On classical and non-classical stochastic path problems

Christel Baier
Technische Universität Dresden

Weighted Markovian models

Weighted Markovian models

natural model to reason about costs, energy, utility, ...

Weighted Markovian models

natural model to reason about costs, energy, utility, ...

- undecidability results for two or more weight functions

  ... close connection to two-counter machines ...

Weighted Markovian models

natural model to reason about costs, energy, utility, ...

- undecidability results for two or more weight functions
- but also various decidability/computability results

Weighted Markovian models

natural model to reason about costs, energy, utility, ...

- undecidability results for two or more weight functions
- but also various decidability/computability results
  - finite-horizon or finite-window objectives

  ... ensure decidability by considering weight constraints
  only for path fragments up to a fixed length

Weighted Markovian models

natural model to reason about costs, energy, utility, ...

- undecidability results for two or more weight functions
- but also various decidability/computability results
  - finite-horizon or finite-window objectives
  - mean-payoff, long-run ratios

Markov chains:	linear equation systems, analysis of BSCCs
MDPs:	linear programs, analysis of end components
	fractional programs for long-run ratios

Weighted Markovian models

natural model to reason about costs, energy, utility, ...

- undecidability results for two or more weight functions
- but also various decidability/computability results
  - finite-horizon or finite-window objectives
  - mean-payoff, long-run ratios
  - for models with non-negative weights
    model checking for various logics

... monotonicity of accumulated weights ...

Weighted Markovian models

natural model to reason about costs, energy, utility, ...

- undecidability results for two or more weight functions
- but also various decidability/computability results
  - ∗ finite-horizon or finite-window objectives
  - ∗ mean-payoff, long-run ratios
  - ∗ for models with non-negative weights
    model checking for various logics
  - ∗ for models with a single weight function
    expected accumulated weights, weight-bounded properties, ...

... close to one-counter machines ...

Outline

- weighted Markov decision processes
- mean-payoff and long-run ratios
- expected accumulated weights
- conditional expected accumulated rewards
- weight-bounded reachability and quantiles
- LTL with weight assertions
- conclusions

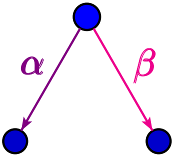
From TS and MC to MDP

TS:	transition system
MC:	Markov chain
MDP:	Markov decision process

From TS and MC to MDP

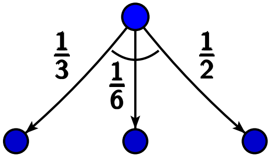
transition system

purely nondeterministic



Markov chain

purely probabilistic



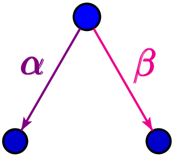
α, β are action names

TS: transition system
MC: Markov chain
MDP: Markov decision process

From TS and MC to MDP

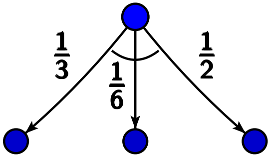
transition system

purely nondeterministic

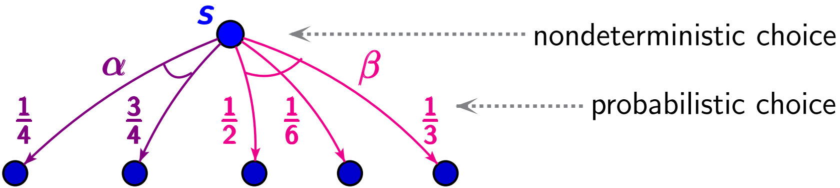


Markov chain

purely probabilistic



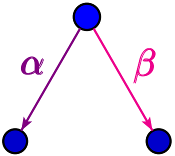
Markov decision process (MDP)



From TS and MC to MDP

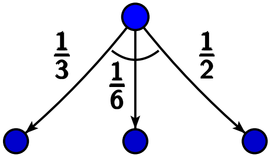
transition system

purely nondeterministic

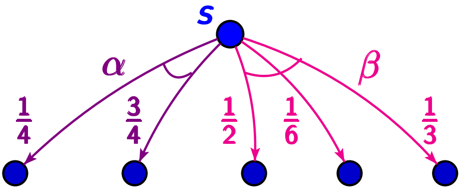


Markov chain

purely probabilistic



Markov decision process (MDP)



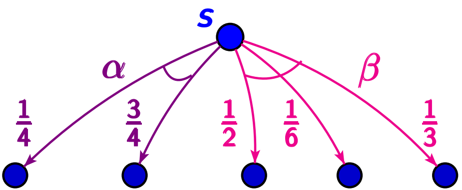
integer weights

$wgt(s,\alpha) \in \mathbb{Z}$

Markov decision process (MDP)

$$\mathcal{M} = (S, Act, P, \ldots)$$

- finite state space S
- Act finite set of actions



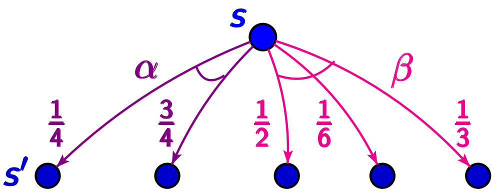
Markov decision process (MDP)

$\mathcal{M} = (S, Act, P, \ldots)$

- finite state space S
- Act finite set of actions
- transition probability fct. $P : S \times Act \times S \to [0,1]$

$$\forall s \in S \ \ \forall \alpha \in Act. \ \sum_{s' \in S} P(s, \alpha, s') \in \{0, 1\}$$

$\alpha \notin Act(s)$ $\alpha \in Act(s)$



nondeterministic choice between enabled actions

$Act(s) = \{\alpha, \beta\}$

Markov decision process (MDP)

$\mathcal{M} = (S, Act, P, rew_1, rew_2, \ldots)$

- finite state space S
- Act finite set of actions
- transition probability fct. $P : S \times Act \times S \to [0, 1]$

$$\forall s \in S \ \ \forall \alpha \in Act. \ \ \sum_{s' \in S} P(s, \alpha, s') \in \{0, 1\}$$

- reward functions rew_1, rew_2, $\ldots : S \times Act \to \mathbb{N}$

energy utility

Weighted MDP

$$\mathcal{M} = (S, Act, P, wgt_1, wgt_2, \ldots)$$

- finite state space S
- Act finite set of actions
- transition probability fct. $P : S \times Act \times S \to [0,1]$

$$\forall s \in S \ \ \forall \alpha \in Act. \ \ \sum_{s' \in S} P(s, \alpha, s') \in \{0,1\}$$

- weight functions wgt_1, wgt_2, $\ldots : S \times Act \to \mathbb{Z}$

energy level
of a battery

win and loss
of a share at the
stock market

Weighted MDP

$$\mathcal{M} = (S, Act, P, wgt_1, wgt_2, \ldots)$$

- finite state space S
- Act finite set of actions
- transition probability fct. $P : S \times Act \times S \to [0,1]$

$$\forall s \in S \ \ \forall \alpha \in Act. \ \sum_{s' \in S} P(s, \alpha, s') \in \{0, 1\}$$

- weight functions wgt_1, wgt_2, $\ldots : S \times Act \to \mathbb{Z}$

accumulated weight of finite paths:

$$wgt_1(s_0 \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_n} s_n) \ = \ \sum_{i=0}^{n-1} wgt_1(s_i, \alpha_{i+1})$$

Weighted MDP

$$\mathcal{M} = (S, Act, P, wgt_1, wgt_2, \ldots)$$

- finite state space S
- Act finite set of actions
- transition probability fct. $P : S \times Act \times S \to [0,1]$

$$\forall s \in S \;\; \forall \alpha \in Act. \;\; \sum_{s' \in S} P(s, \alpha, s') \in \{0,1\}$$

- weight functions wgt_1, wgt_2, $\ldots : S \times Act \to \mathbb{Z}$

ratios of accumulated weights:

$$ratio = \frac{cost}{util} : FinPaths \to \mathbb{Q} \qquad \begin{array}{lcl} cost & = & wgt_1 \\ util & = & wgt_2 \end{array}$$

Probability measure

$$\mathcal{M} = (S, Act, P, wgt_1, wgt_2, \ldots)$$

- finite state space S
- Act finite set of actions
- transition probability fct. $P : S \times Act \times S \to [0,1]$

$$\forall s \in S \;\; \forall \alpha \in Act. \;\; \sum_{s' \in S} P(s, \alpha, s') \in \{0,1\}$$

- weight functions wgt_1, wgt_2, $\ldots : S \times Act \to \mathbb{Z}$

---

probabilities measure $\Pr_s^\sigma$ for given state $s \in S$ and scheduler $\sigma : \underbrace{FinPaths}_{\text{history}} \to \underbrace{Distr(Act)}_{\text{probabilities for next actions}}$

Classification of schedulers

randomized vs deterministic schedulers:

randomized (R):	select a distribution of actions
deterministic (D):	select a unique action

Classification of schedulers

randomized vs deterministic schedulers:

randomized (R):	select a distribution of actions
deterministic (D):	select a unique action

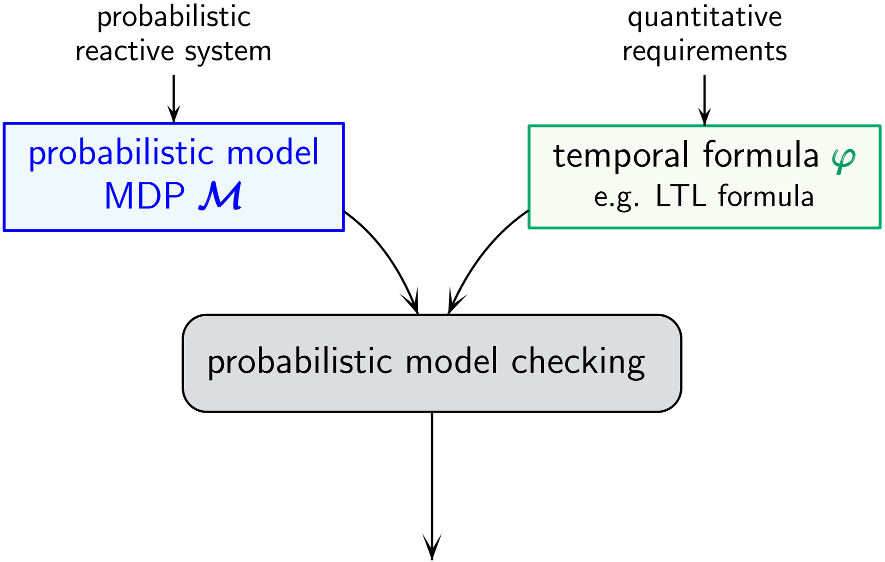
memory requirements:

consider schedulers as triples (Mem, μ, ν)

- Mem is a set of memory cells
- $\mu : \mathit{Mem} \times S \to \mathit{Distr}(\mathit{Act})$ decision function
- $\nu : \mathit{Mem} \times S \to \mathit{Mem}$ memory-update function

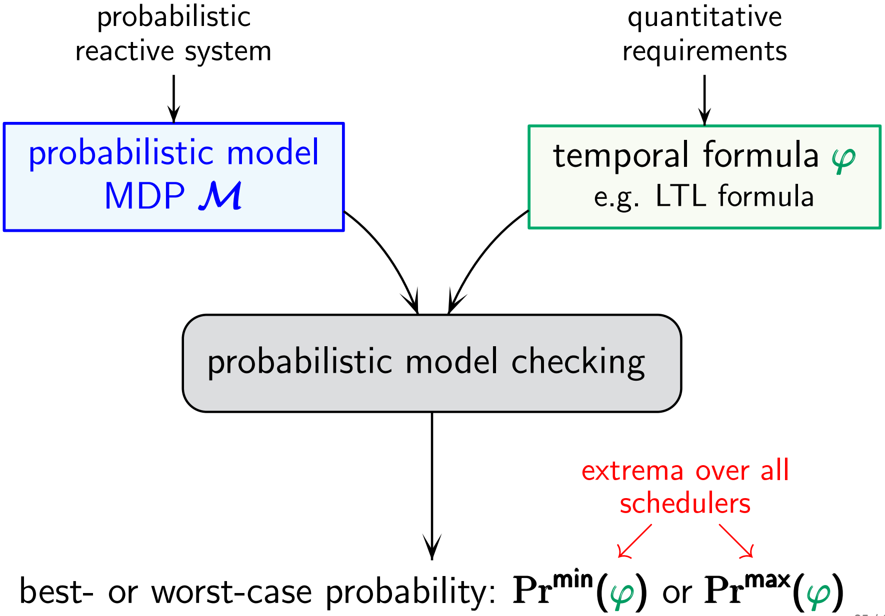
no restriction (H):	possibly infinitely many memory cells
finite-memory (FM):	finitely many memory cells
memoryless (M):	decisions only depend on the current state

Probabilistic model checking

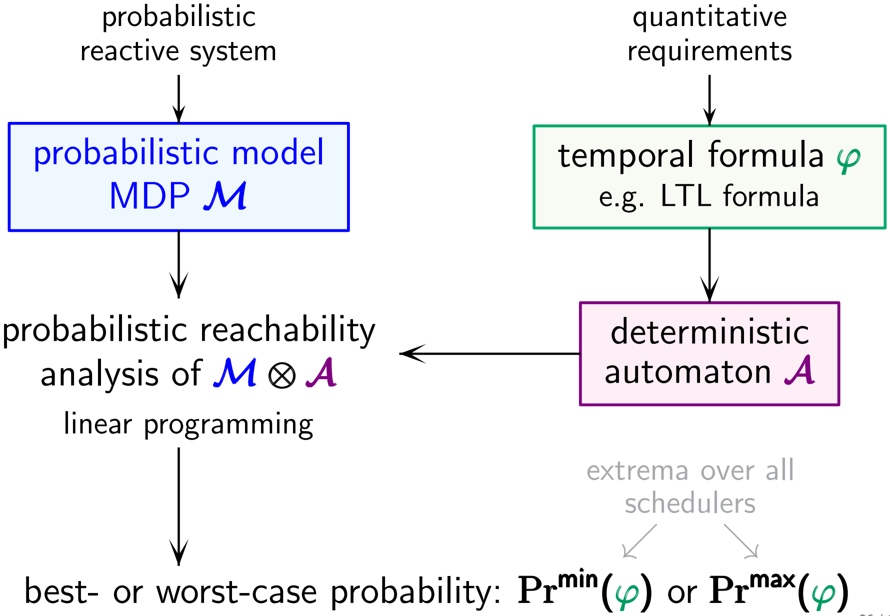


best- or worst-case probability: $\mathrm{Pr}^{\mathsf{min}}(\varphi)$ or $\mathrm{Pr}^{\mathsf{max}}(\varphi)$

Probabilistic model checking



Probabilistic model checking



Probabilistic model checking

probabilistic
reactive system

probabilistic model
MDP $\mathcal{M}$

quantitative
requirements

temporal formula φ
e.g. LTL formula

deterministic
automaton $\mathcal{A}$

probabilistic reachability
analysis of $\mathcal{M} \otimes \mathcal{A}$

linear programming

$$\Pr^{\max}_{\mathcal{M},s}(\varphi) \;=\; \Pr^{\max}_{\mathcal{M}\otimes\mathcal{A},s'}\big(\Diamond\, accEC\,\big)$$

maximal probability
to reach an accepting
end component

End components (EC) [de Alfaro'96]

End components (EC) [de Alfaro'96]

Let $\mathcal{M} = (S, Act, P, \ldots)$ be an MDP.

An *end component* of $\mathcal{M}$ is a strongly connected sub-MDP

End components (EC) [de Alfaro'96]

Let $\mathcal{M} = (S, Act, P, \ldots)$ be an MDP.

An *end component* of $\mathcal{M}$ is a strongly connected sub-MDP, i.e., a pair $\mathcal{E} = (T, A)$ where $\varnothing \neq T \subseteq S$ and $A : T \to 2^{Act}$ s.t.

(1) ...

(2) ...

(3) ...

Let $\mathcal{M} = (S, Act, P, \ldots)$ be an MDP.

An *end component* of $\mathcal{M}$ is a strongly connected sub-MDP, i.e., a pair $\mathcal{E} = (T, A)$ where $\varnothing \neq T \subseteq S$ and $A : T \to 2^{Act}$ s.t.

(1) enabledness of selected actions:

$$\varnothing \neq A(t) \subseteq Act(t) \quad \text{for all } t \in T$$

(2) ...

(3) ...

End components (EC) [de Alfaro'96]

Let $\mathcal{M} = (S, Act, P, \ldots)$ be an MDP.

An *end component* of $\mathcal{M}$ is a strongly connected sub-MDP, i.e., a pair $\mathcal{E} = (T, A)$ where $\varnothing \neq T \subseteq S$ and $A : T \to 2^{Act}$ s.t.

(1) enabledness of selected actions:

$$\varnothing \neq A(t) \subseteq Act(t) \quad \text{for all } t \in T$$

(2) closed under probabilistic branching:

$$\forall t \in T \, \forall \alpha \in A(t). \, \big(P(t, \alpha, u) > 0 \implies u \in T\big)$$

(3) ...

End components (EC) [de Alfaro'96]

Let $\mathcal{M} = (S, Act, P, \ldots)$ be an MDP.

An *end component* of $\mathcal{M}$ is a strongly connected sub-MDP, i.e., a pair $\mathcal{E} = (T, A)$ where $\varnothing \neq T \subseteq S$ and $A : T \to 2^{Act}$ s.t.

(1) enabledness of selected actions:

$$\varnothing \neq A(t) \subseteq Act(t) \quad \text{for all } t \in T$$

(2) closed under probabilistic branching:

$$\forall t \in T\, \forall \alpha \in A(t).\ \big(P(t, \alpha, u) > 0 \Longrightarrow u \in T\big)$$

(3) the underlying graph is strongly connected

End components (EC) [de Alfaro'96]

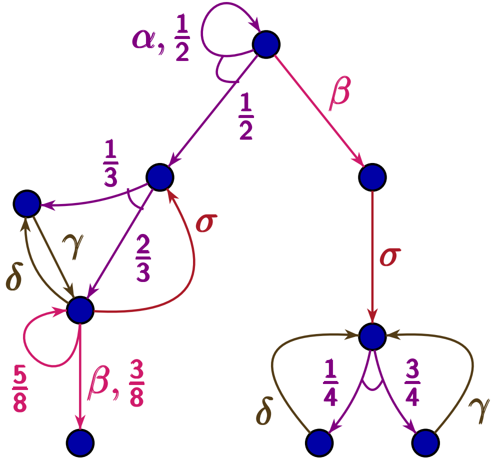
Let $\mathcal{M} = (S, Act, P, \ldots)$ be an MDP.

An *end component* of $\mathcal{M}$ is a strongly connected sub-MDP, i.e., a pair $\mathcal{E} = (T, A)$ where $\varnothing \neq T \subseteq S$ and $A : T \to 2^{Act}$ s.t. $\ldots$

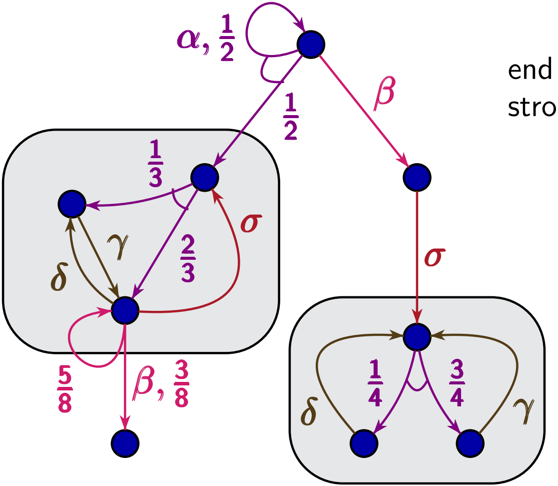
Often viewed as a set of state-action pairs:

$$\mathcal{E} \; = \; \{\, (s, \alpha) \;:\; s \in T,\; \alpha \in A(s) \,\}$$

End components (EC) [de Alfaro'96]

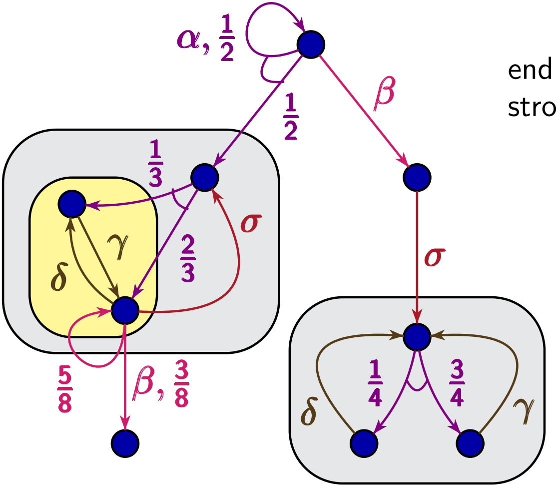


End components (EC) [de Alfaro'96]



end component (EC):
strongly connected sub-MDP

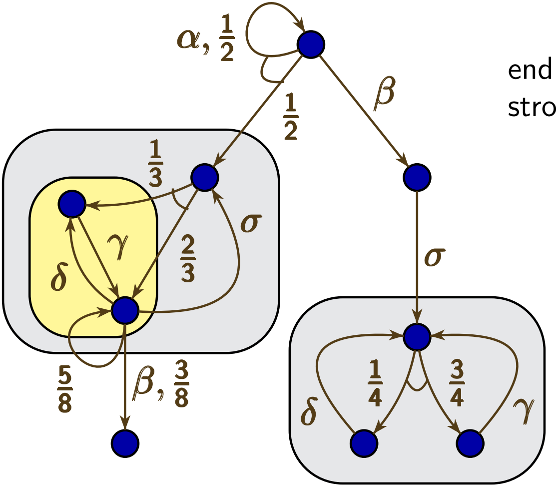
End components (EC) [DE ALFARO'96]



end component (EC):
strongly connected sub-MDP

End components (EC) [de Alfaro'96]

For all schedulers: almost all infinite paths eventually enter an EC and visit all its states infinitely often.



end component (EC):
strongly connected sub-MDP

End components (EC) ... for MDPs without traps

For all schedulers: almost all infinite paths eventually enter an EC and visit all its states infinitely often.

More precisely, for all schedulers σ and states s:

$$\mathrm{Pr}^{\sigma}\left\{ \pi \in \mathit{Paths}(s) : \begin{array}{l} \mathit{limit}(\pi) \text{ is an} \\ \text{end component} \end{array} \right\} = 1$$

limit of an infinite path π:

$$\mathit{limit}(\pi) = \left\{ \begin{array}{l} \text{set of state-action pairs that} \\ \text{appear infinitely often in } \pi \end{array} \right.$$

trap: state without actions

End components (EC) ... for MDPs without traps

For all schedulers: almost all infinite paths eventually enter an EC and visit all its states infinitely often.

More precisely, for all schedulers σ and states s:

$$\Pr^{\sigma}\left\{ \pi \in \mathit{Paths}(s) : \begin{array}{l} \mathit{limit}(\pi) \text{ is an} \\ \text{end component} \end{array} \right\} = 1$$

Let E be a limit property and $T_1, \ldots, T_k \subseteq S$ s.t.

$$\pi \models E \quad \text{iff} \quad \exists i \geqslant 0.\ \underbrace{\inf(\pi)}_{\substack{\text{set of states that appear} \\ \text{infinitely often in } \pi}} = T_i$$

End components (EC) ... for MDPs without traps

For all schedulers: almost all infinite paths eventually enter an EC and visit all its states infinitely often.

More precisely, for all schedulers σ and states s:

$$\Pr^{\sigma}\left\{ \pi \in \mathit{Paths}(s) : \begin{array}{l} \mathit{limit}(\pi) \text{ is an} \\ \text{end component} \end{array} \right\} = 1$$

Let E be a limit property and $T_1, \ldots, T_k \subseteq S$ s.t.

$$\pi \models E \quad \text{iff} \quad \exists i \geqslant 0.\ \inf(\pi) = T_i$$

Then: $\Pr_s^{\max}(E) = \Pr_s^{\max}(\Diamond T)$ where

$$T = \bigcup \left\{ T_i : T_i \text{ constitutes an end component} \right\}$$

Quantitative analysis of Rabin conditions

Quantitative analysis of Rabin conditions

Let E be a Rabin condition $\bigvee_{1 \leqslant i \leqslant k} (\Diamond\Box\neg L_i \wedge \Box\Diamond U_i)$.

$\Diamond$	eventually	$\Diamond\Box$	almost forever
$\Box$	always	$\Box\Diamond$	infinitely often

Quantitative analysis of Rabin conditions

Let E be a Rabin condition $\bigvee_{1 \leqslant i \leqslant k} (\Diamond\Box\neg L_i \wedge \Box\Diamond U_i)$.

$$\Pr_s^{\max}(E) \;=\; \Pr_s^{\max}(\,\Diamond\, accEC\,)$$

$\Diamond$ eventually $\quad$ $\Diamond\Box$ almost forever

$\Box$ always $\quad$ $\Box\Diamond$ infinitely often

Quantitative analysis of Rabin conditions

Let E be a Rabin condition $\bigvee_{1 \leqslant i \leqslant k} (\Diamond\Box\neg L_i \wedge \Box\Diamond U_i)$.

$$\Pr_s^{\max}(E) \;=\; \Pr_s^{\max}(\,\Diamond\, accEC\,)$$

union of all end components T that "meet E", i.e.,

$\exists i \in \{1, \ldots, k\}.\;\; T \cap L_i = \varnothing$ and $T \cap U_i \neq \varnothing$

$\Diamond$	eventually	$\Diamond\Box$	almost forever
$\Box$	always	$\Box\Diamond$	infinitely often

Quantitative analysis of Rabin conditions

Let E be a Rabin condition $\bigvee_{1 \leqslant i \leqslant k} (\Diamond\Box\neg L_i \wedge \Box\Diamond U_i)$.

$$\begin{aligned} \Pr_s^{\max}(E) &= \Pr_s^{\max}(\ \Diamond\, accEC\ ) \\ &= \Pr_s^{\max}(\ \Diamond\, accMEC\ ) \end{aligned}$$

$\bigcup_{1 \leqslant i \leqslant k}$ union of all maximal end components T in $\mathcal{M} \setminus L_i$ s.t. $T \cap U_i \neq \varnothing$

$\Diamond$ eventually $\quad$ $\Diamond\Box$ almost forever

$\Box$ always $\quad$ $\Box\Diamond$ infinitely often

Quantitative analysis of Rabin conditions

Let E be a Rabin condition $\bigvee_{1 \leqslant i \leqslant k} (\Diamond\Box\neg L_i \wedge \Box\Diamond U_i)$.

$$\begin{aligned}
\Pr_s^{\max}(E) &= \Pr_s^{\max}(\,\Diamond\, accEC\,) \\
&= \Pr_s^{\max}(\,\Diamond\, accMEC\,)
\end{aligned}$$

$\bigcup_{1 \leqslant i \leqslant k}$ union of all maximal end components T in $\mathcal{M} \setminus L_i$ s.t. $T \cap U_i \neq \varnothing$

analogous approach for generalized Rabin conditions:

$$\bigvee_{1 \leqslant i \leqslant k} (\,\Diamond\Box\neg L_i \,\wedge\, \Box\Diamond U_{i,1} \,\wedge \ldots \wedge\, \Box\Diamond U_{i,k_i}\,)$$

Quantitative analysis of Rabin conditions

Let E be a Rabin condition $\bigvee_{1 \leqslant i \leqslant k} (\Diamond\Box\neg L_i \wedge \Box\Diamond U_i)$.

$$\begin{aligned}
\Pr_s^{\max}(E) &= \Pr_s^{\max}(\Diamond\, accEC\,) \\
&= \Pr_s^{\max}(\Diamond\, accMEC\,)
\end{aligned}$$

---

model checking algorithm for Rabin condition E:

1. compute the maximal end components
2. check which of them fulfills E
3. compute maximal reachability probabilities

   by linear-programming techniques

Quantitative analysis of Rabin conditions

Let E be a Rabin condition $\bigvee_{1 \leqslant i \leqslant k} (\Diamond\Box\neg L_i \wedge \Box\Diamond U_i)$.

$$\begin{aligned}\Pr_s^{\max}(E) &= \Pr_s^{\max}(\Diamond\, accEC\,) \\ &= \Pr_s^{\max}(\Diamond\, accMEC\,)\end{aligned}$$

---

model checking algorithm for Rabin condition E:

1. compute the maximal end components
2. check which of them fulfills E
3. compute maximal reachability probabilities
   by linear-programming techniques

Computation of maximal end components

maximal end component (MEC):
  end component that is not contained in any other end component

Computation of maximal end components

REPEAT

compute the SCCs of $\mathcal{M}$;

maximal end component (MEC):

end component that is not contained in any other end component

Computation of maximal end components

REPEAT

compute the SCCs of $\mathcal{M}$;

IF there exist states s, t and an action α such that $P(s, \alpha, t) > 0$ and s, t belong to different SCCs

maximal end component (MEC):

end component that is not contained in any other end component

Computation of maximal end components

REPEAT

compute the SCCs of $\mathcal{M}$;

IF there exist states s, t and an action α such that $P(s, \alpha, t) > 0$ and s, t belong to different SCCs

THEN choose such a pair $\langle s, \alpha \rangle$;
remove α from $Act(s)$;

maximal end component (MEC):
end component that is not contained in any other end component

Computation of maximal end components

REPEAT

compute the SCCs of $\mathcal{M}$;

IF there exist states s, t and an action α such that $P(s, \alpha, t) > 0$ and s, t belong to different SCCs

THEN choose such a pair $\langle s, \alpha \rangle$;

remove α from $Act(s)$;

IF s is a trap THEN remove s FI

maximal end component (MEC):

end component that is not contained in any other end component

Computation of maximal end components

REPEAT

compute the SCCs of $\mathcal{M}$;

IF there exist states s, t and an action α such that $P(s, \alpha, t) > 0$ and s, t belong to different SCCs

THEN choose such a pair $\langle s, \alpha \rangle$;

remove α from $Act(s)$;

IF s is a trap THEN remove s FI

FI

UNTIL no further changes

Computation of maximal end components

```
REPEAT
    compute the SCCs of M;
    IF  there exist states s, t and an action α such that
        P(s, α, t) > 0 and s, t belong to different SCCs
    THEN  choose such a pair ⟨s, α⟩;
          remove α from Act(s);
          IF s is a trap  THEN  remove s  FI
    FI
UNTIL no further changes
```

return the non-trivial SCCs as maximal end components

Computation of maximal end components

REPEAT

compute the SCCs of $\mathcal{M}$;

IF there exist states s, t and an action α such that
$P(s, \alpha, t) > 0$ and s, t belong to different SCCs

THEN choose such a pair $\langle s, \alpha \rangle$;
remove α from $Act(s)$;
IF s is a trap THEN remove s FI

FI

UNTIL no further changes

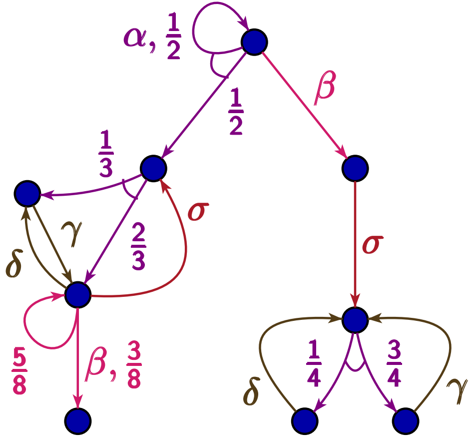
time complexity:
$\mathcal{O}(\, size(\mathcal{M})^2 \,)$

return the non-trivial SCCs as maximal end components

MEC-quotient

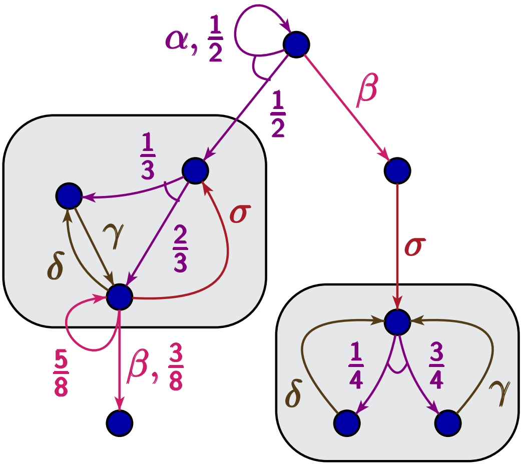
Idea: The MEC-quotient is the MDP $\mathbf{MEC}(\mathcal{M})$ resulting from $\mathcal{M}$ by collapsing all MECs into a single state.

MEC-quotient



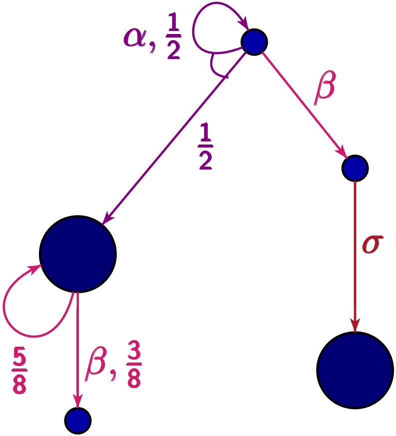
Idea: The MEC-quotient is the MDP $\mathrm{MEC}(\mathcal{M})$ resulting from $\mathcal{M}$ by collapsing all MECs into a single state.

MEC-quotient



Idea: The MEC-quotient is the MDP $\mathrm{MEC}(\mathcal{M})$ resulting from $\mathcal{M}$ by collapsing all MECs into a single state.

MEC-quotient



Idea: The MEC-quotient is the MDP $\mathrm{MEC}(\mathcal{M})$ resulting from $\mathcal{M}$ by collapsing all MECs into a single state.

MEC-quotient

Given MDP $\mathcal{M} = (S, Act, P, \ldots)$ with MECs $\mathcal{E}_1, \ldots, \mathcal{E}_k$ where $\mathcal{E}_i = (T_i, A_i)$.

Idea: The MEC-quotient is the MDP $\mathrm{MEC}(\mathcal{M})$ resulting from $\mathcal{M}$ by collapsing all MECs into a single state.

MEC-quotient

Given MDP $\mathcal{M} = (S, Act, P, \ldots)$ with MECs $\mathcal{E}_1, \ldots, \mathcal{E}_k$ where $\mathcal{E}_i = (T_i, A_i)$. W.l.o.g., if $s, t \in T_i$ then:

$$Act(s) \cap Act(t) \;=\; A_i(s) \cap A_i(t)$$

Idea: The MEC-quotient is the MDP $\mathrm{MEC}(\mathcal{M})$ resulting from $\mathcal{M}$ by collapsing all MECs into a single state.

MEC-quotient

Given MDP $\mathcal{M} = (S, Act, P, \ldots)$ with MECs $\mathcal{E}_1, \ldots, \mathcal{E}_k$ where $\mathcal{E}_i = (T_i, A_i)$. W.l.o.g., if $s, t \in T_i$ then:

$$Act(s) \cap Act(t) \;=\; A_i(s) \cap A_i(t)$$

MEC-quotient $\mathrm{MEC}(\mathcal{M}) = (S', Act, P', \ldots)$ where

$$S' = (S \setminus T) \cup \{\mathcal{E}_1, \ldots, \mathcal{E}_k\} \;\text{ where }\; T = \bigcup_{1 \leqslant i \leqslant k} T_i$$

Idea: The MEC-quotient is the MDP $\mathrm{MEC}(\mathcal{M})$ resulting from $\mathcal{M}$ by collapsing all MECs into a single state.

