# Cartesian Abstractions and Saturated Cost Partitioning in Probabilistic Planning

*Thorsten Klößner, Jendrik Seipp, Marcel Steinmetz*

UNIVERSITÄT DES SAARLANDES

LINKÖPING UNIVERSITY

## Background — Probabilistic Planning

A probabilistic transition system $\langle S, L, C, T, G\rangle$ consists of *states* $S$, *labels* $L$, a *cost function* $C : L \to \mathbb{R}_0^+$, *stochastic transitions*, $T \subseteq S \times L \times Dist(S)$ and goal states $G \subseteq S$, where every set is finite and non-empty.
*Policies* are partial functions $\pi : S \to T$ with $\pi(s) = \langle s, \ell, \delta\rangle$, if $s \in \pi$. If $s \notin \pi$, $\pi$ terminates in $s$. The expected cost of $\pi$ from state $s$ is defined as

$$J^\pi(s) := \mathbb{E}[\sum_{i=0}^{\mathsf{T}-1} C(\pi(\mathsf{S}_i)) \mid \mathsf{S}_0 = s] \quad \text{where} \quad \begin{array}{r} \mathsf{T} := \inf\{i \mid \mathsf{S}_i \notin \pi\} \\ \mathsf{S}_{i+1} \sim \pi(\mathsf{S}_i), \text{ for } i < \mathsf{T}. \end{array}$$

A solution for $s \in Sols(s)$ terminates in a goal state with certainty. A solution $\pi \in Sols(s)$ is *optimal* for $s$ iff $J^\pi(s) = J^*(s) := \inf_{\pi \in Sols(s)} J^\pi(s)$.

A *probabilistic planning task* $\langle \mathcal{V}, \mathcal{O}, I, G\rangle$ consists of variables $\mathcal{V}$ with domains $\mathcal{D}(v)$, *operators* $\mathcal{O}$ with *preconditions* $pre(o)$, cost $cost(o) \in \mathbb{R}_0^+$ and stochastic *effects* $eff_1(o), \ldots, eff_{aty(o)}(o)$ with probabilities $Pr_1(o), \ldots, Pr_{aty(o)}(o)$, initial variable assignment $I$ and partial goal assignment $G$. Every task naturally induces a corresponding PTS.

## Construction of Cartesian Abstractions

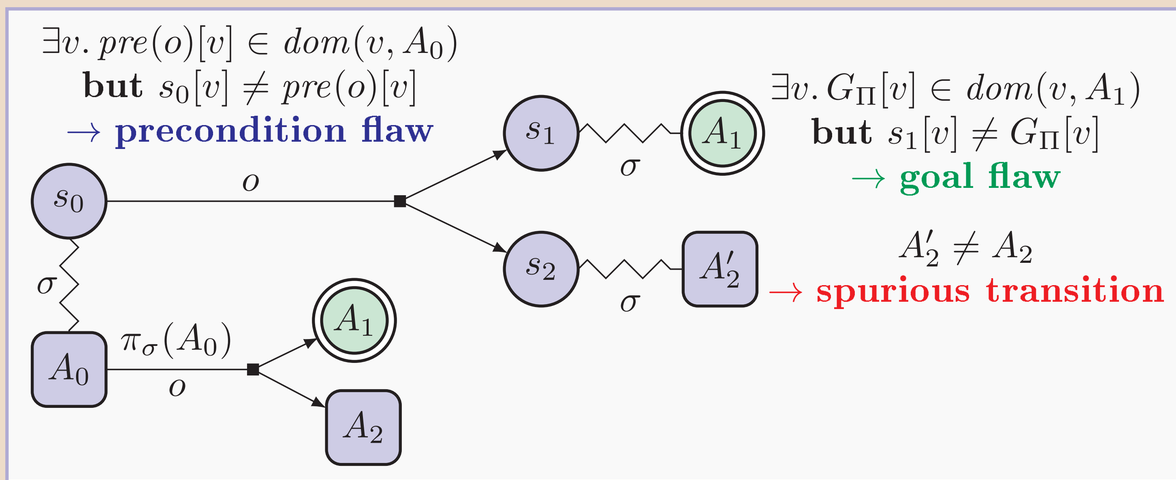
To construct Cartesian abstractions, we employ counter-example guided abstraction refinement (CEGAR), with policies instead of plans.

