Bi-Objective Lexicographic Optimization in Markov Decision Processes with Related Objectives

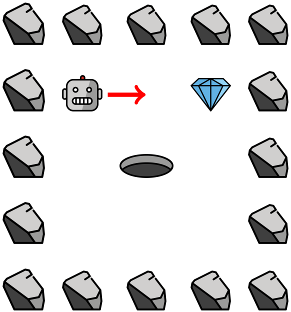
Damien Busatto-Gaston[1] Debraj Chakraborty[2] Anirban Majumdar[3] Sayan Mukherjee[3]
Guillermo A. Pérez[4] Jean-François Raskin[3]

1-Université Paris-Est Créteil, LACL, France
2-Masaryk University, Czech Republic
3-Université Libre de Bruxelles, Belgium
4-University of Antwerp, Belgium

Automated Technology for Verification and Analysis
Singapore, 2023

Example: Frozen lake

Example: Frozen lake



Example: Frozen lake

Example: Frozen lake

-  slips to different direction with some probability

Example: Frozen lake

-  slips to different direction with some probability

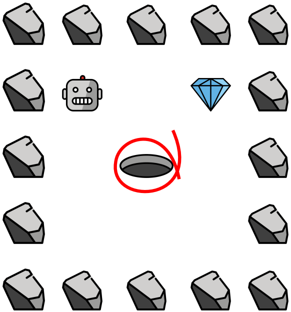
Example: Frozen lake

-  slips to different direction with some probability
-  cannot move through a 

Example: Frozen lake

- slips to different direction with some probability
- cannot move through a

Example: Frozen lake

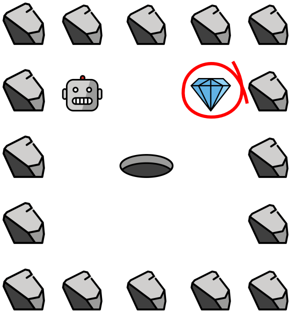


- slips to different direction with some probability
- cannot move through a
- cannot move anymore if it falls in a

Example: Frozen lake

-  slips to different direction with some probability
-  cannot move through a 
-  cannot move anymore if it falls in a 

Example: Frozen lake



- slips to different direction with some probability
- cannot move through a
- cannot move anymore if it falls in a
- wants to reach

Example: Frozen lake

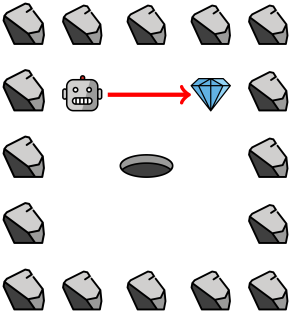
- slips to different direction with some probability
- cannot move through a
- cannot move anymore if it falls in a
- wants to reach

Example: Frozen lake

- slips to different direction with some probability
- cannot move through a
- cannot move anymore if it falls in a
- wants to reach

- Maximize $\mathbb{P}(\ \rightsquigarrow\ )$

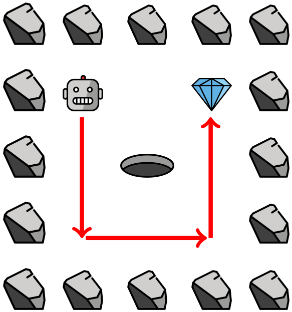
Example: Frozen lake



- slips to different direction with some probability
- cannot move through a
- cannot move anymore if it falls in a
- wants to reach

- Maximize $\mathbb{P}(\rightsquigarrow)$

Example: Frozen lake



- slips to different direction with some probability
- cannot move through a
- cannot move anymore if it falls in a
- wants to reach

- Maximize $\mathbb{P}(\;\rightsquigarrow\;)$

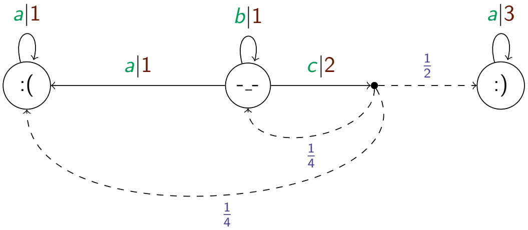
Example: Frozen lake

- slips to different direction with some probability
- cannot move through a
- cannot move anymore if it falls in a
- wants to reach