MEC-quotient

Given MDP $\mathcal{M} = (S, Act, P, \ldots)$ with MECs $\mathcal{E}_1, \ldots, \mathcal{E}_k$ where $\mathcal{E}_i = (T_i, A_i)$. W.l.o.g., if $s, t \in T_i$ then:

$$Act(s) \cap Act(t) \;=\; A_i(s) \cap A_i(t)$$

MEC-quotient $\mathrm{MEC}(\mathcal{M}) = (S', Act, P', \ldots)$ where

$$S' = (S \setminus T) \cup \{\mathcal{E}_1, \ldots, \mathcal{E}_k\} \;\text{ where } T = \bigcup_{1 \leqslant i \leqslant k} T_i$$

enabled actions:

for $s \in S \setminus T$: as in $\mathcal{M}$

for state $\mathcal{E}_i$: all actions in $\bigcup_{s \in T_i} Act(s) \setminus A_i(s)$

MEC-quotient

Given MDP $\mathcal{M} = (S, Act, P, \ldots)$ with MECs $\mathcal{E}_1, \ldots, \mathcal{E}_k$ where $\mathcal{E}_i = (T_i, A_i)$. W.l.o.g., if $s, t \in T_i$ then:

$$Act(s) \cap Act(t) \;=\; A_i(s) \cap A_i(t)$$

MEC-quotient $\mathrm{MEC}(\mathcal{M}) = (S', Act, P', \ldots)$ where

$$S' = (S \setminus T) \cup \{\mathcal{E}_1, \ldots, \mathcal{E}_k\} \quad \text{where } T = \bigcup_{1 \leqslant i \leqslant k} T_i$$

transition probabilities, e.g., if $s \in S \setminus T$, $\alpha \in Act(s)$:

$$P'(s, \alpha, s') \;=\; P(s, \alpha, s') \quad \text{if } s' \in S \setminus T$$

$$P'(s, \alpha, \mathcal{E}_i) \;=\; \sum_{t \in T_i} P(s, \alpha, t)$$

MEC-quotient

Given MDP $\mathcal{M} = (S, Act, P, \ldots)$ with MECs $\mathcal{E}_1, \ldots, \mathcal{E}_k$ where $\mathcal{E}_i = (T_i, A_i)$. W.l.o.g., if $s, t \in T_i$ then:

$$Act(s) \cap Act(t) \;=\; A_i(s) \cap A_i(t)$$

MEC-quotient $\mathrm{MEC}(\mathcal{M}) = (S', Act, P', \ldots)$ where

$$S' = (S \setminus T) \cup \{\mathcal{E}_1, \ldots, \mathcal{E}_k\} \quad \text{where } T = \bigcup_{1 \leqslant i \leqslant k} T_i$$

if $s \in T_i$ and $\alpha \in Act(s) \setminus A_i(s)$:

$$P'(\mathcal{E}_i, \alpha, s') \;=\; P(s, \alpha, s') \quad \text{if } s' \in S \setminus T$$

$$P'(\mathcal{E}_i, \alpha, \mathcal{E}_j) \;=\; \sum_{t \in T_j} P(s, \alpha, t)$$

Properties of the MECs and the MEC-quotient

Properties of the MECs and the MEC-quotient

For all states s, t that belong to the same MEC:

$$\Pr_s^{\max}(\varphi) = \Pr_t^{\max}(\varphi)$$

for each prefix-independent path property φ.

Examples: $\varphi = \Diamond G$ or $\varphi = \Diamond\Box G$ or ...

The same holds for mininimal probabilities for prefix-independent properties and min/max expectations of long-run objectives.

Properties of the MECs and the MEC-quotient

For all states s, t that belong to the same MEC:

$$\Pr_s^{\max}(\varphi) = \Pr_t^{\max}(\varphi)$$

for each prefix-independent path property φ.

Hence: $\mathcal{M}$ and $\mathrm{MEC}(\mathcal{M})$ have the same maximal probabilities for prefix-independent properties.

Examples: $\varphi = \Diamond G$ or $\varphi = \Diamond\Box G$ or ...

The same holds for mininimal probabilities for prefix-independent properties and min/max expectations of long-run objectives.

Properties of the MECs and the MEC-quotient

For all states s, t that belong to the same MEC:

$$\Pr_s^{\max}(\varphi) = \Pr_t^{\max}(\varphi)$$

for each prefix-independent path property φ.

Hence: $\mathcal{M}$ and $\mathrm{MEC}(\mathcal{M})$ have the same maximal probabilities for prefix-independent properties.

$\mathrm{MEC}(\mathcal{M})$ has no end components.

Properties of the MECs and the MEC-quotient

For all states s, t that belong to the same MEC:

$$\Pr^{\max}_{s}(\varphi) = \Pr^{\max}_{t}(\varphi)$$

for each prefix-independent path property φ.

Hence: $\mathcal{M}$ and $\mathrm{MEC}(\mathcal{M})$ have the same maximal probabilities for prefix-independent properties.

$\mathrm{MEC}(\mathcal{M})$ has no end components. Hence:

$$\Pr^{\min}_{\mathrm{MEC}(\mathcal{M}),s}(\Diamond \mathit{Trap}) = 1$$

set of states t with $Act(t) = \varnothing$

Properties of the MECs and the MEC-quotient

For all states s, t that belong to the same MEC:

$$\Pr_s^{\max}(\varphi) = \Pr_t^{\max}(\varphi)$$

for each prefix-independent path property φ.

Hence: $\mathcal{M}$ and $\mathrm{MEC}(\mathcal{M})$ have the same maximal probabilities for prefix-independent properties.

$\mathrm{MEC}(\mathcal{M})$ has no end components. Hence:

$$\Pr^{\min}_{\mathrm{MEC}(\mathcal{M}),s}(\Diamond \mathit{Trap}) = 1$$

... transition probability matrix is contracting ...

Probabilistic model checking

probabilistic
reactive system

probabilistic model
MDP $\mathcal{M}$

quantitative
requirements

temporal formula φ
e.g. LTL formula

deterministic
automaton $\mathcal{A}$

probabilistic reachability
analysis of $\mathcal{M} \otimes \mathcal{A}$

linear programming

$$\Pr^{\max}_{\mathcal{M},s}(\varphi) \;=\; \Pr^{\max}_{\mathcal{M}\otimes\mathcal{A},s'}\big(\Diamond accEC\big)$$

maximal probability
to reach an accepting
end component

Maximal reachability probabilities

Maximal reachability probabilities

given: MDP $\mathcal{M}$ with state space S and $G \subseteq S$

task: compute $x_s = \Pr_s^{\max}(\Diamond G) = \max_{\sigma} \Pr_s^{\sigma}(\Diamond G)$

Maximal reachability probabilities

given: MDP $\mathcal{M}$ with state space S and $G \subseteq S$

task: compute $x_s = \Pr_s^{\max}(\Diamond G) = \max\limits_{\sigma} \Pr_s^{\sigma}(\Diamond G)$

The vector $(x_s)_{s \in S}$ is the least solution in $[0,1]^S$ of the equation system:

$$
\begin{array}{lcll}
x_s & = & 1 & \text{if } s \in G \\
x_s & = & \max\limits_{\alpha} \sum\limits_{t \in S} P(s,\alpha,t) \cdot x_t & \text{otherwise}
\end{array}
$$

α ranges over all actions in $Act(s)$

Maximal reachability probabilities

given: MDP $\mathcal{M}$ with state space S and $G \subseteq S$

task: compute $x_s = \Pr_s^{\max}(\Diamond G) = \max_\sigma \Pr_s^\sigma(\Diamond G)$

The vector $(x_s)_{s \in S}$ is the least solution in $[0,1]^S$ of the equation system:

$$
\begin{array}{lcll}
x_s & = & 1 & \text{if } s \in G \\
x_s & = & 0 & \text{if } s \not\models \exists \Diamond G \\
x_s & = & \max\limits_{\alpha} \sum\limits_{t \in S} P(s, \alpha, t) \cdot x_t & \text{otherwise}
\end{array}
$$

α ranges over all actions in $Act(s)$

Maximal reachability probabilities

given: MDP $\mathcal{M}$ with state space S and $G \subseteq S$

task: compute $x_s = \Pr_s^{\max}(\Diamond G) = \max\limits_{\sigma} \Pr_s^{\sigma}(\Diamond G)$

The vector $(x_s)_{s \in S}$ is the least solution in $[0,1]^S$ of the equation system:

$$
\begin{array}{lcll}
x_s & = & 1 & \text{if } s \in G \\
x_s & = & 0 & \text{if } s \not\models \exists \Diamond G \\
x_s & = & \max\limits_{\alpha} \sum\limits_{t \in S} P(s, \alpha, t) \cdot x_t & \text{otherwise}
\end{array}
$$

"Bellman equations"

Maximal reachability probabilities

given: MDP $\mathcal{M}$ with state space S and $G \subseteq S$

task: compute $x_s = \Pr_s^{\max}(\Diamond G) = \max_\sigma \Pr_s^\sigma(\Diamond G)$

The vector $(x_s)_{s \in S}$ is the least solution in $[0,1]^S$ of the equation system:

$$
\begin{array}{lcll}
x_s & = & 1 & \text{if } s \in G \\
x_s & = & 0 & \text{if } s \not\models \exists \Diamond G \\
x_s & = & \max\limits_{\alpha} \sum\limits_{t \in S} P(s, \alpha, t) \cdot x_t & \text{otherwise}
\end{array}
$$

... induces an optimal MD-scheduler ...

Maximal reachability probabilities

given: MDP $\mathcal{M}$ with state space S and $G \subseteq S$

task: compute $x_s = \Pr_s^{\max}(\Diamond G) = \max\limits_{\sigma} \Pr_s^{\sigma}(\Diamond G)$

The vector $(x_s)_{s \in S}$ is the least solution in $[0,1]^S$ of the equation system:

$$\begin{array}{lcll} x_s & = & 1 & \text{if } s \in G^* \\ x_s & = & 0 & \text{if } s \not\models \exists \Diamond G \\ x_s & = & \max\limits_{\alpha} \sum\limits_{t \in S} P(s,\alpha,t) \cdot x_t & \text{otherwise} \end{array}$$

pre-analysis: $G^* = \{ s \in S : x_s = 1 \}$

Maximal reachability probabilities

given: MDP $\mathcal{M}$ with state space S and $G \subseteq S$

task: compute $x_s = \Pr_s^{\max}(\Diamond G) = \max_\sigma \Pr_s^\sigma(\Diamond G)$

The vector $(x_s)_{s \in S}$ is the unique solution in $[0,1]^S$ of the equation system:

$$
\begin{array}{lcll}
x_s & = & 1 & \text{if } s \in G^* \\
x_s & = & 0 & \text{if } s \not\models \exists \Diamond G \\
x_s & = & \max\limits_{\alpha} \sum\limits_{t \in S} P(s, \alpha, t) \cdot x_t & \text{otherwise}
\end{array}
$$

if $\mathcal{M}$ has no end components

Maximal reachability probabilities

given: MDP $\mathcal{M}$ with state space S and $G \subseteq S$

task: compute $x_s = \Pr_s^{\max}(\Diamond G) = \max\limits_{\sigma} \Pr_s^{\sigma}(\Diamond G)$

value iteration: $x_s = \lim\limits_{n\to\infty} x_s^{(n)}$

$$
\begin{aligned}
x_s^{(n)} &= 1 && \text{if } s \in G^* \\
x_s^{(n)} &= 0 && \text{if } s \not\models \exists \Diamond G \\
x_s^{(n)} &= \max_{\alpha} \sum_{t \in S} P(s, \alpha, t) \cdot x_t^{(n-1)} && \text{else}
\end{aligned}
$$

if $\mathcal{M}$ has no end components

Maximal reachability probabilities

given: MDP $\mathcal{M}$ with state space S and $G \subseteq S$

task: compute $x_s = \Pr_s^{\max}(\Diamond G) = \max_{\sigma} \Pr_s^{\sigma}(\Diamond G)$

value iteration: $x_s = \lim_{n \to \infty} x_s^{(n)}$

$$
\begin{aligned}
x_s^{(n)} &= 1 && \text{if } s \in G^* \\
x_s^{(n)} &= 0 && \text{if } s \not\models \exists \Diamond G \\
x_s^{(n)} &= \max_{\alpha} \sum_{t \in S} P(s, \alpha, t) \cdot x_t^{(n-1)} && \text{else}
\end{aligned}
$$

if $\mathcal{M}$ has no end components or if $x_s^{(0)} \leqslant x_s$

Maximal reachability probabilities

given: MDP $\mathcal{M}$ with state space S and $G \subseteq S$

task: compute $x_s = \Pr^{\max}_s(\Diamond G) = \max\limits_{\sigma} \Pr^{\sigma}_s(\Diamond G)$

value iteration: $x_s = \lim\limits_{n\to\infty} x_s^{(n)}$

$$
\begin{aligned}
x_s^{(n)} &= 1 && \text{if } s \in G^* \\
x_s^{(n)} &= 0 && \text{if } s \not\models \exists\Diamond G \\
x_s^{(n)} &= \max_{\alpha} \sum_{t\in S} P(s,\alpha,t)\cdot x_t^{(n-1)} && \text{else}
\end{aligned}
$$

... termination condition ?

Interval iteration

[Haddad/Monmege'14]

given: MDP $\mathcal{M}$ with state space S and $G \subseteq S$

task: compute $x_s = \Pr^{\max}_s(\Diamond G) = \max\limits_{\sigma} \Pr^{\sigma}_s(\Diamond G)$

value iteration: $x_s = \lim\limits_{n\to\infty} x_s^{(n)}$

$$
\begin{array}{lll}
x_s^{(n)} = 1 & & \text{if } s \in G^* \\
x_s^{(n)} = 0 & & \text{if } s \not\models \exists\Diamond G \\
x_s^{(n)} = \max\limits_{\alpha} \sum\limits_{t\in S} P(s,\alpha,t)\cdot x_t^{(n-1)} & & \text{else}
\end{array}
$$

... use lower and upper iteration in the MEC-quotient ...

Maximal reachability probabilities via LP

given: MDP $\mathcal{M}$ with state space S and $G \subseteq S$

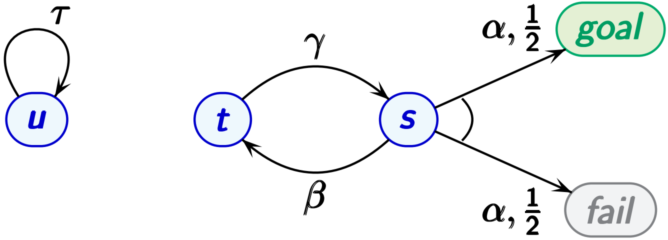
task: compute $x_s = \Pr^{\max}_s(\Diamond G) = \max\limits_{\sigma} \Pr^{\sigma}_s(\Diamond G)$

The vector $(x_s)_{s \in S}$ is the least solution in $[0,1]^S$ of the linear program:

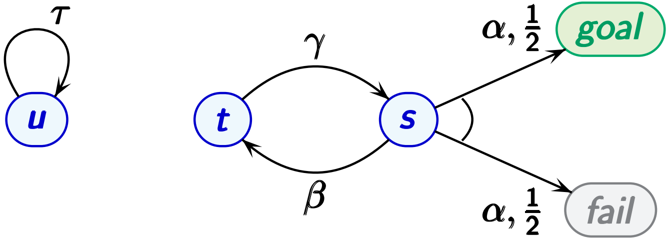
$$
\begin{array}{lcll}
x_s & = & 1 & \text{if } s \in G^* \\
x_s & = & 0 & \text{if } s \not\models \exists \Diamond G \\
x_s & \geqslant & \sum\limits_{t \in S} P(s, \alpha, t) \cdot x_t & \text{for } \alpha \in Act(s)
\end{array}
$$

where $\sum\limits_{s \in S} x_s$ is minimal

Least vs unique solution



Least vs unique solution



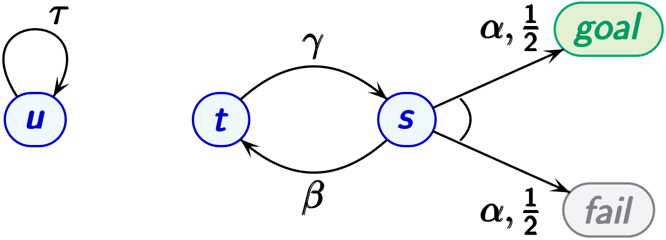
Bellmann equations:

$$x_u = x_u$$

$$x_s = \max\left\{ x_t,\ \frac{1}{2} \right\}$$

$$x_t = x_s$$

Least vs unique solution



Bellmann equations:

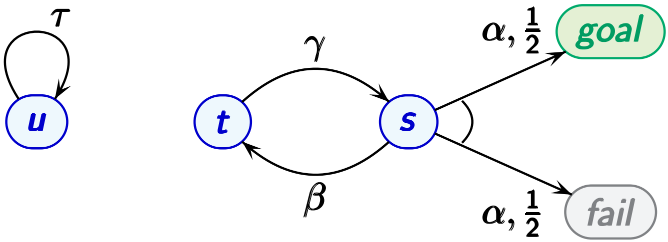
$x_u = 0$

as $u \not\models \exists\Diamond goal$

$x_s = \max\left\{ x_t, \frac{1}{2} \right\}$

$x_t = x_s$

Least vs unique solution



Bellmann equations:

$x_u = 0$

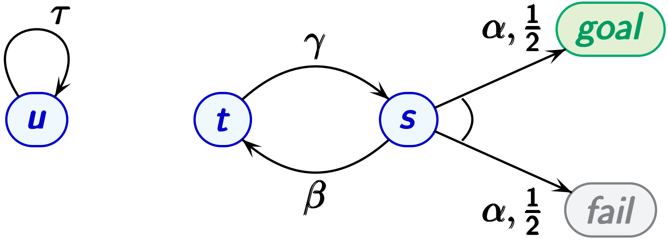
as $u \not\models \exists\Diamond goal$

$$x_s = \max\left\{ x_t, \tfrac{1}{2} \right\}$$
$$x_t = x_s$$

solutions:

$x_t = x_s \geqslant \frac{1}{2}$

Least vs unique solution



Bellmann equations:

$x_u = 0$

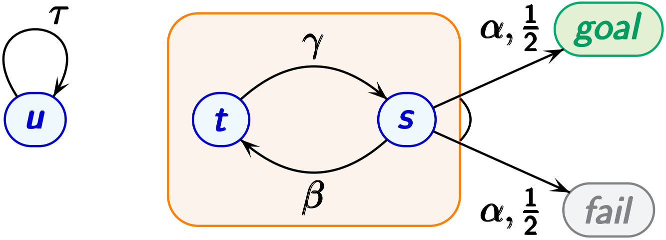
as $u \not\models \exists\Diamond goal$

$$\boxed{\begin{aligned} x_s &= \max\left\{\, x_t,\ \tfrac{1}{2} \,\right\} \\ x_t &= x_s \end{aligned}}$$

least solution:

$x_t = x_s = \frac{1}{2}$

Least vs unique solution



Bellmann equations:

$$x_u = 0$$

as $u \not\models \exists\Diamond goal$

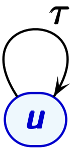
$$x_s = \max\left\{ x_t, \tfrac{1}{2} \right\}$$

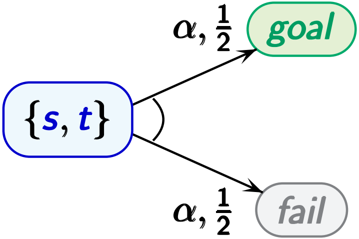
$$x_t = x_s$$

least solution:

$$x_t = x_s = \frac{1}{2}$$

Least vs unique solution





Bellmann equations:

$$x_u = 0$$

as $u \not\models \exists \Diamond goal$

$$\boxed{\begin{aligned} x_s &= \max\left\{ x_t, \tfrac{1}{2} \right\} \\ x_t &= x_s \end{aligned}}$$

unique solution:

$$x_{\{s,t\}} = \frac{1}{2}$$

Stochastic shortest/longest path problem

weighted
MDP $\mathcal{M}$

$\diamondsuit$ *goal*
accumulated weight until
reaching a goal state

best- or worst-case expectation
$\mathbb{E}^{\min}(\diamondsuit\, goal)$ or $\mathbb{E}^{\max}(\diamondsuit\, goal)$

extrema over all schedulers

Stochastic shortest/longest path problem

weighted
MDP $\mathcal{M}$

$\oplus$ *goal*
accumulated weight until
reaching a goal state

requirement for $\mathcal{M}$:
$\Pr^{\min}(\Diamond goal) = 1$

best- or worst-case expectation
$\mathbb{E}^{\min}(\oplus\, goal)$ or $\mathbb{E}^{\max}(\oplus\, goal)$

extrema over all schedulers

Maximal expected accumulated weight

given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$ s.t. $\Pr^{\min}_s(\Diamond G) = 1$ for all states s

task: compute $x_s \;=\; \underbrace{\mathbb{E}^{\max}_s(\oplus G)}_{\text{"stochastic longest path"}}$

Maximal expected accumulated weight

given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$ s.t. $\Pr^{\min}_s(\Diamond G) = 1$ for all states s

task: compute $x_s = \underbrace{\mathbb{E}^{\max}_s(\oplus G)}_{\text{"stochastic longest path"}}$

random variable $\oplus G : MaxPaths \to \mathbb{Z}$

if $\pi = s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} \ldots$ where $s_n \in G$, $s_0, \ldots, s_{n-1} \notin G$:

$$(\oplus G)(\pi) = wgt(s_0 \xrightarrow{\alpha_0} \ldots \xrightarrow{\alpha_n} s_n)$$

if $\pi \not\models \Diamond G$ then $(\oplus G)(\pi) = \perp$ "undefined"

Maximal expected accumulated weight

given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$ s.t.
$\Pr^{\min}_s(\Diamond G) = 1$ for all states s

task: compute $x_s = \mathbb{E}^{\max}_s(\oplus G)$

The vector $(x_s)_{s \in S}$ is the unique solution in $\mathbb{R}^S$ of:

If $s \in G$ then $x_s = 0$. Otherwise:

$$x_s = \max_{\alpha \in Act(s)} \Big(wgt(s, \alpha) + \sum_{t \in S} P(s, \alpha, t) \cdot x_t \Big)$$

"Bellman equations"

Maximal expected accumulated weight

given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$ s.t. $\Pr^{\min}_s(\Diamond G) = 1$ for all states s

task: compute $x_s \;=\; \mathbb{E}^{\max}_s(\oplus G)$

The vector $(x_s)_{s \in S}$ is the unique solution in $\mathbb{R}^S$ of:

If $s \in G$ then $x_s \;=\; 0$. Otherwise:

$$x_s \;=\; \max_{\alpha \in Act(s)} \Big(wgt(s,\alpha) + \sum_{t \in S} P(s,\alpha,t) \cdot x_t \Big)$$

... fixpoint operator is a contracting map ...

[Bertsekas/Tsitsiklis'91]

Maximal expected accumulated weight

given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$ s.t.
$\Pr_s^{\min}(\Diamond G) = 1$ for all states s

task: compute $x_s = \mathbb{E}_s^{\max}(\diamondplus G)$

The vector $(x_s)_{s \in S}$ is the unique solution in $\mathbb{R}^S$ of:

If $s \in G$ then $x_s = 0$. Otherwise:

$$x_s = \max_{\alpha \in Act(s)} \Big(wgt(s, \alpha) + \sum_{t \in S} P(s, \alpha, t) \cdot x_t \Big)$$

... induces an optimal MD-scheduler ...

[Bertsekas/Tsitsiklis'91]

Maximal expected accumulated weight

given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$ s.t. $\Pr^{\min}_s(\Diamond G) = 1$ for all states s

task: compute $x_s = \mathbb{E}^{\max}_s(\oplus G)$

The vector $(x_s)_{s \in S}$ is the unique solution in $\mathbb{R}^S$ of:

If $s \in G$ then $x_s^{(n)} = 0$. Otherwise:

$$x_s^{(n)} = \max_{\alpha \in Act(s)} \Big(wgt(s, \alpha) + \sum_{t \in S} P(s, \alpha, t) \cdot x_t^{(n-1)} \Big)$$

value iteration (arbitrary starting vector)

[Bertsekas/Tsitsiklis'91]

Maximal expected accumulated weight

given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$ s.t. $\Pr_s^{\min}(\Diamond G) = 1$ for all states s

task: compute $x_s \;=\; \mathbb{E}_s^{\max}(\oplus G)$

The vector $(x_s)_{s \in S}$ is the unique solution in $\mathbb{R}^S$ of:

If $s \in G$ then $x_s \;=\; 0$. Otherwise, for $\alpha \in Act(s)$:

$$x_s \;\geqslant\; wgt(s, \alpha) + \sum_{t \in S} P(s, \alpha, t) \cdot x_t$$

where $\sum\limits_{s \in S} x_s$ is minimal

Outline

- weighted Markov decision processes
- mean-payoff and long-run ratios
- expected accumulated weights
- conditional expected accumulated weights
- weight-bounded reachability and quantiles
- LTL with weight assertions
- conclusions

Mean-payoff

Mean-payoff

given: a weighted graph without trap states

mean-payoff functions $\overline{\mathrm{MP}}$, $\underline{\mathrm{MP}}$: *InfPaths* $\to \mathbb{R}$:

$$\overline{\mathrm{MP}}(s_0\, s_1\, s_2 \ldots) = \limsup_{n\to\infty} \frac{1}{n+1} \cdot \sum_{i=0}^{n} wgt(s_i)$$

$$\underline{\mathrm{MP}}(s_0\, s_1\, s_2 \ldots) = \liminf_{n\to\infty} \frac{1}{n+1} \cdot \sum_{i=0}^{n} wgt(s_i)$$

Mean-payoff

given: a weighted graph without trap states

mean-payoff functions $\overline{\mathrm{MP}}$, $\underline{\mathrm{MP}}$: *InfPaths* $\to \mathbb{R}$:

$$\overline{\mathrm{MP}}(s_0\, s_1\, s_2 \ldots) \;=\; \limsup_{n\to\infty} \frac{1}{n+1} \cdot \sum_{i=0}^{n} wgt(s_i)$$

$$\underline{\mathrm{MP}}(s_0\, s_1\, s_2 \ldots) \;=\; \liminf_{n\to\infty} \frac{1}{n+1} \cdot \sum_{i=0}^{n} wgt(s_i)$$

if $wgt(s) = +1$, $wgt(t) = -1$ then there exists $n_1, n_2, \ldots$ and $k_1, k_2, \ldots \in \mathbb{N}$ s.t. for $\pi = s^{n_1}\, t^{k_1}\, s^{n_2}\, t^{k_2} \ldots$:

$$\underline{\mathrm{MP}}(\pi) \;<\; 0 \;<\; \overline{\mathrm{MP}}(\pi)$$

Expected mean-payoff in finite MC or MDP

fundamental results:

in finite MC: $\mathbb{E}_s(\underline{\mathrm{MP}}) = \mathbb{E}_s(\overline{\mathrm{MP}})$

in finite MDP:

$$\mathbb{E}_s^{\max}(\underline{\mathrm{MP}}) = \mathbb{E}_s^{\max}(\overline{\mathrm{MP}})$$
$$\mathbb{E}_s^{\min}(\underline{\mathrm{MP}}) = \mathbb{E}_s^{\min}(\overline{\mathrm{MP}})$$

and optimal MD-scheduler exist

notation: $\mathbb{E}_s^*(\mathrm{MP})$ rather than $\mathbb{E}_s^*(\underline{\mathrm{MP}})$ resp. $\mathbb{E}_s^*(\overline{\mathrm{MP}})$

Expected mean-payoff in finite MC

fundamental results:

in finite MC: $$\mathbb{E}_s(\underline{\mathrm{MP}}) = \mathbb{E}_s(\overline{\mathrm{MP}})$$

for finite MC without traps:

> Almost all paths eventually enter a BSCC and visit all its states infinitely often.

BSCC: bottom strongly connected component

Expected mean-payoff in finite MC

fundamental results:

in finite MC: $$\mathbb{E}_s(\underline{\mathrm{MP}}) = \mathbb{E}_s(\overline{\mathrm{MP}})$$

for finite MC without traps:

Almost all paths eventually enter a BSCC and visit all its states infinitely often ...

... with the same long-run frequencies ...

BSCC: bottom strongly connected component

Long-run frequencies in finite MC

steady-state probabilities in BSCC B of a finite MC:

$$\theta^B(s) \;=\; \lim_{n\to\infty} \frac{1}{n} \cdot \sum_{i=1}^{n} \Pr_t(\bigcirc^i s) \quad \text{for each } t \in B$$

$\bigcirc^i s \;\triangleq\;$ "after i steps in state s"

Long-run frequencies in finite MC

steady-state probabilities in BSCC B of a finite MC:

$$\theta^B(s) \;=\; \lim_{n\to\infty} \frac{1}{n} \cdot \sum_{i=1}^{n} \Pr_t(\bigcirc^i s) \quad \text{for each } t \in B$$

computable by a linear equation system:

$$\theta^B(s) \;=\; \sum_{t\in B} \theta^B(t) \cdot P(t,s)$$

"balance equations"

$\bigcirc^i s \;\triangleq\;$ "after i steps in state s"

Long-run frequencies in finite MC

steady-state probabilities in BSCC B of a finite MC:

$$\theta^B(s) \;=\; \lim_{n\to\infty} \frac{1}{n} \cdot \sum_{i=1}^{n} \Pr_t\big(\bigcirc^i s\big) \quad \text{for each } t \in B$$

computable by a linear equation system:

$$\begin{aligned} \theta^B(s) \;&=\; \sum_{t\in B} \theta^B(t) \cdot P(t,s) \\ \sum_{s\in B} \theta^B(s) \;&=\; 1 \end{aligned}$$

$\bigcirc^i s \;\hat{=}\;$ "after i steps in state s"

Long-run frequencies in finite MC

steady-state probabilities in BSCC B of a finite MC:

$$\theta^B(s) \;=\; \lim_{n\to\infty} \frac{1}{n} \cdot \sum_{i=1}^{n} \Pr_t(\bigcirc^i s) \quad \text{for each } t \in B$$

computable by a linear equation system:

$$\theta^B(s) \;=\; \sum_{t\in B} \theta^B(t) \cdot P(t,s)$$

$$\sum_{s\in B} \theta^B(s) \;=\; 1$$

unique solution of the linear equation system

$$x = x \cdot P|_B$$

$$\sum_{s\in B} x_s = 1$$

$\bigcirc^i s \;\hat{=}\;$ "after i steps in state s"

Long-run frequencies in finite MC

steady-state probabilities in BSCC B of a finite MC:

$$\theta^B(s) \;=\; \lim_{n\to\infty} \frac{1}{n} \cdot \sum_{i=1}^{n} \Pr_t\big(\bigcirc^i s\big) \quad \text{for each } t \in B$$

for almost all paths $\pi = s_0\, s_1\, s_2 \ldots$ with $\pi \models \Diamond B$:

$$\theta^B(s) \;=\; \underbrace{\lim_{n\to\infty} \frac{1}{n+1} \cdot \mathit{freq}(s, s_0\, s_1 \ldots s_n)}_{\substack{\text{long-run frequency of} \\ \text{state } s \text{ in path } \pi}}$$

... limit exists for almost all paths ...

