



Deliverable D1.5

Implementation: Synthesis Techniques

Deliverable

<i>Number and title:</i>	D1.5 – Implementation: Synthesis Techniques		
<i>Work package:</i>	WP1 (Energy and Resources)		
<i>Lead author:</i>	Thomas Noll (RWTH)		
<i>Contributors:</i>	Alexander Bork (RWTH)		
<i>Reviewers:</i>	Pablo Castro (UNRC), Pablo Diale (ASC)		
<i>Due date:</i>	M57 (2026-06-30)	<i>Dissemination level:</i>	Public
		<i>Version:</i>	1.0 (final)
<i>Submission date:</i>	2026-06-26	<i>Pages:</i>	20

Version history

<i>Version</i>	<i>Date</i>	<i>Notes</i>
1.0	2026-06-26	First official release

Project

<i>Title:</i>	Models in Space Systems: Integration, Operation, and Networking		
<i>Acronym:</i>	MISSION	<i>Start date:</i>	01-10-2021
<i>GA no.:</i>	101008233	<i>Duration:</i>	63 months
<i>Call:</i>	H2020-MSCA-RISE-2020	<i>Website:</i>	mission-project.eu



This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 101008233. This document reflects only the authors' view. It does not represent the opinion of the European Union, and the European Union is not responsible for any use that may be made of the information it contains.

1 Executive Summary

This deliverable presents a library of efficient synthesis techniques addressing uncertainty in resource dynamics. This library is as part of the development of strategy synthesis techniques for partially observable stochastic resource-constrained systems. Here, the synthesis of sub-optimal, but simple-to-implement strategies for multi-objective resource optimisation problems has been put in focus. The result is a library of efficient synthesis techniques.

The work presented here is allocated to MISSION Work Package 1, *Energy and Resources*. The project-level motivation is that the in-orbit operation of satellite constellations must be robust against unpredictable circumstances like transient failures of nodes or single-event effects from solar flares that can become a common cause of multiple failures across the constellation. This aspect strongly impacts resource safety, which is studied jointly with Work Package 2. Our contribution is the automated synthesis of sub-optimal strategies for partially observable stochastic systems, to support the safe operation of systems in such unknown, uncertain environments, paired with the synthesis of sub-optimal, but simple-to-implement strategies for multi-objective optimisation problems. This allows us to address the multitude of resource constraints in an effective yet safe manner, backed by theoretical investigation about optimal scheduler characterisation, sub-optimality, and respective approximate values.

2 Introduction

Partially observable Markov decision processes (POMDPs) are a powerful modelling formalism for sequential decision making in uncertain domains where non-determinism is present. They extend *Markov decision processes (MDPs)* [31] for agents that in addition to uncertain transitions also only have incomplete information about the state of the environment [33, 36]. POMDPs have applications in a plethora of domains, including routing in space networks [39], generation of spacecraft operations procedures [14], robotics [38], ecology [11], and cyber security [27].

The classical planning, or synthesis, problem in POMDPs is to compute a *strategy* or *policy*—a plan for resolving non-determinism in the system—that optimises a given objective. This problem is notoriously difficult for many kinds of objectives. Various approaches consider *finite horizons*, where the objective has to be satisfied within a finite amount of steps [36], or *discounting*, where events in later stages become less relevant [24, 35, 37].

In recent years, work focusing on objectives *without discounting* over an *infinite* time horizon has emerged [2, 3, 8, 17, 18, 28]. One commonly considered objective is to compute the maximal probability to reach a set of target states. In the field of probabilistic model checking, this *maximal reachability probability* objective is the basis for the analysis of models with respect to more involved logical specifications such as linear time temporal logic [5].

Expected costs vs. cost-bounded reachability In many practical scenarios, the objective to reach a target is subject to hard constraints on resources. For example, an autonomous vehicle navigating towards a goal position has to consider its fuel level and emission levels of pollutants. A strategy minimising the *expected* fuel and emission costs does not take the resource limits into account, which may lead to the vehicle running out of fuel with unnecessarily high probability. In such scenarios, a strategy that maximises the *probability* of a successful run—where the vehicle reaches the goal without running out of fuel and within pollution limits—is preferable. Thus, objectives that constrain the *expected* costs fail to capture scenarios where satisfying hard constraints on actually incurred costs is key for a run to be successful.

(Multi-)cost-bounded reachability probability objectives characterise such scenarios where a strict adherence to certain resource constraints is crucial. Numerical costs (or rewards) are assigned to transitions in the POMDP. The goal of the agent is to maximise its probability to reach the target states while



Figure 1: Hallway Cleaning Task

satisfying all bounds on the *actual* accumulated costs. In the most general setting, both upper and lower bounds on multiple cost measures can be considered simultaneously. Cost-bounded reachability objectives have mainly been studied for (fully observable) MDPs [13, 15, 23].

Motivating example Consider the scenario depicted in Figure 1. A robot is tasked with cleaning a hallway consisting of six tiles, all initially dirty. The robot starts in the left-most tile with an energy level of 60 units. In every step, the robot can either attempt to clean the current tile or move to the next tile. A cleaning attempt can fail with a probability of 0.2, leaving the tile dirty. The robot, however, is not able to observe whether it has successfully cleaned a tile. Moving to the next tile always consumes one unit of energy. A cleaning attempt consumes either two or four units of energy, each with probability 0.5. The robot is successful if it cleans all six tiles and reaches the target position by moving to the right-most tile of the hallway without running out of energy.