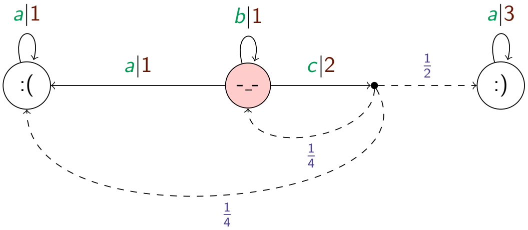
- Maximize $\mathbb{P}(\rightsquigarrow)$
- Reach as fast as possible

Markov Decision Process

Markov Decision Process

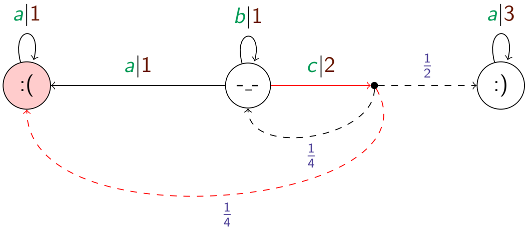


Markov Decision Process



- Path : ⊝ Reward: 0

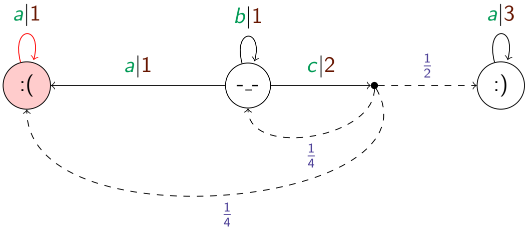
Markov Decision Process



- Path : $\text{-\_-} \xrightarrow{c} \text{:(}$

Reward: 2

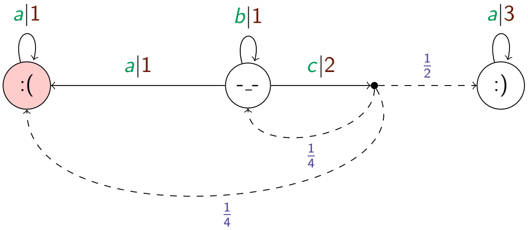
Markov Decision Process



- Path : -_- $\xrightarrow{c}$:($\xrightarrow{a}$:(

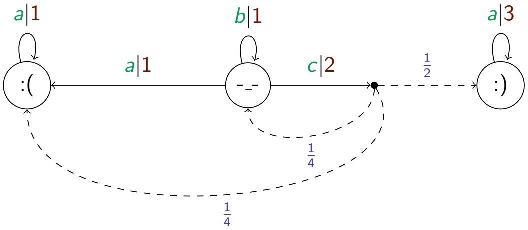
Reward: 3

Markov Decision Process



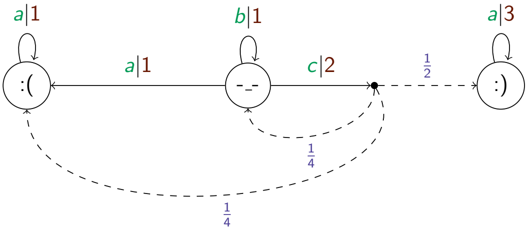
- Path : $\ominus \xrightarrow{c} \odot \xrightarrow{a} \odot \ldots$ Reward: 3
- Mean-payoff (MP): Average reward at limit : $\rho \mapsto \lim_{n\to\infty} \frac{1}{n}\text{Reward}(\rho_n)$