Long-run frequencies in finite MC

steady-state probabilities in BSCC B of a finite MC:

$$\theta^B(s) \;=\; \lim_{n\to\infty} \frac{1}{n} \cdot \sum_{i=1}^{n} \Pr_t\big(\bigcirc^i s\big) \quad \text{for each } t \in B$$

for almost all paths $\pi = s_0\, s_1\, s_2 \ldots$ with $\pi \models \Diamond B$:

$$\theta^B(s) \;=\; \underbrace{\lim_{n\to\infty} \frac{1}{n+1} \cdot freq(s, s_0\, s_1 \ldots s_n)}_{\substack{\text{long-run frequency of} \\ \text{state } s \text{ in path } \pi}}$$

$$freq(s, s_0\, s_1 \ldots s_n) \;=\; \left\{ \begin{array}{l} \text{number of occurrences of } s \\ \text{in the sequence } s_0\, s_1 \ldots s_n \end{array} \right.$$

Mean-payoff in finite weighted MC

steady-state probabilities in BSCC B of a finite MC:

$$\theta^B(s) \;=\; \lim_{n\to\infty} \frac{1}{n} \cdot \sum_{i=1}^{n} \Pr_t\big(\bigcirc^i s\big) \quad \text{for each } t \in B$$

for almost all paths $\pi = s_0\, s_1\, s_2 \ldots$ with $\pi \models \Diamond B$:

$$\theta^B(s) \;=\; \lim_{n\to\infty} \frac{1}{n+1} \cdot \mathit{freq}(s, s_0\, s_1 \ldots s_n)$$

if $\pi \models \Diamond B$ where B is a BSCC then almost surely

$$\mathrm{MP}(\pi) \;=\; \sum_{s\in B} \theta^B(s) \cdot \mathit{wgt}(s)$$

Mean-payoff in finite weighted MC

steady-state probabilities in BSCC B of a finite MC:

$$\theta^B(s) \;=\; \lim_{n\to\infty} \frac{1}{n} \cdot \sum_{i=1}^{n} \Pr_t(\bigcirc^i s) \quad \text{for each } t \in B$$

for almost all paths $\pi = s_0\, s_1\, s_2 \ldots$ with $\pi \models \Diamond B$:

$$\theta^B(s) \;=\; \lim_{n\to\infty} \frac{1}{n+1} \cdot \mathit{freq}(s, s_0\, s_1 \ldots s_n)$$

if $\pi \models \Diamond B$ where B is a BSCC then almost surely

$$\mathrm{MP}(\pi) \;=\; \underbrace{\sum_{s\in B} \theta^B(s) \cdot \mathit{wgt}(s)}_{\text{only depends on } B}$$

Mean-payoff in finite weighted MC

steady-state probabilities in BSCC B of a finite MC:

$$\theta^B(s) \;=\; \lim_{n\to\infty} \frac{1}{n} \cdot \sum_{i=1}^{n} \Pr_t(\bigcirc^i s) \quad \text{for each } t \in B$$

for almost all paths $\pi = s_0\, s_1\, s_2 \ldots$ with $\pi \models \Diamond B$:

$$\theta^B(s) \;=\; \lim_{n\to\infty} \frac{1}{n+1} \cdot \mathit{freq}(s, s_0\, s_1 \ldots s_n)$$

if $\pi \models \Diamond B$ where B is a BSCC then almost surely

$$\mathrm{MP}(\pi) \;=\; \underbrace{\sum_{s\in B} \theta^B(s) \cdot \mathit{wgt}(s)}_{\text{only depends on } B} \;\stackrel{\text{def}}{=}\; \mathrm{MP}(B)$$

Mean-payoff in finite weighted MC

steady-state probabilities in BSCC B of a finite MC:

$$\theta^B(s) \;=\; \lim_{n\to\infty} \frac{1}{n} \cdot \sum_{i=1}^{n} \Pr_t\big(\bigcirc^i s\big) \quad \text{for each } t \in B$$

for almost all paths $\pi = s_0\, s_1\, s_2 \ldots$ with $\pi \models \Diamond B$:

$$\theta^B(s) \;=\; \lim_{n\to\infty} \frac{1}{n+1} \cdot \mathit{freq}(s, s_0\, s_1 \ldots s_n)$$

if $\pi \models \Diamond B$ where B is a BSCC then almost surely

$$\mathrm{MP}(\pi) \;=\; \sum_{s\in B} \theta^B(s) \cdot \mathit{wgt}(s) \;\stackrel{\text{def}}{=}\; \mathrm{MP}(B)$$

expected mean-payoff: $\displaystyle\sum_{B} \Pr_{s_0}(\Diamond B) \cdot \mathrm{MP}(B)$

Long-run ratios in finite MC

given: MC with reward functions $\boldsymbol{cost}, \boldsymbol{util} : S \to \mathbb{N}$

Examples:

- energy-utility ratio
- number of SLA violations per day
- recovery time per failure

Long-run ratios in finite MC

given: MC with reward functions $\mathit{cost}, \mathit{util} : S \to \mathbb{N}$

long-run cost-utility ratio $\mathit{lrrat} : \mathit{InfPaths} \to \mathbb{R}$

$$\mathit{lrrat}(s_0\, s_1\, s_2 \ldots) \;=\; \lim_{n \to \infty} \frac{\mathit{cost}(s_0\, s_1 \ldots s_n)}{\mathit{util}(s_0\, s_1 \ldots s_n)}$$

Examples:

- energy-utility ratio
- number of SLA violations per day
- recovery time per failure

Long-run ratios in finite MC

given: MC with reward functions $cost, util : S \to \mathbb{N}$

long-run cost-utility ratio $lrrat : InfPaths \to \mathbb{R}$

$$lrrat(s_0\, s_1\, s_2 \ldots) \;=\; \lim_{n\to\infty} \frac{cost(s_0\, s_1 \ldots s_n)}{util(s_0\, s_1 \ldots s_n)}$$

does the limit exist for almost all paths **?**

- energy-utility ratio
- number of SLA violations per day
- recovery time per failure

Long-run ratios in finite MC

given: MC with reward functions $cost, util : S \to \mathbb{N}$

long-run cost-utility ratio $lrrat : InfPaths \to \mathbb{R}$

$$
\begin{aligned}
lrrat(s_0\, s_1\, s_2 \ldots) &= \lim_{n\to\infty} \frac{cost(s_0\, s_1 \ldots s_n)}{util(s_0\, s_1 \ldots s_n)} \\
&= \lim_{n\to\infty} \frac{\sum_{i=0}^{n} cost(s_i)}{\sum_{i=0}^{n} util(s_i)}
\end{aligned}
$$

Long-run ratios in finite MC

given: MC with reward functions $\mathit{cost}, \mathit{util} : S \to \mathbb{N}$

long-run cost-utility ratio $\mathit{lrrat} : \mathit{InfPaths} \to \mathbb{R}$

$$\mathit{lrrat}(s_0\, s_1\, s_2 \ldots) \;=\; \lim_{n\to\infty} \frac{\mathit{cost}(s_0\, s_1 \ldots s_n)}{\mathit{util}(s_0\, s_1 \ldots s_n)}$$

$$= \lim_{n\to\infty} \frac{\frac{1}{n+1}\cdot\sum\limits_{i=0}^{n} \mathit{cost}(s_i)}{\frac{1}{n+1}\cdot\sum\limits_{i=0}^{n} \mathit{util}(s_i)}$$

Long-run ratios in finite MC

given: MC with reward functions *cost*, *util* : $S \to \mathbb{N}$

long-run cost-utility ratio *lrrat* : *InfPaths* $\to \mathbb{R}$

$$
\begin{aligned}
\mathit{lrrat}(s_0\, s_1\, s_2 \ldots) &= \lim_{n\to\infty} \frac{\mathit{cost}(s_0\, s_1 \ldots s_n)}{\mathit{util}(s_0\, s_1 \ldots s_n)} \\
&= \lim_{n\to\infty} \frac{\frac{1}{n+1}\cdot \sum\limits_{i=0}^{n} \mathit{cost}(s_i)}{\frac{1}{n+1}\cdot \sum\limits_{i=0}^{n} \mathit{util}(s_i)} \\
&= \frac{\mathrm{MP}[\mathit{cost}](s_0\, s_1\, s_2 \ldots)}{\mathrm{MP}[\mathit{util}](s_0\, s_1\, s_2 \ldots)}
\end{aligned}
$$

Long-run ratios in finite MC

given: MC with reward functions *cost*, *util* : $S \to \mathbb{N}$

long-run cost-utility ratio *lrrat* : *InfPaths* $\to \mathbb{R}$

$$
\begin{aligned}
\mathit{lrrat}(s_0\, s_1\, s_2 \ldots) &= \lim_{n\to\infty} \frac{\mathit{cost}(s_0\, s_1 \ldots s_n)}{\mathit{util}(s_0\, s_1 \ldots s_n)} \\
&= \lim_{n\to\infty} \frac{\frac{1}{n+1}\cdot \sum\limits_{i=0}^{n} \mathit{cost}(s_i)}{\frac{1}{n+1}\cdot \sum\limits_{i=0}^{n} \mathit{util}(s_i)} \\
&= \frac{\mathrm{MP}[\mathit{cost}](s_0\, s_1\, s_2 \ldots)}{\mathrm{MP}[\mathit{util}](s_0\, s_1\, s_2 \ldots)}
\end{aligned}
$$

in particular:
limit exists for
almost all paths

Long-run ratios in finite MC

given: MC with reward functions $cost, util : S \to \mathbb{N}$

long-run cost-utility ratio $lrrat : InfPaths \to \mathbb{R}$

$$lrrat(s_0\, s_1\, s_2 \ldots) \;=\; \lim_{n\to\infty} \frac{cost(s_0\, s_1 \ldots s_n)}{util(s_0\, s_1 \ldots s_n)}$$

if $\pi \models \Diamond B$ where B is a BSCC then almost surely

$$lrrat(\pi) \;=\; \frac{\mathrm{MP}[cost](B)}{\mathrm{MP}[util](B)}$$

$$\mathrm{MP}[wgt](B) = \sum_{s\in B} \theta^B(s) \cdot wgt(s)$$

mean-payoff for weight function

Long-run ratios in finite MC

given: MC with reward functions $cost, util : S \to \mathbb{N}$

long-run cost-utility ratio $lrrat : InfPaths \to \mathbb{R}$

$$lrrat(s_0\, s_1\, s_2 \ldots) \;=\; \lim_{n\to\infty} \frac{cost(s_0\, s_1 \ldots s_n)}{util(s_0\, s_1 \ldots s_n)}$$

if $\pi \models \Diamond B$ where B is a BSCC then almost surely

$$lrrat(\pi) \;=\; \underbrace{\frac{\mathrm{MP}[cost](B)}{\mathrm{MP}[util](B)}}_{\text{only depends on } B} \;\stackrel{\text{def}}{=}\; lrrat(B)$$

Long-run ratios in finite MC

given: MC with reward functions $cost, util : S \to \mathbb{N}$

long-run cost-utility ratio $lrrat : InfPaths \to \mathbb{R}$

$$lrrat(s_0\, s_1\, s_2 \ldots) \;=\; \lim_{n\to\infty} \frac{cost(s_0\, s_1 \ldots s_n)}{util(s_0\, s_1 \ldots s_n)}$$

if $\pi \models \Diamond B$ where B is a BSCC then almost surely

$$lrrat(\pi) \;=\; \frac{\mathrm{MP}[cost](B)}{\mathrm{MP}[util](B)} \stackrel{\text{def}}{=} lrrat(B)$$

expected long-run ratio: $\displaystyle\sum_B \Pr_{s_0}(\Diamond B) \cdot lrrat(B)$

Best threshold for long-run ratios

given: MC with reward functions $cost, util : S \to \mathbb{N}$
rational probability bound p

compute $\quad r_{opt} \quad = \quad \inf\{\, r \in \mathbb{R} : \Pr_{s_0}(\, lrrat \leqslant r\,) > p\,\}$

random variable for the
long-run cost-utility ratio
(as before)

Best threshold for long-run ratios

given: MC with reward functions $cost, util : S \to \mathbb{N}$
rational probability bound p

compute $\quad r_{opt} \;=\; \inf\{\, r \in \mathbb{R} : \Pr_{s_0}(\, lrrat \leqslant r\,) > p\,\}$

$$r_{opt} \;=\; \inf\{\, r \in \mathbb{R} \,:\, \Pr_{s_0}(\Box\Diamond(\tfrac{cost}{util} \leqslant r)) > p\,\}$$

if $\pi = s_0\, s_1\, s_2 \ldots$ is an infinite path then

$$\pi \models \Box\Diamond(\tfrac{cost}{util} \leqslant r) \quad \text{iff} \quad \overset{\infty}{\exists}\, n \text{ s.t. } \frac{cost(s_0\, s_1 \ldots s_n)}{util(s_0\, s_1 \ldots s_n)} \leqslant r$$

Best threshold for long-run ratios

given: MC with reward functions *cost*, *util* : $S \to \mathbb{N}$
rational probability bound p

compute $\quad r_{opt} \;=\; \inf\{\, r \in \mathbb{R} : \Pr_{s_0}(\, \mathit{lrrat} \leqslant r\,) > p \,\}$

$$r_{opt} \;=\; \inf\left\{\, r \in \mathbb{R} \,:\, \Pr_{s_0}\!\left(\Box\Diamond\!\left(\tfrac{cost}{util} \leqslant r\right)\right) > p \,\right\}$$

$$\;=\; \inf\left\{\, r \in \mathbb{R} \,:\, \Pr_{s_0}\!\left(\Diamond\Box\!\left(\tfrac{cost}{util} \leqslant r\right)\right) > p \,\right\}$$

$$\pi \models \Box\Diamond\!\left(\tfrac{cost}{util} \leqslant r\right) \quad \text{iff} \quad \overset{\infty}{\exists}\, n \text{ s.t. } \frac{cost(s_0\, s_1 \ldots s_n)}{util(s_0\, s_1 \ldots s_n)} \leqslant r$$

Best threshold for long-run ratios

given: MC with reward functions $cost, util : S \to \mathbb{N}$
rational probability bound p

compute $\quad r_{opt} \;=\; \inf\{\, r \in \mathbb{R} : \Pr_{s_0}(\, lrrat \leqslant r \,) > p \,\}$

$$
\begin{aligned}
r_{opt} &= \inf\{\, r \in \mathbb{R} : \Pr_{s_0}(\Box\Diamond(\tfrac{cost}{util} \leqslant r)) > p \,\} \\
&= \inf\{\, r \in \mathbb{R} : \Pr_{s_0}(\Diamond\Box(\tfrac{cost}{util} \leqslant r)) > p \,\} \\
&= \min\{\, r \in \mathbb{Q} : \Pr_{s_0}(\Diamond C_r) > p \,\}
\end{aligned}
$$

where C_r = union of all BSCCs B with $lrrat(B) \leqslant r$

Best threshold for long-run ratios

given: MC with reward functions $cost, util : S \to \mathbb{N}$
rational probability bound p

compute $r_{opt} = \inf \{ r \in \mathbb{R} : \Pr_{s_0}(lrrat \leqslant r) > p \}$

$= \min \{ r \in \mathbb{Q} : \Pr_{s_0}(\Diamond C_r) > p \}$

where C_r = union of all BSCCs B with $lrrat(B) \leqslant r$

expected long-run ratio of B

Best threshold for long-run ratios

given: MC with reward functions $cost, util : S \to \mathbb{N}$
rational probability bound p

$$\text{compute} \quad r_{opt} = \inf \{ r \in \mathbb{R} : \Pr_{s_0}(lrrat \leqslant r) > p \}$$
$$= \min \{ r \in \mathbb{Q} : \Pr_{s_0}(\Diamond C_r) > p \}$$

where C_r = union of all BSCCs B with $lrrat(B) \leqslant r$

1. compute the BSCCs $B_1, \ldots, B_k$ and $r_i = lrrat(B_i)$

Best threshold for long-run ratios

given: MC with reward functions $cost, util : S \to \mathbb{N}$
rational probability bound p

compute $$r_{opt} = \inf \{ r \in \mathbb{R} : \Pr_{s_0}(lrrat \leqslant r) > p \}$$
$$= \min \{ r \in \mathbb{Q} : \Pr_{s_0}(\Diamond C_r) > p \}$$

where C_r = union of all BSCCs B with $lrrat(B) \leqslant r$

1. compute the BSCCs $B_1, \ldots, B_k$ and $r_i = lrrat(B_i)$

   w.l.o.g. $r_1 < r_2 < \ldots < r_k$

Best threshold for long-run ratios

given: MC with reward functions $cost, util : S \to \mathbb{N}$
rational probability bound p

compute $r_{opt} = \inf \{ r \in \mathbb{R} : \Pr_{s_0}(lrrat \leqslant r) > p \}$
$= \min \{ r \in \mathbb{Q} : \Pr_{s_0}(\Diamond C_r) > p \}$

where C_r = union of all BSCCs B with $lrrat(B) \leqslant r$

1. compute the BSCCs $B_1, \ldots, B_k$ and $r_i = lrrat(B_i)$
   w.l.o.g. $r_1 < r_2 < \ldots < r_k$
2. determine the minimal $i \in \{1, \ldots, k\}$ such that
   $\Pr_{s_0}(\Diamond B_1) + \ldots + \Pr_{s_0}(\Diamond B_i) > p$ and return r_i

Mean-payoff in MDPs

random variable $\overline{\mathrm{MP}} : \mathit{InfPaths} \to \mathbb{R}$ defined by

$$\overline{\mathrm{MP}}(s_0\, s_1\, s_2 \ldots) = \limsup_{n\to\infty} \frac{1}{n+1} \cdot \sum_{i=0}^{n} \mathit{wgt}(s_i)$$

Mean-payoff in MDPs

Given MDP with weight function $wgt : S \to \mathbb{Q}$, find a scheduler maximizing the expected mean-payoff.

random variable $\overline{\mathrm{MP}} : \mathit{InfPaths} \to \mathbb{R}$ defined by

$$\overline{\mathrm{MP}}(s_0\, s_1\, s_2 \ldots) = \limsup_{n\to\infty} \tfrac{1}{n+1} \cdot \sum_{i=0}^{n} wgt(s_i)$$

Mean-payoff in MDPs

Given MDP with weight function $\boldsymbol{wgt} : \boldsymbol{S} \to \mathbb{Q}$, find a scheduler maximizing the expected mean-payoff.

Results: [Howard'60], [Dernan'66], [Kallenberg'80] ...

- optimal MD-scheduler exist
- computable in polynomial-time via linear program
  to encode the long-run frequencies of MR-scheduler
- value and policy iteration algorithms
- extensions for multiple mean-payoff constraints
  [Brazdil/Brozek/Chatterjee/Foreijt/Kucera'14]

Mean-payoff in MDPs

Given MDP with weight function $\textit{wgt} : S \to \mathbb{Q}$, find a scheduler maximizing the expected mean-payoff.

Results: [Howard'60], [Dernan'66], [Kallenberg'80] ...

- optimal MD-scheduler exist
- computable in polynomial-time via linear program
  to encode the long-run frequencies of MR-scheduler
- value and policy iteration algorithms
- extensions for multiple mean-payoff constraints
  [Brazdil/Brozek/Chatterjee/Foreijt/Kucera'14]

Mean-payoff in strongly connected MDPs

Mean-payoff in strongly connected MDPs

linear program for the maximal expected mean-payoff:

... uses variables $x_{s,\alpha}$ for $s \in S$, $\alpha \in Act(s)$
to encode the long-run frequencies of the
state-action pairs (s, α) in MR-schedulers

Mean-payoff in strongly connected MDPs

linear program for the maximal expected mean-payoff:

... uses variables $x_{s,\alpha}$ for $s \in S$, $\alpha \in Act(s)$
to encode the long-run frequencies of the
state-action pairs (s, α) in MR-schedulers

Given the values $x_{s,\alpha}$, a corresponding MR-scheduler σ can be defined by:

- if $x_s \stackrel{\text{def}}{=} \sum\limits_{\alpha \in Act(s)} x_{s,\alpha} > 0$ then: $\sigma(s)(\alpha) = x_{s,\alpha}/x_s$

Mean-payoff in strongly connected MDPs

linear program for the maximal expected mean-payoff:

... uses variables $x_{s,\alpha}$ for $s \in S$, $\alpha \in Act(s)$
to encode the long-run frequencies of the
state-action pairs (s, α) in MR-schedulers

Given the values $x_{s,\alpha}$, a corresponding MR-scheduler σ can be defined by:

- if $x_s \overset{\text{def}}{=} \sum\limits_{\alpha \in Act(s)} x_{s,\alpha} > 0$ then: $\sigma(s)(\alpha) = x_{s,\alpha}/x_s$
- if $x_s = 0$ then σ behaves an MD-scheduler that reaches a state t with $x_t = 1$ with probability 1

Mean-payoff in strongly connected MDPs

linear program for the maximal expected mean-payoff:

maximize $\sum_{s,\alpha} x_{s,\alpha} \cdot wgt(s,\alpha)$ subject to:

variables for the
long-run frequencies of
state-action pairs

Mean-payoff in strongly connected MDPs

linear program for the maximal expected mean-payoff:

maximize $\underbrace{\sum_{s,\alpha} x_{s,\alpha} \cdot wgt(s,\alpha)}_{\substack{\text{mean-payoff of} \\ \text{MR-scheduler } \sigma \text{ given by}}}$ subject to:

$$\sigma(s)(\alpha) = x_{s,\alpha}/x_s$$

for each state s with $x_s \stackrel{\text{def}}{=} \sum_{\alpha \in Act(s)} x_{s,\alpha} > 0$

Mean-payoff in strongly connected MDPs

linear program for the maximal expected mean-payoff:

maximize $\sum_{s,\alpha} x_{s,\alpha} \cdot wgt(s,\alpha)$ subject to:

$$\underbrace{x_t \;=\; \sum_{s,\alpha} x_{s,\alpha} \cdot P(s,\alpha,t)}_{\text{balance equation for state } t} \quad \text{for } t \in S$$

$$x_t \;=\; \sum_{\beta \in Act(t)} x_{t,\beta}$$

Mean-payoff in strongly connected MDPs

linear program for the maximal expected mean-payoff:

maximize $\sum_{s,\alpha} x_{s,\alpha} \cdot wgt(s,\alpha)$ subject to:

$$x_t = \sum_{s,\alpha} x_{s,\alpha} \cdot P(s,\alpha,t) \quad \text{for } t \in S$$

$$x_{s,\alpha} \geqslant 0 \quad \text{for } s \in S \text{ and } \alpha \in Act(s)$$

long-run frequencies
are non-negative

$$x_t = \sum_{\beta \in Act(t)} x_{t,\beta}$$

Mean-payoff in strongly connected MDPs

linear program for the maximal expected mean-payoff:

maximize $\sum_{s,\alpha} x_{s,\alpha} \cdot wgt(s,\alpha)$ subject to:

$$x_t = \sum_{s,\alpha} x_{s,\alpha} \cdot P(s,\alpha,t) \quad \text{for } t \in S$$

$$x_{s,\alpha} \geqslant 0 \quad \text{for } s \in S \text{ and } \alpha \in Act(s)$$

$$\sum_{s,\alpha} x_{s,\alpha} = 1$$

$$x_t = \sum_{\beta \in Act(t)} x_{t,\beta}$$

long-run frequencies yield a distribution

Mean-payoff in strongly connected MDPs

linear program for the maximal expected mean-payoff:

maximize $\sum_{s,\alpha} x_{s,\alpha} \cdot wgt(s,\alpha)$ subject to:

$$x_t \;=\; \sum_{s,\alpha} x_{s,\alpha} \cdot P(s,\alpha,t) \quad \text{for } t \in S$$

$$x_{s,\alpha} \geqslant 0 \quad \text{for } s \in S \text{ and } \alpha \in Act(s)$$

$$\sum_{s,\alpha} x_{s,\alpha} = 1$$

$$x_t \;=\; \sum_{\beta \in Act(t)} x_{t,\beta}$$

Each solution induces an optimal MR-scheduler.

Mean-payoff in strongly connected MDPs

linear program for the maximal expected mean-payoff:

$$\text{maximize } \sum_{s,\alpha} x_{s,\alpha} \cdot wgt(s,\alpha) \text{ subject to:}$$

$$x_t \;=\; \sum_{s,\alpha} x_{s,\alpha} \cdot P(s,\alpha,t) \quad \text{for } t \in S$$

$$x_{s,\alpha} \geqslant 0 \quad \text{for } s \in S \text{ and } \alpha \in Act(s)$$

$$\sum_{s,\alpha} x_{s,\alpha} = 1 \qquad\qquad x_t = \sum_{\beta \in Act(t)} x_{t,\beta}$$

Each solution induces an optimal MR-scheduler.

But how to obtain an optimal MD-scheduler **?**

Mean-payoff in strongly connected MDPs

linear program for the maximal expected mean-payoff:

maximize $\sum_{s,\alpha} x_{s,\alpha} \cdot wgt(s,\alpha)$ subject to:

$$x_t = \sum_{s,\alpha} x_{s,\alpha} \cdot P(s,\alpha,t) \quad \text{for } t \in S$$

$$x_{s,\alpha} \geqslant 0 \quad \text{for } s \in S \text{ and } \alpha \in Act(s)$$

$$\sum_{s,\alpha} x_{s,\alpha} = 1$$

$$x_s = \sum_{\alpha \in Act(s)} x_{s,\alpha}$$

optimal MD-scheduler: for each state s with $x_s > 0$ pick an action α with $x_{s,\alpha} > 0$

Mean-payoff in MDPs: general case

given: weighted MDP $\mathcal{M}$ without trap states

task: find a scheduler that maximizes the expected mean-payoff

State s is called a trap if $Act(s) = \varnothing$.

Mean-payoff in MDPs: general case

given: weighted MDP $\mathcal{M}$ without trap states

task: find a scheduler that maximizes the expected mean-payoff

method 1: [KALLENBERG'80]

use an LP with two variables for each state-action pair

$x_{s,\alpha}$ long-run frequency

$y_{s,\alpha}$ frequency in the transient part

State s is called a trap if $Act(s) = \varnothing$.

Mean-payoff in MDPs: general case

given: weighted MDP $\mathcal{M}$ without trap states

task: find a scheduler that maximizes the expected mean-payoff

method 1: [KALLENBERG'80]

use an LP with two variables for each state-action pair

$x_{s,\alpha}$ long-run frequency

$y_{s,\alpha}$ frequency in the transient part

method 2:

compute the maximal expected mean-payoff of the MECs and "compose" the result for the full MDP

Mean-payoff in MDPs: general case

step 1: compute the maximal end components $\mathcal{E}_1, ..., \mathcal{E}_k$

Mean-payoff in MDPs: general case

step 1: compute the maximal end components $\mathcal{E}_1, \ldots, \mathcal{E}_k$

step 2: for $i = 1, \ldots, k$, compute the maximal expected mean-payoff mp_i of $\mathcal{E}_i$

Mean-payoff in MDPs: general case

step 1: compute the maximal end components $\mathcal{E}_1, ..., \mathcal{E}_k$

step 2: for $i = 1, ..., k$, compute the maximal expected mean-payoff mp_i of $\mathcal{E}_i$

step 3: construct the modified MEC-quotient $\mathcal{M}'$

Mean-payoff in MDPs: general case

step 1: compute the maximal end components $\mathcal{E}_1, ..., \mathcal{E}_k$

step 2: for $i = 1, ..., k$, compute the maximal expected mean-payoff mp_i of $\mathcal{E}_i$

step 3: construct the modified MEC-quotient $\mathcal{M}'$

$\mathcal{M}'$ arises from $\mathrm{MEC}(\mathcal{M})$ by adding

- a fresh trap state *goal*
- a new action symbol τ
- transitions $\mathcal{E}_i \xrightarrow{\tau}$ *goal* for $i, ..., k$

Mean-payoff in MDPs: general case

step 1: compute the maximal end components $\mathcal{E}_1, ..., \mathcal{E}_k$

step 2: for $i = 1, ..., k$, compute the maximal expected mean-payoff mp_i of $\mathcal{E}_i$

step 3: construct the modified MEC-quotient $\mathcal{M}'$ with weight mp_i for the transitions $\mathcal{E}_i \xrightarrow{\tau} goal$ and weight 0 for all other states

Mean-payoff in MDPs: general case

step 1: compute the maximal end components $\mathcal{E}_1, ..., \mathcal{E}_k$

step 2: for $i = 1, ..., k$, compute the maximal expected mean-payoff mp_i of $\mathcal{E}_i$

step 3: construct the modified MEC-quotient $\mathcal{M}'$ with weight mp_i for the transitions $\mathcal{E}_i \xrightarrow{\tau} goal$ and weight 0 for all other states

step 4: compute the maximal expected total weight in $\mathcal{M}'$

Mean-payoff in MDPs: general case

step 1: compute the maximal end components $\mathcal{E}_1, ..., \mathcal{E}_k$

step 2: for $i = 1, ..., k$, compute the maximal expected mean-payoff mp_i of $\mathcal{E}_i$

step 3: construct the modified MEC-quotient $\mathcal{M}'$ with weight mp_i for the transitions $\mathcal{E}_i \xrightarrow{\tau} goal$ and weight 0 for all other states

step 4: compute the maximal expected total weight

$\Pr^{\min}_{\mathcal{M}'}(\Diamond goal) = 1$ maximal expected total weight and optimal MD-scheduler exist

Mean-payoff in MDPs: general case

step 1: compute the maximal end components $\mathcal{E}_1, ..., \mathcal{E}_k$

step 2: for $i = 1, ..., k$, compute the maximal expected mean-payoff mp_i of $\mathcal{E}_i$

step 3: construct the modified MEC-quotient $\mathcal{M}'$ with weight mp_i for the transitions $\mathcal{E}_i \xrightarrow{\tau} goal$ and weight 0 for all other states

step 4: compute the maximal expected total weight in $\mathcal{M}'$

$$\boxed{\mathbb{E}^{\max}_{\mathcal{M}'}(\text{"total weight"}) \;=\; \mathbb{E}^{\max}_{\mathcal{M}}(\mathrm{MP})}$$

Mean-payoff in MDPs: general case

step 1: compute the maximal end components $\mathcal{E}_1, \ldots, \mathcal{E}_k$

step 2: for $i = 1, \ldots, k$, compute the maximal expected mean-payoff mp_i of $\mathcal{E}_i$

step 3: construct the modified MEC-quotient $\mathcal{M}'$ with weight mp_i for the transitions $\mathcal{E}_i \xrightarrow{\tau} goal$ and weight 0 for all other states

step 4: compute the maximal expected total weight in $\mathcal{M}'$

question: how to compute an optimal scheduler ?

Mean-payoff in MDPs: general case

step 1: compute the maximal end components $\mathcal{E}_1, ..., \mathcal{E}_k$

step 2: for $i = 1, ..., k$, compute the maximal expected mean-payoff mp_i of $\mathcal{E}_i$
... and an optimal MD-scheduler σ_i

step 3: construct the modified MEC-quotient $\mathcal{M}'$ with weight mp_i for the transitions $\mathcal{E}_i \xrightarrow{\tau} goal$ and weight 0 for all other states

step 4: compute the maximal expected total weight in $\mathcal{M}'$ ... and an optimal MD-scheduler ν

optimal MD-scheduler arises by combining $\nu, \sigma_1, \ldots, \sigma_k$

Expected long-run ratios

$\textit{ratio} = \frac{\textit{cost}}{\textit{util}}$ where $\textit{cost}$, $\textit{util}$ are reward functions

Expected long-run ratios

for Markov chains:

trivially computable in each BSCC as the quotient of the mean-payoff of both reward functions

$$\sum_{B} \Pr_s(\Diamond B) \cdot \frac{\mathrm{MP}[\mathit{cost}](B)}{\mathrm{MP}[\mathit{util}](B)}$$

B ranges over all BSCCs of the MC

$\mathit{ratio} = \frac{\mathit{cost}}{\mathit{util}}$ where cost, util are reward functions

Expected long-run ratios

for Markov chains:

trivially computable in each BSCC as the quotient of the mean-payoff of both reward functions

for MDPs:

- optimal MD-schedulers exist [GIMBERT'07]
- LP-based approach [DE ALFARO'98]

$\textit{\textbf{ratio}} = \frac{\textit{\textbf{cost}}}{\textit{\textbf{util}}}$ where ***cost***, ***util*** are reward functions

Expected long-run ratios

for Markov chains:

trivially computable in each BSCC as the quotient of the mean-payoff of both reward functions

for MDPs:

- optimal MD-schedulers exist [Gimbert'07]
- LP-based approach [de Alfaro'98]

minimize y subject to

$$x_s \geqslant cost(s,\alpha) - y \cdot util(s,\alpha) + \sum_{t \in S} P(s,\alpha,t) \cdot x_t$$

for all states s and $\alpha \in Act(s)$

Expected long-run ratios

for Markov chains:

trivially computable in each BSCC as the quotient of the mean-payoff of both reward functions

for MDPs:

- optimal MD-schedulers exist [GIMBERT'07]
- LP-based approach [DE ALFARO'98]
- fractional LP for uni-chain MDPs [ESSEN/JOBSTMANN'11]

  using an encoding of MR-scheduler as for mean-payoff; synthesis of an MD-scheduler maximizing the long-run ratio

Outline

- weighted Markov decision processes
- mean-payoff and long-run ratios
- expected accumulated weights
- conditional expected accumulated rewards
- weight-bounded reachability and quantiles
- LTL with weight assertions
- conclusions

Stochastic shortest/longest path problem

weighted
MDP $\mathcal{M}$

$\diamondplus$ *goal*
accumulated weight until
reaching a goal state

requirement for $\mathcal{M}$:
$\Pr^{\min}(\Diamond \mathit{goal}) = 1$

best- or worst-case expectation
$\mathbb{E}^{\min}(\diamondplus\, \mathit{goal})$ or $\mathbb{E}^{\max}(\diamondplus\, \mathit{goal})$

extrema over all schedulers

Stochastic shortest/longest path problem

weighted MDP $\mathcal{M}$

$\Diamond\!\!\!\!+\, \mathit{goal}$

accumulated weight until reaching a goal state

relaxed requirement:

$\mathrm{Pr}^{\max}(\Diamond \mathit{goal}) = 1$

Bertsekas/Tsitsiklis'91
de Alfaro'99

best- or worst-case expectation

$\mathbb{E}^{\min}(\Diamond\!\!\!\!+\, \mathit{goal})$ or $\mathbb{E}^{\max}(\Diamond\!\!\!\!+\, \mathit{goal})$

extrema over all proper schedulers

Maximal expected accumulated weight

given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$ s.t.

$$T = \{s \in S : \Pr_s^{\max}(\Diamond G) = 1\} \neq \varnothing$$

task: compute $x_s = \underbrace{\mathbb{E}_s^{\max}(\oplus G)}_{\text{maximum over all proper schedulers}}$ for $s \in T$

σ is proper iff $\Pr_s^{\sigma}(\Diamond G) = 1$ for all $s \in T$

Maximal expected accumulated weight

given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$ s.t.

$$T = \{s \in S : \Pr_s^{\max}(\Diamond G) = 1\} \neq \varnothing$$

task: compute $x_s = \mathbb{E}_s^{\max}(\oplus G)$ for $s \in T$

W.l.o.g. $T = S$.

σ is proper iff $\Pr_s^{\sigma}(\Diamond G) = 1$ for all $s \in T$

Maximal expected accumulated weight

given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$ s.t.

$$T = \{ s \in S : \Pr^{\max}_s(\lozenge G) = 1 \} \neq \varnothing$$

task: compute $x_s = \mathbb{E}^{\max}_s(\oplus G)$ for $s \in T$

W.l.o.g. $T = S$.

replace $\mathcal{M}$ with the sub-MDP consisting of

- the states in T and
- the state-action pairs (s, α) where $s \in T \setminus G$, $\alpha \in Act(s)$ and

$$\Pr^{\max}_s(\lozenge G) = \sum_{t \in S} P(s, \alpha, t) \cdot \Pr^{\max}_t(\lozenge G)$$

Maximal expected accumulated weight

given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$ s.t.
$T = \{s \in S : \Pr_s^{\max}(\Diamond G) = 1\} \neq \varnothing$

task: compute $x_s = \mathbb{E}_s^{\max}(\oplus G)$ for $s \in T$

W.l.o.g. $T = S$. In particular, $s \models \exists \Diamond G$ for all $s \in S$.

replace $\mathcal{M}$ with the sub-MDP consisting of

- the states in T and
- the state-action pairs (s, α) where $s \in T \setminus G$, $\alpha \in Act(s)$ and

$$\Pr_s^{\max}(\Diamond G) = \sum_{t \in S} P(s, \alpha, t) \cdot \Pr_t^{\max}(\Diamond G)$$

Maximal expected accumulated weight

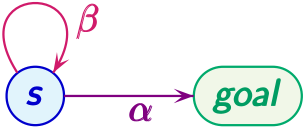
given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$ s.t.
$T = \{s \in S : \Pr_s^{\max}(\Diamond G) = 1\} \neq \varnothing$

task: compute $x_s = \mathbb{E}_s^{\max}(\oplus G)$ for $s \in T$

W.l.o.g. $T = S$. In particular, $s \models \exists \Diamond G$ for all $s \in S$.

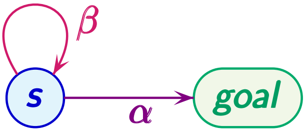
$\mathbb{E}_s^{\max}(\oplus goal)$ can be infinite !

Maximal expected accumulated weight



$$wgt(s, \alpha) = 0$$
$$wgt(s, \beta) = 1$$

Maximal expected accumulated weight



$$wgt(s,\alpha) = 0$$
$$wgt(s,\beta) = 1$$

maximal expected acumulated weight:

$$\mathbb{E}^{\max}_{s}(\oplus goal) = +\infty$$

note that $\mathbb{E}^{\sigma_n}_{s}(\oplus goal) = n$ where σ_n schedules

- β for the first n visits of s
- α for the $(n+1)$-st visit of s

Maximal expected accumulated weight

given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$ s.t.
$\Pr_s^{\max}(\Diamond G) = 1$ for all $s \in S$

task: compute $x_s = \mathbb{E}_s^{\max}(\oplus G)$ for $s \in S$

If $\mathbb{E}_s^{\sigma}($"total weight"$) = -\infty$ for each improper scheduler σ then: [Bertsekas/Tsitsiklis'91]

$x_s < +\infty$ for all $s \in S$

Maximal expected accumulated weight

given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$ s.t.
$\Pr^{\max}_s(\Diamond G) = 1$ for all $s \in S$

task: compute $x_s = \mathbb{E}^{\max}_s(\oplus G)$ for $s \in S$

If $\mathbb{E}^{\sigma}_s($"total weight"$) = -\infty$ for each improper scheduler σ then: [Bertsekas/Tsitsiklis'91]

- $x_s < +\infty$ for all $s \in S$
- the vector $(x_s)_{s \in S}$ is computable via the Bellman equations

Maximal expected accumulated weight

given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$ s.t. $\Pr^{\max}_s(\Diamond G) = 1$ for all $s \in S$

task: compute $x_s = \mathbb{E}^{\max}_s(\oplus G)$ for $s \in S$

If $\mathbb{E}^{\sigma}_s($"total weight"$) = -\infty$ for each improper scheduler σ then: [BERTSEKAS/TSITSIKLIS'91]

If $s \in G$ then $x_s = 0$. Otherwise:

$$x_s = \max_{\alpha \in Act(s)} \Big(wgt(s, \alpha) + \sum_{t \in S} P(s, \alpha, t) \cdot x_t \Big)$$

Maximal expected accumulated weight

given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$ s.t. $\Pr^{\max}_s(\Diamond G) = 1$ for all $s \in S$

task: compute $x_s = \mathbb{E}^{\max}_s(\oplus G)$ for $s \in S$

If $\mathbb{E}^{\sigma}_s($"total weight"$) = -\infty$ for each improper scheduler σ then: [Bertsekas/Tsitsiklis'91]

If $s \in G$ then $x_s = 0$. Otherwise:

$$x_s = \max_{\alpha \in Act(s)} \Bigl(wgt(s,\alpha) + \sum_{t \in S} P(s,\alpha,t) \cdot x_t \Bigr)$$

... unique fixpoint

Maximal expected accumulated weight

given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$ s.t.
$\Pr^{\max}_s(\Diamond G) = 1$ for all $s \in S$

task: compute $x_s = \mathbb{E}^{\max}_s(\oplus G)$ for $s \in S$

If $\mathbb{E}^{\sigma}_s(\text{"total weight"}) = -\infty$ for each improper scheduler σ then: [Bertsekas/Tsitsiklis'91]

If $s \in G$ then $x_s = 0$. Otherwise:

$$x_s = \max_{\alpha \in Act(s)} \Big(wgt(s,\alpha) + \sum_{t \in S} P(s,\alpha,t) \cdot x_t \Big)$$

... unique fixpoint, optimal MD-scheduler exist ...

