v}_s$ = expectation from s to T under an optimal strategy

Optimal pure memoryless strategy $\sigma^{\mathbf{v}}$:

$$\sigma^{\mathbf{v}}(s) = \arg \min_{a \in A(s)} \left[w(a) + \sum_{s' \in S \setminus T} \delta(s, a, s') \cdot \mathbf{v}_{s'} \right].$$

$\rightsquigarrow$ **playing optimally = locally optimizing present + future**

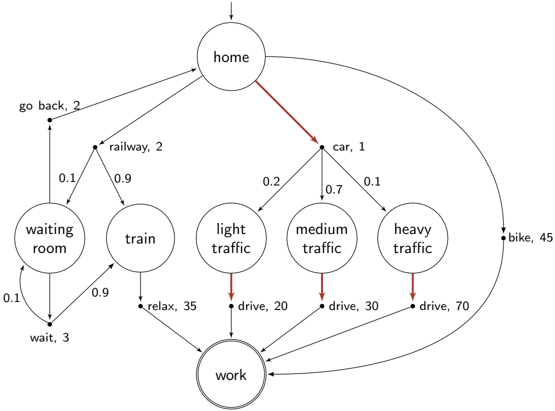
SSP-E: PTIME algorithm

1. Graph analysis (linear time)
   - s not connected to $T \Rightarrow \infty$ and remove
   - $s \in T \Rightarrow 0$
2. **Linear programming** (**LP**, polynomial time)

In practice, **value and strategy iteration** algorithms often used

- best performance in most cases but exponential in the worst-case
- fixed point algorithms, successive solution improvements [BT91, dA99, HM14]

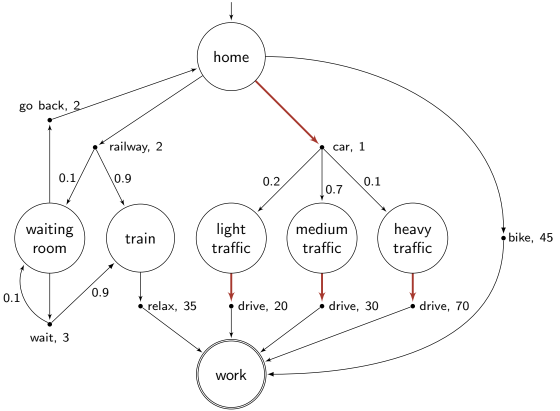
Travelling without taking too many risks



Minimizing the *expected time* to destination makes sense **if** we travel often and it is not a problem to be late.

With car, in 10% of the cases, the journey takes 71 minutes.

Travelling without taking too many risks



Most bosses will not be happy if we are late too often. . .

$\rightsquigarrow$ what if we are risk-averse and want to avoid that?

SSP-P: forcing short paths with high probability

SSP-P problem

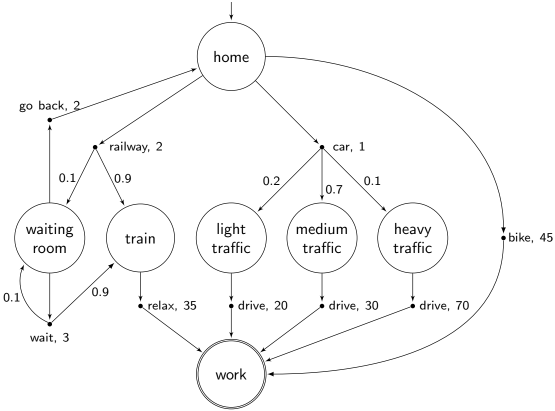
Given MDP $D = (S, s_{\text{init}}, A, \delta, w)$, target set T, threshold $\ell \in \mathbb{N}$, and probability threshold $\alpha \in [0, 1] \cap \mathbb{Q}$, decide if there exists a strategy σ such that $\mathbb{P}^{\sigma}_{D}\big[\{\rho \in \mathcal{R}_{s_{\text{init}}}(D) \mid \mathsf{TS}^{T}(\rho) \leq \ell\}\big] \geq \alpha$.

Theorem

The SSP-P problem can be decided in pseudo-polynomial time, and it is PSPACE-hard. Optimal pure strategies with pseudo-polynomial memory always exist and can be constructed in pseudo-polynomial time.

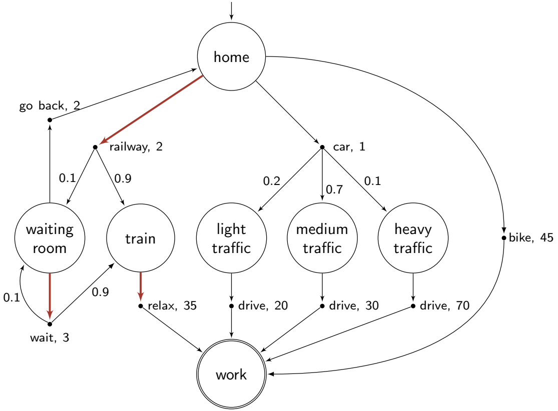
See [HK14] for hardness and for example [RRS14a] for algorithm.

SSP-P: illustration



Specification: reach work within 40 minutes with 0.95 probability

SSP-P: illustration



Specification: reach work within 40 minutes with 0.95 probability

Sample strategy: take the **train** $\rightsquigarrow \mathbb{P}^{\sigma}_{D}\left[\mathsf{TS}^{\mathsf{work}} \leq 40\right] = 0.99$

Bad choices: car (0.9) and bike (0.0)

SSP-P: pseudo-PTIME algorithm (1/2)

Key idea: pseudo-PTIME reduction to the **stochastic reachability problem** (**SR**)

SSP-P: pseudo-PTIME algorithm (1/2)

Key idea: pseudo-PTIME reduction to the **stochastic reachability problem** (**SR**)

SR problem

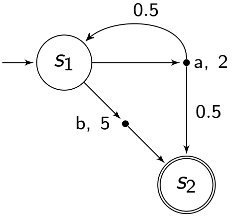
Given unweighted MDP $D = (S, s_{\mathsf{init}}, A, \delta)$, target set T and probability threshold $\alpha \in [0, 1] \cap \mathbb{Q}$, decide if there exists a strategy σ such that $\mathbb{P}^{\sigma}_{D}\left[\Diamond T\right] \geq \alpha$.

Theorem

The SR problem can be decided in polynomial time. Optimal pure memoryless strategies always exist and can be constructed in polynomial time.

- ▷ linear programming (similar to SSP-E)

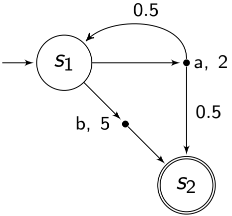
SSP-P: pseudo-PTIME algorithm (2/2)



Sketch of the reduction

1. Start from D, $T = \{s_2\}$, and $\ell = 7$.

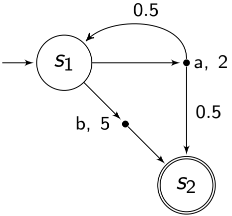
SSP-P: pseudo-PTIME algorithm (2/2)

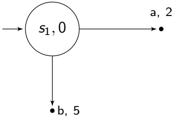


Sketch of the reduction

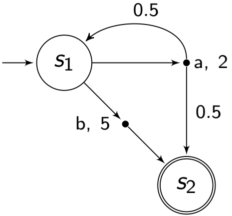
1. Start from D, $T = \{s_2\}$, and $\ell = 7$.
2. Build D_ℓ by unfolding D, tracking the current sum *up to the threshold* ℓ, and integrating it in the states of the expanded MDP.

