Pattern Databases for Goal-Probability Maximization in Probabilistic Planning

Thorsten Klößner, Jörg Hoffmann, Marcel Steinmetz, Álvaro Torralba

Universität des Saarlandes

Aalborg Universitet

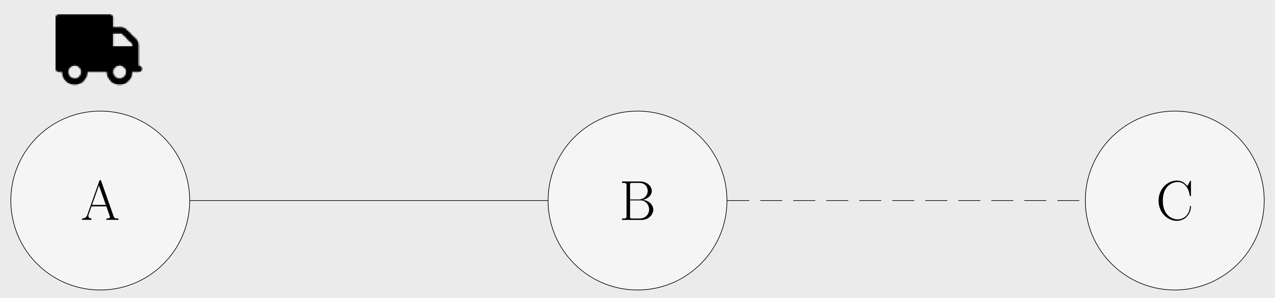
icaps Guangzhou 2021

Probabilistic SAS$^+$ Tasks

A *probabilistic SAS$^+$ task* $\Pi = (\mathbb{V}, A, I, G)$ consists of:

- Variables $\mathbb{V}$
- Actions A, each $a \in A$ associated with
  - Precondition pre(a), partial variable assignment
  - Effects eff(a), partial variable assignments
  - Probability distribution P_a : eff$(a) \to (0, 1]$ over its effects
- Initial State I, complete variable assignment
- Goal G, partial variable assignment

Example Task:



- $\mathbb{V} = \{at, crash\}$
- $A = \{drive\_safe\_road(A, B), drive\_bumpy\_road(B, C)\}$:

drive_safe_road(x, y) :
- pre : $\{at = x, crash = \bot\}$
- eff : $\Pr(\{at = y\}) = 100\%$

drive_bumpy_road(x, y) :
- pre : $\{at = x, crash = \bot\}$
- eff : $\Pr(\{at = y\}) = 50\%$
  $\Pr(\{crash = \top\}) = 50\%$

- $I = \{at = A, crash = \bot\}$
- $G = \{at = C\}$

Probabilistic Planning – MaxProb

Objective: Maximize goal probability in the long run (”MaxProb”)

Multiple Heuristic search algorithms are applicable

- Acyclic Problems: AO*, LAO*, LRTDP…
- Cyclic problems: Run within FRET framework (Kolobov et al. 2011)

Required: Admissible (upper-bounding) heuristic on goal probability

Heuristics for MaxProb

Prior Work: Dead-end detection with classical heuristics (e.g. Steinmetz et al. 2016)

Weakness: Trivial estimate for states with goal probability > 0

Constrained SSPs: Occupation measures, relaxed operator counting (Trevizan et al. 2017)

Weakness: Designed for a different setting, works poorly in MaxProb

Our approach: Pattern Databases for MaxProb

Projection of a Probabilistic SAS$^+$ Task

Construct projection to a variable subset $V \subseteq \mathbb{V}$ from the input task

Similar to classical planning, some effects may become identical
$\to$ Merge their probabilities

$$\begin{matrix} \Pr(\{A = a, B = b_1\}) = 25\% \\ \Pr(\{A = a, B = b_2\}) = 25\% \end{matrix} \quad \xrightarrow{V=\{A\}} \quad \Pr(\{A = a\}) = 50\%$$

The new projected task induces a state space abstraction

The goal probability of an abstract state in this state space is higher
$\to$ **Admissible heuristic!**

MaxProb Pattern Databases

Multiple projections form a pattern database

Obvious combination strategy: Use the **minimum** heuristic estimate

$$\begin{matrix} \{v_1, v_2\} & \{v_3, v_4\} & \{v_5, v_6\} \\ h(s) = 1 & h(s) = \underline{0.6} & h(s) = 0.7 \end{matrix}$$

$$\to h_{\min}(s) = \underline{0.6}$$

In Classical Planning: Additivity Constraints (Haslum et al. 2007)

Are there similar combination strategies in this setting?