Maximal expected accumulated weight

given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$ s.t.
$\Pr^{\max}_s(\Diamond G) = 1$ for all $s \in S$

task: compute $x_s = \mathbb{E}^{\max}_s(\oplus G)$ for $s \in S$

If $\mathbb{E}^{\sigma}_s($ "total weight" $) = -\infty$ for each improper scheduler σ then: [Bertsekas/Tsitsiklis'91]

If $s \in G$ then $x_s = 0$. Otherwise:

$$x_s \geqslant \max_{\alpha \in Act(s)} \Big(wgt(s, \alpha) + \sum_{t \in S} P(s, \alpha, t) \cdot x_t \Big)$$

unique solution where $\sum\limits_{s \in S} x_s$ is minimal

Maximal expected accumulated weight

given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$ s.t.
$\Pr^{\max}_s(\Diamond G) = 1$ for all $s \in S$

task: compute $x_s = \mathbb{E}^{\max}_s(\oplus G)$ for $s \in S$

If $\mathbb{E}^{\sigma}_s($ "total weight" $) = -\infty$ for each improper scheduler σ then: [Bertsekas/Tsitsiklis'91]

If $s \in G$ then $x_s^{(n)} = 0$. Otherwise:

$$x_s^{(n)} = \max_{\alpha \in Act(s)} \Big(wgt(s, \alpha) + \sum_{t \in S} P(s, \alpha, t) \cdot x_t^{(n-1)} \Big)$$

value iteration (arbitrary starting vector)

Maximal expected accumulated weight

given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$ s.t.
$\Pr_s^{\max}(\Diamond G) = 1$ for all $s \in S$

task: compute $x_s = \mathbb{E}_s^{\max}(\oplus G)$ for $s \in S$

If $\mathbb{E}_s^{\sigma}($"total weight"$) = -\infty$ for each improper scheduler σ then: [Bertsekas/Tsitsiklis'91]

- $x_s < +\infty$ for all $s \in S$
- the vector $(x_s)_{s \in S}$ is computable via the Bellman equations

How to compute x_s if $\mathbb{E}_s^{\sigma}($"total weight"$) > -\infty$ for some improper scheduler σ ?

Maximal expected accumulated weight

given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$ s.t.
$\Pr_s^{\max}(\lozenge G) = 1$ for all $s \in S$

task: compute $x_s = \mathbb{E}_s^{\max}(\oplus G)$ for $s \in S$

If $\mathbb{E}_s^{\sigma}($"total weight"$) = -\infty$ for each improper scheduler σ then: [BERTSEKAS/TSITSIKLIS'91]

- $x_s < +\infty$ for all $s \in S$
- the vector $(x_s)_{s \in S}$ is computable via the Bellman equations

How to compute x_s if $\mathbb{E}_s^{\sigma}($"total weight"$) > -\infty$ for some improper scheduler σ ? How to check finiteness ?

Non-negative weights

given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$ s.t. $\Pr^{\max}_s(\Diamond G) = 1$ for all $s \in S$

task: compute $x_s = \mathbb{E}^{\max}_s(\oplus G)$ for $s \in S$

consider the case of non-negative weights, i.e., $wgt(s, \alpha) \geqslant 0$ for all state-action pairs

Non-negative weights

given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$ s.t. $\Pr^{\max}_s(\Diamond G) = 1$ for all $s \in S$

task: compute $x_s = \mathbb{E}^{\max}_s(\oplus G)$ for $s \in S$

results: [De Alfaro'99]

- $\mathbb{E}^{\max}_s(\oplus G) = \infty$ iff s can reach a $\underbrace{\text{positive EC}}_{\text{end component that contains a state-action pair with positive weight}}$

Non-negative weights

given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$ s.t. $\Pr^{\max}_s(\Diamond G) = 1$ for all $s \in S$

task: compute $x_s = \mathbb{E}^{\max}_s(\oplus G)$ for $s \in S$

results: [De Alfaro'99]

- $\mathbb{E}^{\max}_s(\oplus G) = \infty$ iff s can reach a positive EC
- if $\mathcal{M}$ has no positive ECs and $\mathcal{N} = \mathrm{MEC}(\mathcal{M})$ then:

$$\mathbb{E}^{\max}_{\mathcal{M},s}(\oplus G) = \mathbb{E}^{\max}_{\mathcal{N},s}(\oplus G)$$

The MEC-quotient has no end components and maximal expected accumulated weights are computable using the Bellman equations.

Non-negative weights

given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$ s.t. $\Pr^{\max}_s(\Diamond G) = 1$ for all $s \in S$

task: compute $x_s = \mathbb{E}^{\max}_s(\oplus G)$ for $s \in S$

results: [De Alfaro'99]

- $\mathbb{E}^{\max}_s(\oplus G) = \infty$ iff s can reach a positive EC
- if $\mathcal{M}$ has no positive ECs and $\mathcal{N} = \mathrm{MEC}(\mathcal{M})$ then:

$$\mathbb{E}^{\max}_{\mathcal{M},s}(\oplus G) = \mathbb{E}^{\max}_{\mathcal{N},s}(\oplus G)$$

Hence: $\mathbb{E}^{\max}_{\mathcal{M},s}(\oplus G)$ is computable in polynomial time.

Non-positive weights

given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$ s.t. $\Pr^{\max}_s(\Diamond G) = 1$ for all $s \in S$

task: compute $x_s = \mathbb{E}^{\max}_s(\oplus G)$ for $s \in S$

results: [De Alfaro'99]

- $\mathbb{E}^{\max}_s(\oplus G)$ is finite ... and non-positive

Non-positive weights

given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$ s.t.
$\Pr^{\max}_s(\Diamond G) = 1$ for all $s \in S$

task: compute $x_s = \mathbb{E}^{\max}_s(\oplus G)$ for $s \in S$

results: [De Alfaro'99]

- $\mathbb{E}^{\max}_s(\oplus G)$ is finite ... and non-positive
- if $\mathcal{N}$ is the MDP arising from $\mathcal{M}$ by collapsing all zero-ECs then ...

end component $\mathcal{E}$ with $wgt(s, \alpha) = 0$
for all state-action pairs (s, α) in $\mathcal{E}$

Non-positive weights

given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$ s.t. $\Pr^{\max}_s(\Diamond G) = 1$ for all $s \in S$

task: compute $x_s = \mathbb{E}^{\max}_s(\oplus G)$ for $s \in S$

results: [De Alfaro'99]

- $\mathbb{E}^{\max}_s(\oplus G)$ is finite ... and non-positive
- if $\mathcal{N}$ is the MDP arising from $\mathcal{M}$ by collapsing all zero-ECs then $\mathbb{E}^{\max}_{\mathcal{M},s}(\oplus G) = \mathbb{E}^{\max}_{\mathcal{N},s}(\oplus G)$

$\uparrow$

end component $\mathcal{E}$ with $wgt(s, \alpha) = 0$ for all state-action pairs (s, α) in $\mathcal{E}$

Non-positive weights

given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$ s.t. $\Pr^{\max}_s(\Diamond G) = 1$ for all $s \in S$

task: compute $x_s = \mathbb{E}^{\max}_s(\diamondplus G)$ for $s \in S$

results: [De Alfaro'99]

- $\mathbb{E}^{\max}_s(\diamondplus G)$ is finite ... and non-positive
- if $\mathcal{N}$ is the MDP arising from $\mathcal{M}$ by collapsing all zero-ECs then $\mathbb{E}^{\max}_{\mathcal{M},s}(\diamondplus G) = \mathbb{E}^{\max}_{\mathcal{N},s}(\diamondplus G)$

$\uparrow$

computable as the MECs of the MDP $\mathcal{M}_0$ consisting of the state-action pairs in $\mathcal{M}$ with weight $\mathbf{0}$

Non-positive weights

given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$ s.t. $\Pr^{\max}_s(\Diamond G) = 1$ for all $s \in S$

task: compute $x_s = \mathbb{E}^{\max}_s(\oplus G)$ for $s \in S$

results: [De Alfaro'99]

- $\mathbb{E}^{\max}_s(\oplus G)$ is finite ... and non-positive
- if $\mathcal{N}$ is the MDP arising from $\mathcal{M}$ by collapsing all zero-ECs then $\mathbb{E}^{\max}_{\mathcal{M},s}(\oplus G) \ = \ \mathbb{E}^{\max}_{\mathcal{N},s}(\oplus G)$
- $\mathbb{E}^{\max}_{\mathcal{N},s}(\oplus G)$ computable via Bellman equations

  ... expected total weight of each improper scheduler is $-\infty$

Maximal expected accumulated weight

given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$ s.t. $\Pr^{\max}_s(\Diamond G) = 1$ for all $s \in S$

task: compute $x_s = \mathbb{E}^{\max}_s(\oplus G)$ for $s \in S$

If $\mathbb{E}^{\sigma}_s($"total weight"$) = -\infty$ for each improper scheduler σ then: [Bertsekas/Tsitsiklis'91]

- $x_s < +\infty$ for all $s \in S$
- $(x_s)_{s \in S}$ is computable via the Bellman equations

Treatment of non-negative or non-positive weights: $\surd$

General case: **???**

Maximal expected accumulated weight

given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$ s.t.
$\Pr^{\max}_s(\Diamond G) = 1$ for all $s \in S$

task: compute $x_s = \mathbb{E}^{\max}_s(\oplus G)$ for $s \in S$

If $\mathbb{E}^{\sigma}_s($"total weight"$) = -\infty$ for each improper scheduler σ then: [Bertsekas/Tsitsiklis'91]

- $x_s < +\infty$ for all $s \in S$
- $(x_s)_{s \in S}$ is computable via the Bellman equations

Treatment of non-negative or non-positive weights: $\surd$

General case: ... consider the MECs separately ...

Let $\mathcal{E}$ be an end component of $\mathcal{M}$.

$$\mathbb{E}^{\max}_{\mathcal{E}}(\diamondplus G) = \infty$$

iff $\quad \ldots$

Let $\mathcal{E}$ be an end component of $\mathcal{M}$.

$$\mathbb{E}^{\max}_{\mathcal{E}}(\Diamond G) = \infty$$

iff $\mathcal{E}$ is weight-divergent

Let $\mathcal{E}$ be an end component of $\mathcal{M}$.

$$\mathbb{E}^{\max}_{\mathcal{E}}(\oplus G) = \infty$$

iff $\mathcal{E}$ is weight-divergent, i.e., for all states s in $\mathcal{E}$:

$$\sup \left\{ r \in \mathbb{N} : \Pr^{\max}_{\mathcal{E},s}(\Diamond(wgt \geqslant r)) = 1 \right\} = \infty$$

Let $\mathcal{E}$ be an end component of $\mathcal{M}$.

$$\mathbb{E}^{\max}_{\mathcal{E}}(\diamondplus G) = \infty$$

iff $\mathcal{E}$ is weight-divergent, i.e., for all states s in $\mathcal{E}$:

$$\sup \{ r \in \mathbb{N} : \Pr^{\max}_{\mathcal{E},s}(\Diamond(wgt \geqslant r)) = 1 \} = \infty$$

iff $\Pr^{\max}_{\mathcal{E}}\{ \pi : \limsup_{n\to\infty} wgt(pref(\pi, n)) = \infty \} = 1$

$pref(\pi, n)$ = prefix of π of length n

Let $\mathcal{E}$ be an end component of $\mathcal{M}$.

$$\mathbb{E}^{\max}_{\mathcal{E}}(\oplus G) = \infty$$

iff $\mathcal{E}$ is weight-divergent, i.e., for all states s in $\mathcal{E}$:

$$\sup \left\{ r \in \mathbb{N} \,:\, \Pr^{\max}_{\mathcal{E},s}(\,\Diamond(wgt \geqslant r)\,) = 1 \right\} = \infty$$

iff $\Pr^{\max}_{\mathcal{E}}\left\{ \pi \,:\, \limsup\limits_{n\to\infty} wgt(pref(\pi, n)) = \infty \right\} = 1$

iff $\mathbb{E}^{\max}_{\mathcal{E}}(\mathrm{MP}) > 0$ or $\ldots$

$pref(\pi, n)$ = prefix of π of length n

Let $\mathcal{E}$ be an end component of $\mathcal{M}$.

$$\mathbb{E}^{\max}_{\mathcal{E}}(\oplus G) = \infty$$

iff $\mathcal{E}$ is weight-divergent, i.e., for all states s in $\mathcal{E}$:

$$\sup \{ r \in \mathbb{N} : \Pr^{\max}_{\mathcal{E},s}(\lozenge(wgt \geqslant r)) = 1 \} = \infty$$

iff $\Pr^{\max}_{\mathcal{E}}\{ \pi : \limsup_{n \to \infty} wgt(pref(\pi, n)) = \infty \} = 1$

iff $\mathbb{E}^{\max}_{\mathcal{E}}(\mathrm{MP}) > 0$ or $\mathbb{E}^{\max}_{\mathcal{E}}(\mathrm{MP}) = 0$ & $\mathcal{E}$ is gambling

$pref(\pi, n) =$ prefix of π of length n

Let $\mathcal{E}$ be an end component of $\mathcal{M}$.

$$\mathbb{E}^{\max}_{\mathcal{E}}(\oplus G) = \infty$$

iff $\mathcal{E}$ is weight-divergent, i.e., for all states s in $\mathcal{E}$:

$$\sup \{ r \in \mathbb{N} : \Pr^{\max}_{\mathcal{E},s}(\Diamond(wgt \geqslant r)) = 1 \} = \infty$$

iff $\Pr^{\max}_{\mathcal{E}}\{ \pi : \limsup_{n\to\infty} wgt(pref(\pi, n)) = \infty \} = 1$

iff $\mathbb{E}^{\max}_{\mathcal{E}}(\mathrm{MP}) > 0$ or $\mathbb{E}^{\max}_{\mathcal{E}}(\mathrm{MP}) = 0$ & $\mathcal{E}$ is gambling

there exists scheduler s.t. almost surely:

$$\limsup_{n\to\infty} wgt(pref(\pi, n)) = +\infty$$

$$\liminf_{n\to\infty} wgt(pref(\pi, n)) = -\infty$$

Let $\mathcal{E}$ be an end component of $\mathcal{M}$.

$$\mathbb{E}^{\max}_{\mathcal{E}}(\oplus G) = \infty$$

iff $\mathcal{E}$ is weight-divergent, i.e., for all states s in $\mathcal{E}$:

$$\sup \{ r \in \mathbb{N} : \Pr^{\max}_{s}(\Diamond(wgt \geqslant r)) = 1 \} = \infty$$

iff $\Pr^{\max}_{\mathcal{E}} \{ \pi : \limsup_{n \to \infty} wgt(pref(\pi, n)) = \infty \} = 1$

iff $\mathbb{E}^{\max}_{\mathcal{E}}(\mathrm{MP}) > 0$ or $\mathbb{E}^{\max}_{\mathcal{E}}(\mathrm{MP}) = 0$ & $\mathcal{E}$ is gambling

can be checked in
polynomial time

exists scheduler s.t. almost surely:

$$\limsup_{n \to \infty} \; wgt(pref(\pi, n)) = +\infty$$

$$\liminf_{n \to \infty} \; wgt(pref(\pi, n)) = -\infty$$

Let $\mathcal{E}$ be an end component of $\mathcal{M}$.

$$\mathbb{E}^{\max}_{\mathcal{E}}(\oplus G) = \infty$$

iff $\mathcal{E}$ is weight-divergent, i.e., for all states s in $\mathcal{E}$:

$$\sup \{ r \in \mathbb{N} : \Pr^{\max}_{s}(\Diamond(wgt \geqslant r)) = 1 \} = \infty$$

iff $\Pr^{\max}_{\mathcal{E}}\{ \pi : \limsup_{n\to\infty} wgt(pref(\pi, n)) = \infty \} = 1$

iff $\mathbb{E}^{\max}_{\mathcal{E}}(\mathrm{MP}) > 0$ or $\mathbb{E}^{\max}_{\mathcal{E}}(\mathrm{MP}) = 0$ & $\mathcal{E}$ is gambling

can be checked in polynomial time

how to check whether an EC is gambling ?

Let $\mathcal{E}$ be an end component of $\mathcal{M}$.

$$\mathbb{E}^{\max}_{\mathcal{E}}(\oplus G) = \infty$$

iff $\mathcal{E}$ is weight-divergent, i.e., for all states s in $\mathcal{E}$:

$$\sup \left\{ r \in \mathbb{N} : \Pr^{\max}_{s}(\Diamond(wgt \geqslant r)) = 1 \right\} = \infty$$

iff $\Pr^{\max}_{\mathcal{E}}\left\{ \pi : \limsup_{n \to \infty} wgt(pref(\pi, n)) = \infty \right\} = 1$

iff $\mathbb{E}^{\max}_{\mathcal{E}}(\mathrm{MP}) > 0$ or $\mathbb{E}^{\max}_{\mathcal{E}}(\mathrm{MP}) = 0$ & $\mathcal{E}$ is gambling

The problem to check whether a given EC is gambling is NP-hard

Let $\mathcal{E}$ be an end component of $\mathcal{M}$.

$$\mathbb{E}^{\max}_{\mathcal{E}}(\oplus G) = \infty$$

iff $\mathcal{E}$ is weight-divergent, i.e., for all states s in $\mathcal{E}$:

$$\sup \left\{ r \in \mathbb{N} : \Pr^{\max}_{s}(\Diamond(wgt \geqslant r)) = 1 \right\} = \infty$$

iff $\Pr^{\max}_{\mathcal{E}}\left\{ \pi : \limsup_{n\to\infty} wgt(pref(\pi, n)) = \infty \right\} = 1$

iff $\mathbb{E}^{\max}_{\mathcal{E}}(\mathrm{MP}) > 0$ or $\mathbb{E}^{\max}_{\mathcal{E}}(\mathrm{MP}) = 0$ & $\mathcal{E}$ is gambling

The problem to check whether a given EC is gambling

- is NP-hard
- solvable in polynomial-time if $\mathbb{E}^{\max}_{\mathcal{E}}(\mathrm{MP}) = 0$

Non-gambling EC with zero mean-payoff

Let $\mathcal{E}$ be an end component of $\mathcal{M}$ with $\mathbb{E}^{\max}_{\mathcal{E}}(\mathrm{MP}) = 0$.

Non-gambling EC with zero mean-payoff

Let $\mathcal{E}$ be an end component of $\mathcal{M}$ with $\mathbb{E}^{\max}_{\mathcal{E}}(\mathrm{MP}) = 0$.

Pick an MD-scheduler σ s.t. $\mathbb{E}^{\sigma}_{\mathcal{E},s}(\mathrm{MP}) = 0$ for $s \in \mathcal{E}$

Non-gambling EC with zero mean-payoff

Let $\mathcal{E}$ be an end component of $\mathcal{M}$ with $\mathbb{E}^{\max}_{\mathcal{E}}(\mathrm{MP}) = 0$.

Pick an MD-scheduler σ s.t. $\mathbb{E}^{\sigma}_{\mathcal{E},s}(\mathrm{MP}) = 0$ for $s \in \mathcal{E}$ and a BSCC $\mathcal{E}'$ of σ.

$\mathcal{E}'$ is a finite strongly connected Markov chain

Non-gambling EC with zero mean-payoff

Let $\mathcal{E}$ be an end component of $\mathcal{M}$ with $\mathbb{E}^{\max}_{\mathcal{E}}(\mathrm{MP}) = 0$.

Pick an MD-scheduler σ s.t. $\mathbb{E}^{\sigma}_{\mathcal{E},s}(\mathrm{MP}) = 0$ for $s \in \mathcal{E}$ and a BSCC $\mathcal{E}'$ of σ. W.l.o.g. $\mathcal{E} = \mathcal{E}'$.

$\mathcal{E}$ is a finite strongly connected Markov chain

Non-gambling EC with zero mean-payoff

Let $\mathcal{E}$ be an end component of $\mathcal{M}$ with $\mathbb{E}^{\max}_{\mathcal{E}}(\mathrm{MP}) = \mathbf{0}$.

Pick an MD-scheduler σ s.t. $\mathbb{E}^{\sigma}_{\mathcal{E},s}(\mathrm{MP}) = \mathbf{0}$ for $s \in \mathcal{E}$ and a BSCC $\mathcal{E}'$ of σ. W.l.o.g. $\mathcal{E} = \mathcal{E}'$.

> If $\mathcal{E}$ is not gambling then $\mathcal{E}$ is a zero-EC

$\mathcal{E}$ is a finite strongly connected Markov chain

Non-gambling EC with zero mean-payoff

Let $\mathcal{E}$ be an end component of $\mathcal{M}$ with $\mathbb{E}^{\max}_{\mathcal{E}}(\mathrm{MP}) = \mathbf{0}$.

Pick an MD-scheduler σ s.t. $\mathbb{E}^{\sigma}_{\mathcal{E},s}(\mathrm{MP}) = \mathbf{0}$ for $s \in \mathcal{E}$ and a BSCC $\mathcal{E}'$ of σ. W.l.o.g. $\mathcal{E} = \mathcal{E}'$.

If $\mathcal{E}$ is not gambling then $\mathcal{E}$ is a zero-EC, i.e., the total weight of all cycles in $\mathcal{E}$ is 0.

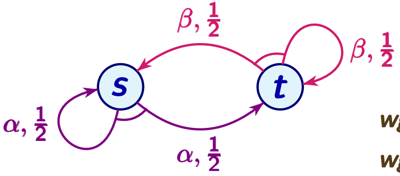
$\mathcal{E}$ is a finite strongly connected Markov chain

Non-gambling EC with zero mean-payoff

Let $\mathcal{E}$ be an end component of $\mathcal{M}$ with $\mathbb{E}^{\max}_{\mathcal{E}}(\mathrm{MP}) = 0$.

Pick an MD-scheduler σ s.t. $\mathbb{E}^{\sigma}_{\mathcal{E},s}(\mathrm{MP}) = 0$ for $s \in \mathcal{E}$ and a BSCC $\mathcal{E}'$ of σ. W.l.o.g. $\mathcal{E} = \mathcal{E}'$.

> If $\mathcal{E}$ is not gambling then $\mathcal{E}$ is a zero-EC, i.e., the total weight of all cycles in $\mathcal{E}$ is 0.



$wgt(s, \alpha) = +1$
$wgt(t, \beta) = -1$

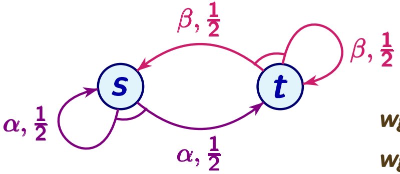
gambling

Non-gambling EC with zero mean-payoff

Let $\mathcal{E}$ be an end component of $\mathcal{M}$ with $\mathbb{E}^{\max}_{\mathcal{E}}(\mathrm{MP}) = 0$.

Pick an MD-scheduler σ s.t. $\mathbb{E}^{\sigma}_{\mathcal{E},s}(\mathrm{MP}) = 0$ for $s \in \mathcal{E}$ and a BSCC $\mathcal{E}'$ of σ. W.l.o.g. $\mathcal{E} = \mathcal{E}'$.

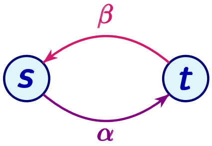
> If $\mathcal{E}$ is not gambling then $\mathcal{E}$ is a zero-EC, i.e., the total weight of all cycles in $\mathcal{E}$ is 0.



gambling

$wgt(s, \alpha) = +1$

$wgt(t, \beta) = -1$



zero-EC

Non-gambling EC with zero mean-payoff

Let $\mathcal{E}$ be an end component of $\mathcal{M}$ with $\mathbb{E}^{\max}_{\mathcal{E}}(\mathrm{MP}) = 0$.

Pick an MD-scheduler σ s.t. $\mathbb{E}^{\sigma}_{\mathcal{E},s}(\mathrm{MP}) = 0$ for $s \in \mathcal{E}$ and a BSCC $\mathcal{E}'$ of σ. W.l.o.g. $\mathcal{E} = \mathcal{E}'$.

If $\mathcal{E}$ is not gambling then $\mathcal{E}$ is a zero-EC, i.e., the total weight of all cycles in $\mathcal{E}$ is 0.

Let $\mathcal{E}$ be a zero-EC and s, t states in $\mathcal{E}$. There exists $w(s,t) \in \mathbb{Z}$ such that:

$$w(s,t) = wgt(\pi) \quad \text{for all paths } \pi \text{ from } s \text{ to } t$$

Non-gambling EC with zero mean-payoff

Let $\mathcal{E}$ be an end component of $\mathcal{M}$ with $\mathbb{E}^{\max}_{\mathcal{E}}(\mathrm{MP}) = 0$.

Pick an MD-scheduler σ s.t. $\mathbb{E}^{\sigma}_{\mathcal{E},s}(\mathrm{MP}) = 0$ for $s \in \mathcal{E}$ and a BSCC $\mathcal{E}'$ of σ. W.l.o.g. $\mathcal{E} = \mathcal{E}'$.

> If $\mathcal{E}$ is not gambling then $\mathcal{E}$ is a zero-EC, i.e., the total weight of all cycles in $\mathcal{E}$ is 0.

Let $\mathcal{E}$ be a zero-EC and s, t states in $\mathcal{E}$. There exists $w(s,t) \in \mathbb{Z}$ such that:

$$w(s,t) = wgt(\pi) \quad \text{for all paths } \pi \text{ from } s \text{ to } t$$

Then: $w(t,s) = -w(s,t)$

Non-gambling EC with zero mean-payoff

Let $\mathcal{E}$ be an end component of $\mathcal{M}$ with $\mathbb{E}^{\max}_{\mathcal{E}}(\mathrm{MP}) = \mathbf{0}$.

Pick an MD-scheduler σ s.t. $\mathbb{E}^{\sigma}_{\mathcal{E},s}(\mathrm{MP}) = \mathbf{0}$ for $s \in \mathcal{E}$ and a BSCC $\mathcal{E}'$ of σ. W.l.o.g. $\mathcal{E} = \mathcal{E}'$.

> If $\mathcal{E}$ is not gambling then $\mathcal{E}$ is a zero-EC, i.e., the total weight of all cycles in $\mathcal{E}$ is 0.

Let $\mathcal{E}$ be a zero-EC and s, t states in $\mathcal{E}$. There exists $w(s,t) \in \mathbb{Z}$ such that:

$$w(s,t) = wgt(\pi) \quad \text{for all paths } \pi \text{ from } s \text{ to } t$$

Then: $w(t,s) = -w(s,t)$

... remove $\mathcal{E}$ from $\mathcal{M}$ ...

Spider construction ... for removing zero-ECs

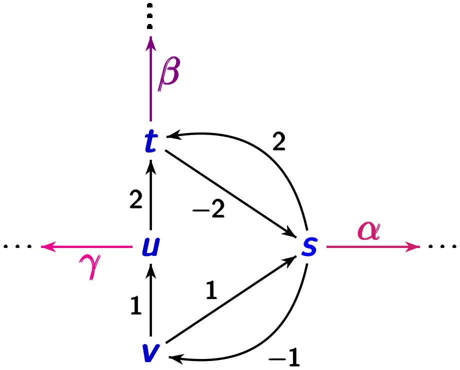
given: MDP $\mathcal{M}$ and a zero-EC $\mathcal{E}$ of $\mathcal{M}$

task: construct an MDP $\mathcal{N}$ with the same non-zero ECs and where $\mathcal{E}$ is no longer a zero-EC

Spider construction ... for removing zero-ECs

given: MDP $\mathcal{M}$ and a zero-EC $\mathcal{E}$ of $\mathcal{M}$

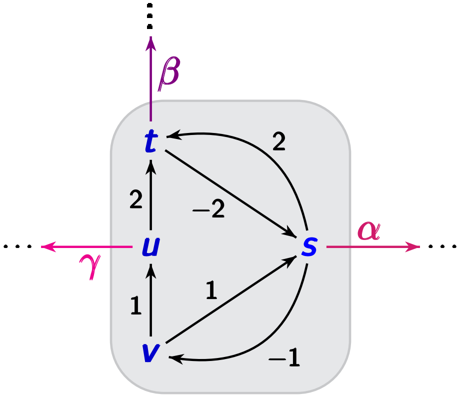
task: construct an MDP $\mathcal{N}$ with the same non-zero ECs and where $\mathcal{E}$ is no longer a zero-EC



Spider construction … for removing zero-ECs

given: MDP $\mathcal{M}$ and a zero-EC $\mathcal{E}$ of $\mathcal{M}$

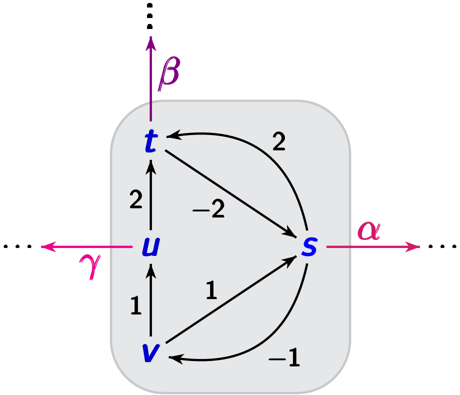
task: construct an MDP $\mathcal{N}$ with the same non-zero ECs and where $\mathcal{E}$ is no longer a zero-EC

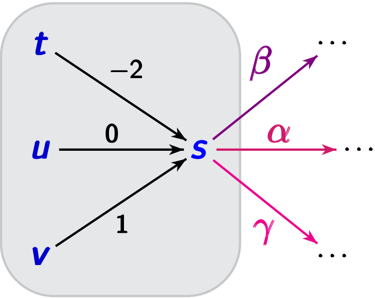


Spider construction ... for removing zero-ECs

given: MDP $\mathcal{M}$ and a zero-EC $\mathcal{E}$ of $\mathcal{M}$

task: construct an MDP $\mathcal{N}$ with the same non-zero ECs and where $\mathcal{E}$ is no longer a zero-EC

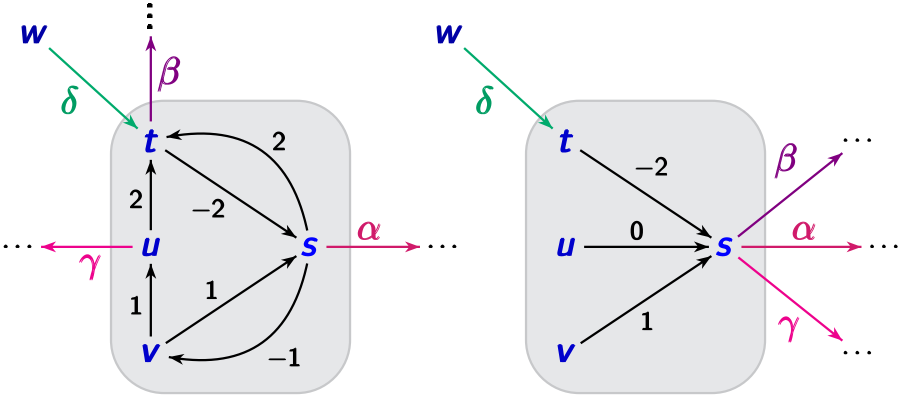




Spider construction ... for removing zero-ECs

given: MDP $\mathcal{M}$ and a zero-EC $\mathcal{E}$ of $\mathcal{M}$

task: construct an MDP $\mathcal{N}$ with the same non-zero ECs and where $\mathcal{E}$ is no longer a zero-EC



Spider construction ... for removing zero-ECs

given: MDP $\mathcal{M}$ and a zero-EC $\mathcal{E}$ of $\mathcal{M}$

task: construct an MDP $\mathcal{N}$ with the same non-zero ECs and where $\mathcal{E}$ is no longer a zero-EC

W.l.o.g: $Act(s) \cap Act(t) = \varnothing$ if $s \neq t$.