**Algorithm 1:** CEGAR refinement loop (policy-based).

**Input:** Probabilistic Planning Task $\Pi$
$\langle \Theta_\sigma, \sigma\rangle \leftarrow$ TrivialAbstraction($\Pi$)
**while not** TerminationCondition() **do**
  $\pi_\sigma \leftarrow$ FindOptimalSolution($\Theta_\sigma$, $\sigma(I_\Pi)$)
  **if not** $\pi_\sigma$ exists**: return** "*Task unsolvable*"
  $\phi \leftarrow$ FindFlaw($\Pi$, $\Theta_\sigma$, $\pi_\sigma$)
  **if not** $\phi$ exists**: return** "*Optimal solution found*"
  $\langle \Theta_\sigma, \sigma\rangle \leftarrow$ Refine($\Theta_\sigma$, $\sigma$, $\phi$)
**return** $\Theta_\sigma$

## Finding Flaws in the Abstraction

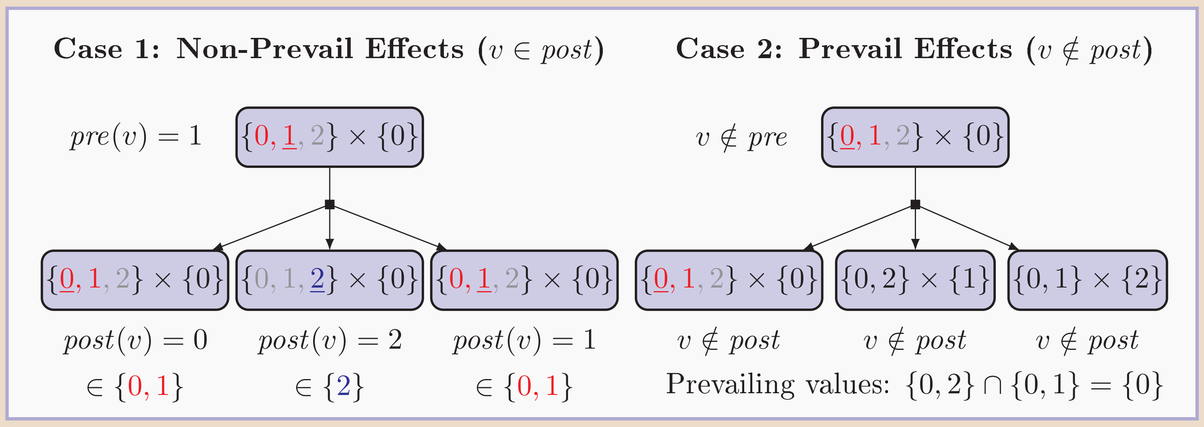
Execute abstract policy $\pi_\sigma$ as a concrete policy in the concrete state space, by choosing for $s$ the operator of the corresponding abstract transition $\pi_\sigma(\sigma(s))$. Can fail or be suboptimal for three reasons:




Expand the policy graph of this policy from the initial state, checking for flaws on-the-fly. If no flaw is found, the policy is optimal. Otherwise, return a pair $\langle s, c\rangle$ of state $s$ and cartesian set $c$ such that $s \notin c \subsetneq \sigma(s)$. $c$ represents the property $s$ was expected to satisfy, but violated instead.

## Refining the abstract PTS

Given flaw $\langle s, c\rangle$, pick a split variable $v$ with $s[v] \notin dom(v, c) \subsetneq dom(v, \sigma(s))$. Split $\sigma(s)$ along $v$ into two split states $A_1$ and $A_2$ by defining $dom(v, A_1) := dom(v, c)$, $dom(v, A_2) = dom(v, \sigma(s)) \setminus dom(v, c)$.
Next, rewire abstract transitions involving $\sigma(s)$ to the split states. Which of $A_1$, $A_2$ depends on which values for $v$ can occur for the outcome:




**Algorithm 2:** Rewire an abstract transition.

**Input:** Split $\langle \sigma(s), A_1, A_2, v\rangle$
Abstract Transition $\langle B_0, o, \langle B_1, \ldots, B_{aty(o)}\rangle\rangle$
$B_0', \ldots, B_{aty(o)}' \leftarrow B_0, \ldots, B_{aty(o)}$ `// Rewired targets`
**for** $i = 0$ **to** $aty(o)$ **do** `// Rewire non-prevail effects`
  **if** $B_i = \sigma(s)$ **and** $v \in post_i(o)$**:**
    $B_i' \leftarrow \begin{cases} A_1 & post_i(o)[v] \in dom(v, A_1) \\ A_2 & \text{else, i.e., } post_i(o)[v] \in dom(v, A_2) \end{cases}$
$PrevailingEffectTargets \leftarrow \{i \mid B_i = \sigma(s) \wedge v \notin post_i(o)\}$
**if** $PrevailingEffectTargets = \emptyset$**:**
  **return** $\langle B_0', o, \langle B_1', \ldots, B_{aty(o)}'\rangle\rangle$
$PrevailingValues \leftarrow \bigcap_{i: B_i \neq \sigma(s) \wedge v \notin post_i(o)} dom(v, B_i)$
**for** $A \in \{A_1, A_2\}$ **do** `// Uniformly rewire prevail effects`
  **if** $PrevailingValues \cap dom(v, A) \neq \emptyset$**:**
    **for** $i \in PrevailingEffectTargets$ **do** $B_i' \leftarrow A$
    **output** $\langle B_0', o, \langle B_1', \ldots, B_{aty(o)}'\rangle\rangle$