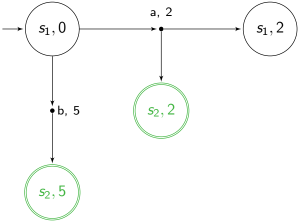
SSP-P: pseudo-PTIME algorithm (2/2)



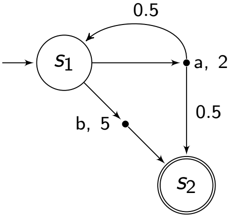


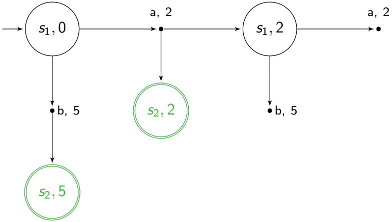
SSP-P: pseudo-PTIME algorithm (2/2)



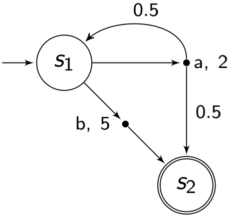


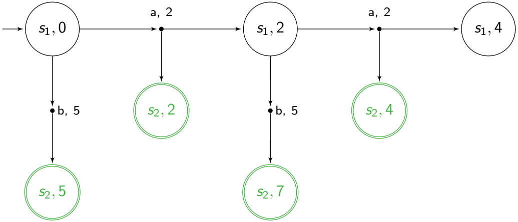
SSP-P: pseudo-PTIME algorithm (2/2)



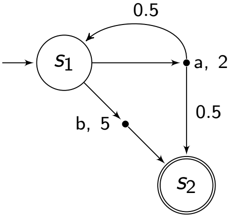


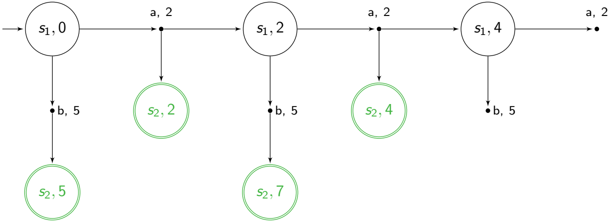
SSP-P: pseudo-PTIME algorithm (2/2)



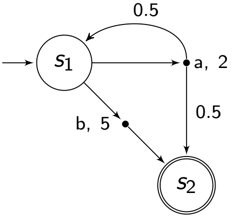


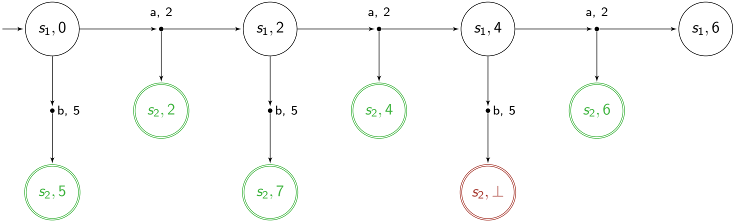
SSP-P: pseudo-PTIME algorithm (2/2)



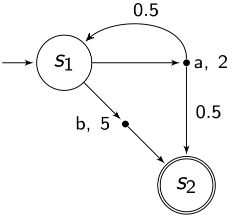


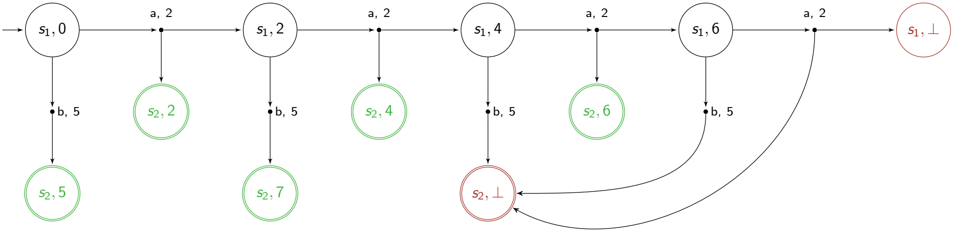
SSP-P: pseudo-PTIME algorithm (2/2)





SSP-P: pseudo-PTIME algorithm (2/2)

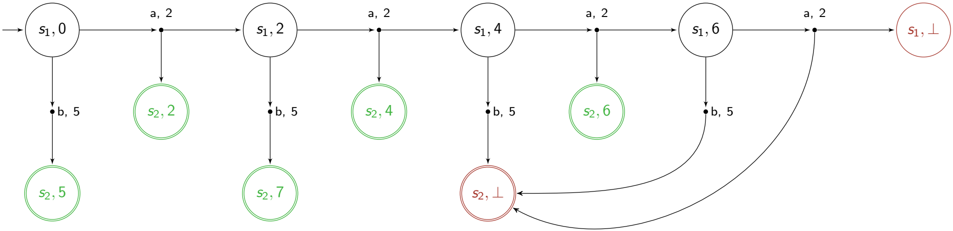




SSP-P: pseudo-PTIME algorithm (2/2)

3 Bijection between runs of D and D_ℓ

$$\mathsf{TS}^T(\rho) \leq \ell \quad \Leftrightarrow \quad \rho' \models \Diamond T',\ T' = T \times \{0, 1, \ldots, \ell\}$$

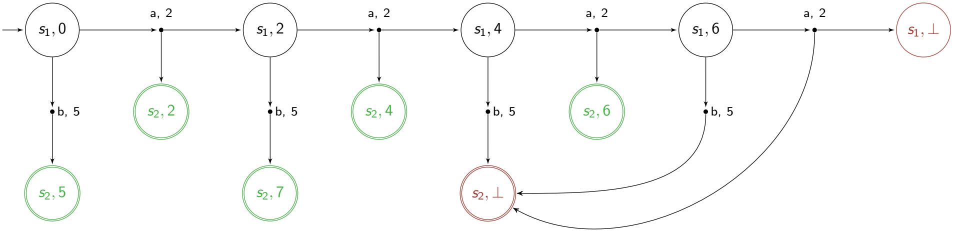


SSP-P: pseudo-PTIME algorithm (2/2)

3. Bijection between runs of D and D_ℓ

$$\mathsf{TS}^T(\rho) \leq \ell \quad \Leftrightarrow \quad \rho' \models \Diamond T',\ T' = T \times \{0, 1, \ldots, \ell\}$$

4. Solve the SR problem on D_ℓ
   - ▷ Memoryless strategy in $D_\ell \rightsquigarrow$ pseudo-polynomial memory in D in general



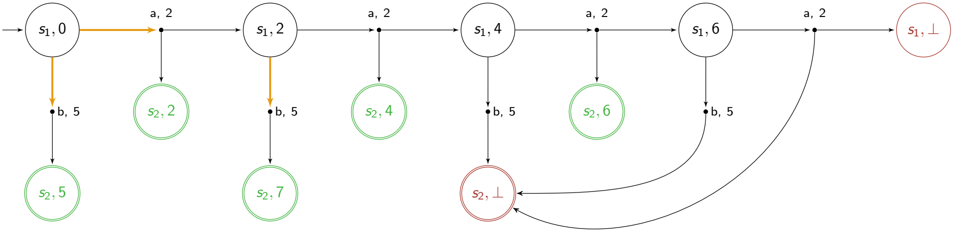
SSP-P: pseudo-PTIME algorithm (2/2)

If we just want to minimize the risk of exceeding $\ell = 7$,

- an obvious possibility is to play *b* directly,
- playing *a* only once is also acceptable.