Spider construction ... for removing zero-ECs

given: MDP $\mathcal{M}$ and a zero-EC $\mathcal{E}$ of $\mathcal{M}$

1. pick a state s in $\mathcal{E}$

Spider construction ... for removing zero-ECs

given: MDP $\mathcal{M}$ and a zero-EC $\mathcal{E}$ of $\mathcal{M}$

1. pick a state s in $\mathcal{E}$
2. remove all state-action pairs in $\mathcal{E}$

Spider construction ... for removing zero-ECs

given: MDP $\mathcal{M}$ and a zero-EC $\mathcal{E}$ of $\mathcal{M}$

1. pick a state s in $\mathcal{E}$
2. remove all state-action pairs in $\mathcal{E}$
3. for each state t in $\mathcal{E}$ with $t \neq s$:
   add transition $t \xrightarrow{\tau} s$ with $wgt(t, \tau) = \underbrace{-w(s,t)}_{w(t,s)}$

Spider construction ... for removing zero-ECs

given: MDP $\mathcal{M}$ and a zero-EC $\mathcal{E}$ of $\mathcal{M}$

1. pick a state s in $\mathcal{E}$
2. remove all state-action pairs in $\mathcal{E}$
3. for each state t in $\mathcal{E}$ with $t \neq s$:
   add transition $t \xrightarrow{\tau} s$ with $wgt(t, \tau) = -w(s, t)$
4. replace each state-action pair (t, β) in $\mathcal{M} \setminus \mathcal{E}$
   where $t \neq s$ with the pair (s, β)

$$s \overset{\text{in } \mathcal{E}}{\dashrightarrow} t \xrightarrow{\beta} \ldots$$

Spider construction ... for removing zero-ECs

given: MDP $\mathcal{M}$ and a zero-EC $\mathcal{E}$ of $\mathcal{M}$

1. pick a state s in $\mathcal{E}$
2. remove all state-action pairs in $\mathcal{E}$
3. for each state t in $\mathcal{E}$ with $t \neq s$:
   add transition $t \xrightarrow{\tau} s$ with $wgt(t, \tau) = -w(s, t)$
4. replace each state-action pair (t, β) in $\mathcal{M} \setminus \mathcal{E}$ where $t \neq s$ with the pair (s, β):

$$wgt(s, \beta) \;=\; w(s, t) + wgt(t, \beta)$$

$$s \xrightarrow{\text{in } \mathcal{E}} t \xrightarrow{\beta} \ldots$$

Spider construction ... for removing zero-ECs

given: MDP $\mathcal{M}$ and a zero-EC $\mathcal{E}$ of $\mathcal{M}$

1. pick a state s in $\mathcal{E}$
2. remove all state-action pairs in $\mathcal{E}$
3. for each state t in $\mathcal{E}$ with $t \neq s$:
   add transition $t \xrightarrow{\tau} s$ with $wgt(t,\tau) = -w(s,t)$
4. replace each state-action pair (t,β) in $\mathcal{M} \setminus \mathcal{E}$
   where $t \neq s$ with the pair (s,β):

$$wgt(s,\beta) = w(s,t) + wgt(t,\beta)$$

$$P(s,\beta,u) = P(t,\beta,u) \text{ for all states } u \text{ in } \mathcal{M}$$

Spider construction ... for removing zero-ECs

given: MDP $\mathcal{M}$ and a zero-EC $\mathcal{E}$ of $\mathcal{M}$

spider construction yields a new MDP $\mathcal{N} = \mathcal{M}_{\setminus\mathcal{E}}$

$$\mathcal{M} \text{ is weight-divergent iff } \mathcal{N} \text{ is weight-divergent}$$

Spider construction ... for removing zero-ECs

given: MDP $\mathcal{M}$ and a zero-EC $\mathcal{E}$ of $\mathcal{M}$

spider construction yields a new MDP $\mathcal{N} = \mathcal{M}_{\setminus\mathcal{E}}$

- $\mathcal{M}$ is weight-divergent iff $\mathcal{N}$ is weight-divergent
- $\mathbb{E}^{\max}_{\mathcal{M},s}(\oplus G) = \mathbb{E}^{\max}_{\mathcal{N},s}(\oplus G)$ for all states s in $\mathcal{M}$

Spider construction ... for removing zero-ECs

given: MDP $\mathcal{M}$ and a zero-EC $\mathcal{E}$ of $\mathcal{M}$

spider construction yields a new MDP $\mathcal{N} = \mathcal{M}_{\setminus \mathcal{E}}$

- $\mathcal{M}$ is weight-divergent iff $\mathcal{N}$ is weight-divergent
- $\mathbb{E}^{\max}_{\mathcal{M},s}(\diamondplus G) = \mathbb{E}^{\max}_{\mathcal{N},s}(\diamondplus G)$ for all states s in $\mathcal{M}$
- $\|\mathcal{N}\| \leqslant \|\mathcal{M}\| - 1$

where $\|\mathcal{M}\|$ = number of state-action pairs in $\mathcal{M}$

Spider construction ... for removing zero-ECs

given: MDP $\mathcal{M}$ and a zero-EC $\mathcal{E}$ of $\mathcal{M}$

spider construction yields a new MDP $\mathcal{N} = \mathcal{M}_{\backslash\mathcal{E}}$

- $\mathcal{M}$ is weight-divergent iff $\mathcal{N}$ is weight-divergent
- $\mathbb{E}^{\max}_{\mathcal{M},s}(\oplus G) = \mathbb{E}^{\max}_{\mathcal{N},s}(\oplus G)$ for all states s in $\mathcal{M}$
- $\|\mathcal{N}\| \leqslant \|\mathcal{M}\| - 1$

where $\|\mathcal{M}\|$ = number of state-action pairs in $\mathcal{M}$

idea: apply the spider construction recursively to check weight-divergence of strongly connected MDPs

Checking weight-divergence

given: strongly connected MDP $\mathcal{M}$ with $\mathbb{E}^{\max}_{\mathcal{M}}(\mathrm{MP}) \leqslant 0$

task: check if $\mathcal{M}$ is weight-divergent

Checking weight-divergence

given: strongly connected MDP $\mathcal{M}$ with $\mathbb{E}^{\max}_{\mathcal{M}}(\mathrm{MP}) \leqslant \mathbf{0}$

task: check if $\mathcal{M}$ is weight-divergent

1. compute $\mathbb{E}^{\max}_{\mathcal{M}}(\mathrm{MP})$ and an optimal MD-scheduler σ

Checking weight-divergence

given: strongly connected MDP $\mathcal{M}$ with $\mathbb{E}^{\max}_{\mathcal{M}}(\mathrm{MP}) \leqslant 0$

task: check if $\mathcal{M}$ is weight-divergent

1. compute $\mathbb{E}^{\max}_{\mathcal{M}}(\mathrm{MP})$ and an optimal MD-scheduler σ
2. if $\mathbb{E}^{\max}_{\mathcal{M}}(\mathrm{MP}) < 0$ then return "no"

$\uparrow$

$\mathcal{M}$ is not weight-divergent

as the total weight of almost all paths tends to $-\infty$

Checking weight-divergence

given: strongly connected MDP $\mathcal{M}$ with $\mathbb{E}^{\max}_{\mathcal{M}}(\mathrm{MP}) \leqslant 0$

task: check if $\mathcal{M}$ is weight-divergent

1. compute $\mathbb{E}^{\max}_{\mathcal{M}}(\mathrm{MP})$ and an optimal MD-scheduler σ
2. if $\mathbb{E}^{\max}_{\mathcal{M}}(\mathrm{MP}) < 0$ then return "no"
3. pick a $\underbrace{\text{BSCC } \mathcal{E}}_{\substack{\text{strongly connected MC with} \\ \text{expected mean-payoff } 0}}$ of the MC induced by σ

Checking weight-divergence

given: strongly connected MDP $\mathcal{M}$ with $\mathbb{E}^{\max}_{\mathcal{M}}(\mathrm{MP}) \leqslant 0$

task: check if $\mathcal{M}$ is weight-divergent

1. compute $\mathbb{E}^{\max}_{\mathcal{M}}(\mathrm{MP})$ and an optimal MD-scheduler σ
2. if $\mathbb{E}^{\max}_{\mathcal{M}}(\mathrm{MP}) < 0$ then return "no"
3. pick a BSCC $\mathcal{E}$ of the MC induced by σ
4. if $\mathcal{E}$ is a zero-EC then apply the procedure recursively to the MDP $\mathcal{M}_{\setminus\mathcal{E}}$.

   ↑

   spider construction

Checking weight-divergence

given: strongly connected MDP $\mathcal{M}$ with $\mathbb{E}^{\max}_{\mathcal{M}}(\mathrm{MP}) \leqslant 0$

task: check if $\mathcal{M}$ is weight-divergent

1. compute $\mathbb{E}^{\max}_{\mathcal{M}}(\mathrm{MP})$ and an optimal MD-scheduler σ
2. if $\mathbb{E}^{\max}_{\mathcal{M}}(\mathrm{MP}) < 0$ then return "no"
3. pick a BSCC $\mathcal{E}$ of the MC induced by σ
4. if $\mathcal{E}$ is a zero-EC then apply the procedure recursively to the MDP $\mathcal{M}_{\backslash\mathcal{E}}$.

   Otherwise ... $\mathcal{E}$ is gambling

Checking weight-divergence

given: strongly connected MDP $\mathcal{M}$ with $\mathbb{E}^{\max}_{\mathcal{M}}(\mathrm{MP}) \leqslant 0$

task: check if $\mathcal{M}$ is weight-divergent

1. compute $\mathbb{E}^{\max}_{\mathcal{M}}(\mathrm{MP})$ and an optimal MD-scheduler σ
2. if $\mathbb{E}^{\max}_{\mathcal{M}}(\mathrm{MP}) < 0$ then return "no"
3. pick a BSCC $\mathcal{E}$ of the MC induced by σ
4. if $\mathcal{E}$ is a zero-EC then apply the procedure recursively to the MDP $\mathcal{M}_{\backslash \mathcal{E}}$.

   Otherwise return "yes, $\mathcal{M}$ is weight-divergent".

Checking weight-divergence

given: strongly connected MDP $\mathcal{M}$ with $\mathbb{E}^{\max}_{\mathcal{M}}(\mathrm{MP}) \leqslant 0$

task: check if $\mathcal{M}$ is weight-divergent

1. compute $\mathbb{E}^{\max}_{\mathcal{M}}(\mathrm{MP})$ and an optimal MD-scheduler σ
2. if $\mathbb{E}^{\max}_{\mathcal{M}}(\mathrm{MP}) < 0$ then return "no"

If $\mathcal{M}$ is not weight-divergent then the algorithm has generated an MDP $\mathcal{N}$ with $\mathbb{E}^{\max}_{\mathcal{M},s}(\oplus G) = \mathbb{E}^{\max}_{\mathcal{N},s}(\oplus G)$

recursively to the MDP $\mathcal{M}_{\setminus \mathcal{E}}$.

Otherwise return "yes, $\mathcal{M}$ is weight-divergent".

Checking weight-divergence

given: strongly connected MDP $\mathcal{M}$ with $\mathbb{E}^{\max}_{\mathcal{M}}(\mathrm{MP}) \leqslant 0$

task: check if $\mathcal{M}$ is weight-divergent

1. compute $\mathbb{E}^{\max}_{\mathcal{M}}(\mathrm{MP})$ and an optimal MD-scheduler σ
2. if $\mathbb{E}^{\max}_{\mathcal{M}}(\mathrm{MP}) < 0$ then return "no"

If $\mathcal{M}$ is not weight-divergent then the algorithm has generated an MDP $\mathcal{N}$ with $\mathbb{E}^{\max}_{\mathcal{M},s}(\oplus G) = \mathbb{E}^{\max}_{\mathcal{N},s}(\oplus G)$ and $\mathbb{E}^{\sigma}_{\mathcal{N},s}(\text{"total weight"}) = -\infty$ for each improper scheduler σ. ... as $\mathcal{N}$ has no zero-ECs ...

Checking weight-divergence

given: strongly connected MDP $\mathcal{M}$ with $\mathbb{E}^{\max}_{\mathcal{M}}(\mathrm{MP}) \leqslant 0$

task: check if $\mathcal{M}$ is weight-divergent

1. compute $\mathbb{E}^{\max}_{\mathcal{M}}(\mathrm{MP})$ and an optimal MD-scheduler σ
2. if $\mathbb{E}^{\max}_{\mathcal{M}}(\mathrm{MP}) < 0$ then return "no"

If $\mathcal{M}$ is not weight-divergent then the algorithm has generated an MDP $\mathcal{N}$ with $\mathbb{E}^{\max}_{\mathcal{M},s}(\diamondplus G) = \mathbb{E}^{\max}_{\mathcal{N},s}(\diamondplus G)$ and $\mathbb{E}^{\sigma}_{\mathcal{N},s}(\text{"total weight"}) = -\infty$ for each improper scheduler σ.

... $\mathbb{E}^{\max}_{\mathcal{N},s}(\diamondplus G)$ computable via Bellman equations ...

Maximal expected accumulated weight

given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$
s.t. $\Pr_s^{\max}(\Diamond G) = 1$ for all $s \in S$

task: compute $x_s = \mathbb{E}_s^{\max}(\oplus G)$ for $s \in S$

If $\mathbb{E}_s^{\sigma}($"total weight"$) = -\infty$ for each improper scheduler σ then: [Bertsekas/Tsitsiklis'91]

- $x_s < +\infty$ for all $s \in S$
- $(x_s)_{s \in S}$ is computable via the Bellman equations

Maximal expected accumulated weight

given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$
s.t. $\Pr_s^{\max}(\Diamond G) = 1$ for all $s \in S$

task: compute $x_s = \mathbb{E}_s^{\max}(\oplus G)$ for $s \in S$

If $\mathbb{E}_s^{\sigma}($"total weight"$) = -\infty$ for each improper scheduler σ then: [Bertsekas/Tsitsiklis'91]

- $x_s < +\infty$ for all $s \in S$
- $(x_s)_{s \in S}$ is computable via the Bellman equations

Recursive application of the spider construction ...

- to check that there is no weight-divergent MEC

[Baier/Bertrand/Dubslaff/Gburek/Sankur'17]

Maximal expected accumulated weight

given: MDP $\mathcal{M} = (S, Act, P, wgt)$ and $G \subseteq S$
s.t. $\Pr_s^{\max}(\Diamond G) = 1$ for all $s \in S$

task: compute $x_s = \mathbb{E}_s^{\max}(\oplus G)$ for $s \in S$

If $\mathbb{E}_s^{\sigma}($ "total weight" $) = -\infty$ for each improper scheduler σ then: [Bertsekas/Tsitsiklis'91]

- $x_s < +\infty$ for all $s \in S$
- $(x_s)_{s \in S}$ is computable via the Bellman equations

Recursive application of the spider construction ...

- to check that there is no weight-divergent MEC
- to generate a new MDP $\mathcal{N}$ where $x_s = \mathbb{E}_{\mathcal{N},s}^{\max}(\oplus G)$ and the above criterion applies

Outline

- weighted Markov decision processes
- mean-payoff and long-run ratios
- expected accumulated weights
- conditional expected accumulated rewards
- weight-bounded reachability and quantiles
- LTL with weight assertions
- conclusions

Stochastic longest path problem

weighted MDP $\mathcal{M}$

$\oplus$ *goal*
accumulated weight until reaching a goal state

relaxed requirement:
$\Pr^{\max}(\Diamond goal) = 1$

BERTSEKAS/TSITSIKLIS'91
DE ALFARO'99

maximal expectation
$\mathbb{E}^{\max}(\oplus goal)$

maximum over all proper schedulers

Maximal conditional expectations

weighted
MDP $\mathcal{M}$

$\Diamond\!\!\!\!\!+\, goal$
accumulated weight until
reaching a goal state

relaxed requirement:
$\Pr^{\max}(\Diamond goal) > 0$

BAIER/KLEIN/
KLÜPPELHOLZ/
WUNDERLICH'17

maximal conditional expectation
$\mathbb{E}^{\max}(\, \Diamond\!\!\!\!\!+\, green \mid \Diamond goal\,)$

maximum over all positive schedulers

Maximal conditional expectations

weighted MDP $\mathcal{M}$

$\oplus\!\!\!\!\!\Diamond\,$ *goal*
accumulated weight until reaching a goal state

relaxed requirement:
$\Pr^{\max}(\Diamond \mathit{goal}) > 0$

assumption:
non-negative weights

maximal conditional expectation
$\mathbb{E}^{\max}(\ \oplus\!\!\!\!\!\Diamond\, \mathit{green} \mid \Diamond \mathit{goal}\ )$

maximum over all positive schedulers

Why should we be interested in ..?

Why should we be interested in ..?

- termination time of probabilistic programs

  conditional expected number of steps until termination, under the condition that the program terminates

- failure diagnosis and resilience analysis

  e.g. cost of repair protocols for a certain failure scenario

- various forms of multi-objective reasoning

  e.g., expected utility level, assuming a fixed energy budget

- conditional value-at-risk

  expected loss in worst case scenarios, under the assumption that these scenarios indeed occur

Why is it more difficult ...?

Why is it more difficult ...?

unconditional expected accumulated rewards

- optimal memoryless schedulers exists that maximize the expected reward from every state
- computable via linear programs with one variable per state

Why is it more difficult ...?

unconditional expected accumulated rewards

- optimal memoryless schedulers exists that maximize the expected reward from every state
- computable via linear programs with one variable per state

conditional expected accumulated rewards

- optimal schedulers require memory
- local reasoning not sufficient

Why is it more difficult ...?

unconditional expected accumulated rewards

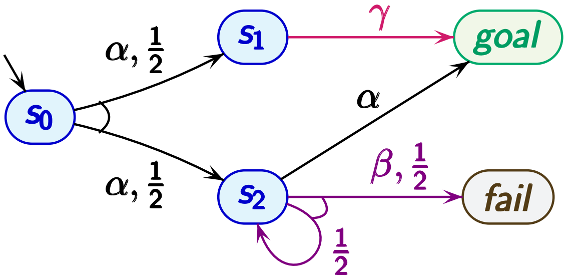
- optimal memoryless schedulers exists that maximize the expected reward from every state
- computable via linear programs with one variable per state

conditional expected accumulated rewards

- optimal schedulers require memory
- local reasoning not sufficient

... let's have a look at an example ...

Maximal conditional expected reward



$$rew(s_1, \gamma) = r$$

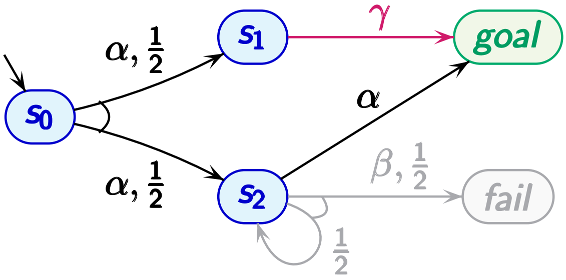
$$rew(s_2, \beta) = 1$$

$$rew(s_i, \alpha) = 0$$

maximal conditional expected reward:

$$\mathbb{E}^{\max}(\, \Diamond\!\!\!+ \mathit{goal} \mid \Diamond \mathit{goal}\,) = ???$$

Maximal conditional expected reward



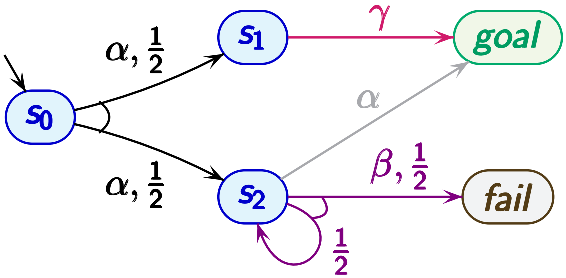
$rew(s_1, \gamma) = r$

$rew(s_2, \beta) = 1$

$rew(s_i, \alpha) = 0$

"choose always α in state s_2": $\dfrac{\frac{1}{2} \cdot r + \frac{1}{2} \cdot 0}{\frac{1}{2} + \frac{1}{2}} = \dfrac{r}{2}$

Maximal conditional expected reward



$$rew(s_1, \gamma) = r$$

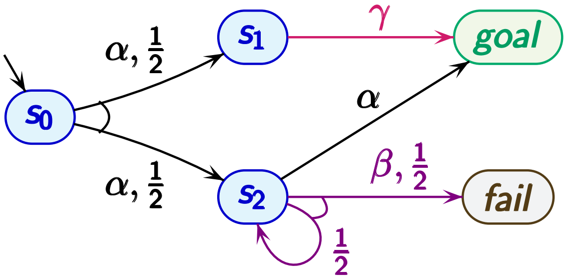
$$rew(s_2, \beta) = 1$$

$$rew(s_i, \alpha) = 0$$

"choose always α in state s_2": $\dfrac{\frac{1}{2}\cdot r + \frac{1}{2}\cdot 0}{\frac{1}{2} + \frac{1}{2}} = \dfrac{r}{2}$

"choose always β in state s_2": $\dfrac{\frac{1}{2}\cdot r + 0}{\frac{1}{2} + 0} = r$

Maximal conditional expected reward



$rew(s_1, \gamma) = r$

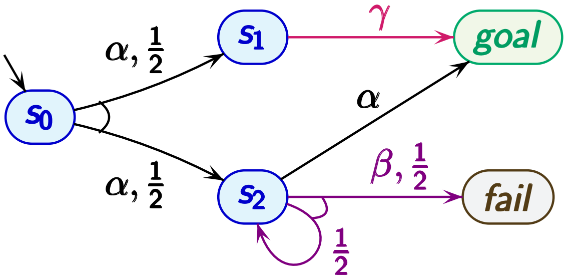
$rew(s_2, \beta) = 1$

$rew(s_i, \alpha) = 0$

"choose β exacly for the first n visits of s_2"

$$\frac{\frac{1}{2}\cdot r + \frac{1}{2}\cdot\frac{1}{2^n}\cdot n}{\frac{1}{2} + \frac{1}{2}\cdot\frac{1}{2^n}}$$

Maximal conditional expected reward



$rew(s_1, \gamma) = r$

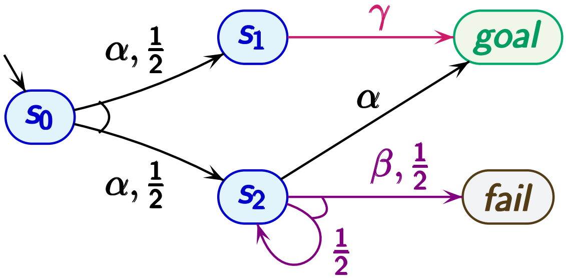
$rew(s_2, \beta) = 1$

$rew(s_i, \alpha) = 0$

"choose β exacly for the first n visits of s_2"

$$\frac{\frac{1}{2} \cdot r + \frac{1}{2} \cdot \frac{1}{2^n} \cdot n}{\frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2^n}} = r + \frac{n - r}{2^n + 1}$$

Maximal conditional expected reward

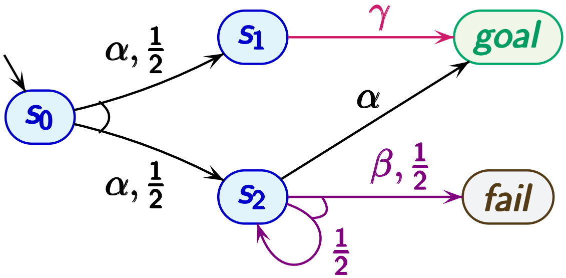


$$rew(s_1, \gamma) = r$$
$$rew(s_2, \beta) = 1$$
$$rew(s_i, \alpha) = 0$$

"choose β exacly for the first n visits of s_2"

$$\frac{\frac{1}{2}\cdot r + \frac{1}{2}\cdot\frac{1}{2^n}\cdot n}{\frac{1}{2} + \frac{1}{2}\cdot\frac{1}{2^n}} = r + \frac{n-r}{2^n+1} > r \quad \text{iff} \quad n > r$$

Maximal conditional expected reward



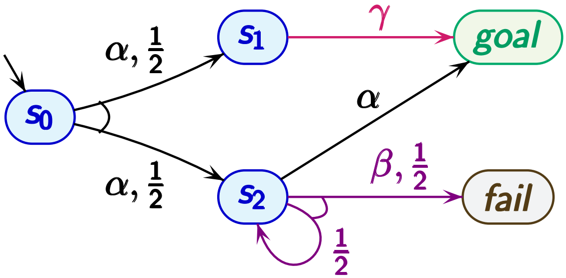
$$rew(s_1, \gamma) = r$$
$$rew(s_2, \beta) = 1$$
$$rew(s_i, \alpha) = 0$$

"choose β exacly for the first n visits of s_2"

$$\frac{\frac{1}{2}\cdot r + \frac{1}{2}\cdot\frac{1}{2^n}\cdot n}{\frac{1}{2} + \frac{1}{2}\cdot\frac{1}{2^n}} = r + \frac{n-r}{2^n+1} > r \quad \text{iff} \quad n > r$$

optimal value is achieved for $n = r+2$

Maximal conditional expected reward



$rew(s_1, \gamma) = r$

$rew(s_2, \beta) = 1$

$rew(s_i, \alpha) = 0$

maximal conditional reward until *goal*:

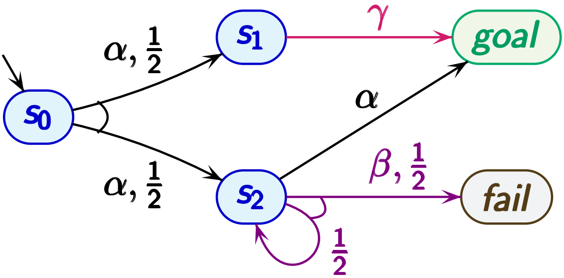
* memory required for optimal schedulers

  optimal scheduler needs counter for the number of visits in s_2

* local reasoning not sufficient

  ... as optimal decisions in s_2 depend on r

Maximal conditional expected reward



$rew(s_1, \gamma) = r$

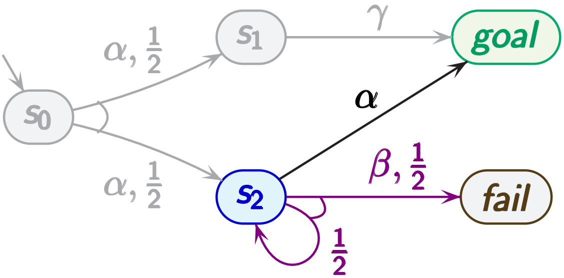
$rew(s_2, \beta) = 1$

$rew(s_i, \alpha) = 0$

maximal conditional reward until *goal*

... is finite for state s_0, namely $r + \frac{2}{2^{r+2}+1}$

Maximal conditional expected reward



$rew(s_1, \gamma) = r$

$rew(s_2, \beta) = 1$

$rew(s_i, \alpha) = 0$

maximal conditional reward until *goal*

... is finite for state s_0, namely $r + \frac{2}{2^{r+2}+1}$

... but infinite for s_2

$$\sup_{n \in \mathbb{N}} \frac{\frac{n}{2^n}}{\frac{1}{2^n}} = \infty$$

Problem statement

given: MDP $\mathcal{M} = (S, Act, P, rew, s_0)$ and F, $G \subseteq S$
such that $\Pr^{\max}_{s_0}(\ \Diamond F \mid \Diamond G\ ) = 1$

task: ...

$\Pr^{\max}_{s_0}(\Diamond F \mid \Diamond G\ ) = 1$ iff there is scheduler σ s.t.

1. $\Pr^{\sigma}_{s_0}(\Diamond G\ ) > 0$ and
2. $\Pr^{\sigma}_{s_0}(\ \Diamond F \mid \Diamond G\ ) = 1$

Problem statement

given: MDP $\mathcal{M} = (S, Act, P, rew, s_0)$ and F, $G \subseteq S$
such that $\Pr^{\max}_{s_0}(\,\Diamond F \mid \Diamond G\,) = 1$

task: compute $\mathbb{E}^{\max}_{s_0}(\,\Diamond F \mid \Diamond G\,)$

maximal conditional accumulated reward to reach F

under all schedulers σ where $\Pr^{\sigma}_{s_0}(\,\Diamond G\,) > 0$
and $\Pr^{\sigma}_{s_0}(\,\Diamond F \mid \Diamond G\,) = 1$

Problem statement

given: MDP $\mathcal{M} = (S, Act, P, rew, s_0)$ and $F, G \subseteq S$
such that $\mathrm{Pr}^{\max}_{s_0}(\,\Diamond F \mid \Diamond G\,) = 1$

task: compute $\mathbb{E}^{\max}_{s_0}(\,\oplus F \mid \Diamond G\,)$

---

after some preprocessing and cleaning-up:

1. all states are reachable from s_0
2. $F = G = \{goal\}$ for a trap state $goal$
3. there is another trap state $fail$ with
$\mathrm{Pr}^{\min}_{s}(\,\Diamond(goal \vee fail)\,) = 1$ for all states s

Shortform notation used in the sequel

Given a scheduler σ with $\mathrm{Pr}^{\sigma}_{s_0}(\Diamond goal) > 0$, let:

$$\mathbb{CE}^{\sigma} = \mathbb{E}^{\sigma}_{s_0}(\,\oplus goal \mid \Diamond goal\,)$$

Maximal conditional expectation:

$$\mathbb{CE}^{\max} = \sup_{\sigma} \mathbb{CE}^{\sigma}$$

ranging over all schedulers σ
with $\mathrm{Pr}^{\sigma}_{s_0}(\Diamond goal) > 0$

Shortform notation used in the sequel

Given a scheduler σ with $\Pr^{\sigma}_{s_0}(\Diamond goal) > 0$, let:

$$\mathbb{CE}^{\sigma} = \mathbb{E}^{\sigma}_{s_0}(\, \oplus goal \mid \Diamond goal \,)$$

Maximal conditional expectation:

$$\mathbb{CE}^{\max} = \sup_{\sigma} \mathbb{CE}^{\sigma}$$

$\uparrow$

supremum over all
deterministic reward-based schedulers
$\sigma : S \times \mathbb{N} \to Act$

Checking finiteness

Given a scheduler σ with $\Pr^{\sigma}_{s_0}(\Diamond goal) > 0$, let:

$$\mathbb{CE}^{\sigma} = \mathbb{E}^{\sigma}_{s_0}(\oplus goal \mid \Diamond goal)$$

Maximal conditional expectation:

$$\mathbb{CE}^{\max} = \sup_{\sigma} \mathbb{CE}^{\sigma}$$

Checking finiteness in polynomial time:

$$\mathbb{CE}^{\max} < \infty \quad \text{iff} \quad \left\{ \begin{array}{l} \text{there is no scheduler } \sigma \text{ s.t.} \\ \Pr^{\sigma}_{s_0}(\Diamond goal) = 0 \text{ and there is a} \\ \text{reachable positive } \sigma\text{-cycle} \end{array} \right.$$

If $\mathbb{CE}^{\mathsf{max}} < \infty$ then ...

If $\mathbb{CE}^{\mathsf{max}} < \infty$ then ...

- pseudo-polynomial algorithm to compute an upper bound $\mathbb{CE}^{\mathbf{ub}}$ for $\mathbb{CE}^{\mathsf{max}}$

pseudo-polynomial: time complexity is polynomial in the

* size of the graph structure and
* length of an unary encoding of the probability/reward values

If $\mathbb{CE}^{\mathsf{max}} < \infty$ then ...

- pseudo-polynomial algorithm to compute an upper bound $\mathbb{CE}^{\mathrm{ub}}$ for $\mathbb{CE}^{\mathsf{max}}$
- threshold problem "is $\mathbb{CE}^{\mathsf{max}} \trianglerighteq \vartheta$?" is PSPACE-hard, and PSPACE-complete for acyclic MDPs

... same for upper bounds by duality ...

threshold problem:

given: MDP $\mathcal{M}$, $\vartheta \in \mathbb{Q}$ and $\trianglerighteq \in \{>, \geqslant\}$

task: check whether $\mathbb{CE}^{\mathsf{max}} \trianglerighteq \vartheta$

If $\mathbb{CE}^{\mathsf{max}} < \infty$ then ...

- pseudo-polynomial algorithm to compute an upper bound $\mathbb{CE}^{\mathrm{ub}}$ for $\mathbb{CE}^{\mathsf{max}}$
- threshold problem "is $\mathbb{CE}^{\mathsf{max}} \trianglerighteq \vartheta$?" is PSPACE-hard, and PSPACE-complete for acyclic MDPs
- there exists a saturation point $\wp$ such that optimal schedulers behave memoryless from reward $\wp$ on

  ... and maximize the probability to reach the goal state

If $\mathbb{CE}^{\mathsf{max}} < \infty$ then ...

- pseudo-polynomial algorithm to compute an upper bound $\mathbb{CE}^{\mathrm{ub}}$ for $\mathbb{CE}^{\mathsf{max}}$
- threshold problem "is $\mathbb{CE}^{\mathsf{max}} \trianglerighteq \vartheta$?" is PSPACE-hard, and PSPACE-complete for acyclic MDPs
- there exists a saturation point $\wp$ such that optimal schedulers behave memoryless from reward $\wp$ on
- pseudo-polynomial threshold algorithm

If $\mathbb{CE}^{\max} < \infty$ then ...

- pseudo-polynomial algorithm to compute an upper bound $\mathbb{CE}^{\mathrm{ub}}$ for $\mathbb{CE}^{\max}$
- threshold problem "is $\mathbb{CE}^{\max} \trianglerighteq \vartheta$?" is PSPACE-hard, and PSPACE-complete for acyclic MDPs
- there exists a saturation point $\wp$ such that optimal schedulers behave memoryless from reward $\wp$ on
- pseudo-polynomial threshold algorithm: generates a scheduler σ s.t. $\mathbb{CE}^{\sigma} > \vartheta$ or $\mathbb{CE}^{\max} = \mathbb{CE}^{\sigma} = \vartheta$ (if existent)

If $\mathbb{CE}^{\max} < \infty$ then ...

- pseudo-polynomial algorithm to compute an upper bound $\mathbb{CE}^{\mathrm{ub}}$ for $\mathbb{CE}^{\max}$
- threshold problem "is $\mathbb{CE}^{\max} \trianglerighteq \vartheta$?" is PSPACE-hard, and PSPACE-complete for acyclic MDPs
- there exists a saturation point $\wp$ such that optimal schedulers behave memoryless from reward $\wp$ on
- pseudo-polynomial threshold algorithm: generates a scheduler σ s.t. $\mathbb{CE}^{\sigma} > \vartheta$ or $\mathbb{CE}^{\max} = \mathbb{CE}^{\sigma} = \vartheta$
- exponential-time algorithm to compute $\mathbb{CE}^{\max}$
  interleaves scheduler-improvement steps with threshold algorithm

Computing an upper bound

Computing an upper bound

unconditional total expected reward in a new MDP

Computing an upper bound

unconditional total expected reward in a new MDP $\mathcal{N}$ that simulates $\mathcal{M}$ under the condition $\Diamond goal$

Computing an upper bound

unconditional total expected reward in a new MDP $\mathcal{N}$ that simulates $\mathcal{M}$ under the condition $\Diamond goal$

first mode:

- augments states with the reward accumulated so far up to $R^{max} = \sum_{s} \max_{\alpha} rew(s, \alpha)$
- reward 0 for all state-actions in the first mode
- mode switch from (s, r) via action α with reward r' if $r' \stackrel{\text{def}}{=} r + rew(s, \alpha) > R^{max}$

second mode: simulation of $\mathcal{M}$ (without reward-annotations)

Computing an upper bound

unconditional total expected reward in a new MDP $\mathcal{N}$ that simulates $\mathcal{M}$ under the condition $\Diamond goal$

first mode:

- augments states with the reward accumulated so far up to $R^{max} = \sum_{s} \max_{\alpha} rew(s, \alpha)$
- reward 0 for all state-actions in the first mode
- mode switch from (s, r) via action α with reward r' if $r' \stackrel{\text{def}}{=} r + rew(s, \alpha) > R^{\max}$

second mode: simulation of $\mathcal{M}$ (without reward-annotations)

reset transitions:

from all fail states to $\mathcal{N}$'s initial state $(s_0, 0)$

Sketch of the threshold algorithm

compute the saturation point $\wp$ and optimal decisions for state-reward pairs (s, r) with $r \geqslant \wp$

```
FOR r = ℘−1, ℘−2, ..., 0 DO
```

compute most feasible actions for the state-reward pairs (s, r) using

- decisions for (s', r') with $r' > r$
- a linear program to treat zero-reward actions

```
OD
```

check if $\mathbb{CE}^{\sigma} \trianglerighteq \vartheta$ for the generated scheduler σ

Sketch of the threshold algorithm

compute the saturation point $\wp$ and optimal decisions for state-reward pairs (s, r) with $r \geqslant \wp$

FOR $r = \wp - 1, \wp - 2, \ldots, 0$ DO

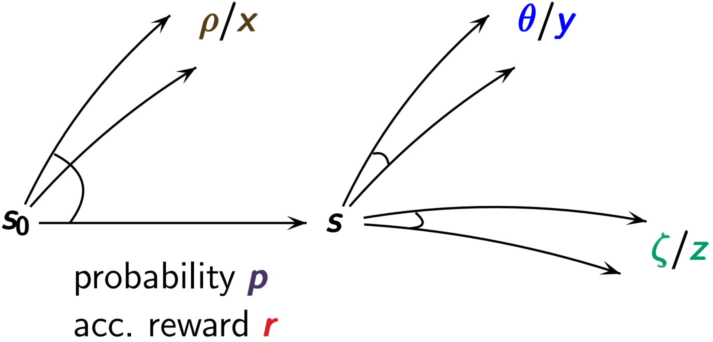
> compute most feasible actions for the state-reward pairs (s, r) using
>
> - decisions for (s', r') with $r' > r$
> - a linear program to treat zero-reward actions

OD

check if $\mathbb{CE}^{\sigma} \trianglerighteq \vartheta$ for the generated scheduler σ

Let $\rho, \theta, \zeta, r \in \mathbb{R}$, $p, x, y, z \in [0, 1]$ such that $y > z$ and $x + py > 0$, $x + pz > 0$.

$$\mathbb{CE}^{\sigma} = \frac{\rho + p(ry + \theta)}{x + py} \qquad \mathbb{CE}^{\tau} = \frac{\rho + p(rz + \zeta)}{x + pz}$$



Let $\rho, \theta, \zeta, r \in \mathbb{R}$, $p, x, y, z \in [0,1]$ such that $y > z$ and $x + py > 0$, $x + pz > 0$.

$$\mathbb{CE}^{\sigma} = \frac{\rho + p(ry + \theta)}{x + py} \qquad \mathbb{CE}^{\tau} = \frac{\rho + p(rz + \zeta)}{x + pz}$$

$$\mathbb{CE}^{\sigma} > \mathbb{CE}^{\tau} \quad \text{iff} \quad r + \frac{\theta - \zeta}{y - z} > \max\left\{\mathbb{CE}^{\sigma}, \mathbb{CE}^{\tau}\right\}$$

Let $\rho, \theta, \zeta, r \in \mathbb{R}$, $p, x, y, z \in [0,1]$ such that $y > z$ and $x + py > 0$, $x + pz > 0$.

$$\mathbb{CE}^{\sigma} = \frac{\rho + p(ry + \theta)}{x + py} \qquad \mathbb{CE}^{\tau} = \frac{\rho + p(rz + \zeta)}{x + pz}$$

$$\mathbb{CE}^{\sigma} > \mathbb{CE}^{\tau} \quad \text{iff} \quad \boxed{r + \frac{\theta - \zeta}{y - z}} > \max\left\{\mathbb{CE}^{\sigma}, \mathbb{CE}^{\tau}\right\}$$

does not depend on ρ, x, p

Let $\rho, \theta, \zeta, r \in \mathbb{R}$, $p, x, y, z \in [0,1]$ such that $y > z$ and $x + py > 0$, $x + pz > 0$.

$$\mathbb{CE}^{\sigma} = \frac{\rho + p(ry + \theta)}{x + py} \qquad \mathbb{CE}^{\tau} = \frac{\rho + p(rz + \zeta)}{x + pz}$$

$$\mathbb{CE}^{\sigma} > \mathbb{CE}^{\tau} \quad \text{iff} \quad r + \frac{\theta - \zeta}{y - z} > \max\left\{\mathbb{CE}^{\sigma}, \mathbb{CE}^{\tau}\right\}$$

threshold algorithm:

$$r + \frac{\theta - \zeta}{y - z} \geqslant \vartheta \quad \text{iff} \quad \theta - (\vartheta - r)y \geqslant \zeta - (\vartheta - r)z$$

Let $\rho, \theta, \zeta, r \in \mathbb{R}$, $p, x, y, z \in [0,1]$ such that $y > z$ and $x + py > 0$, $x + pz > 0$.

$$\mathbb{CE}^{\sigma} = \frac{\rho + p(ry + \theta)}{x + py} \qquad \mathbb{CE}^{\tau} = \frac{\rho + p(rz + \zeta)}{x + pz}$$

$$\mathbb{CE}^{\sigma} > \mathbb{CE}^{\tau} \quad \text{iff} \quad r + \frac{\theta - \zeta}{y - z} > \max\left\{\mathbb{CE}^{\sigma}, \mathbb{CE}^{\tau}\right\}$$

threshold algorithm:

$$r + \frac{\theta - \zeta}{y - z} \geqslant \vartheta \quad \text{iff} \quad \theta - (\vartheta - r)y \ \geqslant \ \zeta - (\vartheta - r)z$$

... use LP-techniques to maximize $\theta - (\vartheta - r)y$

Let $\rho, \theta, \zeta, r \in \mathbb{R}$, $p, x, y, z \in [0,1]$ such that $y > z$ and $x + py > 0$, $x + pz > 0$.

$$\mathbb{CE}^{\sigma} = \frac{\rho + p(ry + \theta)}{x + py} \qquad \mathbb{CE}^{\tau} = \frac{\rho + p(rz + \zeta)}{x + pz}$$

$$\mathbb{CE}^{\sigma} > \mathbb{CE}^{\tau} \quad \text{iff} \quad r + \frac{\theta - \zeta}{y - z} > \max\left\{\mathbb{CE}^{\sigma}, \mathbb{CE}^{\tau}\right\}$$

saturation point: smallest value r such that

$$r + \frac{\theta - \zeta}{y - z} \geqslant \mathbb{CE}^{\max} \qquad \text{for all } \tau$$

where σ maximizes the probabilities for reaching the goal

Let $\rho, \theta, \zeta, r \in \mathbb{R}$, $p, x, y, z \in [0,1]$ such that $y > z$ and $x + py > 0$, $x + pz > 0$.

$$\mathbb{CE}^{\sigma} = \frac{\rho + p(ry + \theta)}{x + py} \qquad \mathbb{CE}^{\tau} = \frac{\rho + p(rz + \zeta)}{x + pz}$$

$$\mathbb{CE}^{\sigma} > \mathbb{CE}^{\tau} \quad \text{iff} \quad r + \frac{\theta - \zeta}{y - z} > \max\left\{\mathbb{CE}^{\sigma}, \mathbb{CE}^{\tau}\right\}$$

saturation point: smallest value r such that

$$r + \frac{\theta - \zeta}{y - z} \geqslant \mathbb{CE}^{\text{ub}} \qquad \text{for all } \tau$$

where σ maximizes the probabilities for reaching the goal

Let $\rho, \theta, \zeta, r \in \mathbb{R}$, $p, x, y, z \in [0,1]$ such that $y > z$ and $x + py > 0$, $x + pz > 0$.

$$\mathbb{CE}^{\sigma} = \frac{\rho + p(ry + \theta)}{x + py} \qquad \mathbb{CE}^{\tau} = \frac{\rho + p(rz + \zeta)}{x + pz}$$

$$\mathbb{CE}^{\sigma} > \mathbb{CE}^{\tau} \quad \text{iff} \quad r + \frac{\theta - \zeta}{y - z} > \max\left\{\mathbb{CE}^{\sigma}, \mathbb{CE}^{\tau}\right\}$$

saturation point: smallest value r such that

$$r + \frac{\theta - \zeta}{y - z} \geqslant \mathbb{CE}^{\mathrm{ub}} \qquad \text{for all } \tau$$

... it suffices to consider "one-step variants" τ of σ

Computing the maximal conditional expectation

using a scheduler-improvement approach with iterative calls of the threshold algorithm

If $\mathbb{CE}^{\max} \geqslant \vartheta$ then the threshold algorithm generates a scheduler σ s.t. $\mathbb{CE}^{\sigma} > \vartheta$ or $\mathbb{CE}^{\max} = \mathbb{CE}^{\sigma} = \vartheta$.