Markov Decision Process



- Strategy : Paths → Actions

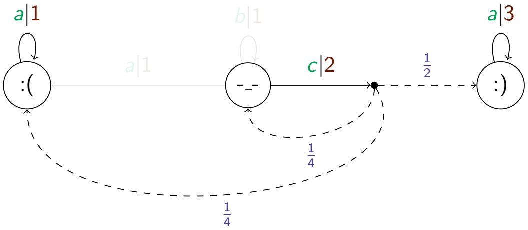
Markov Decision Process



- Strategy : Paths $\rightarrow$ Actions

$$\{\text{-\_-} \mapsto c, \text{:)} \mapsto a, \text{:(} \mapsto a\}$$

Markov Decision Process



- Strategy : Paths $\to$ Actions

$$\{\text{-\_-} \mapsto c, \text{:)} \mapsto a, \text{:(} \mapsto a\}$$

- Fixing a strategy creates a Markov chain

Objectives

Find a strategy σ that optimizes

Objectives

Find a strategy σ that optimizes

- expected mean payoff $\mathbb{E}_\sigma(\mathsf{MP})$

Objectives

Find a strategy σ that optimizes

- expected mean payoff $\mathbb{E}_\sigma(\mathrm{MP})$
- reachability : probability to reach the target $T \subseteq S$

Objectives

Find a strategy σ that optimizes

- expected mean payoff $\mathbb{E}_\sigma(\mathsf{MP})$
- reachability : probability to reach the target $T \subseteq S$

$$\mathbb{P}_\sigma(s_0 \models \Diamond T)$$

Objectives

Find a strategy σ that optimizes

- expected mean payoff $\mathbb{E}_\sigma(\mathsf{MP})$
- reachability : probability to reach the target $T \subseteq S$

$$\mathbb{P}_\sigma(s_0 \models \Diamond T)$$

- Safety : probability to avoid the states $T \subseteq S$

Objectives

Find a strategy σ that optimizes

- expected mean payoff $\mathbb{E}_\sigma(\mathsf{MP})$
- reachability : probability to reach the target $T \subseteq S$

$$\mathbb{P}_\sigma(s_0 \models \Diamond T)$$

- Safety : probability to avoid the states $T \subseteq S$

$$\mathbb{P}_\sigma(s_0 \models \Box \neg T)$$

Objectives

Find a strategy σ that optimizes

- expected mean payoff $\mathbb{E}_\sigma(\mathrm{MP})$
- reachability : probability to reach the target $T \subseteq S$

$$\mathbb{P}_\sigma(s_0 \models \Diamond T)$$

- Safety : probability to avoid the states $T \subseteq S$

$$\mathbb{P}_\sigma(s_0 \models \Box \neg T)$$

- Lexicographic multi-objective :

Objectives

Find a strategy σ that optimizes

- expected mean payoff $\mathbb{E}_\sigma(\mathrm{MP})$
- reachability : probability to reach the target $T \subseteq S$

$$\mathbb{P}_\sigma(s_0 \models \Diamond T)$$

- Safety : probability to avoid the states $T \subseteq S$

$$\mathbb{P}_\sigma(s_0 \models \Box \neg T)$$

- Lexicographic multi-objective :
  - Optimize objective 1

Objectives

Find a strategy σ that optimizes

- expected mean payoff $\mathbb{E}_\sigma(\mathrm{MP})$
- reachability : probability to reach the target $T \subseteq S$

$$\mathbb{P}_\sigma(s_0 \models \Diamond T)$$

- Safety : probability to avoid the states $T \subseteq S$

$$\mathbb{P}_\sigma(s_0 \models \Box \neg T)$$

- Lexicographic multi-objective :
  - Optimize objective 1
  - Optimize objective 2 among strategies optimizing objective 1

Objectives

Find a strategy σ that optimizes

- expected mean payoff $\mathbb{E}_\sigma(\mathsf{MP})$
- reachability : probability to reach the target $T \subseteq S$

$$\mathbb{P}_\sigma(s_0 \models \Diamond T)$$

- Safety : probability to avoid the states $T \subseteq S$

$$\mathbb{P}_\sigma(s_0 \models \Box \neg T)$$

- Lexicographic multi-objective :
  - Optimize objective 1
  - Optimize objective 2 among strategies optimizing objective 1
  - ...