For the SSP-P problem, **both strategies are equivalent**

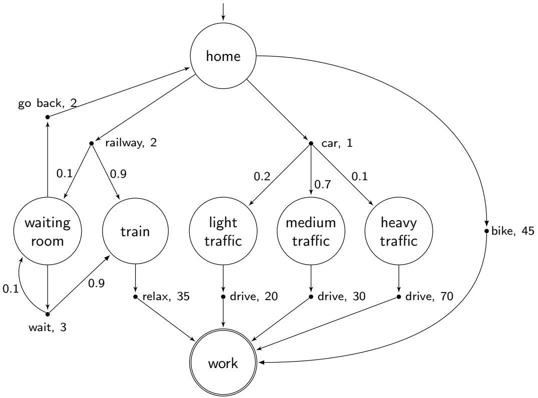
$\rightsquigarrow$ need richer models to discriminate them!



Related work

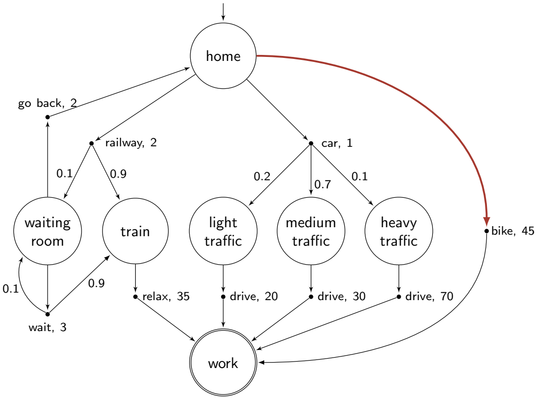
- SSP-P problem [Oht04, SO13].
- *Quantile queries* [UB13]: minimizing the value ℓ of an SSP-P problem for some fixed α. Recently extended to *cost problems* [HK14].
- SSP-E problem in **multi-dimensional** MDPs [FKN+11].

SP-G: strict worst-case guarantees



Specification: *guarantee* that work is reached within 60 minutes (to avoid missing an important meeting)

SP-G: strict worst-case guarantees

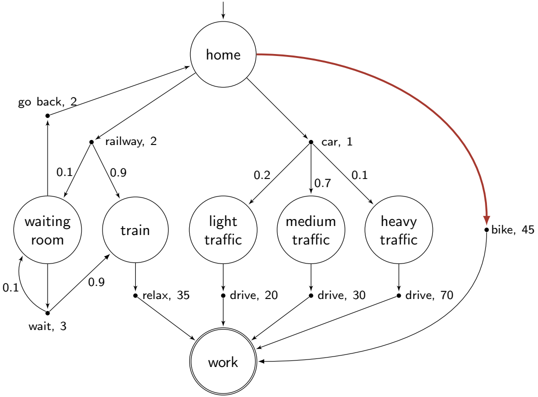


Specification: *guarantee* that work is reached within 60 minutes (to avoid missing an important meeting)

Sample strategy: take the **bike** $\rightsquigarrow \forall\, \rho \in \mathsf{Out}^{\sigma}_{D} \colon\ \mathsf{TS}^{\mathsf{work}}(\rho) \leq 60$

Bad choices: train ($wc = \infty$) and car ($wc = 71$)

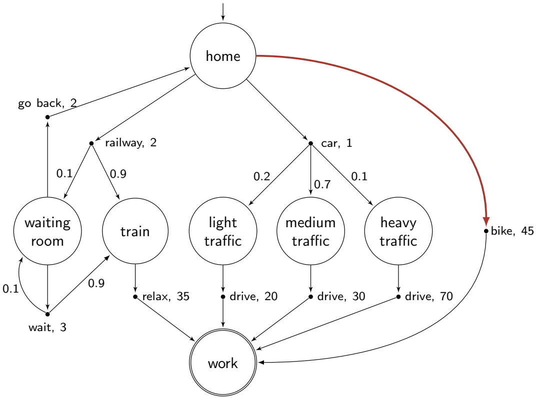
SP-G: strict worst-case guarantees



Winning surely (worst-case) $\neq$ almost-surely (proba. 1)

- ▷ train ensures reaching work with probability one, but does not prevent runs where work is never reached

SP-G: strict worst-case guarantees



Worst-case analysis $\rightsquigarrow$ **two-player game** against an antagonistic adversary

- ▷ forget about probabilities and give the choice of transitions to the adversary

SP-G: shortest path game problem

SP-G problem

Given MDP $D = (S, s_{\text{init}}, A, \delta, w)$, target set T and threshold $\ell \in \mathbb{N}$, decide if there exists a strategy σ such that for all $\rho \in \text{Out}^{\sigma}_{D}$, we have that $\text{TS}^{T}(\rho) \leq \ell$.

Theorem [KBB+08]

The SP-G problem can be decided in polynomial time. Optimal pure memoryless strategies always exist and can be constructed in polynomial time.

- ▷ Does not hold for arbitrary weights.

Related work

- Pseudo-PTIME for arbitrary weights [BGHM14, FGR12].
- Arbitrary weights + multiple dimensions $\rightsquigarrow$ undecidable (by adapting the proof of [CDRR13] for total-payoff).