Computing the maximal conditional expectation

using a scheduler-improvement approach with iterative calls of the threshold algorithm

let σ be an arbitrary scheduler;

REPEAT

$\vartheta := \mathbb{CE}^{\sigma}$;

$\sigma :=$ outcome of the algorithm for threshold ϑ

UNTIL $\mathbb{CE}^{\sigma} = \vartheta$

computation of an optimal scheduler

If $\mathbb{CE}^{\max} \geqslant \vartheta$ then the threshold algorithm generates a scheduler σ s.t. $\mathbb{CE}^{\sigma} > \vartheta$ or $\mathbb{CE}^{\max} = \mathbb{CE}^{\sigma} = \vartheta$.

Computing the maximal conditional expectation

using a scheduler-improvement approach with iterative calls of the threshold algorithm

let σ be ...

REPEAT

$\vartheta := \mathbb{CE}^{\sigma}$;

$\sigma :=$ outcome of the algorithm for threshold ϑ

UNTIL $\mathbb{CE}^{\sigma} = \vartheta$

time complexity: double exponential

If $\mathbb{CE}^{\max} \geqslant \vartheta$ then the threshold algorithm generates a scheduler σ s.t. $\mathbb{CE}^{\sigma} > \vartheta$ or $\mathbb{CE}^{\max} = \mathbb{CE}^{\sigma} = \vartheta$.

Computing the maximal conditional expectation

using a scheduler-improvement approach with iterative calls of the threshold algorithm

let σ be ...

REPEAT

$\vartheta := \mathbb{CE}^{\sigma}$;

$\sigma :=$ outcome of the algorithm for threshold ϑ

UNTIL $\mathbb{CE}^{\sigma} = \vartheta$

time complexity: double exponential

in the worst-case: $|\,\mathrm{MD}\,|^{\wp}$ iterations where the saturation point $\wp$ can be exponential in *size*$(\mathcal{M})$

Computing the maximal conditional expectation

exponential-time algorithm for computing $\mathbb{CE}^{\max}$

* freezes level-wise optimal decisions
* uses threshold algorithm for scheduler-improvement steps
* maintains an interval of feasible threshold candidates

Computing the maximal conditional expectation

exponential-time algorithm for computing $\mathbb{CE}^{\max}$

* freezes level-wise optimal decisions
* uses threshold algorithm for scheduler-improvement steps
* maintains an interval of feasible threshold candidates

$$\mathbb{CE}^{\sigma} = \frac{\rho + p(ry + \theta)}{x + py} \qquad \mathbb{CE}^{\tau} = \frac{\rho + p(rz + \zeta)}{x + pz}$$

$$\mathbb{CE}^{\sigma} > \mathbb{CE}^{\tau} \quad \text{iff} \quad r + \frac{\theta - \zeta}{y - z} > \max\left\{\mathbb{CE}^{\sigma}, \mathbb{CE}^{\tau}\right\}$$

If this holds for all τ then σ is optimal for level r.

Computing the maximal conditional expectation

exponential-time algorithm for computing $\mathbb{CE}^{\max}$

* freezes level-wise optimal decisions
* uses threshold algorithm for scheduler-improvement steps
* maintains an interval of feasible threshold candidates

$$\mathbb{CE}^{\sigma} = \frac{\rho + p(ry + \theta)}{x + py} \qquad \mathbb{CE}^{\tau} = \frac{\rho + p(rz + \zeta)}{x + pz}$$

$$\mathbb{CE}^{\sigma} > \mathbb{CE}^{\tau} \quad \text{iff} \quad \boxed{r + \frac{\theta - \zeta}{y - z}} > \max\left\{\mathbb{CE}^{\sigma}, \mathbb{CE}^{\tau}\right\}$$

use these values as threshold values

Computing the maximal conditional expectation

exponential-time algorithm for computing $\mathbb{CE}^{\max}$

- freezes level-wise optimal decisions
- uses threshold algorithm for scheduler-improvement steps
- maintains an interval of feasible threshold candidates

$$\mathbb{CE}^{\sigma} = \frac{\rho + p(ry + \theta)}{x + py} \qquad \mathbb{CE}^{\tau} = \frac{\rho + p(rz + \zeta)}{x + pz}$$

$$\mathbb{CE}^{\sigma} > \mathbb{CE}^{\tau} \quad \text{iff} \quad r + \frac{\theta - \zeta}{y - z} > \max\left\{\mathbb{CE}^{\sigma}, \mathbb{CE}^{\tau}\right\}$$

in total: $\mathcal{O}(\wp \cdot |\mathrm{MD}|)$ scheduler-improvement steps

Outline

- weighted Markov decision processes
- mean-payoff and long-run ratios
- expected accumulated weights
- conditional expected accumulated rewards
- weight-bounded reachability and quantiles
- LTL with weight assertions
- conclusions

Weight-bounded reachability probabilities

Weight-bounded reachability probabilities

given: weighted MDP $\mathcal{M}$, set $G \subseteq S$ and weight bound $r \in \mathbb{Z}$

task: compute $\Pr_s^{\max}(\Diamond^{\leqslant r} G)$

Weight-bounded reachability probabilities

given: weighted MDP $\mathcal{M}$, set $G \subseteq S$ and
weight bound $r \in \mathbb{Z}$

task: compute $\Pr_s^{\max}(\Diamond^{\leqslant r} G)$

If π is an infinite path then:

$$\pi \models \Diamond^{\leqslant r} G \quad \text{iff} \quad \begin{cases} \text{there exists a finite prefix } \pi' \text{ of } \pi \\ \text{s.t. } \underbrace{wgt(\pi')}_{\text{accumulated weight}} \leqslant r \text{ and } last(\pi') \in G \end{cases}$$

Weight-bounded reachability probabilities

given: weighted MDP $\mathcal{M}$, set $G \subseteq S$ and
weight bound $r \in \mathbb{Z}$

task: compute $\Pr^{\max}_s(\Diamond^{\leqslant r} G)$

for non-negative weights (rewards):

- computable, but computationally expensive
- algorithms rely on the monotonicty of accumulated rewards along the prefixes of paths

for arbitrary weights (negative and positive values)

- much more difficult, even for Markov chains

Reward-bounded reachability

for models with a single $\underbrace{\text{non-negative weight function}}_{\text{“reward function”}}$

Reward-bounded reachability

for models with a single non-negative weight function

Markov chains: "check whether $\Pr_s(\Diamond^{\leqslant r} G) > \frac{1}{2}$"

- NP-hard [Laroussinie/Sproston'05]
- PosSLP-hard [Haase/Kiefer'15]
- solvable in pseudo-polynomial time

PosSLP:

given: an arithmetic circuit with operators $(+, -, *)$
and distinguished output gate

question: can the circuit output a positive integer ?

Reward-bounded reachability

for models with a single non-negative weight function

Markov chains: "check whether $\Pr_s(\Diamond^{\leqslant r} G) > \frac{1}{2}$"

- NP-hard [Laroussinie/Sproston'05]
- PosSLP-hard [Haase/Kiefer'15]
- solvable in pseudo-polynomial time

MDPs: "check whether $\Pr_s^{\max}(\Diamond^{\leqslant r} G) > \frac{1}{2}$"

Reward-bounded reachability

for models with a single non-negative weight function

Markov chains: "check whether $\mathrm{Pr}_s(\Diamond^{\leqslant r} G) > \frac{1}{2}$"

- NP-hard [Laroussinie/Sproston'05]
- PosSLP-hard [Haase/Kiefer'15]
- solvable in pseudo-polynomial time

MDPs: "check whether $\mathrm{Pr}_s^{\max}(\Diamond^{\leqslant r} G) > \frac{1}{2}$"

- PSPACE-complete for acyclic MDPs [Haase/Kiefer'15]

Reward-bounded reachability

for models with a single non-negative weight function

Markov chains: "check whether $\mathbf{Pr}_s(\Diamond^{\leqslant r} G) > \frac{1}{2}$"

- NP-hard [Laroussinie/Sproston'05]
- PosSLP-hard [Haase/Kiefer'15]
- solvable in pseudo-polynomial time

MDPs: "check whether $\mathbf{Pr}_s^{\max}(\Diamond^{\leqslant r} G) > \frac{1}{2}$"

- PSPACE-complete for acyclic MDPs [Haase/Kiefer'15]
- solvable in pseudo-polynomial time
  using an iterative LP-based approach
  [Ummels/Baier'13], [Baier/Daum/Dubslaff/Klein/Klüppelholz'14]

Reward-bounded reachability

for models with a single non-negative weight function

Markov chains: "check whether $\Pr_s(\Diamond^{\leqslant r} G) > \frac{1}{2}$"

- NP-hard [Laroussinie/Sproston'05]
- PosSLP-hard [Haase/Kiefer'15]
- solvable in pseudo-polynomial time

MDPs: "check whether $\Pr^{\max}_s(\Diamond^{\leqslant r} G) > \frac{1}{2}$"

- PSPACE-complete for acyclic MDPs [Haase/Kiefer'15]
- solvable in pseudo-polynomial time
- even $\Pr^{\max}_s(\Diamond^{\leqslant r} G)$ and quantiles are computable in pseudo-polynomial time

Computing quantitative quantiles

$$r_{\min} \ = \ \min \{ \, r \in \mathbb{N} \, : \, \Pr^{\max}_s(\Diamond^{\leqslant r} G) > q \, \}$$

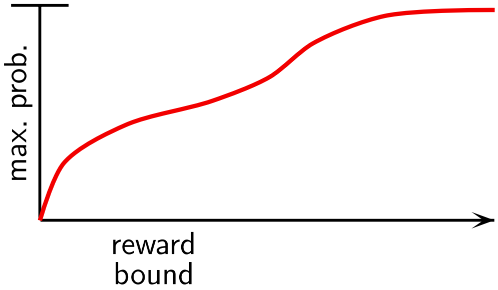
existential quantile for

- upper reward-bounded reachability
- lower probability bound

$$\Pr^{\max}_s(\varphi) > q \quad \text{iff} \quad \left\{ \begin{array}{l} \text{there exists a scheduler } \sigma \\ \text{with } \Pr^{\sigma}_s(\varphi) > q \end{array} \right.$$

Computing quantitative quantiles

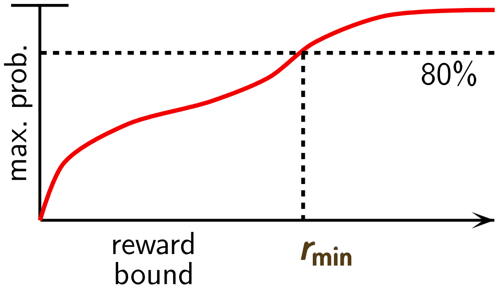
$$r_{\min} \;=\; \min\left\{\, r \in \mathbb{N} \,:\, \Pr_s^{\max}(\Diamond^{\leqslant r} G) > q \,\right\}$$



$\Pr_s^{\max}(\varphi) > q$ iff $\begin{cases} \text{there exists a scheduler } \sigma \\ \text{with } \Pr_s^{\sigma}(\varphi) > q \end{cases}$

Computing quantitative quantiles

$$r_{\min} \;=\; \min\left\{\, r \in \mathbb{N} \;:\; \Pr^{\max}_{s}(\Diamond^{\leqslant r} G) > q \,\right\}$$



$q = 0.8$

$\Pr^{\max}_{s}(\varphi) > q$ iff $\begin{cases} \text{there exists a scheduler } \sigma \\ \text{with } \Pr^{\sigma}_{s}(\varphi) > q \end{cases}$

Computing quantitative quantiles

$$r_{\min} \;=\; \min\left\{\, r \in \mathbb{N} \;:\; \Pr^{\max}_{s_0}(\Diamond^{\leqslant r} G) > q \,\right\}$$

1. compute $p \;=\; \Pr^{\max}_{s_0}(\,\Diamond G\,)$
2. return $r_{\min} = \infty$ if $p \leqslant q$
3. ...

[Ummels/Baier'13] [Baier/Daum/Dubslaff/Klein/Klüppelholz'13]

Computing quantitative quantiles

$$r_{\min} \;=\; \min\left\{\, r \in \mathbb{N} \;:\; \underbrace{\Pr^{\max}_{s_0}(\Diamond^{\leqslant r} G)}_{p_{s_0,r}} > q \,\right\}$$

1. compute $p \;=\; \Pr^{\max}_{s_0}(\,\Diamond G\,)$
2. return $r_{\min} = \infty$ if $p \leqslant q$
3. for $r = 0, 1, 2, \ldots$ compute the values $p_{s,r}$ for all states $s \in S$ and return the smallest value r such that $p_{s_0,r} > q$

[Ummels/Baier'13] [Baier/Daum/Dubslaff/Klein/Klüppelholz'13]

Computing quantitative quantiles

$$r_{\min} = \min \left\{ r \in \mathbb{N} : \underbrace{\Pr^{\max}_{s_0}(\Diamond^{\leqslant r} G)}_{p_{s_0,r}} > q \right\}$$

1. compute $p = \Pr^{\max}_{s_0}(\Diamond G)$
2. return $r_{\min} = \infty$ if $p \leqslant q$
3. for $r = 0, 1, 2, \ldots$ compute the values $p_{s,r}$ for all states $s \in S$ and return the smallest value r such that $p_{s_0,r} > q$

exponential bound on the number of required iterations

(in practice much faster)

Computing quantitative quantiles

$$r_{\min} \;=\; \min\left\{\, r \in \mathbb{N} \;:\; \underbrace{\Pr^{\max}_{s_0}(\Diamond^{\leqslant r} G)}_{p_{s_0,r}} > q \,\right\}$$

1. compute $p \;=\; \Pr^{\max}_{s_0}(\,\Diamond G\,)$
2. return $r_{\min} = \infty$ if $p \leqslant q$
3. for $r = 0, 1, 2, \ldots$ compute the values $p_{s,r}$ for all states $s \in S$ and return the smallest value r such that $p_{s_0,r} > q$

computation of $p_{s,r}$ by an iterative linear-programming approach with back propagation

linear program for the values $p_{s,r} = \Pr^{\max}_s(\Diamond^{\leqslant r} G)$

$$x_{s,r} = 0 \quad \text{if } s \not\models \exists\Diamond G$$

$$x_{s,r} = 1 \quad \text{if } s \in G$$

If $s \notin G$, $s \models \exists\Diamond G$ and $\alpha \in Act(s)$ then:

$$x_{s,r} \geqslant \sum_{t\in S} P(s,\alpha,t) \cdot x_{t,r} \quad \text{if } rew(s,\alpha) = 0$$

$$x_{s,r} \geqslant \sum_{t\in S} P(s,\alpha,t) \cdot x_{t,r-\ell} \quad \text{if } \ell = rew(s,\alpha) > 0$$

solution: $x_{s,r} = p_{s,r} = \Pr^{\max}_s(\Diamond^{\leqslant r} G)$

linear program for the values $p_{s,r} = \Pr^{\max}_s(\Diamond^{\leqslant r} G)$

$$x_{s,r} = 0 \quad \text{if } s \not\models \exists\Diamond G$$

$$x_{s,r} = 1 \quad \text{if } s \in G$$

$$\boxed{\text{minimize } \sum_s x_{s,r}}$$

If $s \notin G$, $s \models \exists\Diamond G$ and $\alpha \in Act(s)$ then:

$$x_{s,r} \geqslant \sum_{t \in S} P(s,\alpha,t) \cdot x_{t,r} \quad \text{if } rew(s,\alpha) = 0$$

$$x_{s,r} \geqslant \sum_{t \in S} P(s,\alpha,t) \cdot x_{t,r-\ell} \quad \text{if } \ell = rew(s,\alpha) > 0$$

unique solution: $x_{s,r} = p_{s,r} = \Pr^{\max}_s(\Diamond^{\leqslant r} G)$

linear program for the values $p_{s,r} \; = \; \Pr^{\max}_s(\Diamond^{\leqslant r} G)$

$$x_{s,r} \;=\; 0 \quad \text{if } s \not\models \exists\Diamond G$$

$$x_{s,r} \;=\; 1 \quad \text{if } s \in G$$

minimize $\sum_s x_{s,r}$

If $s \notin G$, $s \models \exists\Diamond G$ and $\alpha \in Act(s)$ then:

$$x_{s,r} \;\geqslant\; \sum_{t \in S} P(s,\alpha,t) \cdot x_{t,r} \qquad \text{if } rew(s,\alpha) = 0$$

$$x_{s,r} \;\geqslant\; \sum_{t \in S} P(s,\alpha,t) \cdot x_{t,r-\ell} \qquad \text{if } \ell = rew(s,\alpha) > 0$$

use the solutions $p_{t,i} = \Pr^{\max}_s(\Diamond^{\leqslant i} G)$ for $i < r$

computed in previous iterations

linear program for the values $p_{s,r} \;=\; \Pr^{\max}_s(\Diamond^{\leqslant r} G)$

$$x_{s,r} \;=\; 0 \qquad \text{if } s \not\models \exists\Diamond G$$

$$x_{s,r} \;=\; 1 \qquad \text{if } s \in G$$

minimize $\sum_s x_{s,r}$

If $s \notin G$, $s \models \exists\Diamond G$ and $\alpha \in Act(s)$ then:

$$x_{s,r} \;\geqslant\; \sum_{t\in S} P(s,\alpha,t)\cdot x_{t,r} \qquad \text{if } rew(s,\alpha) = 0$$

$$x_{s,r} \;\geqslant\; const$$

use the solutions $p_{t,i} = \Pr^{\max}_s(\Diamond^{\leqslant i} G)$ for $i < r$

computed in previous iterations

linear program for the values $p_{s,r} = \Pr^{\max}_s(\Diamond^{\leqslant r} G)$

$$x_{s,r} = 0 \quad \text{if } s \not\models \exists\Diamond G$$

$$x_{s,r} = 1 \quad \text{if } s \in G$$

minimize $\sum_s x_{s,r}$

If $s \notin G$, $s \models \exists\Diamond G$ and $\alpha \in Act(s)$ then:

$$x_{s,r} \geqslant \sum_{t \in S} P(s,\alpha,t) \cdot x_{t,r} \quad \text{if } rew(s,\alpha) = 0$$

$$x_{s,r} \geqslant const$$

linear in the size of the MDP

linear program to be solved in the r-th iteration

linear program for the values $p_{s,r} = \Pr^{\max}_s(\Diamond^{\geqslant r} G)$

lower
reward
bound

If π is an infinite path then:

$$\pi \models \Diamond^{\geqslant r} G \quad \text{iff} \quad \left\{ \begin{array}{l} \text{there exists a finite prefix } \pi' \text{ of } \pi \\ \text{s.t. } \underbrace{rew(\pi')}_{\substack{\text{accumulated}\\ \text{reward}}} \geqslant r \text{ and } last(\pi') \in G \end{array} \right.$$

linear program for the values $p_{s,r} = \Pr^{\max}_{s}(\lozenge^{\geqslant r} G)$

$$
\left.
\begin{array}{lcl}
x_{s,0} & = & \Pr^{\max}_{s}(\lozenge G) \\
x_{s,i} & = & 0 \quad \text{if } s \not\models \exists\lozenge G \\
x_{s,i} & \geqslant & 0 \quad \text{if } s \models \exists\lozenge G
\end{array}
\right\} \text{ for } 0 \leqslant i \leqslant r
$$

linear program for the values $p_{s,r} = \Pr^{\max}_s(\Diamond^{\geqslant r} G)$

$$x_{s,0} = \Pr^{\max}_s(\Diamond G)$$

$$\left.\begin{array}{lll} x_{s,i} & = & 0 \quad \text{if } s \not\models \exists\Diamond G \\ x_{s,i} & \geqslant & 0 \quad \text{if } s \models \exists\Diamond G \end{array}\right\} \text{ for } 0 \leqslant i \leqslant r$$

If $s \models \exists\Diamond G$ and $\alpha \in Act(s)$ then:

$$x_{s,i} \geqslant \sum_{t \in S} P(s,\alpha,t) \cdot x_{t,i-rew(s,\alpha)}$$

if $rew(s,\alpha) \leqslant i \leqslant r$

linear program for the values $p_{s,r} = \Pr^{\max}_s(\Diamond^{\geqslant r} G)$

$$\left.\begin{array}{lll} x_{s,0} & = & \Pr^{\max}_s(\Diamond G) \\ x_{s,i} & = & 0 \quad \text{if } s \not\models \exists\Diamond G \\ x_{s,i} & \geqslant & 0 \quad \text{if } s \models \exists\Diamond G \end{array}\right\} \text{ for } 0 \leqslant i \leqslant r$$

If $s \models \exists\Diamond G$ and $\alpha \in Act(s)$ then:

$$x_{s,i} \geqslant \sum_{t \in S} P(s,\alpha,t) \cdot x_{t,i-rew(s,\alpha)}$$

if $rew(s,\alpha) \leqslant i \leqslant r$

minimize $\sum_{s,i} x_{s,i}$

unique solution: $x_{s,i} = p_{s,i} = \Pr^{\max}_s(\Diamond^{\geqslant i} G)$

Summary: reward-bounded reachability in MDPs

Summary: reward-bounded reachability in MDPs

- threshold problem "$\mathrm{Pr}^{\max}_{s}(\Diamond^{\leqslant r} G) > \frac{1}{2}$?" is PSPACE-complete for acyclic MDPs [Haase/Kiefer'15]
- $\mathrm{Pr}^{\max}_{s}(\Diamond^{\leqslant r} G)$ computable in pseudo-polynomial time using an iterative LP-based approach
- analogous results for minimal probabilities, upper probability bounds and lower reward bounds

Summary: reward-bounded reachability in MDPs

- threshold problem "$\Pr_s^{\max}(\Diamond^{\leqslant r} G) > \frac{1}{2}$?" is PSPACE-complete for acyclic MDPs [Haase/Kiefer'15]
- $\Pr_s^{\max}(\Diamond^{\leqslant r} G)$ computable in pseudo-polynomial time using an iterative LP-based approach
- analogous results for minimal probabilities, upper probability bounds and lower reward bounds
- EXPTIME-completeness for the task to check whether $\Pr_s^{\max}(\Diamond^{= r} G) = 1$ [Haase/Kiefer'15]

Summary: reward-bounded reachability in MDPs

- threshold problem "$\Pr_s^{\max}(\Diamond^{\leqslant r} G) > \frac{1}{2}$?" is PSPACE-complete for acyclic MDPs [Haase/Kiefer'15]
- $\Pr_s^{\max}(\Diamond^{\leqslant r} G)$ computable in pseudo-polynomial time using an iterative LP-based approach
- analogous results for minimal probabilities, upper probability bounds and lower reward bounds
- EXPTIME-completeness for the task to check whether $\Pr_s^{\max}(\Diamond^{=r} G) = 1$ [Haase/Kiefer'15]
- qualitative quantiles, e.g., minimal $r \in \mathbb{N}$ with $\Pr_s^{\max}(\Diamond^{\leqslant r} G) = 1$, computable in polynomial time
  ... using a Dijkstra-like Greedy method [Ummels/Baier'13]

Weight-bounded reachability

Weight-bounded reachability

given: MDP $\mathcal{M}$ with weight fct $wgt : S \times Act \to \mathbb{Z}$,
$s \in S$, $G \subseteq S$ and weight bound $r \in \mathbb{Z}$

Weight-bounded reachability

given: MDP $\mathcal{M}$ with weight fct $wgt : S \times Act \to \mathbb{Z}$,
$s \in S$, $G \subseteq S$ and weight bound $r \in \mathbb{Z}$

qualitative decision problems:

$$\Pr_s^{\min}(\Diamond^{\leqslant r} G) = 1 \ ? \qquad \Pr_s^{\min}(\Diamond^{\leqslant r} G) > 0 \ ?$$

$$\Pr_s^{\max}(\Diamond^{\leqslant r} G) > 0 \ ? \qquad \Pr_s^{\max}(\Diamond^{\leqslant r} G) = 1 \ ?$$

Weight-bounded reachability

given: MDP $\mathcal{M}$ with weight fct $wgt : S \times Act \to \mathbb{Z}$,
$s \in S$, $G \subseteq S$ and weight bound $r \in \mathbb{Z}$

qualitative decision problems:

$$\Pr^{\min}_s(\Diamond^{\leqslant r} G) = 1 \ ? \qquad \Pr^{\min}_s(\Diamond^{\leqslant r} G) > 0 \ ?$$

$$\boxed{\Pr^{\max}_s(\Diamond^{\leqslant r} G) > 0 \ ?} \qquad \Pr^{\max}_s(\Diamond^{\leqslant r} G) = 1 \ ?$$

$\uparrow$

solvable by standard
shortest-path algorithms

Weight-bounded reachability

given: MDP $\mathcal{M}$ with weight fct $wgt : S \times Act \to \mathbb{Z}$,
$s \in S$, $G \subseteq S$ and weight bound $r \in \mathbb{Z}$

qualitative decision problems:

$\Pr_s^{\min}(\lozenge^{\leqslant r} G) = 1$? $\Pr_s^{\min}(\lozenge^{\leqslant r} G) > 0$?

$\boxed{\Pr_s^{\max}(\lozenge^{\leqslant r} G) > 0 \text{ ?}}$ $\Pr_s^{\max}(\lozenge^{\leqslant r} G) = 1$?

Compute the minimal weight $w \in \mathbb{Z} \cup \{-\infty\}$ of a shortest ("lightest") path from s to G.

If $w \leqslant r$ then return "yes" else return "no".

Weight-bounded reachability

given: MDP $\mathcal{M}$ with weight fct $wgt : S \times Act \to \mathbb{Z}$,
$s \in S$, $G \subseteq S$ and weight bound $r \in \mathbb{Z}$

qualitative decision problems:

$$\boxed{\begin{array}{l} \Pr^{\min}_s(\Diamond^{\leqslant r} G) = 1 \ ? \\ \Pr^{\max}_s(\Diamond^{\leqslant r} G) > 0 \ ? \end{array}}$$

$$\Pr^{\min}_s(\Diamond^{\leqslant r} G) > 0 \ ?$$

$$\Pr^{\max}_s(\Diamond^{\leqslant r} G) = 1 \ ?$$

$\uparrow$

solvable in
polynomial time

assuming G consists of trap states

Weight-bounded reachability

given: MDP $\mathcal{M}$ with weight fct $wgt : S \times Act \to \mathbb{Z}$,
$s \in S$, $G \subseteq S$ and weight bound $r \in \mathbb{Z}$

qualitative decision problems:

$\boxed{\Pr_s^{\min}(\Diamond^{\leqslant r} G) = 1 \ ?}$

$\Pr_s^{\min}(\Diamond^{\leqslant r} G) > 0 \ ?$

$\Pr_s^{\max}(\Diamond^{\leqslant r} G) = 1 \ ?$

$\uparrow$

Return "no" if $\Pr_s^{\min}(\Diamond G) < 1$ or ...

assuming G consists of trap states

Weight-bounded reachability

given: MDP $\mathcal{M}$ with weight fct $wgt : S \times Act \to \mathbb{Z}$,
$s \in S$, $G \subseteq S$ and weight bound $r \in \mathbb{Z}$

qualitative decision problems:

$\boxed{\Pr_s^{\min}(\Diamond^{\leqslant r} G) = 1 \ ?}$

$\Pr_s^{\min}(\Diamond^{\leqslant r} G) > 0 \ ?$

$\Pr_s^{\max}(\Diamond^{\leqslant r} G) = 1 \ ?$

Return "no" if $\Pr_s^{\min}(\Diamond G) < 1$ or there are positive cycles "between s and G"

assuming G consists of trap states

Weight-bounded reachability

given: MDP $\mathcal{M}$ with weight fct $wgt : S \times Act \to \mathbb{Z}$,
$s \in S$, $G \subseteq S$ and weight bound $r \in \mathbb{Z}$

qualitative decision problems:

$$\boxed{\Pr_s^{\min}(\Diamond^{\leqslant r} G) = 1 \ ?}$$

$$\Pr_s^{\min}(\Diamond^{\leqslant r} G) > 0 \ ?$$

$$\Pr_s^{\max}(\Diamond^{\leqslant r} G) = 1 \ ?$$

$\uparrow$

Return "no" if $\Pr_s^{\min}(\Diamond G) < 1$ or there are positive cycles "between s and G"

Compute the maximal weight w of a path from s to G.

Weight-bounded reachability

given: MDP $\mathcal{M}$ with weight fct $wgt : S \times Act \to \mathbb{Z}$,
$s \in S$, $G \subseteq S$ and weight bound $r \in \mathbb{Z}$

qualitative decision problems:

$$\boxed{\Pr_s^{\min}(\Diamond^{\leqslant r} G) = 1 \ ?}$$

$$\Pr_s^{\min}(\Diamond^{\leqslant r} G) > 0 \ ?$$

$$\Pr_s^{\max}(\Diamond^{\leqslant r} G) = 1 \ ?$$

Return "no" if $\Pr_s^{\min}(\Diamond G) < 1$ or there are positive cycles "between s and G"

Compute the maximal weight w of a path from s to G.

If $w > r$ then return "no" else return "yes".

Weight-bounded reachability

given: MDP $\mathcal{M}$ with weight fct $wgt : S \times Act \to \mathbb{Z}$,
$s \in S$, $G \subseteq S$ and weight bound $r \in \mathbb{Z}$

qualitative decision problems:

$$\Pr_s^{\min}(\Diamond^{\leqslant r} G) = 1 \ ?$$

$$\Pr_s^{\max}(\Diamond^{\leqslant r} G) > 0 \ ?$$

↑

solvable in polynomial time

$$\boxed{\begin{array}{c} \Pr_s^{\min}(\Diamond^{\leqslant r} G) > 0 \ ? \\ \Pr_s^{\max}(\Diamond^{\leqslant r} G) = 1 \ ? \end{array}}$$

↑

in NP ∩ coNP

Weight-bounded reachability

given: MDP $\mathcal{M}$ with weight fct $wgt : S \times Act \to \mathbb{Z}$,
$s \in S$, $G \subseteq S$ and weight bound $r \in \mathbb{Z}$

qualitative decision problems:

$\Pr^{\min}_s(\Diamond^{\leqslant r} G) = 1$?

$\Pr^{\max}_s(\Diamond^{\leqslant r} G) > 0$?

$\Pr^{\min}_s(\Diamond^{\leqslant r} G) > 0$?

$\Pr^{\max}_s(\Diamond^{\leqslant r} G) = 1$?