Problem statement

Find a strategy σ that

Problem statement

Find a strategy σ that

- Maximize the probability to reach the target $T \subseteq S$

Problem statement

Find a strategy σ that

- Maximize the probability to reach the target $T \subseteq S$:

$$\sigma \in \arg\max_{\sigma'} \mathbb{P}_{\sigma'}(s_0 \models \Diamond T)$$

Problem statement

Find a strategy σ that

- Maximize the probability to reach the target $T \subseteq S$:

$$\sigma \in \arg\max_{\sigma'} \mathbb{P}_{\sigma'}(s_0 \models \Diamond T)$$

- Reach the target as soon as possible

Problem statement

Find a strategy σ that

- Maximize the probability to reach the target $T \subseteq S$:

$$\sigma \in \arg\max_{\sigma'} \mathbb{P}_{\sigma'}(s_0 \models \Diamond T) = \Sigma_{\Diamond T}$$

- Reach the target as soon as possible

Problem statement

Find a strategy σ that

- Maximize the probability to reach the target $T \subseteq S$:

$$\sigma \in \arg\max_{\sigma'} \mathbb{P}_{\sigma'}(s_0 \models \Diamond T) = \Sigma_{\Diamond T}$$

- Reach the target as soon as possible:
  - Among $\sigma \in \Sigma_{\Diamond T}$, expected conditional distance to target is minimum

Problem statement

Find a strategy σ that

- Maximize the probability to reach the target $T \subseteq S$:

$$\sigma \in \arg\max_{\sigma'} \mathbb{P}_{\sigma'}(s_0 \models \Diamond T) = \Sigma_{\Diamond T}$$

- Reach the target as soon as possible:
  - Among $\sigma \in \Sigma_{\Diamond T}$, expected conditional distance to target is minimum:

$$\sigma \in \arg\min_{\sigma' \in \Sigma_{\Diamond T}} \mathbb{E}_{\sigma'}(\text{len}_T \mid \Diamond T)$$

Problem statement

Find a strategy σ that

- Maximize the probability to reach the target $T \subseteq S$:

$$\sigma \in \arg\max_{\sigma'} \mathbb{P}_{\sigma'}(s_0 \models \Diamond T) = \Sigma_{\Diamond T}$$

- Reach the target as soon as possible:
  - Among $\sigma \in \Sigma_{\Diamond T}$, expected conditional distance to target is minimum:

$$\sigma \in \arg\min_{\sigma' \in \Sigma_{\Diamond T}} \mathbb{E}_{\sigma'}(\mathsf{len}_T \mid \Diamond T)$$

$\mathsf{len}_T : \rho \mapsto$ length of the shortest prefix of ρ reaching T

Algorithm

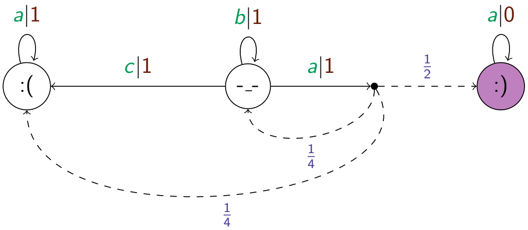
Find a strategy that

- maximizes the probability to reach the target,
- minimizes the conditional expected length to the target

Algorithm

Find a strategy that

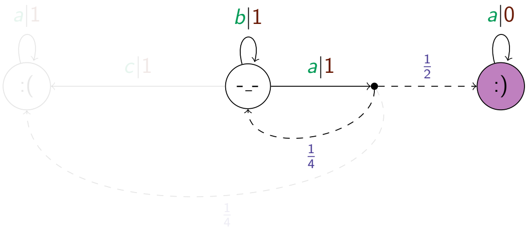
- maximizes the probability to reach the target,
- minimizes the conditional expected length to the target



Algorithm

Find a strategy that

- maximizes the probability to reach the target,
- minimizes the conditional expected length to the target

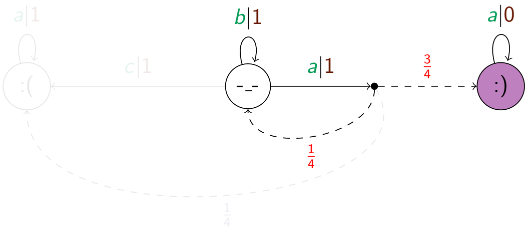


- Remove all "bad" actions and states

Algorithm

Find a strategy that

- maximizes the probability to reach the target,
- minimizes the conditional expected length to the target