SP-G: PTIME algorithm

1. Cycles are bad ⇒ must reach target within $n = |S|$ steps
2. $\forall\, s \in S$, $\forall\, i$, $0 \leq i \leq n$, compute $\mathbb{C}(s, i)$
   - ▷ lowest bound on cost to T from s that we can ensure in i steps
   - ▷ **dynamic programming** (polynomial time)

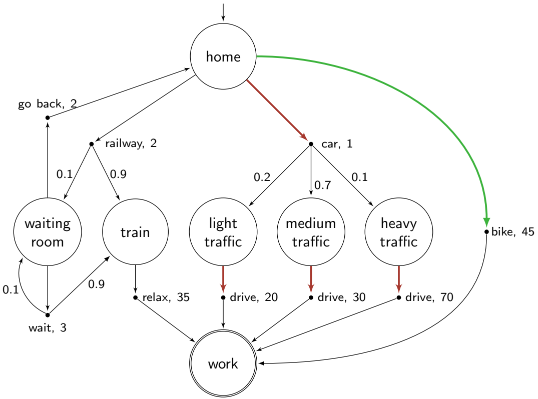
   Initialize

   $$\forall\, s \in T,\ \mathbb{C}(s, 0) = 0 \qquad \forall\, s \in S \setminus T,\ \mathbb{C}(s, 0) = \infty$$

   Then, $\forall\, s \in S$, $\forall\, i$, $1 \leq i \leq n$,

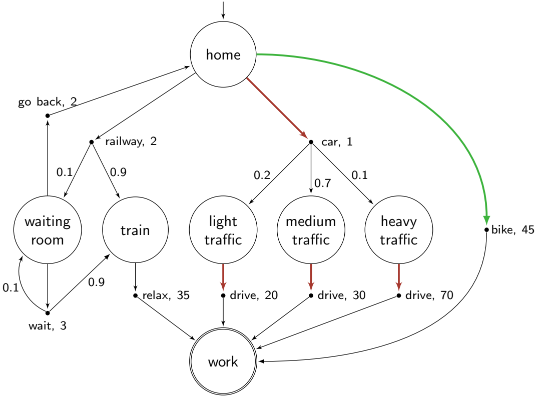
   $$\mathbb{C}(s, i) = \min\left[\mathbb{C}(s, i-1), \min_{a \in A(s)} \max_{s' \in \mathrm{Supp}(\delta(s,a))} w(a) + \mathbb{C}(s', i-1)\right]$$
3. Winning strategy iff $\mathbb{C}(s_{\mathrm{init}}, n) \leq \ell$

SSP-WE = SP-G ∩ SSP-E - illustration



- SSP-E: car $\rightsquigarrow \mathbb{E} = 33$ but $wc = 71 > 60$
- SP-G: bike $\rightsquigarrow wc = 45 < 60$ but $\mathbb{E} = 45 >>> 33$

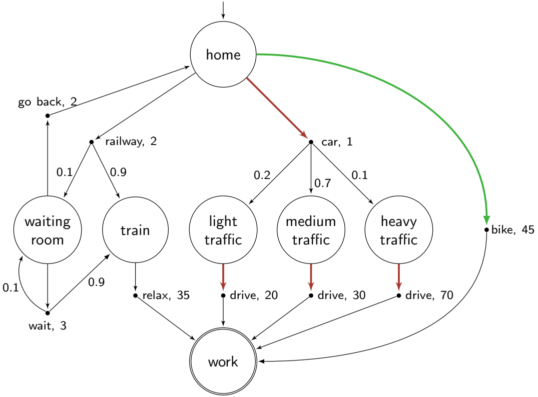
SSP-WE = SP-G ∩ SSP-E - illustration



Can we do better?

▷ **Beyond worst-case synthesis** [BFRR14b, BFRR14a]: minimize the expected time under the worst-case constraint.

SSP-WE $=$ SP-G $\cap$ SSP-E - illustration



Sample strategy: try train up to 3 delays then switch to bike.

- $\rightsquigarrow$ $wc = 58 < 60$ and $\mathbb{E} \approx 37.34 << 45$
- $\rightsquigarrow$ pure *finite-memory* strategy

SSP-WE: beyond worst-case synthesis

SSP-WE problem

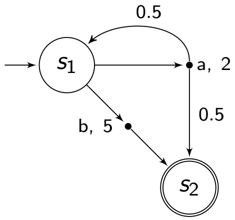
Given MDP $D = (S, s_{\text{init}}, A, \delta, w)$, target set T, and thresholds $\ell_1, \ell_2 \in \mathbb{N}$, decide if there exists a strategy σ such that:

1. $\forall\, \rho \in \mathsf{Out}^{\sigma}_{D}\colon\ \mathsf{TS}^{T}(\rho) \leq \ell_1$,
2. $\mathbb{E}^{\sigma}_{D}(\mathsf{TS}^{T}) \leq \ell_2$.

Theorem [BFRR14b]

The SSP-WE problem can be decided in pseudo-polynomial time and is NP-hard. Pure pseudo-polynomial-memory strategies are always sufficient and in general necessary, and satisfying strategies can be constructed in pseudo-polynomial time.

SSP-WE: pseudo-PTIME algorithm

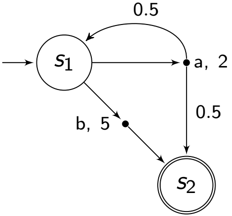


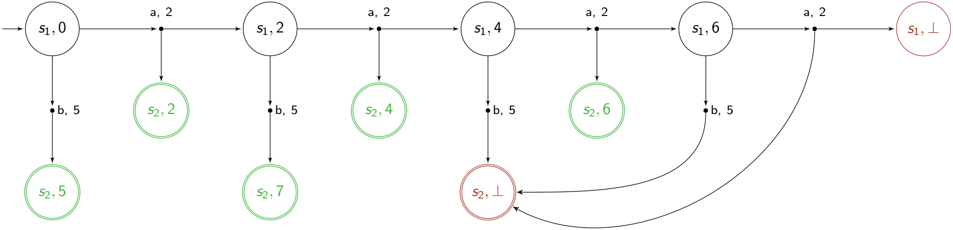
Consider SSP-WE problem for $\ell_1 = 7$ (*wc*), $\ell_2 = 4.8$ ($\mathbb{E}$).

▷ Reduction to the SSP-E problem on a pseudo-polynomial-size expanded MDP.

1. Build unfolding as for SSP-P problem w.r.t. worst-case threshold ℓ_1.

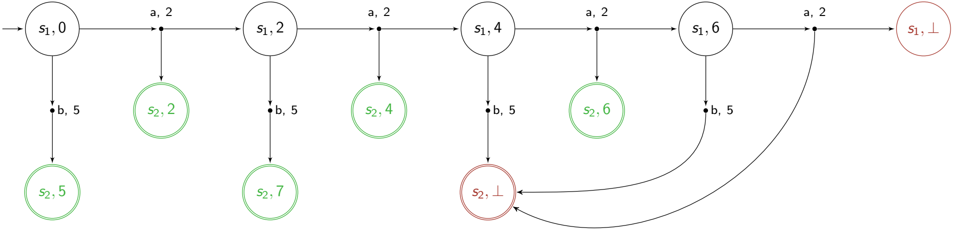
SSP-WE: pseudo-PTIME algorithm





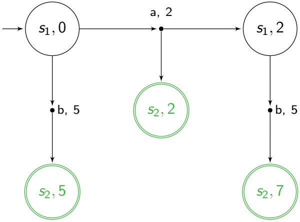
SSP-WE: pseudo-PTIME algorithm

2. Compute R, the attractor of $T' = T \times \{0, 1, \ldots, \ell_1\}$.
3. Restrict MDP to $D' = D_{\ell_1} \downharpoonright R$, the *safe* part w.r.t. SP-G.