$\uparrow$

in **NP** $\cap$ **coNP**

using algorithms for energy-MDPs

[Chatterjee/Doyen'11]

[Mayer/Schewe/Totzke/Wojtczak'17]

Reduction to energy-MDPs

given: MDP $\mathcal{M}$ with weight fct $wgt : S \times Act \to \mathbb{Z}$,
$s \in S$, $G \subseteq S$ and weight bound $r \in \mathbb{Z}$

Suppose $G = S$.

Reduction to energy-MDPs

given: MDP $\mathcal{M}$ with weight fct $wgt : S \times Act \to \mathbb{Z}$,
$s \in S$, $G \subseteq S$ and weight bound $r \in \mathbb{Z}$

Suppose $G = S$.

$$\Pr_s^{\min}(\lozenge(wgt \geqslant r)) = 0$$

$$\text{iff} \quad \Pr_s^{\max}(\square(wgt < r)) = 1$$

Reduction to energy-MDPs

given: MDP $\mathcal{M}$ with weight fct $wgt : S \times Act \to \mathbb{Z}$,
$s \in S$, $G \subseteq S$ and weight bound $r \in \mathbb{Z}$

Suppose $G = S$.

$$\Pr_s^{\min}(\Diamond(wgt \geqslant r)) = 0$$

$$\text{iff} \quad \Pr_s^{\max}(\Box(wgt < r)) = 1$$

$$\text{iff} \quad \Pr_s^{\max}(\Box(-wgt > -r)) = 1$$

Reduction to energy-MDPs

given: MDP $\mathcal{M}$ with weight fct $wgt : S \times Act \to \mathbb{Z}$,
$s \in S$, $G \subseteq S$ and weight bound $r \in \mathbb{Z}$

Suppose $G = S$.

$$\Pr_s^{\min}(\Diamond(wgt \geqslant r)) = 0$$

$$\text{iff} \quad \Pr_s^{\max}(\Box(wgt < r)) = 1$$

$$\text{iff} \quad \underbrace{\Pr_s^{\max}(\Box(-wgt > -r)) = 1}_{\substack{\text{energy condition in the MDP} \\ \text{with weight fct } -wgt}}$$

Reduction to energy-MDPs

given: MDP $\mathcal{M}$ with weight fct $wgt : S \times Act \to \mathbb{Z}$,
$s \in S$, $G \subseteq S$ and weight bound $r \in \mathbb{Z}$

Suppose $G = S$.

$$\Pr^{\min}_s(\Diamond(wgt \geqslant r)) = 0$$

$$\text{iff} \quad \Pr^{\max}_s(\Box(wgt < r)) = 1$$

$$\text{iff} \quad \underbrace{\Pr^{\max}_s(\Box(-wgt > -r)) = 1}_{\substack{\text{interreducible with non-stochastic} \\ \text{two-player mean payoff games}}}$$

Reduction to mean payoff games

given: MDP $\mathcal{M}$ with weight fct $wgt : S \times Act \to \mathbb{Z}$,
and initial state $s \in S$

Suppose all states are reachable from s.

$$\exists w \in \mathbb{Z} \text{ s.t. } \Pr_s^{\max}(\Box(wgt > w)) = 1$$

iff ...

Reduction to mean payoff games

given: MDP $\mathcal{M}$ with weight fct $wgt : S \times Act \to \mathbb{Z}$,
and initial state $s \in S$

Suppose all states are reachable from s.

$$\exists w \in \mathbb{Z} \text{ s.t. } \Pr_s^{\max}(\Box(wgt > w)) = 1$$

iff there is an MD-scheduler without negative cycles

Reduction to mean payoff games

given: MDP $\mathcal{M}$ with weight fct $wgt : S \times Act \to \mathbb{Z}$,
and initial state $s \in S$

Suppose all states are reachable from s.

$$\exists w \in \mathbb{Z} \text{ s.t. } \Pr^{\max}_s(\Box(wgt > w)) = 1$$

iff there is an MD-scheduler without negative cycles

iff the induced 2-player game has a winning strategy for the objective "non-negative mean payoff"

Reduction to mean payoff games

given: MDP $\mathcal{M}$ with weight fct $wgt : S \times Act \to \mathbb{Z}$,
and initial state $s \in S$

Suppose all states are reachable from s.

$$\exists w \in \mathbb{Z} \text{ s.t. } \Pr^{\max}_s(\Box(wgt > w)) = 1$$

iff there is an MD-scheduler without negative cycles

iff the induced 2-player game has a winning strategy for the objective "non-negative mean payoff"

If so, the minimal w is computable in polynomial time.

Weight-bounded reachability

given: MDP $\mathcal{M}$ with weight fct $wgt : S \times Act \to \mathbb{Z}$,
a state $s \in S$ and weight bound $r \in \mathbb{Z}$

qualitative decision problems:

$\Pr_s^{\min}(\Diamond^{\leqslant r} G) = 1$? $\qquad$ $\Pr_s^{\min}(\Diamond^{\leqslant r} G) > 0$?

$\Pr_s^{\max}(\Diamond^{\leqslant r} G) > 0$? $\qquad$ $\Pr_s^{\max}(\Diamond^{\leqslant r} G) = 1$?

solvable in polynomial time

as hard as two-player mean payoff games

not known to be solvable in poly-time

Weight-bounded reachability

given: MDP $\mathcal{M}$ with weight fct $wgt : S \times Act \to \mathbb{Z}$,
a state $s \in S$ and weight bound $r \in \mathbb{Z}$

qualitative decision problems:

$\Pr_s^{\min}(\lozenge^{\leqslant r} G) = 1$? $\qquad$ $\Pr_s^{\min}(\lozenge^{\leqslant r} G) > 0$?

$\Pr_s^{\max}(\lozenge^{\leqslant r} G) > 0$? $\qquad$ $\Pr_s^{\max}(\lozenge^{\leqslant r} G) = 1$?

Is there an algorithm to compute $\Pr_s^{\max}(\lozenge^{\leqslant r} G)$?

Weight-bounded reachability

given: MDP $\mathcal{M}$ with weight fct $wgt : S \times Act \to \mathbb{Z}$,
a state $s \in S$ and weight bound $r \in \mathbb{Z}$

qualitative decision problems:

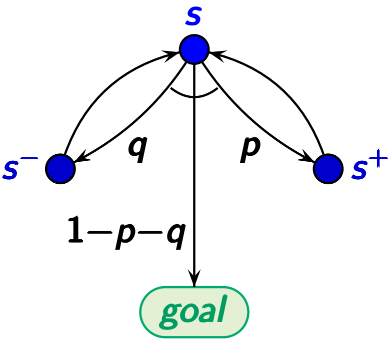
$$\Pr^{\min}_{s}(\Diamond^{\leqslant r} G) = 1 \ ? \qquad \Pr^{\min}_{s}(\Diamond^{\leqslant r} G) > 0 \ ?$$

$$\Pr^{\max}_{s}(\Diamond^{\leqslant r} G) > 0 \ ? \qquad \Pr^{\max}_{s}(\Diamond^{\leqslant r} G) = 1 \ ?$$

Is there an algorithm to compute $\Pr^{\max}_{s}(\Diamond^{\leqslant r} G)$?

Maybe, but the probabilities can be irrational,
even for Markov chains ...

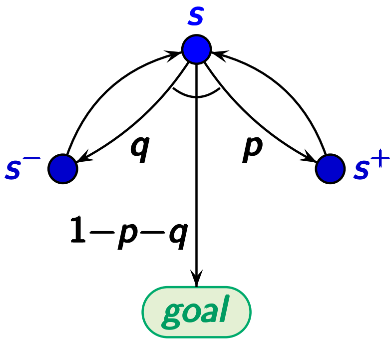
Weight-bounded reachability in MC [KIEFER'17]



$$wgt(s) = 0$$
$$wgt(s^-) = -1$$
$$wgt(s^+) = +1$$

probability parameters
p and q with $0 < p, q < 1$
and $p + q < 1$

Weight-bounded reachability in MC [KIEFER'17]



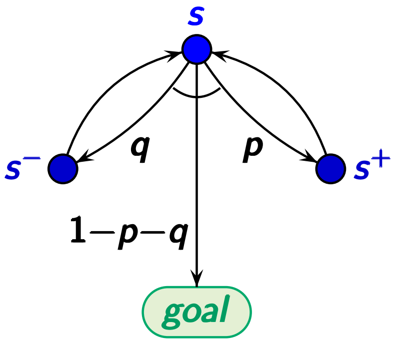
$wgt(s) = 0$

$wgt(s^-) = -1$

$wgt(s^+) = +1$

$$\Pr_s(\Diamond^{=0} goal) = (1-p-q) \cdot \sum_{n=0}^{\infty} \binom{2n}{n} \cdot p^n \cdot q^n$$

Weight-bounded reachability in MC [Kiefer'17]

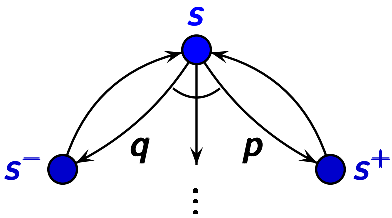


$$wgt(s) = 0$$
$$wgt(s^-) = -1$$
$$wgt(s^+) = +1$$

$$\Pr_s(\Diamond^{=0} goal) = (1-p-q) \cdot \sum_{n=0}^{\infty} \binom{2n}{n} \cdot p^n \cdot q^n$$

$$= \frac{1-p-q}{\sqrt{1-4 \cdot p \cdot q}} \quad \text{... irrational}$$

Weight-bounded reachability in MC



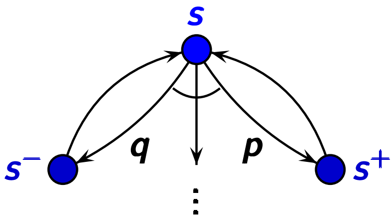
$$wgt(s) = 0$$
$$wgt(s^-) = -1$$
$$wgt(s^+) = +1$$

What we have so far:

- weight-bounded reachability prob. can be irrational
- qualitative decision problems solvable in poly-time

$$\left.\begin{array}{l} \Pr_s(\Diamond^{\leqslant r} goal) > 0 \; ? \\[2ex] \Pr_s(\Diamond^{\leqslant r} goal) = 1 \; ? \\ \text{where } goal \text{ is a trap} \end{array}\right\} \quad \text{shortest-path algorithm}$$

Weight-bounded reachability in MC



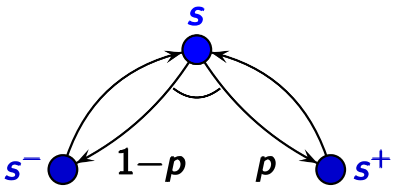
$$
\begin{aligned}
wgt(s) &= 0 \\
wgt(s^-) &= -1 \\
wgt(s^+) &= +1
\end{aligned}
$$

What we have so far:

- weight-bounded reachability prob. can be irrational
- qualitative decision problems solvable in poly-time

$$
\left.
\begin{array}{l}
\Pr_s(\Diamond^{\leqslant r} goal) > 0 \ ? \\[1ex]
\Pr_s(\Diamond^{\leqslant r} goal) = 1 \ ? \\
\text{where } goal \text{ is a trap}
\end{array}
\right\}
\begin{array}{l}
\text{do not depend} \\
\text{on the concrete} \\
\text{probabilities}
\end{array}
$$

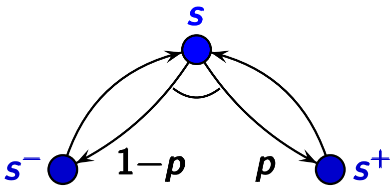
Markov chain with weight function



$$wgt(s) = +1$$
$$wgt(s^-) = -2$$
$$wgt(s^+) = 0$$

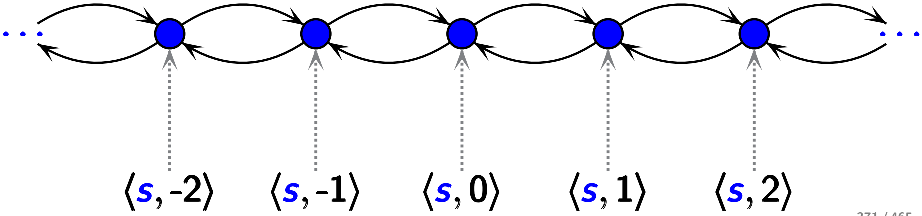
does $\Pr_s\big(\Diamond(wgt < 0)\big) = 1$ hold ?

Markov chain with weight function

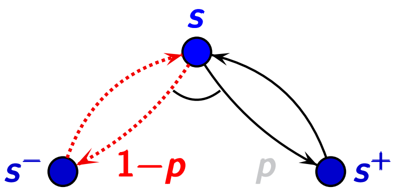


$wgt(s) = +1$
$wgt(s^-) = -2$
$wgt(s^+) = 0$

random walk:



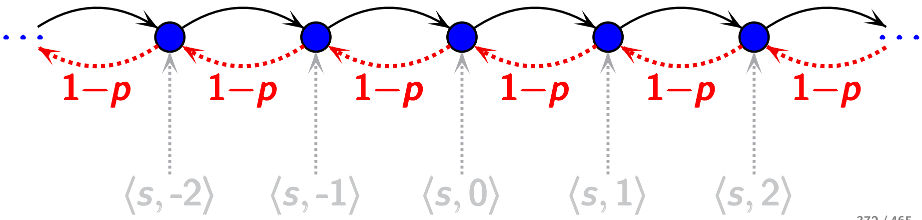
Markov chain with weight function



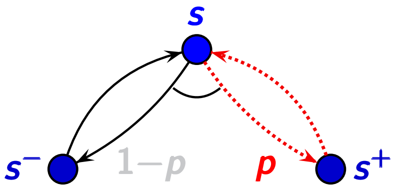
$$
\begin{aligned}
wgt(s) &= +1\\
wgt(s^-) &= -2\\
wgt(s^+) &= 0
\end{aligned}
$$

weight $-\mathbf{1}$ for the cycle $s\ s^-\ s$

random walk:



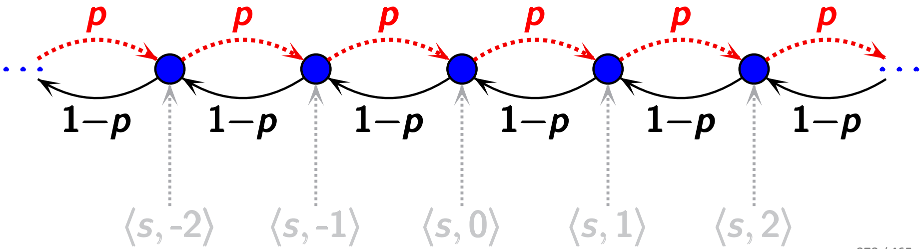
Markov chain with weight function

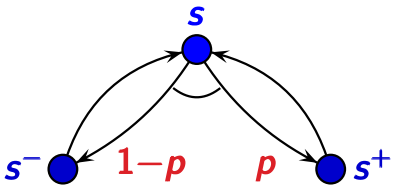


$$wgt(s) = +1$$
$$wgt(s^-) = -2$$
$$wgt(s^+) = 0$$

weight $+1$ for the cycle $s\ s^+\ s$

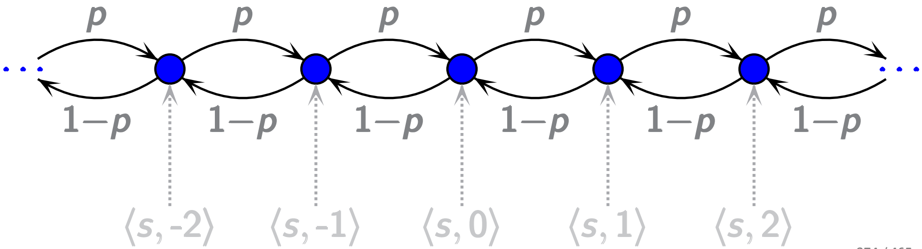
random walk:



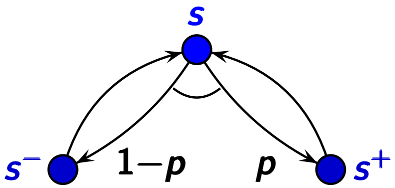


$$wgt(s) = +1$$
$$wgt(s^-) = -2$$
$$wgt(s^+) = 0$$

$$\Pr_s(\Box(wgt \geqslant 0)) > 0 \quad \text{iff} \quad p > \frac{1}{2}$$



Markov chain with weight function

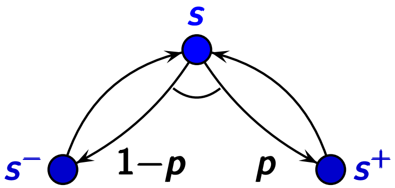


$$
\begin{aligned}
wgt(s) &= +1\\
wgt(s^-) &= -2\\
wgt(s^+) &= 0
\end{aligned}
$$

$$\boxed{\Pr_s(\Box(wgt \geqslant r)) > 0 \quad \text{iff} \quad p > \frac{1}{2}}$$

for each $r \in \mathbb{Z}$, $r \leqslant 0$

Markov chain with weight function

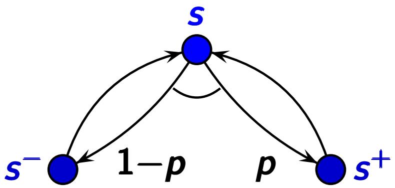


$$wgt(s) = +1$$
$$wgt(s^-) = -2$$
$$wgt(s^+) = 0$$

$$\Pr_s(\Box(wgt \geqslant r)) = 0 \quad \text{iff} \quad p \leqslant \frac{1}{2}$$

for each $r \in \mathbb{Z}$, $r \leqslant 0$

Markov chain with weight function

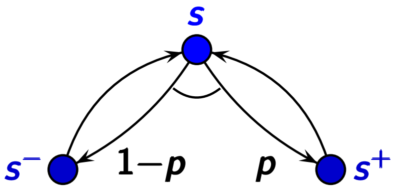


$$\begin{aligned} wgt(s) &= +1 \\ wgt(s^-) &= -2 \\ wgt(s^+) &= 0 \end{aligned}$$

$$\Pr_s(\,\Box(wgt \geqslant r)\,) = 0 \quad \text{iff} \quad p \leqslant \frac{1}{2}$$

$$\text{iff} \quad \Pr_s(\,\Diamond(wgt < r)\,) = 1$$

Markov chain with weight function

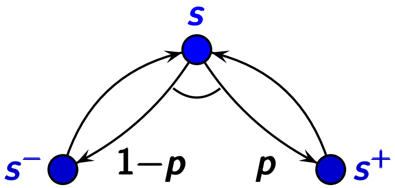


$$\begin{aligned} wgt(s) &= +1 \\ wgt(s^-) &= -2 \\ wgt(s^+) &= 0 \end{aligned}$$

$$\Pr_s(\Box(wgt \geqslant r)) = 0 \quad \text{iff} \quad p \leqslant \frac{1}{2}$$

$$\text{iff} \quad \underbrace{\Pr_s(\Diamond(wgt < r)) = 1}_{\substack{\text{depends on the concrete} \\ \text{transition probabilities}}}$$

Markov chain with weight function



$$wgt(s) = +1$$
$$wgt(s^-) = -2$$
$$wgt(s^+) = 0$$

$$\Pr_s(\Box(wgt \geqslant r)) = 0 \quad \text{iff} \quad p \leqslant \frac{1}{2}$$

$$\text{iff} \quad \underbrace{\Pr_s(\Diamond(wgt < r)) = 1}_{\text{depends on the concrete transition probabilities}}$$

But: "does $\Pr_s(\Diamond^{<r} goal) = 1$ hold ?", where *goal* is a trap, does not depend on the transition probabilities.

Weight invariance problem: positive case

The problem "does $\mathrm{Pr}_s(\Box(wgt > r) \wedge \varphi) > 0$ hold ?"

- depends on the concrete transition probabilities

where φ is a ω-regular property and $0 \leqslant q < 1$

Weight invariance problem: positive case

The problem "does $\mathbf{Pr}_s(\Box(wgt > r) \wedge \varphi) > 0$ hold ?"

- depends on the concrete transition probabilities
- is solvable in polynomial time

  BSCC-analysis and variants of shortest-paths algorithms, assuming φ is a Rabin or Streett or reachability condition

  [Brázdil/Kiefer/Kučera/Novotný/Katoen'14]
  [Krähmann/Schubert/Baier/Dubslaff'15]

where φ is a ω-regular property and $0 \leqslant q < 1$

Weight invariance problem: positive case

The problem "does $\mathrm{Pr}_s(\,\Box(wgt > r) \wedge \varphi\,) > \mathbf{0}$ hold ?"

- depends on the concrete transition probabilities
- is solvable in polynomial time

  BSCC-analysis and variants of shortest-paths algorithms, assuming φ is a Rabin or Streett condition

  [Brázdil/Kiefer/Kučera/Novotný/Katoen'14]
  [Krähmann/Schubert/Baier/Dubslaff'15]

check whether there exists a good BSCC B s.t. either $\mathrm{MP}(B) > \mathbf{0}$ or $\mathrm{MP}(B) = \mathbf{0}$ and there is a path from s to B with sufficiently high weight

Weight invariance problem: quantitative case

The problem "does $\Pr_s(\Box(wgt > r) \wedge \varphi) > 0$ hold ?"

- depends on the concrete transition probabilities
- is solvable in polynomial time

  BSCC-analysis and variants of shortest-paths algorithms, assuming φ is a Rabin or Streett condition

The problem "does $\Pr_s(\Box(wgt > 0) \wedge \varphi) > q$ hold ?"

- is reducible to the threshold problem for probabilistic pushdown automata (exponential blowup)

Weight invariance problem: quantitative case

The problem "does $\Pr_s(\Box(wgt > r) \land \varphi) > 0$ hold ?"

- depends on the concrete transition probabilities
- is solvable in polynomial time

  BSCC-analysis and variants of shortest-paths algorithms, assuming φ is a Rabin or Streett condition

The problem "does $\Pr_s(\Box(wgt > 0) \land \varphi) > q$ hold ?"

- is reducible to the threshold problem for probabilistic pushdown automata (exponential blowup)
- is PosSLP-hard, even for unit weights and $\varphi = \mathit{true}$

  [Etessami/Yannak.'09], [Brázdil/Brozek/Etes./Kučera/Wojt.'10]

Weight invariance problem: almost-sure case

The problem "does $\mathrm{Pr}_s(\Box(wgt > r) \wedge \varphi) = 1$ hold ?"

- independent from the concrete transition probabilities
- is solvable in polynomial time

Weight invariance problem: almost-sure case

The problem "does $\Pr_s(\Box(wgt > r) \wedge \varphi) = 1$ hold ?"

- independent from the concrete transition probabilities
- is solvable in polynomial time

$$\Pr_s(\Box(wgt > r) \wedge \varphi) = 1$$

$$\text{iff} \quad \Pr_s(\Box(wgt > r)) = 1 \quad \text{and} \quad \Pr_s(\varphi) = 1$$

Weight invariance problem: almost-sure case

The problem "does $\mathrm{Pr}_s(\Box(wgt > r) \wedge \varphi) = 1$ hold ?"

- independent from the concrete transition probabilities
- is solvable in polynomial time

$$\mathrm{Pr}_s(\Box(wgt > r) \wedge \varphi) = 1$$

iff $\quad \mathrm{Pr}_s(\Box(wgt > r)) = 1 \quad$ and $\boxed{\mathrm{Pr}_s(\varphi) = 1}$

standard algorithm

polynomial-time for

reachability, Rabin or Streett

Weight invariance problem: almost-sure case

The problem "does $\Pr_s(\Box(wgt > r) \wedge \varphi) = 1$ hold ?"

- independent from the concrete transition probabilities
- is solvable in polynomial time

$$\Pr_s(\Box(wgt > r) \wedge \varphi) = 1$$

iff $\boxed{\Pr_s(\Box(wgt > r)) = 1}$ and $\Pr_s(\varphi) = 1$

shortest-path algorithm

check chether the weight of a shortest path from s is at least r+1

Quantiles for ratio constraints

Given a Markov chain $\mathcal{M}$ with two reward functions $rew_1, rew_2 : S \to \mathbb{N}$ with $rew_2 > \mathbf{0}$, consider their ratio

examples:

- energy-utility ratio
- cost of repair mechanisms per failure
- SLA violations per day

Quantiles for ratio constraints

Given a Markov chain $\mathcal{M}$ with two reward functions $rew_1, rew_2 : S \to \mathbb{N}$ with $rew_2 > \mathbf{0}$, consider their ratio:

$$ratio : FinPaths \to \mathbb{Q}, \quad ratio(\pi) \;=\; \frac{rew_1(\pi)}{rew_2(\pi)}$$

examples:

- energy-utility ratio
- cost of repair mechanisms per failure
- SLA violations per day

Quantiles for ratio constraints

Given a Markov chain $\mathcal{M}$ with two reward functions $rew_1, rew_2 : S \to \mathbb{N}$ with $rew_2 > \mathbf{0}$, consider their ratio:

$$ratio : FinPaths \to \mathbb{Q}, \quad ratio(\pi) \;=\; \frac{rew_1(\pi)}{rew_2(\pi)}$$

qualitative quantiles for ratio invariances:

$$\mathbf{sup}\ \{\ r \in \mathbb{Q} :\ \Pr_s(\Box(ratio > r)) > 0\ \}$$

$$\mathbf{sup}\ \{\ r \in \mathbb{Q} :\ \Pr_s(\Box(ratio > r)) = 1\ \}$$

examples:

- energy-utility ratio
- cost of repair mechanisms per failure
- SLA violations per day

Quantiles for ratio constraints

Given a Markov chain $\mathcal{M}$ with two reward functions $rew_1, rew_2 : S \to \mathbb{N}$ with $rew_2 > \mathbf{0}$, consider their ratio:

$$ratio : FinPaths \to \mathbb{Q}, \quad ratio(\pi) \; = \; \frac{rew_1(\pi)}{rew_2(\pi)}$$

qualitative quantiles for ratio invariances:

$$\sup \{ \; r \in \mathbb{Q} : \; \Pr_s(\Box(ratio > r)) > 0 \; \}$$

$$\sup \{ \; r \in \mathbb{Q} : \; \Pr_s(\Box(ratio > r)) = 1 \; \}$$

... are computable in polynomial time ...

[Krähmann/Schubert/Baier/Dubslaff'15]

Quantiles for ratio constraints

Given a Markov chain $\mathcal{M}$ with two reward functions $rew_1, rew_2 : S \to \mathbb{N}$ with $rew_2 > \mathbf{0}$, consider their ratio:

$$ratio : FinPaths \to \mathbb{Q}, \quad ratio(\pi) \;=\; \frac{rew_1(\pi)}{rew_2(\pi)}$$

qualitative quantiles for ratio invariances:

$$\sup \left\{ r \in \mathbb{Q} : \ \Pr_s(\Box(ratio > r)) > 0 \ \right\}$$

$$\sup \left\{ r \in \mathbb{Q} : \ \Pr_s(\Box(ratio > r)) = 1 \ \right\}$$

... are computable in polynomial time ...

[Krähmann/Schubert/Baier/Dubslaff'15]

Positive ratio quantiles

$$\sup \left\{ r \in \mathbb{Q} : \Pr_s(\Box(ratio > r)) > 0 \right\}$$

$$ratio = \frac{rew_1}{rew_2} : FinPaths \to \mathbb{Q} \quad \text{where } rew_2 > 0$$

Positive ratio quantiles

$$\sup \left\{ r \in \mathbb{Q} : \boxed{\Pr_s(\Box(\mathit{ratio} > r)) > 0} \right\}$$

- inner decision problem for fixed r is solvable in polynomial time

$$\mathit{ratio} = \frac{\mathit{rew}_1}{\mathit{rew}_2} : \mathit{FinPaths} \to \mathbb{Q} \quad \text{where } \mathit{rew}_2 > 0$$

Positive ratio quantiles

$$\text{sup} \left\{ r \in \mathbb{Q} : \boxed{\Pr_s(\Box(\mathit{ratio} > r)) > 0} \right\}$$

- inner decision problem for fixed r is solvable in polynomial time

reduction to positive weight invariances:

$$\mathit{ratio} > r \quad \text{iff} \quad \mathit{rew}_1 - r \cdot \mathit{rew}_2 > 0$$

$\mathit{ratio} = \frac{\mathit{rew}_1}{\mathit{rew}_2} : \mathit{FinPaths} \to \mathbb{Q}$ where $\mathit{rew}_2 > 0$

Positive ratio quantiles

$$\sup \left\{ r \in \mathbb{Q} : \Pr_s(\Box(ratio > r)) > 0 \right\}$$

- inner decision problem for fixed r is solvable in polynomial time

reduction to positive weight invariances:

$$ratio > r \quad \text{iff} \quad \underbrace{rew_1 - r \cdot rew_2}_{\text{weight function}} > 0$$

$ratio = \frac{rew_1}{rew_2} : FinPaths \to \mathbb{Q}$ where $rew_2 > 0$

Positive ratio quantiles

$$\sup \left\{ \, r \in \mathbb{Q} : \; \Pr_s(\Box(\mathit{ratio} > r)) > 0 \, \right\}$$

- inner decision problem for fixed r is solvable in polynomial time

reduction to positive weight invariances:

$$\mathit{ratio} > r \quad \text{iff} \quad \underbrace{\mathit{rew}_1 - r \cdot \mathit{rew}_2}_{\text{weight function}} > 0$$

If $r \in \mathbb{Q}$ then pick some $c \in \mathbb{N}$ such that $(\mathit{rew}_1 - r \cdot \mathit{rew}_2) \cdot c$ is an integer weight function.

Positive ratio quantiles

$$\mathbf{sup}\ \{\ r \in \mathbb{Q} :\ \Pr_s(\Box(\mathit{ratio} > r)) > 0\ \}$$

- inner decision problem for fixed r is solvable in polynomial time
- quantile can be approximated using a binary search

$$\mathit{ratio} = \frac{\mathit{rew}_1}{\mathit{rew}_2} : \mathit{FinPaths} \to \mathbb{Q} \quad \text{where } \mathit{rew}_2 > 0$$

Positive ratio quantiles

$$\sup \left\{ r \in \mathbb{Q} : \Pr_s(\Box(ratio > r)) > 0 \right\}$$

- inner decision problem for fixed r is solvable in polynomial time
- quantile can be approximated using a binary search

for all finite paths π:

$$0 \leqslant ratio(\pi) \leqslant \frac{\max rew_1}{\min rew_2}$$

$$ratio = \frac{rew_1}{rew_2} : FinPaths \to \mathbb{Q} \quad \text{where } rew_2 > 0$$

Positive ratio quantiles

$$\sup \left\{ \, r \in \mathbb{Q} : \; \Pr_s(\, \Box(\mathit{ratio} > r)\,) > 0 \; \right\}$$

- inner decision problem for fixed r is solvable in polynomial time
- quantile can be approximated using a binary search and is one of the values
  - expected long-run ratio of a BSCC

If B is a BSCC then the expected long-run ratio is:

$$\frac{\mathrm{MP}_B[\mathit{rew}_1]}{\mathrm{MP}_B[\mathit{rew}_2]} \quad \text{where} \quad \mathrm{MP}_B[\mathit{rew}] = \left\{ \begin{array}{l} \text{mean-payoff} \\ \text{of } \mathit{rew} \text{ in } B \end{array} \right.$$

Positive ratio quantiles

$$\textbf{sup}\ \{\ r \in \mathbb{Q} :\ \Pr_s(\Box(\mathit{ratio} > r)) > 0\ \}$$

- inner decision problem for fixed r is solvable in polynomial time
- quantile can be approximated using a binary search and is one of the values
  - expected long-run ratio of a BSCC
  - $\mathit{ratio}(\pi)$ for a simple path π from s
  - $\mathit{ratio}(\pi)$ for a simple cycle π reachable from s

Positive ratio quantiles

$$\sup \left\{ r \in \mathbb{Q} : \Pr_s(\Box(ratio > r)) > 0 \right\}$$

- inner decision problem for fixed r is solvable in polynomial time
- quantile can be approximated using a binary search and is one of the values
  - expected long-run ratio of a BSCC
  - $ratio(\pi)$ for a simple path π from s
  - $ratio(\pi)$ for a simple cycle π reachable from s

finitely many values

Positive ratio quantiles

$$\sup \{ r \in \mathbb{Q} : \Pr_s(\Box(ratio > r)) > 0 \}$$

- inner decision problem for fixed r is solvable in polynomial time
- quantile can be approximated using a binary search and is one of the values ... and therefore rational
  - expected long-run ratio of a BSCC
  - $ratio(\pi)$ for a simple path π from s
  - $ratio(\pi)$ for a simple cycle π reachable from s

finitely many values

Positive ratio quantiles

$$\sup\left\{\, r \in \mathbb{Q} : \ \Pr_s(\Box(\mathit{ratio} > r)) > 0 \,\right\}$$

- inner decision problem for fixed r is solvable in polynomial time
- quantile can be approximated using a binary search and is one of the values ... and therefore rational
  - expected long-run ratio of a BSCC
  - $\mathit{ratio}(\pi)$ for a simple path π from s
  - $\mathit{ratio}(\pi)$ for a simple cycle π reachable from s

finitely many values

- computation using the continued-fraction method

Positive ratio quantiles

$$\sup \left\{ \, r \in \mathbb{Q} : \Pr_s(\Box(\mathit{ratio} > r)) > 0 \, \right\} \; = \; \frac{c}{d}$$

where $c, d \in \mathbb{N}$ with $d > 0$

- quantile can be approximated using a binary search and is one of the values ... and therefore rational
  - $*$ expected long-run ratio of a BSCC
  - $*$ $\mathit{ratio}(\pi)$ for a simple path π from s
  - $*$ $\mathit{ratio}(\pi)$ for a simple cycle π reachable from s

  finitely many values
- computation using the continued-fraction method

Positive ratio quantiles

$$\sup \left\{ r \in \mathbb{Q} : \Pr_s(\Box(ratio > r)) > 0 \right\} = \frac{c}{d}$$

where $d \leqslant D = \max\left\{ \max_B d_B, \ |S| \cdot \max rew_2 \right\}$

- quantile can be approximated using a binary search and is one of the values ... and therefore rational
  - $*$ expected long-run ratio c_B/d_B of BSCC B
  - $*$ $ratio(\pi)$ for a simple path π from s
  - $*$ $ratio(\pi)$ for a simple cycle π reachable from s
- computation using the continued-fraction method

Positive ratio quantiles

$$\sup \left\{ r \in \mathbb{Q} : \Pr_s(\Box(ratio > r)) > 0 \right\} = \frac{c}{d}$$

where $d \leqslant D = \max\left\{ \max_B d_B, \ |S| \cdot \max rew_2 \right\}$

1. compute an approximation p of the quantile up to precision $\varepsilon = 1/2D^2$

$$\left|\frac{c}{d} - p\right| < \varepsilon$$

Positive ratio quantiles

$$\textbf{sup}\ \{\ r \in \mathbb{Q} :\ \Pr_s(\Box(\mathit{ratio} > r))\ > 0\ \}\ =\ \frac{c}{d}$$

where $d \leqslant D = \max\{\max_B d_B,\ |S| \cdot \max \mathit{rew}_2\}$

1. compute an approximation p of the quantile up to precision $\varepsilon = 1/2D^2$

The quantile is the best rational approximation of p with denominator at most D

$$\left|\frac{c}{d} - p\right| < \varepsilon$$

Positive ratio quantiles

$$\sup \left\{ r \in \mathbb{Q} : \ \Pr_s(\Box(ratio > r)) > 0 \right\} \ = \ \frac{c}{d}$$

where $d \leqslant D = \max\left\{ \max_B d_B, \ |S| \cdot \max rew_2 \right\}$

1. compute an approximation p of the quantile up to precision $\varepsilon = 1/2D^2$