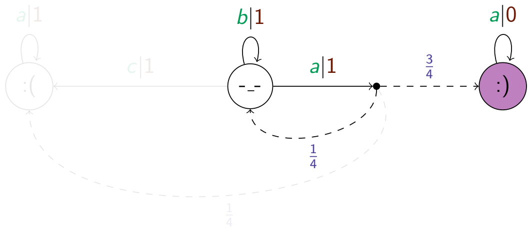


- Remove all "bad" actions and states
- "Redistribute" the probabilities

Algorithm

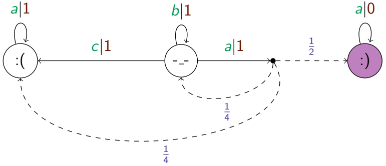
Find a strategy that

- maximizes the probability to reach the target,
- minimizes the conditional expected length to the target

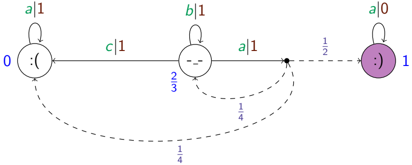


- Remove all "bad" actions and states
- "Redistribute" the probabilities
- Find strategy that minimizes the ~~conditional~~ expected length to the target

Algorithm

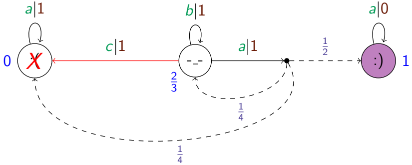


Algorithm



- Calculate values: $\mathsf{Val}(s) = \max_\sigma \mathbb{P}_\sigma(s \models \Diamond \text{:)})$

Algorithm

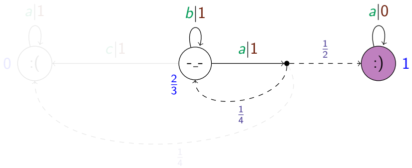


- Calculate values: $\mathrm{Val}(s) = \max_\sigma \mathbb{P}_\sigma(s \models \Diamond \text{☺})$
- Remove bad action a from state s s.t.

$$\mathrm{Val}(s) \neq \sum_{s'} P(s, a, s') \cdot \mathrm{Val}(s')$$

- Remove bad states s such that $\mathrm{Val}(s) = 0$

Algorithm

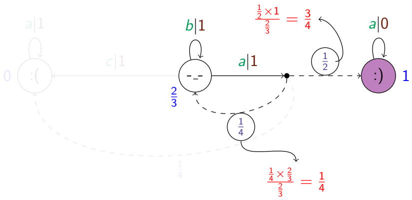


- Calculate values: $\mathrm{Val}(s) = \max_\sigma \mathbb{P}_\sigma(s \models \Diamond \text{:)})$
- Remove bad action a from state s s.t.

$$\mathrm{Val}(s) \neq \sum_{s'} P(s, a, s') \cdot \mathrm{Val}(s')$$

- Remove bad states s such that $\mathrm{Val}(s) = 0$

Algorithm



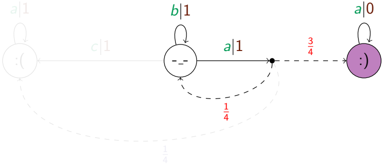
- Calculate values: $\mathsf{Val}(s) = \max_\sigma \mathbb{P}_\sigma(s \models \Diamond \text{:)})$
- Remove bad action a from state s s.t.

$$\mathsf{Val}(s) \neq \sum_{s'} P(s,a,s') \cdot \mathsf{Val}(s')$$

- Remove bad states s such that $\mathsf{Val}(s) = 0$
- Redistribute probabilities:

$$P'(s,a,s') = P(s,a,s') \cdot \frac{\mathsf{Val}(s')}{\mathsf{Val}(s)}$$

Algorithm



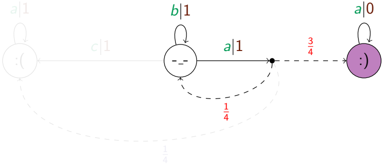
- Calculate values: $\mathrm{Val}(s) = \max_\sigma \mathbb{P}_\sigma(s \models \Diamond \text{:)})$
- Remove bad action a from state s s.t.