Multiplicativity Constraints

As we operate with probabilities, multiplicativity seems natural
$\to$ **Multiplicativity Constraints**

$$\begin{matrix} \{v_1, v_2\} & \{v_3, v_4\} & \{v_5, v_6\} \\ h(s) = 1 & h(s) = 0.6 & h(s) = 0.7 \end{matrix}$$

$$\to h_{\text{mul}}(s) = 1 \cdot 0.6 \cdot 0.7 = 0.42 < h_{\min}(s) = 0.6$$

An action a *affects* V iff a is stochastic after projecting a to V

If no action affects both V and W, then they are multiplicative.

Experimental Results

Time Limit: 30 minutes, Memory Limit: 4GB

Coverage

All acyclic problems (AO*)

Pattern Size	Classical PDBs	Minimum	Multiplicative
$K = 2$	944	940	942
$K = 3$	981	991	**994**
$K = 4$	932	964	961

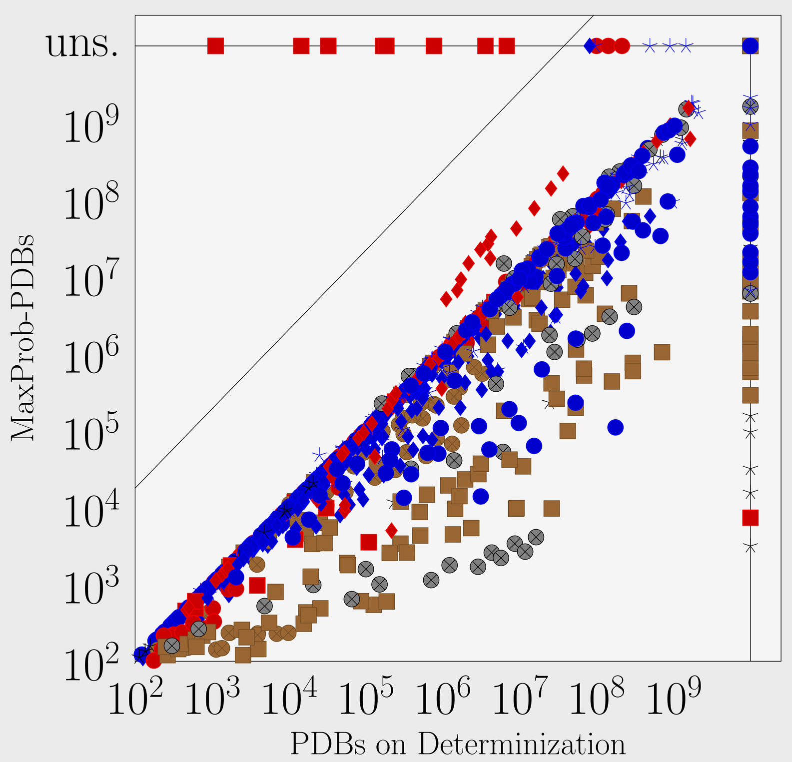
All cyclic problems (LAO* + FRET)

Pattern Size	Classical PDBs	Minimum	Multiplicative
$K = 2$	263	263	264
$K = 3$	317	318	318
$K = 4$	357	**373**	**373**

Only Automated Pentesting

Pattern Size	Classical PDBs	Minimum	Multiplicative
$K = 2$	12	14	16
$K = 3$	12	14	16
$K = 4$	12	14	**17**

Generated States with K=4



Conclusions

- Improvement over classical PDBs when patterns are sufficiently large
- Multiplicativity useful in automated penetration testing
- Open Question: How to generate the initial collection intelligently?