The quantile is the best rational approximation of p with denominator at most D, i.e., if $a, b \in \mathbb{N}$ with $0 < b \leqslant D$ then:

$$\left|\frac{a}{b} - p\right| < \varepsilon \quad \text{iff} \quad \frac{a}{b} = \frac{c}{d}$$

Positive ratio quantiles

$$\sup \left\{ r \in \mathbb{Q} : \Pr_s(\Box(\mathit{ratio} > r)) > 0 \right\} = \frac{c}{d}$$

where $d \leqslant D = \max\left\{ \max_B d_B,\ |S| \cdot \max \mathit{rew}_2 \right\}$

1. compute an approximation p of the quantile up to precision $\varepsilon = 1/2D^2$
2. apply the continued-fraction method to p

The quantile is the best rational approximation of p with denominator at most D, i.e., if $a, b \in \mathbb{N}$ with $0 < b \leqslant D$ then:

$$\left|\frac{a}{b} - p\right| < \varepsilon \quad \text{iff} \quad \frac{a}{b} = \frac{c}{d}$$

Positive ratio quantiles

$$\sup \left\{ r \in \mathbb{Q} : \Pr_s(\Box(ratio > r)) > 0 \right\} = \frac{c}{d}$$

where $d \leqslant D = \max\left\{ \max_B d_B, \ |S| \cdot \max rew_2 \right\}$

1. compute an approximation p of the quantile up to precision $\varepsilon = 1/2D^2$

2. apply the continued-fraction method to p

$$p = p_1 + \cfrac{1}{p_2 + \cfrac{1}{p_3 + \cfrac{1}{p_4 + \cfrac{1}{\cdots}}}}$$

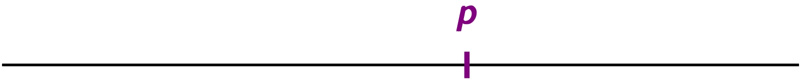
Positive ratio quantiles

$$\textbf{sup}\left\{ r \in \mathbb{Q} : \Pr_s(\Box(\mathit{ratio} > r)) > 0 \right\} = \frac{c}{d}$$

where $d \leqslant D = \max\left\{ \max_B d_B, \; |S| \cdot \max \mathit{rew}_2 \right\}$

1. compute an approximation p of the quantile up to precision $\varepsilon = 1/2D^2$

2. apply the continued-fraction method to p
   [Grötschel/Lovász/Schrijver'87]



Positive ratio quantiles

$$\sup \left\{ r \in \mathbb{Q} : \Pr_s(\Box(ratio > r)) > 0 \right\} = \frac{c}{d}$$

where $d \leqslant D = \max\left\{ \max_B d_B, \ |S| \cdot \max rew_2 \right\}$

1. compute an approximation p of the quantile up to precision $\varepsilon = 1/2D^2$

2. apply the continued-fraction method to p
   [Grötschel/Lovász/Schrijver'87]

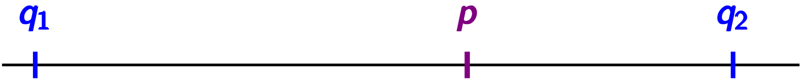
Positive ratio quantiles

$$\sup \left\{ r \in \mathbb{Q} : \Pr_s(\Box(ratio > r)) > 0 \right\} = \frac{c}{d}$$

where $d \leqslant D = \max\left\{ \max_B d_B, \; |S| \cdot \max rew_2 \right\}$

1. compute an approximation p of the quantile up to precision $\varepsilon = 1/2D^2$

2. apply the continued-fraction method to p
[Grötschel/Lovász/Schrijver'87]



Positive ratio quantiles

$$\sup \left\{ r \in \mathbb{Q} : \Pr_s(\Box(ratio > r)) > 0 \right\} = \frac{c}{d}$$

where $d \leqslant D = \max\left\{ \max_B d_B, \ |S| \cdot \max rew_2 \right\}$

1. compute an approximation p of the quantile up to precision $\varepsilon = 1/2D^2$

2. apply the continued-fraction method to p
   [Grötschel/Lovász/Schrijver'87]

Positive ratio quantiles

$$\sup \left\{ r \in \mathbb{Q} : \Pr_s(\Box(\mathit{ratio} > r)) > 0 \right\} = \frac{c}{d}$$

where $d \leqslant D = \max\left\{ \max_B d_B,\ |S| \cdot \max \mathit{rew}_2 \right\}$

1. compute an approximation p of the quantile up to precision $\varepsilon = 1/2D^2$

2. apply the continued-fraction method to p

   [Grötschel/Lovász/Schrijver'87]

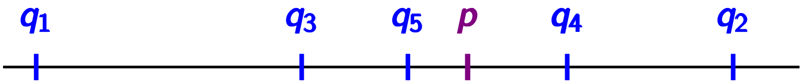
Positive ratio quantiles

$$\sup\left\{\, r \in \mathbb{Q} : \ \Pr_s(\Box(ratio > r)) > 0 \,\right\} \ = \ \frac{c}{d}$$

where $d \leqslant D = \max\left\{ \max_B d_B, \ |S| \cdot \max rew_2 \right\}$

1. compute an approximation p of the quantile up to precision $\varepsilon = 1/2D^2$
2. apply the continued-fraction method to p

   [Grötschel/Lovász/Schrijver'87]



Positive ratio quantiles

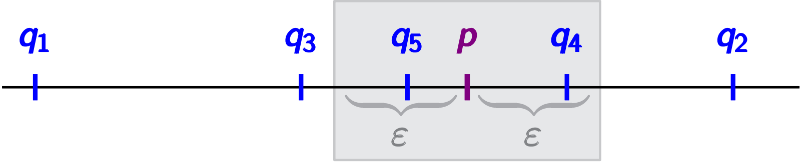
$$\sup \left\{ r \in \mathbb{Q} : \Pr_s(\Box(ratio > r)) > 0 \right\} = \frac{c}{d}$$

where $d \leqslant D = \max\left\{ \max_B d_B,\ |S| \cdot \max rew_2 \right\}$

1. compute an approximation p of the quantile up to precision $\varepsilon = 1/2D^2$

2. apply the continued-fraction method to p

[Grötschel/Lovász/Schrijver'87]



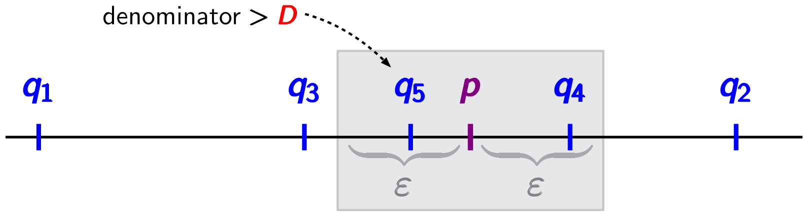
Positive ratio quantiles

$$\sup \left\{ r \in \mathbb{Q} : \Pr_s(\Box(\mathit{ratio} > r)) > 0 \right\} = \frac{c}{d}$$

where $d \leqslant D = \max\left\{ \max_B d_B, \ |S| \cdot \max \mathit{rew}_2 \right\}$

1. compute an approximation p of the quantile up to precision $\varepsilon = 1/2D^2$

2. apply the continued-fraction method to p



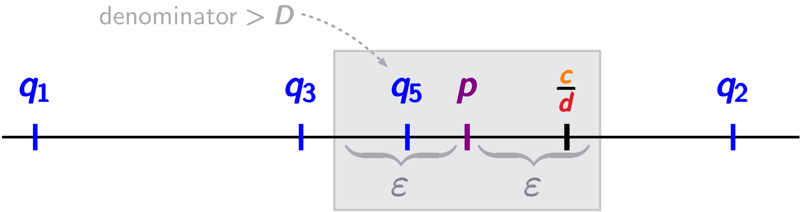
Positive ratio quantiles

$$\sup \left\{ r \in \mathbb{Q} : \ \Pr_s(\Box(ratio > r)) > 0 \ \right\} = \frac{c}{d}$$

where $d \leqslant D = \max\left\{ \max_B d_B, \ |S| \cdot \max rew_2 \right\}$

1. compute an approximation p of the quantile up to precision $\varepsilon = 1/2D^2$

2. apply the continued-fraction method to p



Polynomially computable ratio quantiles

qualitative quantiles for ratio invariances:

$$\sup\left\{ r \in \mathbb{Q} : \Pr_s\big(\Box(ratio > r) \wedge \varphi\big) > 0 \right\}$$

$$\sup\left\{ r \in \mathbb{Q} : \Pr_s\big(\Box(ratio > r) \wedge \varphi\big) = 1 \right\}$$

where φ is a reachability, Rabin or Streett condition

Polynomially computable ratio quantiles

qualitative quantiles for ratio invariances:

$$\mathbf{sup}\left\{ r \in \mathbb{Q} \;:\; \Pr_s\big(\Box(\mathit{ratio} > r) \wedge \varphi\big) > 0 \right\}$$

$$\mathbf{sup}\left\{ r \in \mathbb{Q} \;:\; \Pr_s\big(\Box(\mathit{ratio} > r) \wedge \varphi\big) = 1 \right\}$$

qualitative and quantitative quantiles for long-run ratios:

$$\mathbf{sup}\left\{ r \in \mathbb{Q} \;:\; \Pr_s\big(\Box\Diamond(\mathit{ratio} > r) \wedge \varphi\big) = 1 \right\}$$

$$\mathbf{sup}\left\{ r \in \mathbb{Q} \;:\; \Pr_s\big(\Box\Diamond(\mathit{ratio} > r) \wedge \varphi\big) > q \right\}$$

where φ is a reachability, Rabin or Streett condition

Polynomially computable ratio quantiles

qualitative quantiles for ratio invariances:

$$\textbf{sup}\,\{\, r \in \mathbb{Q} \,:\, \Pr_s\big(\Box(\mathit{ratio} > r) \wedge \varphi\big) \,>\, 0 \,\}$$

$$\textbf{sup}\,\{\, r \in \mathbb{Q} \,:\, \Pr_s\big(\Box(\mathit{ratio} > r) \wedge \varphi\big) \,=\, 1 \,\}$$

qualitative and quantitative quantiles for long-run ratios:

$$\textbf{sup}\,\{\, r \in \mathbb{Q} \,:\, \Pr_s\big(\Box\Diamond(\mathit{ratio} > r) \wedge \varphi\big) \,=\, 1 \,\}$$

$$\textbf{sup}\,\{\, r \in \mathbb{Q} \,:\, \Pr_s\big(\Box\Diamond(\mathit{ratio} > r) \wedge \varphi\big) \,>\, q \,\}$$

universal almost-sure quantiles in MDPs:

$$\textbf{sup}\,\{\, r \in \mathbb{Q} \,:\, \Pr_s^{\min}\big(\Box(\mathit{ratio} > r) \wedge \varphi\big) \,=\, 1 \,\}$$

Outline

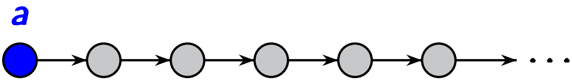
- weighted Markov decision processes
- mean-payoff and long-run ratios
- expected accumulated weights
- conditional expected accumulated rewards
- weight-bounded reachability and quantiles
- LTL with weight assertions
- conclusions

LTL: linear temporal logic

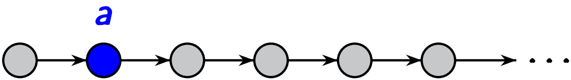
$$\varphi \quad ::= \quad \mathit{true} \;\big|\; a \;\big|\; \varphi_1 \wedge \varphi_2 \;\big|\; \neg\varphi \;\big|\; \bigcirc\varphi \;\big|\; \varphi_1 \,\mathbf{U}\, \varphi_2$$

where $a \in AP$ $\quad \bigcirc \mathrel{\widehat{=}}$ next $\quad \mathbf{U} \mathrel{\widehat{=}}$ until

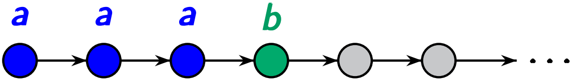
atomic proposition
$a \in AP$



next operator
$\bigcirc a$



until operator
$a \,\mathbf{U}\, b$

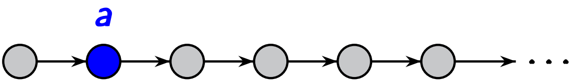


LTL: linear temporal logic

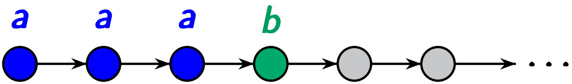
$$\varphi \quad ::= \quad \mathit{true} \mid a \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \bigcirc\varphi \mid \varphi_1 \,\mathbf{U}\, \varphi_2$$

$\Diamond\varphi$	$\stackrel{\text{def}}{=}$	$\mathit{true}\,\mathbf{U}\,\varphi$	eventually
$\Box\varphi$	$\stackrel{\text{def}}{=}$	$\neg\Diamond\neg\varphi$	always

next operator
$\bigcirc a$



until operator
$a\,\mathbf{U}\,b$



LTL: semantics over path-position pairs

for $\pi = s_0\, s_1\, s_2 \ldots \in S^\omega$ and $k \in \mathbb{N}$:

$$
\begin{array}{lll}
(\pi, k) \models \mathit{true} & & \\
(\pi, k) \models a & \text{iff} & s_k \models a \\
(\pi, k) \models \varphi_1 \wedge \varphi_2 & \text{iff} & (\pi, k) \models \varphi_1 \text{ and } (\pi, k) \models \varphi_2 \\
(\pi, k) \models \neg\varphi & \text{iff} & (\pi, k) \not\models \varphi \\
(\pi, k) \models \bigcirc\varphi & \text{iff} & (\pi, k+1) \models \varphi \\
(\pi, k) \models \varphi_1 \,\mathsf{U}\, \varphi_2 & \text{iff} & \text{there exists } j \geqslant k \text{ such that} \\
 & & \quad (\pi, j) \models \varphi_2 \quad \text{and} \\
 & & \quad (\pi, i) \models \varphi_1 \quad \text{for } k \leqslant i < j
\end{array}
$$

LTL with monitored weight assertions

extends LTL by weight assertions

$$\underbrace{\Diamond^{\mathcal{A}}\ \mathit{wcon}}_{\text{future formula}} \qquad\qquad \underbrace{\Diamond^{\mathcal{A}}\ \mathit{wcon}}_{\text{past formula}}$$

[Baier/Klein/Klüppelholz/Wunderlich'14]

LTL with monitored weight assertions

extends LTL by weight assertions

$$\underbrace{\oplus^{\mathcal{A}}\, wcon}_{\text{future formula}} \qquad \underbrace{\ominus^{\mathcal{A}}\, wcon}_{\text{past formula}}$$

- $\mathcal{A}$ is a DFA, called weight monitor
- $wcon$ is a weight constraint for finite paths

DFA: deterministic finite automaton

LTL with monitored weight assertions

extends LTL by weight assertions

$$\underbrace{\diamondplus^{\mathcal{A}}\, \mathit{wcon}}_{\text{future formula}} \qquad \underbrace{\diamondminus^{\mathcal{A}}\, \mathit{wcon}}_{\text{past formula}}$$

- $\mathcal{A}$ is a DFA, called weight monitor
- wcon is a weight constraint for finite paths

---

interpretation over path-position pairs:

$$(\pi, k) \models \diamondplus^{\mathcal{A}}\, \mathit{wcon}$$

$$\text{iff} \quad \exists \ell > k.\ \mathit{trace}(\pi[k \ldots \ell]) \in \mathcal{L}(\mathcal{A}) \wedge \pi[k \ldots \ell] \models \mathit{wcon}$$

LTL with monitored weight assertions

extends LTL by weight assertions

$$\underbrace{\diamondplus^{\mathcal{A}}\ \mathit{wcon}}_{\text{future formula}} \qquad \underbrace{\diamondminus^{\mathcal{A}}\ \mathit{wcon}}_{\text{past formula}}$$

- $\mathcal{A}$ is a DFA, called weight monitor
- wcon is a weight constraint for finite paths

---

interpretation over path-position pairs:

$$(\pi, k) \models \diamondminus^{\mathcal{A}}\ \mathit{wcon}$$

$$\text{iff} \quad \exists \ell < k.\ \mathit{trace}(\pi[\ell \ldots k]) \in \mathcal{L}(\mathcal{A}) \wedge \pi[\ell \ldots k] \models \mathit{wcon}$$

LTL with monitored weight assertions

extends LTL by weight assertions

$$\underbrace{\oplus^{\mathcal{A}}\ \mathit{wcon}}_{\text{future formula}} \qquad \underbrace{\ominus^{\mathcal{A}}\ \mathit{wcon}}_{\text{past formula}}$$

- $\mathcal{A}$ is a DFA, called weight monitor
- wcon is a conjunction of basic weight constraints

$$\sum_{i=1}^{d} q_i \cdot \mathit{wgt}_i \quad \bowtie \quad r \qquad \text{where } q_i, r \in \mathbb{Q}$$

symbols for the accumulated values of weight functions in semantic structures

comparison operator $\bowtie\ \in \{\leqslant, <, \geqslant, >\}$

LTL with monitored weight assertions

extends LTL by weight assertions

$$\underbrace{\oplus^{\mathcal{A}}\, wcon}_{\text{future formula}} \qquad \underbrace{\ominus^{\mathcal{A}}\, wcon}_{\text{past formula}}$$

- $\mathcal{A}$ is a DFA, called weight monitor
- $wcon$ is a conjunction of basic weight constraints

$$\sum_{i=1}^{d} q_i \cdot wgt_i \quad \bowtie \quad r \qquad \text{where } q_i, r \in \mathbb{Q}$$

cost-utility constraints as basic weight constraints:

$$\frac{cost}{util} > \rho \quad \text{iff} \quad cost - \rho \cdot util > 0$$

LTL with monitored weight assertions

extends LTL by weight assertions

$$\underbrace{\oplus^{\mathcal{A}}\ wcon}_{\text{future formula}} \qquad\qquad \underbrace{\ominus^{\mathcal{A}}\ wcon}_{\text{past formula}}$$

- $\mathcal{A}$ is a DFA, called weight monitor
- $wcon$ is a conjunction of basic weight constraints

$$\sum_{i=1}^{d} q_i \cdot wgt_i \quad \bowtie \quad r \qquad\qquad \text{where } q_i, r \in \mathbb{Q}$$

weight-bounded until as a weight assertion:

$$a\,\mathsf{U}^{\leqslant r}\,b \quad \equiv \quad \oplus^{a\,\mathsf{U}\,b}(wgt \leqslant r)$$

LTL with monitored weight assertions

extends LTL by weight assertions

$$\underbrace{\Diamond\!\!\!\!+^{\mathcal{A}}\ \mathit{wcon}}_{\text{future formula}} \qquad\qquad \underbrace{\Diamond\!\!\!\!-^{\mathcal{A}}\ \mathit{wcon}}_{\text{past formula}}$$

- $\mathcal{A}$ is a DFA, called weight monitor
- wcon is a conjunction of basic weight constraints

$$\sum_{i=1}^{d} q_i \cdot \mathit{wgt}_i \quad \bowtie \quad r \qquad\qquad \text{where } q_i, r \in \mathbb{Q}$$

prefix-accumulation assertions

$$\Diamond\!\!\!\!+^{\mathit{first}=\mathit{init}}\ \mathit{wcon}$$

Boker/Chatterjee/
Henzinger/Kupferman'11

LTL with monitored weight assertions

extends LTL by weight assertions

$$\underbrace{\oplus^{\mathcal{A}}\ \mathit{wcon}}_{\text{future formula}} \qquad\qquad \underbrace{\ominus^{\mathcal{A}}\ \mathit{wcon}}_{\text{past formula}}$$

- $\mathcal{A}$ is a DFA, called weight monitor
- wcon is a conjunction of basic weight constraints or average-weight constraints

$$\sum_{i=1}^{d} q_i \cdot \underbrace{\mathit{avg}[\mathit{wgt}_i]}_{\substack{\text{average of the}\\ \text{accumulated values}}} \ \bowtie\ r \qquad \text{where } q_i, r \in \mathbb{Q}$$

LTL with monitored weight assertions

extends LTL by weight assertions

$$\underbrace{\oplus^{\mathcal{A}} \, wcon}_{\text{future formula}} \qquad \underbrace{\ominus^{\mathcal{A}} \, wcon}_{\text{past formula}}$$

- $\mathcal{A}$ is a DFA, called weight monitor
- $wcon$ is a conjunction of basic weight constraints or average-weight constraints

$$\sum_{i=1}^{d} q_i \cdot avg[wgt_i] \;\bowtie\; r \qquad \text{where } q_i, r \in \mathbb{Q}$$

$$avg[wgt]\left(s_0 \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_n} s_n\right) \;=\; \frac{1}{n} \cdot \sum_{j=0}^{n-1} wgt\left(s_j, \alpha_{j+1}\right)$$

LTL with monitored weight assertions

extends LTL by weight assertions

$$\underbrace{\oplus^{\mathcal{A}}\, wcon}_{\text{future formula}} \qquad \underbrace{\ominus^{\mathcal{A}}\, wcon}_{\text{past formula}}$$

- $\mathcal{A}$ is a DFA, called weight monitor
- $wcon$ is a conjunction of basic weight constraints or average-weight constraints

$$\sum_{i=1}^{d} q_i \cdot avg[wgt_i] \;\bowtie\; r \qquad \text{where } q_i, r \in \mathbb{Q}$$

mean-payoff assertions, e.g.,

$$\Diamond\Box\oplus^{first=init}(avg[wgt] \geqslant r)$$

Boker/Chat./Henz./Kup.'11

Tomita/Hiu./Hag./Yon.'12

LTL with monitored weight assertions

extends LTL by weight assertions

$$\underbrace{\Diamond\!\!\!\!+^{\mathcal{A}}\ \mathit{wcon}}_{\text{future formula}} \qquad\qquad \underbrace{\ominus^{\mathcal{A}}\ \mathit{wcon}}_{\text{past formula}}$$

- $\mathcal{A}$ is a DFA, called weight monitor
- wcon is a conjunction of basic weight constraints or average-weight constraints

$$\sum_{i=1}^{d} q_i \cdot \mathit{avg}[\mathit{wgt}_i] \quad \bowtie \quad r \qquad\qquad \text{where } q_i, r \in \mathbb{Q}$$

fixed-window assertions, e.g.

$$\Diamond\Box\Diamond\!\!\!\!+^{\mathit{length}=\ell}\,(\mathit{avg}[\mathit{wgt}] \geqslant r)$$

Chatterjee/Doyen/Randour/Raskin'13

LTL with weight assertions

extends LTL by weight assertions

$$\oplus^{\mathcal{A}}(\varphi_1; \textit{wcon}; \varphi_2) \qquad \ominus^{\mathcal{A}}(\varphi_1; \textit{wcon}; \varphi_2)$$

generalized weight assertions

- $\mathcal{A}$ is a DFA, called weight monitor
- *wcon* is a conjunction of basic weight constraints or average-weight constraints
- φ_1, φ_2 pre- and postconditions (arbitrary formulas)

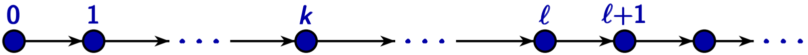
LTL with weight assertions

extends LTL by weight assertions

$$\Diamond\!\!\!\!\!\!+^{\mathcal{A}}(\varphi_1; \mathit{wcon}; \varphi_2) \qquad \Diamond\!\!\!\!\!\!-^{\mathcal{A}}(\varphi_1; \mathit{wcon}; \varphi_2)$$

generalized weight assertions

- $\mathcal{A}$ is a DFA, called weight monitor
- *wcon* is a conjunction of basic weight constraints or average-weight constraints
- φ_1, φ_2 pre- and postconditions (arbitrary formulas)



path π

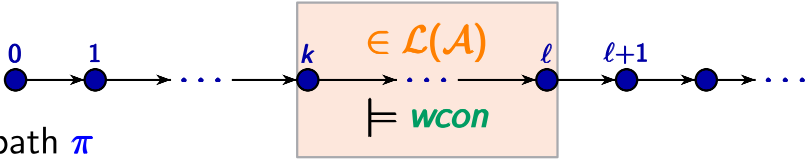
LTL with weight assertions

extends LTL by weight assertions

$$\diamondplus^{\mathcal{A}}(\varphi_1; \mathit{wcon}; \varphi_2) \qquad \diamondminus^{\mathcal{A}}(\varphi_1; \mathit{wcon}; \varphi_2)$$

generalized weight assertions

- $\mathcal{A}$ is a DFA, called weight monitor
- *wcon* is a conjunction of basic weight constraints or average-weight constraints
- φ_1, φ_2 pre- and postconditions (arbitrary formulas)



path π

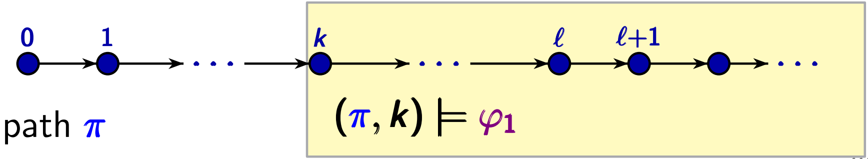
LTL with weight assertions

extends LTL by weight assertions

$$\diamondplus^{\mathcal{A}}(\varphi_1; \mathit{wcon}; \varphi_2) \qquad \diamondminus^{\mathcal{A}}(\varphi_1; \mathit{wcon}; \varphi_2)$$

generalized weight assertions

- $\mathcal{A}$ is a DFA, called weight monitor
- wcon is a conjunction of basic weight constraints or average-weight constraints
- φ_1, φ_2 pre- and postconditions (arbitrary formulas)



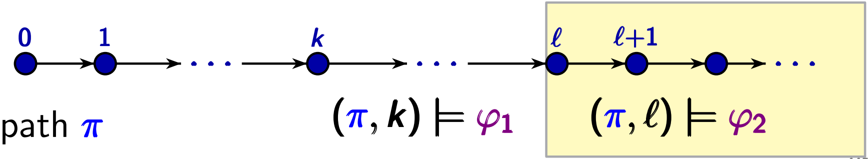
LTL with weight assertions

extends LTL by weight assertions

$$\diamondplus^{\mathcal{A}}(\varphi_1; \mathit{wcon}; \varphi_2) \qquad \diamondminus^{\mathcal{A}}(\varphi_1; \mathit{wcon}; \varphi_2)$$

generalized weight assertions

- $\mathcal{A}$ is a DFA, called weight monitor
- wcon is a conjunction of basic weight constraints or average-weight constraints
- φ_1, φ_2 pre- and postconditions (arbitrary formulas)



LTL with weight assertions

extends LTL by weight assertions

$$\diamondplus^{\mathcal{A}}(\varphi_1; \mathit{wcon}; \varphi_2) \qquad \diamondminus^{\mathcal{A}}(\varphi_1; \mathit{wcon}; \varphi_2)$$

generalized weight assertions

- $\mathcal{A}$ is a DFA, called weight monitor
- *wcon* is a conjunction of basic weight constraints or average-weight constraints
- φ_1, φ_2 pre- and postconditions (arbitrary formulas)

weight-bounded PSL-like suffix conjunction:

$$\mathit{regexpr} \overset{\mathit{wcon}}{\diamond\!\!\longrightarrow} \varphi \quad \equiv \quad \diamondplus^{\mathcal{A}}(\mathit{true}; \mathit{wcon}; \varphi)$$

LTL with weight assertions

extends LTL by weight assertions

$$\oplus^{\mathcal{A}}(\varphi_1; \textit{wcon}; \varphi_2) \qquad \ominus^{\mathcal{A}}(\varphi_1; \textit{wcon}; \varphi_2)$$

generalized weight assertions

- $\mathcal{A} \in \mathcal{C}$ for some class $\mathcal{C}$ of DFA (weight monitors)
- *wcon* is a conjunction of basic weight constraints or average-weight constraints
- φ_1, φ_2 pre- and postconditions (arbitrary formulas)

LTL($\oplus, \ominus : \mathcal{C}$): full logic with weight monitors in $\mathcal{A} \in \mathcal{C}$

PL($\oplus : \mathcal{C}$): prop. logic with weight assertions $\oplus^{\mathcal{A}}\, \textit{wcon}$

Model checking problems

Given formula φ and state s of a weighted structure.

weighted MDP (WMDP):

PMC-problem: compute $\mathrm{Pr}_s^{\max}(\varphi)$

qualitative decision probl.: does $\mathrm{Pr}_s^{\max}(\varphi) > 0$ hold ?

weighted Markov chain (WMC):

PMC-problem: compute $\mathrm{Pr}_s(\varphi)$

qualitative decision problem: does $\mathrm{Pr}_s(\varphi) > 0$ hold ?

weighted transition system (WTS):

does $s \models \exists\varphi$ hold ?

Model checking problems

	general case
$\mathsf{LTL}(\diamondplus, \diamondminus : \mathsf{All})$	
$\mathsf{PL}(\diamondplus : \mathsf{All})$	

Model checking problems

	general case
LTL($\oplus, \ominus$: All)	undecidable
PL($\oplus$: All)	

temporal logics with prefix accumulation over WTS

- undecidability result for LTL
- decidability for the $\exists\Diamond$-fragment of CTL

[Boker/Chatterjee/Henzinger/Kupferman'11]

Model checking problems

	general case
LTL($\oplus$, $\ominus$: All)	undecidable
PL($\oplus$: All) PL($\oplus$: Reach)	undecidable

[Baier/Klein/Klüppelholz/Wunderlich'14]

temporal logics with prefix accumulation over WTS

- undecidability result for LTL
- decidability for the $\exists\Diamond$-fragment of CTL

[Boker/Chatterjee/Henzinger/Kupferman'11]

Model checking problems

	general case
LTL($\oplus, \ominus$: All) LTL($\oplus, \ominus$: Reach)	undecidable
PL($\oplus$: All) PL($\oplus$: Reach)	undecidable

[Baier/Klein/Klüppelholz/Wunderlich'14]

temporal logics with prefix accumulation over WTS

- undecidability result for LTL
- decidability for the $\exists\Diamond$-fragment of CTL

[Boker/Chatterjee/Henzinger/Kupferman'11]

Model checking problems

	general case	
LTL($\oplus, \ominus$: All) LTL($\oplus, \ominus$: Reach)	undecidable	
PL($\oplus$: All) PL($\oplus$: Reach)	undecidable	$\}$ even for models with non-negative weights

simulation of a weight function wgt by two non-negative weight fct. wgt^+ and wgt^-

s.t. $\mathit{wgt} = \mathit{wgt}^+ - \mathit{wgt}^-$

Model checking problems

	general case	non-negative weight fct. simple weight constr.
LTL($\oplus, \ominus$: All) LTL($\oplus, \ominus$: Reach)	undecidable	
PL($\oplus$: All) PL($\oplus$: Reach)	undecidable	even for models with non-negative weights

basic weight constraint

$$\sum_{j=1}^{d} q_j \cdot wgt_j \bowtie r$$

simple basic weight constraint

$$wgt_j \bowtie r$$

Model checking problems

	general case	non-negative weight fct. simple weight constr.
$\mathrm{LTL}(\diamondplus, \diamondminus : \mathrm{All})$ $\mathrm{LTL}(\diamondplus, \diamondminus : \mathrm{Reach})$	undecidable	decidable
$\mathrm{PL}(\diamondplus : \mathrm{All})$ $\mathrm{PL}(\diamondplus : \mathrm{Reach})$	undecidable	decidable

basic weight constraint

$$\sum_{j=1}^{d} q_j \cdot wgt_j \bowtie r$$

simple basic weight constraint

$$wgt_j \bowtie r$$

Model checking problems

	general case	non-negative weight fct. simple weight constr.
$\mathrm{LTL}(\diamondplus, \diamondminus : \mathrm{All})$ $\mathrm{LTL}(\diamondplus, \diamondminus : \mathrm{Reach})$	undecidable	decidable
$\mathrm{PL}(\diamondplus : \mathrm{All})$ $\mathrm{PL}(\diamondplus : \mathrm{Reach})$	undecidable	decidable
$\mathrm{LTL}(\diamondplus, \diamondminus : \mathrm{Acyclic})$ $\mathrm{LTL}(\diamondplus, \diamondminus : \mathrm{Window})$		

acyclic weight monitors: finite accepted language

window monitors: specify length restrictions on the accepted words

Model checking problems

	general case	non-negative weight fct. simple weight constr.
LTL($\diamondplus, \diamondminus$: All) **LTL($\diamondplus, \diamondminus$: Reach)**	undecidable	decidable
PL($\diamondplus$: All) **PL($\diamondplus$: Reach)**	undecidable	decidable
LTL($\diamondplus, \diamondminus$: Acyclic) **LTL($\diamondplus, \diamondminus$: Window)**	WMDP: WMC, WTS:	2EXP-complete PSPACE-complete

acyclic weight monitors: finite accepted language

window monitors: specify length restrictions on the accepted words

Model checking problems

	general case	non-negative weight fct. simple weight constr.
LTL($\oplus$, $\ominus$: All) **LTL($\oplus$, $\ominus$: Reach)**	undecidable	decidable
PL($\oplus$: All) **PL($\oplus$: Reach)**	undecidable	decidable
LTL($\oplus$, $\ominus$: Acyclic) **LTL($\oplus$, $\ominus$: Window)**	WMDP: WMC, WTS:	2EXP-complete PSPACE-complete
PL($\oplus$: Acyclic) **PL($\oplus$: Window)**	NP-complete	NP-complete

Model checking problems

	general case	non-negative weight fct. simple weight constr.
LTL($\diamondplus, \diamondminus$: All) LTL($\diamondplus, \diamondminus$: Reach)	undecidable	decidable
PL($\diamondplus$: All) PL($\diamondplus$: Reach)	undecidable	decidable

LTL($\diamondplus, \diamondminus$: Acyclic)
LTL($\diamondplus, \diamondminus$: Window)

PL($\diamondplus$: Acyclic)
PL($\diamondplus$: Window)

computation of $\Pr_s^{\max}(\varphi)$ via reduction to the PMC-problem for standard LTL

Model checking problems

	general case	non-negative weight fct. simple weight constr.
LTL($\diamondplus, \diamondminus$: All) LTL($\diamondplus, \diamondminus$: Reach)	undecidable	decidable
PL($\diamondplus$: All) PL($\diamondplus$: Reach)	undecidable	decidable

LTL($\diamondplus, \diamondminus$: Acyclic) LTL($\diamondplus, \diamondminus$: Window)	states of the new MDP are tuples $\langle s, \underbrace{f_1, \ldots, f_k}_{} \rangle$ functions that track the information about the monitor states and accumulated weights for each weight constraint
PL($\diamondplus$: Acyclic) PL($\diamondplus$: Window)	

Model checking problems

	general case	non-negative weight fct. simple weight constr.
LTL($\diamondplus, \diamondminus$: **All**) **LTL**($\diamondplus, \diamondminus$: **Reach**)	undecidable	decidable
PL($\diamondplus$: **All**) **PL**($\diamondplus$: **Reach**)	undecidable	decidable

decidability crucially relies on the monotonicity of accumulated weights along the prefixes of paths

Outline

- weighted Markov decision processes
- mean-payoff and long-run ratios
- expected accumulated weights
- conditional expected accumulated rewards
- weight-bounded reachability and quantiles
- LTL with weight assertions
- conclusions

Conclusion

reasoning about “weight objectives” in probabilistic models often more difficult than it seems

Conclusion

reasoning about "weight objectives" in probabilistic models often more difficult than it seems

- polynomial-time algorithms for expectations
  accumulated weights, mean-payoff & other long-run objectives
- conditional expected accumulated weights
  so far: exponential-time algorithm for non-negative weights
- probabilities for weight-bounded properties
  * algorithms/tools for non-negative weights in MC and MDPs
    ... use monotocity, computationally hard
  * arbitrary weights: difficult, even for Markov chains

THANK YOU