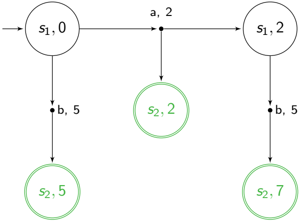
SSP-WE: pseudo-PTIME algorithm

2. Compute R, the attractor of $T' = T \times \{0, 1, \ldots, \ell_1\}$.
3. Restrict MDP to $D' = D_{\ell_1} \downharpoonleft R$, the *safe* part w.r.t. SP-G.



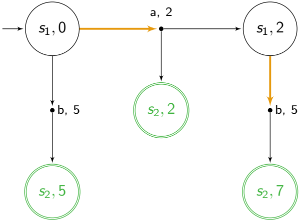
SSP-WE: pseudo-PTIME algorithm

4. Compute memoryless optimal strategy σ in D' for SSP-E.
5. Answer is $\textsc{Yes}$ iff $\mathbb{E}^{\sigma}_{D'}(\mathsf{TS}^{T'}) \leq \ell_2$.



SSP-WE: pseudo-PTIME algorithm

4. Compute memoryless optimal strategy σ in D' for SSP-E.
5. Answer is YES iff $\mathbb{E}^{\sigma}_{D'}(\mathsf{TS}^{T'}) \leq \ell_2$.



Here,
$\mathbb{E}^{\sigma}_{D'}(\mathsf{TS}^{T'}) = 9/2$

SSP-WE: wrap-up

SSP	complexity	strategy
SSP-E	PTIME	pure memoryless
SSP-P	pseudo-PTIME / PSPACE-h.	pure pseudo-poly.
SSP-G	PTIME	pure memoryless
SSP-WE	pseudo-PTIME / NP-h.	pure pseudo-poly.

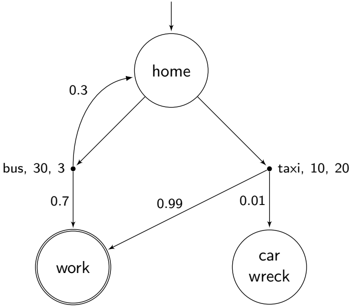
▷ NP-hardness ⇒ inherently harder than SSP-E and SSP-G.

Beyond worst-case synthesis for mean-payoff

MP	complexity	strategy
MP-E	PTIME	pure memoryless
MP-G	NP ∩ coNP	pure memoryless
MP-WE	NP ∩ coNP	pure pseudo-poly.

- ▷ Long-run average of weights [EM79], subsumes parity games [Jur98].
- ▷ Additional modeling power **for free**.
- ▷ Much more involved technically [BFRR14b, BFRR14a].

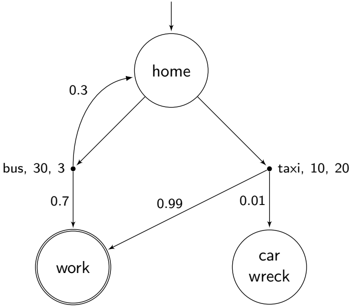
Multiple objectives ⇒ trade-offs



Two-dimensional weights on actions: *time* and *cost*.

Often necessary to consider trade-offs: e.g., between the probability to reach work in due time and the risks of an expensive journey.

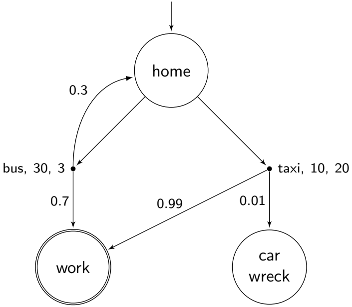
Multiple objectives ⇒ trade-offs



SSP-P problem considers a **single percentile constraint**.

- **C1**: 80% of runs reach work in at most 40 minutes.
  - ▷ Taxi $\rightsquigarrow \leq 10$ minutes with probability $0.99 > 0.8$.

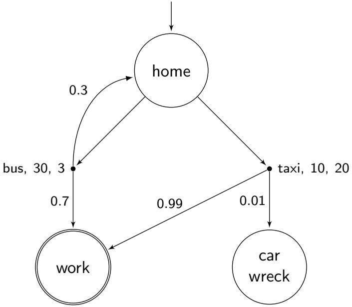
Multiple objectives ⇒ trade-offs



SSP-P problem considers a **single percentile constraint**.

- **C1**: 80% of runs reach work in at most 40 minutes.
  - ▷ Taxi $\rightsquigarrow \leq 10$ minutes with probability $0.99 > 0.8$.
- **C2**: 50% of them cost at most 10$ to reach work.
  - ▷ Bus $\rightsquigarrow \geq 70\%$ of the runs reach work for 3$.

Multiple objectives ⇒ trade-offs

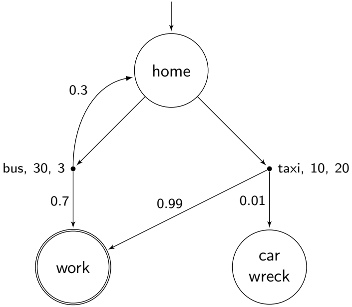


SSP-P problem considers a **single percentile constraint**.

- **C1**: 80% of runs reach work in at most 40 minutes.
  - ▷ Taxi $\rightsquigarrow \leq 10$ minutes with probability $0.99 > 0.8$.
- **C2**: 50% of them cost at most 10\$ to reach work.
  - ▷ Bus $\rightsquigarrow \geq 70\%$ of the runs reach work for 3\$.

Taxi $\not\models$ C2, bus $\not\models$ C1. What if we want C1 $\wedge$ C2?

Multiple objectives ⇒ trade-offs

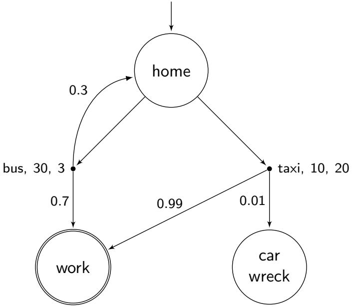


- **C1**: 80% of runs reach work in at most 40 minutes.
- **C2**: 50% of them cost at most 10$ to reach work.

Study of **multi-constraint percentile queries** [RRS14a].

- ▷ Sample strategy: bus once, then taxi. Requires *memory*.
- ▷ Another strategy: bus with probability 3/5, taxi with probability 2/5. Requires *randomness*.

Multiple objectives ⇒ trade-offs



- **C1**: 80% of runs reach work in at most 40 minutes.
- **C2**: 50% of them cost at most 10$ to reach work.

Study of **multi-constraint percentile queries** [RRS14a].

In general, *both* memory *and* randomness are required.

≠ previous problems

SSP-PQ: multi-constraint percentile queries (1/2)

SSP-PQ problem

Given d-dimensional MDP $D = (S, s_{\text{init}}, A, \delta, w)$, and $q \in \mathbb{N}$ percentile constraints described by target sets $T_i \subseteq S$, dimensions $k_i \in \{1, \ldots, d\}$, value thresholds $\ell_i \in \mathbb{N}$ and probability thresholds $\alpha_i \in [0, 1] \cap \mathbb{Q}$, where $i \in \{1, \ldots, q\}$, decide if there exists a strategy σ such that

$$\forall\, i \in \{1, \ldots, q\},\ \mathbb{P}^{\sigma}_{D}\left[\mathsf{TS}^{T_i}_{k_i} \leq \ell_i\right] \geq \alpha_i,$$

where $\mathsf{TS}^{T_i}_{k_i}$ denotes the truncated sum on dimension k_i and w.r.t. target set T_i.