$$\mathrm{Val}(s) \neq \sum_{s'} P(s,a,s') \cdot \mathrm{Val}(s')$$

- Remove bad states s such that $\mathrm{Val}(s) = 0$
- Redistribute probabilities:

$$P'(s,a,s') = P(s,a,s') \cdot \frac{\mathrm{Val}(s')}{\mathrm{Val}(s)}$$

Algorithm



- Calculate values: $\mathrm{Val}(s) = \max_\sigma \mathbb{P}_\sigma(s \models \Diamond \text{:)})$
- Remove bad action a from state s s.t.

$$\mathrm{Val}(s) \neq \sum_{s'} P(s, a, s') \cdot \mathrm{Val}(s')$$

- Remove bad states s such that $\mathrm{Val}(s) = 0$
- Redistribute probabilities:

$$P'(s, a, s') = P(s, a, s') \cdot \frac{\mathrm{Val}(s')}{\mathrm{Val}(s)}$$

- Find the strategy $\arg\min_{\sigma'} \mathbb{E}_{\sigma'}(\mathrm{len}_{\text{:)}})$ in M'

Explanation

Explanation

- Change the probability measure by redistributing probabilities.

Explanation

- Change the probability measure by redistributing probabilities.

$$\mathbb{P}'_\sigma(\cdot) = \mathbb{P}_\sigma(\cdot \mid \Diamond \text{☺})$$

Explanation

- Change the probability measure by redistributing probabilities.

$$\mathbb{P}'_\sigma(\cdot) = \mathbb{P}_\sigma(\cdot \mid \Diamond \text{☺})$$

Probability in M'

Explanation

- Change the probability measure by redistributing probabilities.

$$\mathbb{P}'_\sigma(\cdot) = \mathbb{P}_\sigma(\cdot \mid \Diamond \text{☺})$$

Probability in M' Conditional probability in M

Explanation

- Change the probability measure by redistributing probabilities.

$$\mathbb{P}'_\sigma(\cdot) = \mathbb{P}_\sigma(\cdot \mid \Diamond \text{☺})$$

Probability in M' Conditional probability in M

- Remove some suboptimal strategies

Explanation

- Change the probability measure by redistributing probabilities.

$$\mathbb{P}'_\sigma(\cdot) = \mathbb{P}_\sigma(\cdot \mid \Diamond ☺)$$

Probability in M' Conditional probability in M

- Remove some suboptimal strategies such that optimizing objective 2 in M' automatically optimizes objective 1 in M.

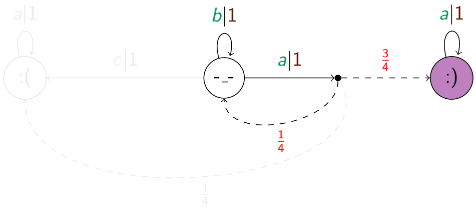
Explanation

- Change the probability measure by redistributing probabilities.

$$\mathbb{P}'_\sigma(\cdot) = \mathbb{P}_\sigma(\cdot \mid \Diamond \text{:)})$$

Probability in M' Conditional probability in M

- Remove some suboptimal strategies such that optimizing objective 2 in M' automatically optimizes objective 1 in M.