## Saturated Cost Partitioning

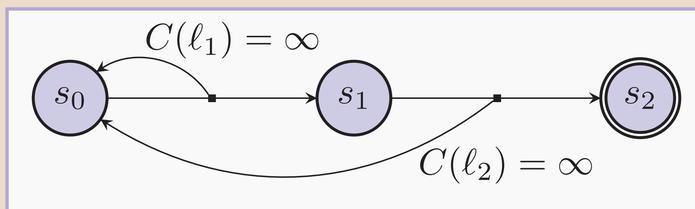
We use *saturated cost partitioning* to combine multiple cartesian abstractions, constructed in succession. To compute the saturated and remaining costs during the procedure, we use the following saturated cost function.

**Theorem.** *Let $\Theta$ be a PTS and $T^{fin} := \{\langle s, \ell, \delta\rangle \in T \mid \forall t \in supp(\delta). J^*_\Theta(t) \neq \infty\}$. The cost function $scf(\ell) := \infty$ if $C(\ell) = \infty$ and otherwise*

$$scf(\ell) := \sup_{\langle s,\ell,\delta\rangle \in T^{fin}} J^*_\Theta(s) - \sum_{t \in S} \delta(t) \cdot J^*_\Theta(t)$$

*preserves the optimal state costs of $\Theta$.*

**Observation:** The minimal saturated cost function is **not unique!**




## Experiments

| | $h^{\text{blind}}$ and single abstraction | | | | | multiple abstractions | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Domains | $h^{\text{blind}}$ | $h^{\text{PDB}}_{\text{CEGAR}}$ | $h^{\text{Cart}_1}_{\text{det}}$ | $h^{\text{Cart}_1}_{\text{iLAO}^\star}$ | $h^{\text{Cart}_1}_{\text{A}^\star}$ | $h^{\text{roc}}$ | $h^{\text{SYS-1}}_{\text{Can}}$ | $h^{\text{SYS-2}}_{\text{Can}}$ | $h^{\text{SYS-3}}_{\text{Can}}$ | $h^{\text{HC}}_{\text{Can}}$ | $h^{\text{DC}}_{\text{Max}}$ | $h^{\text{SYS-1}}_{\text{SCP}}$ | $h^{\text{SYS-2}}_{\text{SCP}}$ | $h^{\text{SYS-3}}_{\text{SCP}}$ | $h^{\text{HC}}_{\text{SCP}}$ | $h^{\text{DC}}_{\text{SCP}}$ | $h^{\text{Cart}_k}_{\text{det}}$ | $h^{\text{Cart}_k}_{\text{iLAO}^\star}$ | $h^{\text{Cart}_k}_{\text{A}^\star}$ |
| BlocksW | 7 | **8** | 7 | 7 | 7 | 7 | **9** | **9** | 8 | **9** | 8 | **9** | **9** | **9** | **9** | **9** | 7 | 7 | 7 |
| Boxworld | 4 | 5 | **7** | 5 | **7** | 4 | 5 | 5 | 4 | 6 | 4 | 5 | **7** | **7** | 6 | 6 | 6 | **7** | 6 |
| Elevators | 10 | **11** | 10 | **11** | 10 | 10 | 13 | 12 | 8 | **17** | 15 | 13 | 15 | **17** | **17** | **17** | 15 | 14 | 16 |
| PARCPr | **8** | **8** | **8** | **8** | **8** | **20** | 13 | 8 | 8 | 19 | 11 | 14 | **20** | 6 | **20** | 18 | 12 | 14 | 12 |
| Random | 14 | 17 | **18** | 17 | **18** | 14 | 16 | 17 | 13 | **18** | 17 | **18** | **18** | 13 | **18** | 16 | **18** | 16 | **18** |
| Schedule | 12 | **15** | 13 | 13 | 12 | 11 | 12 | 13 | 12 | 14 | 13 | 14 | 12 | 12 | 13 | 13 | 13 | 12 | 12 |
| SysAdmin | **12** | **12** | **12** | 11 | 11 | **12** | **12** | **12** | 8 | **12** | **12** | **12** | **12** | **12** | **12** | **12** | **12** | **12** | **12** |
| TTireworld | 5 | **9** | 7 | **9** | 7 | 7 | 7 | 7 | 8 | **9** | **9** | 7 | 7 | 8 | **9** | **9** | 7 | **9** | 7 |
| ZenoTravel | 5 | 10 | 8 | **11** | 10 | 7 | 9 | 10 | 9 | 10 | 9 | 9 | 9 | 9 | 10 | 10 | 8 | 10 | 10 |
| Sum (of 180) | 77 | **95** | 90 | 92 | 90 | 92 | 96 | 93 | 78 | **114** | 98 | 101 | 109 | 93 | **114** | 110 | 98 | 100 | 101 |