Very general framework allowing for: multiple constraints related to $\neq$ dimensions, and $\neq$ target sets.

$\rightsquigarrow$ Great flexibility in modeling applications.

SSP-PQ: multi-constraint percentile queries (2/2)

Theorem [RRS14a]

The SSP-PQ problem can be decided in

- exponential time in general,
- pseudo-polynomial time for single-dimension single-target multi-contraint queries.

It is PSPACE-hard even for single-constraint queries. Randomized exponential-memory strategies are always sufficient and in general necessary, and satisfying strategies can be constructed in exponential time.

▷ PSPACE-hardness already true for SSP-P [HK14].

$\rightsquigarrow$ SSP-PQ = wide extension for basically no price in complexity.

SSP-PQ: EXPTIME / pseudo-PTIME algorithm

1. Build an unfolded MDP D_ℓ similar to SSP-P case:
   - stop unfolding when *all* dimensions reach sum $\ell = \max_i \ell_i$.

SSP-PQ: EXPTIME / pseudo-PTIME algorithm

1. Build an unfolded MDP D_ℓ similar to SSP-P case:
   - ▷ stop unfolding when *all* dimensions reach sum $\ell = \max_i \ell_i$.

2. Maintain *single*-exponential size by defining an equivalence relation between states of D_ℓ:
   - ▷ $S_\ell \subseteq S \times (\{0, \ldots, \ell\} \cup \{\bot\})^d$,
   - ▷ pseudo-poly. if $d = 1$.

SSP-PQ: EXPTIME / pseudo-PTIME algorithm

1. Build an unfolded MDP D_ℓ similar to SSP-P case:
   - ▷ stop unfolding when *all* dimensions reach sum $\ell = \max_i \ell_i$.
2. Maintain *single*-exponential size by defining an equivalence relation between states of D_ℓ:
   - ▷ $S_\ell \subseteq S \times (\{0, \ldots, \ell\} \cup \{\bot\})^d$,
   - ▷ pseudo-poly. if $d = 1$.
3. For each constraint i, compute a target set R_i in D_ℓ:
   - ▷ $\rho \models$ constraint i in $D \Leftrightarrow \rho' \models \Diamond R_i$ in D_ℓ.

SSP-PQ: EXPTIME / pseudo-PTIME algorithm

1. Build an unfolded MDP D_ℓ similar to SSP-P case:
   - ▷ stop unfolding when *all* dimensions reach sum $\ell = \max_i \ell_i$.
2. Maintain *single*-exponential size by defining an equivalence relation between states of D_ℓ:
   - ▷ $S_\ell \subseteq S \times (\{0, \ldots, \ell\} \cup \{\bot\})^d$,
   - ▷ pseudo-poly. if $d = 1$.
3. For each constraint i, compute a target set R_i in D_ℓ:
   - ▷ $\rho \models$ constraint i in $D \Leftrightarrow \rho' \models \Diamond R_i$ in D_ℓ.
4. Solve a **multiple reachability problem** on D_ℓ.
   - ▷ Generalizes the SR problem [EKVY08, RRS14a].
   - ▷ Time polynomial in $|D_\ell|$ but exponential in q.
   - ▷ Single-dim. single target queries $\Rightarrow$ absorbing targets $\Rightarrow$ polynomial-time algorithm.

SSP-PQ: wrap-up

SSP	complexity	strategy
SSP-E	PTIME	pure memoryless
SSP-P	pseudo-PTIME / PSPACE-h.	pure pseudo-poly.
SSP-G	PTIME	pure memoryless
SSP-WE	pseudo-PTIME / NP-h.	pure pseudo-poly.
SSP-PQ	EXPTIME (p.-PTIME) / PSPACE-h.	randomized exponential

Related work and additional results

- *Cost problems* [HK14]: $\exists ?\, \sigma,\ \mathbb{P}^{\sigma}_{D}\big[\mathsf{TS}^{T} \models \varphi\big] \geq \alpha$.
  - ▷ Boolean combination of inequalities φ.
  - ▷ Orthogonal to percentiles queries.
  - ▷ Single-dimensional MDPs and single target T.
  - ▷ Threshold α bounds the probability of the whole event φ whereas SSP-PQ analyze each event independently.
  - ▷ Incomparable in general, SSP-P as a common subclass.

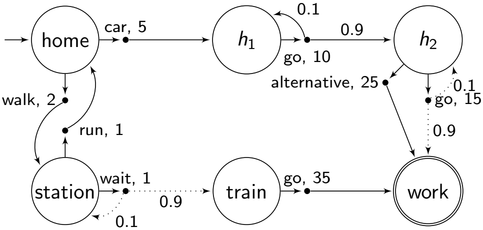
- SSP-PQ is undecidable for arbitrary weights in multi-dimensional MDPs, even with a unique target set [RRS14a].

Percentile queries: other payoff functions

In [RRS14a], we study a wide range of payoffs: reachability, inf, sup, $\liminf$, $\limsup$, mean-payoff, shortest path (truncated sum), discounted sum.

- In the most general setting, complexity is at most EXPTIME.
- Only PTIME for *fixed query size* for all payoffs but the discounted sum.
- Reduced complexity for single-dimension or single-constraint queries.
- Most technically involved cases are infimum mean-payoff and discounted sum.

Imperfect *a priori* knowledge of the environment

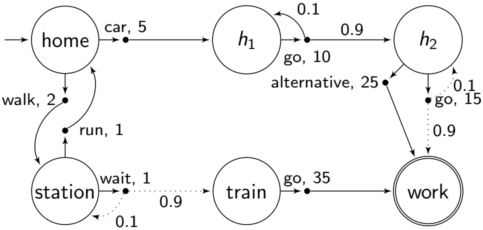


Probabilities represent a *model* of the environment.

- ▷ Probability of a train coming $\neq$ when there is a strike.
- ▷ We may not know about the strike...

How to synthesize strategies with guarantees against several $\neq$ environments (e.g., strike or not)?

Imperfect *a priori* knowledge of the environment



Four possible environments, no a priori knowledge of which one we face:

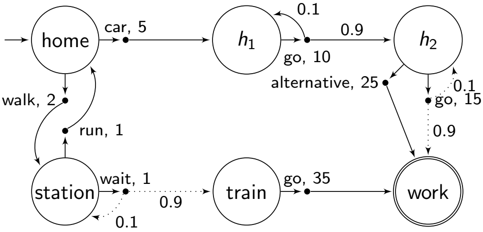
() no problem,

(S) strike (no train) ⇒ wait always leads back to station,

(A) accident (highway blocked) ⇒ go from h_2 always stays in h_2,

(AS) both.