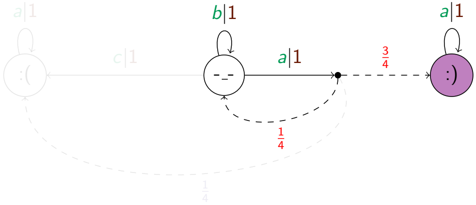
Explanation

- Change the probability measure by redistributing probabilities.

$$\mathbb{P}'_\sigma(\cdot) = \mathbb{P}_\sigma(\cdot \mid \Diamond \text{:)})$$

Probability in M' Conditional probability in M

- Remove some suboptimal strategies such that optimizing objective 2 in M' automatically optimizes objective 1 in M.



$$\sigma \notin \arg\max_{\sigma'} \mathbb{P}^M_{\sigma'}(s_0 \models \Diamond \text{:)})$$

$$\Downarrow$$

$$\mathbb{E}^{M'}_\sigma(\text{len}_{\text{:)}}) = \infty$$

Experimental results

layouts	$\mathbb{P}$(🤖 $\rightsquigarrow$ 💎)	min(🤖 $\rightsquigarrow$ 💎)	$\mathbb{E}_{opt}$(🤖 $\rightsquigarrow$ 💎)	$\mathbb{E}_{storm}$(🤖 $\rightsquigarrow$ 💎)
1	0.66	9	76.48	76.48
2	0.52	18	299.75	629.16
3	1.00	2	2.40	12.12
4	1.00	3	3.44	34.47
5	1.00	6	7.71	137.56
6	0.68	10	264.04	9598.81
7	1.00	5	112.69	9367.02
8	0.91	10	11.49	5879.63
9	1.00	3	3.66	5711.76
10	0.91	5	12.89	149357.57

Table: Comparison of expected conditional distance to the target given by the strategy optimizing distance and the strategy given by model checker STORM

General problem

General problem

Find a strategy σ such that

General problem

Find a strategy σ such that

- Optimize objective f :

$$\sigma \in \arg\,\mathrm{opt}_{\sigma'} \mathbb{P}_{\sigma'}(f) = \Sigma_f$$

General problem

Find a strategy σ such that

- Optimize objective f :

$$\sigma \in \arg\operatorname{opt}_{\sigma'} \mathbb{P}_{\sigma'}(f) = \Sigma_f$$

- Optimize objective g conditional to f among strategies that optimize f:

$$\sigma \in \arg\operatorname{opt}_{\sigma' \in \Sigma_f} \mathbb{E}_{\sigma'}(g \mid f)$$

General problem

Find a strategy σ such that

- Optimize objective f :

$$\sigma \in \arg\operatorname{opt}_{\sigma'} \mathbb{P}_{\sigma'}(f) = \Sigma_f$$

- Optimize objective g conditional to f among strategies that optimize f:

$$\sigma \in \arg\operatorname{opt}_{\sigma' \in \Sigma_f} \mathbb{E}_{\sigma'}(g \mid f)$$

We gave a prunning-based algorithm where:

f	g

General problem

Find a strategy σ such that

- Optimize objective f :

$$\sigma \in \arg\operatorname{opt}_{\sigma'} \mathbb{P}_{\sigma'}(f) = \Sigma_f$$

- Optimize objective g conditional to f among strategies that optimize f:

$$\sigma \in \arg\operatorname{opt}_{\sigma' \in \Sigma_f} \mathbb{E}_{\sigma'}(g \mid f)$$

We gave a prunning-based algorithm where:

f	g
reachability	distance/cost to target

General problem

Find a strategy σ such that

- Optimize objective f :

$$\sigma \in \arg\operatorname{opt}_{\sigma'} \mathbb{P}_{\sigma'}(f) = \Sigma_f$$

- Optimize objective g conditional to f among strategies that optimize f:

$$\sigma \in \arg\operatorname{opt}_{\sigma' \in \Sigma_f} \mathbb{E}_{\sigma'}(g \mid f)$$

We gave a prunning-based algorithm where:

f	g
reachability	distance/cost to target
safety	mean payoff

General problem

Find a strategy σ such that

- Optimize objective f :

$$\sigma \in \arg\operatorname{opt}_{\sigma'} \mathbb{P}_{\sigma'}(f) = \Sigma_f$$

- Optimize objective g conditional to f among strategies that optimize f:

$$\sigma \in \arg\operatorname{opt}_{\sigma' \in \Sigma_f} \mathbb{E}_{\sigma'}(g \mid f)$$

We gave a prunning-based algorithm where:

f	g
reachability	distance/cost to target
safety	mean payoff
safety	any measurable function

Thank You!