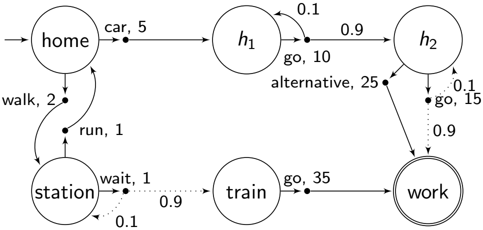
Imperfect *a priori* knowledge of the environment



Specification: we want σ such that

- $\mathbb{P}^{\sigma}_{D}[\mathsf{TS}^{T} \leq 40] \geq 0.95$,
- $\mathbb{P}^{\sigma}_{D(S)}[\mathsf{TS}^{T} \leq 50] \geq 0.95$,
- $\mathbb{P}^{\sigma}_{D(A)}[\mathsf{TS}^{T} \leq 40] \geq 0.95$,
- $\mathbb{P}^{\sigma}_{D(SA)}[\mathsf{TS}^{T} \leq 75] \geq 0.95$.

Imperfect *a priori* knowledge of the environment

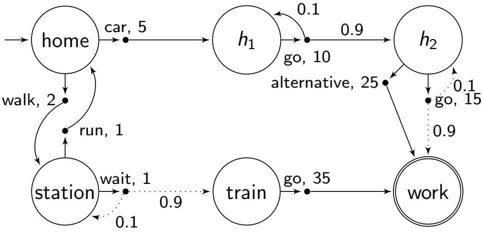


Specification: we want σ such that

- $\mathbb{P}^{\sigma}_{D}[\mathsf{TS}^{T} \leq 40] \geq 0.95$,
- $\mathbb{P}^{\sigma}_{D(S)}[\mathsf{TS}^{T} \leq 50] \geq 0.95$,
- $\mathbb{P}^{\sigma}_{D(A)}[\mathsf{TS}^{T} \leq 40] \geq 0.95$,
- $\mathbb{P}^{\sigma}_{D(SA)}[\mathsf{TS}^{T} \leq 75] \geq 0.95$.

Taking the car right away *does not* ensure to reach work within 40 minutes with probability ≥ 0.95 even when no accident.

Imperfect *a priori* knowledge of the environment



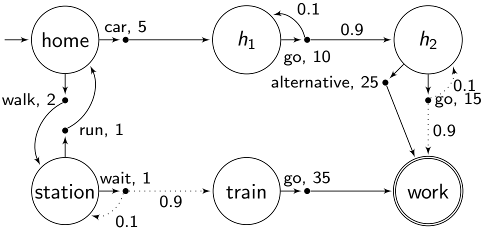
Specification: we want σ such that

- $\mathbb{P}^{\sigma}_{D}[\mathsf{TS}^{T} \leq 40] \geq 0.95$,
- $\mathbb{P}^{\sigma}_{D(A)}[\mathsf{TS}^{T} \leq 40] \geq 0.95$,
- $\mathbb{P}^{\sigma}_{D(S)}[\mathsf{TS}^{T} \leq 50] \geq 0.95$,
- $\mathbb{P}^{\sigma}_{D(SA)}[\mathsf{TS}^{T} \leq 75] \geq 0.95$.

Taking the car right away *does not* ensure to reach work within 40 minutes with probability ≥ 0.95 even when no accident.

Never switching to car means certain doom if strike.

Imperfect *a priori* knowledge of the environment



Specification: we want σ such that

- $\mathbb{P}^{\sigma}_{D}[\mathsf{TS}^{T} \leq 40] \geq 0.95$,
- $\mathbb{P}^{\sigma}_{D(S)}[\mathsf{TS}^{T} \leq 50] \geq 0.95$,
- $\mathbb{P}^{\sigma}_{D(A)}[\mathsf{TS}^{T} \leq 40] \geq 0.95$,
- $\mathbb{P}^{\sigma}_{D(SA)}[\mathsf{TS}^{T} \leq 75] \geq 0.95$.

Sample strategy:

▷ go to the station and wait twice,

▷ if no train, go back and take car,

▷ take alternative road *if* we failed to progress twice using go.

SSP-ME: multi-environment MDPs (1/2)

SSP-ME problem

Given single-dimensional multi-environment MDP $D = (S, s_{\text{init}}, A, (\delta_i)_{1 \leq i \leq k}, (w_i)_{1 \leq i \leq k})$, target set T, thresholds $\ell_1, \ldots, \ell_k \in \mathbb{N}$, and probabilities $\alpha_1, \ldots, \alpha_k \in [0,1] \cap \mathbb{Q}$, decide if there exists a strategy σ satisfying

$$\forall i \in \{1, \ldots, k\},\ \mathbb{P}^{\sigma}_{D_i}[\mathsf{TS}^T \leq \ell_i] \geq \alpha_i.$$

Focus on qualitative variants.

- Almost-sure: $\alpha_1 = \ldots = \alpha_k = 1$.
- Limit-sure: answer is $\textsc{Yes}$ for all $(\alpha_1, \ldots, \alpha_k) \in]0,1[^k$

SSP-ME: multi-environment MDPs (2/2)

Theorem [RS14]

The almost-sure and limit-sure SSP-ME problems can be solved in pseudo-polynomial time for a fixed number of environments. Pure finite memory suffices for the almost-sure case, and a family of finite-memory strategies that witnesses the limit-sure problem can be computed.

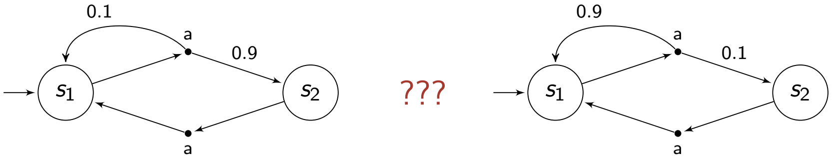
In the quantitative case, *approximate* version of the problem.

Theorem [RS14]

The SSP-ME problem and the ε-gap SSP-ME are NP-hard. For any $\varepsilon > 0$, there is a procedure for the ε-gap SSP-ME problem.

SSP-ME: learning components

Key idea: identify learning components that can be used to determine almost-surely (resp. limit-surely) the current environment.



- By playing long enough, one can guess the environment with arbitrarily high probability (but < 1).

SSP-ME: learning components

Key idea: identify learning components that can be used to determine almost-surely (resp. limit-surely) the current environment.

- ▷ One move suffices to determine the environment with certainty.

SSP-ME: wrap-up

SSP	complexity	strategy
SSP-E	PTIME	pure memoryless
SSP-P	pseudo-PTIME / PSPACE-h.	pure pseudo-poly.
SSP-G	PTIME	pure memoryless
SSP-WE	pseudo-PTIME / NP-h.	pure pseudo-poly.
SSP-PQ	EXPTIME (p.-PTIME) / PSPACE-h.	randomized exponential
SSP-ME (qual. fixed #)	pseudo-PTIME	pure pseudo-poly.

- Study of [RS14] limited to reachability, safety and parity objectives with two environments.

Summary: stochastic shortest path problem

- **SSP-E:** minimize the expected sum to target.
  - ▷ Actual outcomes may vary greatly.

Summary: stochastic shortest path problem

- **SSP-E:** minimize the expected sum to target.
  - ▷ Actual outcomes may vary greatly.
- **SSP-P:** maximize the probability of acceptable performance.
  - ▷ No control over the quality of bad runs, no average-case performance.

Summary: stochastic shortest path problem

- **SSP-E:** minimize the expected sum to target.
  - ▷ Actual outcomes may vary greatly.
- **SSP-P:** maximize the probability of acceptable performance.
  - ▷ No control over the quality of bad runs, no average-case performance.
- **SP-G:** maximize the worst-case performance, extreme risk-aversion.
  - ▷ Strict worst-case guarantees, no average-case performance.

Summary: stochastic shortest path problem

- **SSP-E:** minimize the expected sum to target.
  - ▷ Actual outcomes may vary greatly.
- **SSP-P:** maximize the probability of acceptable performance.
  - ▷ No control over the quality of bad runs, no average-case performance.
- **SP-G:** maximize the worst-case performance, extreme risk-aversion.
  - ▷ Strict worst-case guarantees, no average-case performance.
- **SSP-WE:** SSP-E $\cap$ SP-G.
  - ▷ Based on beyond worst-case synthesis [BFRR14b, BFRR14a].

Summary: stochastic shortest path problem

- **SSP-E:** minimize the expected sum to target.
  - ▷ Actual outcomes may vary greatly.
- **SSP-P:** maximize the probability of acceptable performance.
  - ▷ No control over the quality of bad runs, no average-case performance.
- **SP-G:** maximize the worst-case performance, extreme risk-aversion.
  - ▷ Strict worst-case guarantees, no average-case performance.
- **SSP-WE:** SSP-E ∩ SP-G.
  - ▷ Based on beyond worst-case synthesis [BFRR14b, BFRR14a].
- **SSP-PQ:** extends SSP-P to multi-constraint percentile queries [RRS14a].
  - ▷ Multi-dimensional, flexible, trade-offs.

Summary: stochastic shortest path problem

- **SSP-E:** minimize the expected sum to target.
  - ▷ Actual outcomes may vary greatly.
- **SSP-P:** maximize the probability of acceptable performance.
  - ▷ No control over the quality of bad runs, no average-case performance.
- **SP-G:** maximize the worst-case performance, extreme risk-aversion.
  - ▷ Strict worst-case guarantees, no average-case performance.
- **SSP-WE:** SSP-E ∩ SP-G.
  - ▷ Based on beyond worst-case synthesis [BFRR14b, BFRR14a].
- **SSP-PQ:** extends SSP-P to multi-constraint percentile queries [RRS14a].
  - ▷ Multi-dimensional, flexible, trade-offs.
- **SSP-ME:** multi-environment MDPs [RS14].
  - ▷ Overcomes uncertainty about the stochastic model.

Thank you! Any question?

References I

Véronique Bruyère, Emmanuel Filiot, Mickael Randour, and Jean-François Raskin.
Expectations or guarantees? I want it all! A crossroad between games and MDPs.
In Proc. of SR, EPTCS 146, pages 1–8, 2014.

Véronique Bruyère, Emmanuel Filiot, Mickael Randour, and Jean-François 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.

Thomas Brihaye, Gilles Geeraerts, Axel Haddad, and Benjamin Monmege.
To reach or not to reach? Efficient algorithms for total-payoff games.
CoRR, abs/1407.5030, 2014.

Dimitri P. Bertsekas and John N. Tsitsiklis.
An analysis of stochastic shortest path problems.
Mathematics of Operations Research, 16(3):580–595, 1991.

Krishnendu Chatterjee, Laurent Doyen, Mickael Randour, and Jean-François Raskin.
Looking at mean-payoff and total-payoff through windows.
In Proc. of ATVA, LNCS 8172, pages 118–132. Springer, 2013.

Boris V. Cherkassky, Andrew V. Goldberg, and Tomasz Radzik.
Shortest paths algorithms: Theory and experimental evaluation.
Math. programming, 73(2):129–174, 1996.

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

References II

Kousha Etessami, Marta Z. Kwiatkowska, Moshe Y. Vardi, and Mihalis Yannakakis.
Multi-objective model checking of Markov decision processes.
LMCS, 4(4), 2008.

Andrzej Ehrenfeucht and Jan Mycielski.
Positional strategies for mean payoff games.
International Journal of Game Theory, 8:109–113, 1979.

Emmanuel Filiot, Raffaella Gentilini, and Jean-François Raskin.
Quantitative languages defined by functional automata.
In Proc. of CONCUR, LNCS 7454, pages 132–146. Springer, 2012.

Vojtěch Forejt, Marta Kwiatkowska, Gethin Norman, David Parker, and Hongyang Qu.
Quantitative multi-objective verification for probabilistic systems.
In Proc. of TACAS, LNCS 6605, pages 112–127. Springer, 2011.

Christoph Haase and Stefan Kiefer.
The odds of staying on budget.
CoRR, abs/1409.8228, 2014.

Serge Haddad and Benjamin Monmege.
Reachability in MDPs: Refining convergence of value iteration.
In Joël Ouaknine, Igor Potapov, and James Worrell, editors, Reachability Problems - 8th International Workshop, RP 2014, Oxford, UK, September 22-24, 2014. Proceedings, volume 8762 of Lecture Notes in Computer Science, pages 125–137. Springer, 2014.

References III

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

Leonid Khachiyan, Endre Boros, Konrad Borys, Khaled M. Elbassioni, Vladimir Gurvich, Gábor Rudolf, and Jihui Zhao.
On short paths interdiction problems: Total and node-wise limited interdiction.
pages 204–233, 2008.

Yoshio Ohtsubo.
Optimal threshold probability in undiscounted Markov decision processes with a target set.
Applied Math. and Computation, 149(2):519 – 532, 2004.

Mickael Randour, Jean-François Raskin, and Ocan Sankur.
Percentile queries in multi-dimensional Markov decision processes.
CoRR, abs/1410.4801, 2014.

Mickael Randour, Jean-François Raskin, and Ocan Sankur.
Variations on the stochastic shortest path problem.
CoRR, abs/1411.0835, 2014.

Mickael Randour, Jean-François Raskin, and Ocan Sankur.
Variations on the stochastic shortest path problem.
In Proc. of VMCAI, LNCS. Springer, 2015.

References IV

Jean-François Raskin and Ocan Sankur.
Multiple-environment Markov decision processes.
In Proc. of FSTTCS, LIPIcs. Schloss Dagstuhl - LZI, 2014.

Masahiko Sakaguchi and Yoshio Ohtsubo.
Markov decision processes associated with two threshold probability criteria.
Journal of Control Theory and Applications, 11(4):548–557, 2013.

Michael Ummels and Christel Baier.
Computing quantiles in Markov reward models.
In Proc. of FOSSACS, LNCS 7794, pages 353–368. Springer, 2013.