Our goal is to find a strategy for the robot that schedules in each step the best action based on the available information in order to maximise the probability that the task succeeds. We can model the scenario as a POMDP with two cost measures C_{energy} and C_{clean} (the latter assigning a cost of one when a tile is successfully cleaned and zero otherwise), and the cost-bounded reachability query “*maximise the probability to reach the target while accumulating at most 60 cost units for C_{energy} and at least six cost units for C_{clean} .*”

Usual POMDP objectives fail to capture the objective we are interested in. In particular, minimising the expected energy usage or maximising the expected number of cleaned tiles fails to consider the respective other constraint and does not accurately reflect the hard requirements on incurred costs. A multi-cost-bounded objective with cost bounds on both measures, however, adequately reflects our aims.

Our approach We formalise the problem of synthesising strategies for multi-cost-bounded reachability probability objectives on POMDPs and observe undecidability of its decision variant, i.e., in the case of given threshold probabilities. We then consider three variations of this problem that all differ in the degree of observability of the cost accumulated so far. (As usual, we assume that the observations in the POMDP are visible to the decision-making agent.) For the extreme case in which the accumulated costs are completely invisible, we provide a transformation to an equivalent unbounded reachability problem on an (often larger) POMDP. The key concept is to encode accumulation of costs in the state space. The resulting *cost-unfolding POMDP* can then be analysed using existing approximation methods for unbounded infinite-horizon reachability problems.

We then consider the other extreme case in which the accumulated cost is fully observable by the agent and can be used in its decision making. We show that the sequential approach of [15] for multi-cost-bounded reachability in MDPs, i.e., fully observable POMDPs, can be readily lifted to this setting of *cost-aware* POMDPs. Furthermore, we consider the novel setting where the agent cannot observe the exact costs gathered so far, but only certain *cost levels* (e.g., high, mid and low). We show that this setting can often be reduced to the analysis of a cost-aware POMDP.

Our algorithmic solutions are designed such that they leverage strengths of existing (time- and space-) efficient techniques, and are thus able to directly profit from future advancements in unbounded reachability in POMDPs and cost-bounded reachability in MDPs.

Implementation Our approach has been implemented as a prototypical extension of the model-checking tool STORM¹ [16], which supports the analysis of systems involving random or probabilistic phenomena. This extension provides the required unfolding operations on POMDPs, along with the capabilities for exploring MDPs with cost bounds. Our experimental evaluation demonstrate the suitability of cost-bounded reachability analysis for practical applications; details are provided in Section 7. The implementation is currently being integrated into the existing POMDP verification framework of STORM and will become part of the official release in the course of 2026.

2.1 Related Work

Multi-cost-bounded reachability objectives for (fully observable) MDPs are well-studied [4, 13, 15, 23, 29, 32]. We focus on related research dedicated to partially observable models.

Most closely related to our study is the work on *risk-sensitive POMDPs* [19] where a special case of a cost-bounded reachability objective is considered. In particular, the authors present an unfolding of cost-bounds on the level of beliefs, similar to our unfolding on the level of the POMDP. They also consider the case of observable costs. However, our framework is more general by allowing for *multiple* bounds over different cost measures as well as mixtures of upper and lower bounds. In addition, our unfolding enables the use of non-belief-based solution methods for the problem.

The authors of [41] consider an unfolding that encodes cost and step bounds in the belief space to solve a classification problem for *hidden-model MDPs*, which are a special class of POMDPs. In that setup, the goal of an agent is to find out in which specific instance of structurally similar MDP environments it is located while not exceeding certain costs.

Another related formalism are constrained POMDPs (CPOMDPs) [20, 30, 34]. In a CPOMDP, the objective is to maximise the expected sum of rewards subject to a bound on the *expected* cumulative costs. While classically the setting considers discounted values with upper bounds on expected costs, there are extensions to an undiscounted setting [22]. Furthermore, [40] consider a scenario where violation of bounds for expected and actual costs coincide. In contrast, our work incorporates mixtures of different bound types on the actual incurred costs and does not make assumptions on the POMDP or the cost bounds.

The authors of [12] aim at minimising expected costs over all strategies in a POMDP that reach a target *almost surely*, i.e., with probability one. This can be considered a related problem where a strict bound is placed on the reachability probability rather than the incurred costs.

3 Preliminaries

We briefly outline the theoretical background for POMDPs. Further details can be found in [33]. The textbook [5, Chapter 10] gives an introduction to MDPs from a formal methods perspective.

Let $X \neq \emptyset$ be a countable set. A (probability) *distribution* over X is a function $\mu : X \rightarrow [0, 1]$ with $\sum_{x \in X} \mu(x) = 1$. $\text{Dist}(X)$ is the set of distributions over X . We write $x \in \mu$ if $\mu(x) > 0$. The *support* of μ is $\text{supp}(\mu) := \{x \mid x \in \mu\}$. For $k \in \mathbb{N}$ and *vector* $\mathbf{x} = \langle x_1, \dots, x_k \rangle \in X^k$, we write $\mathbf{x}[i] = x_i$ for the i -th element (where $1 \leq i \leq k$).

MDPs A *Markov decision process (MDP)* is a tuple $M = \langle S, \text{Act}, \mathbf{P}, s_{\text{init}} \rangle$ with a (finite or countably infinite) set S of states, a finite set Act of actions, a transition function $\mathbf{P} : S \times \text{Act} \rightarrow \text{Dist}(S)$, and an initial state $s_{\text{init}} \in S$. In every state s , an agent making decisions in the MDP chooses an action $a \in \text{Act}$ and the state is updated to state s' with probability $\mathbf{P}(s, a)(s')$. If $s' \in \mathbf{P}(s, a)$, we call (s, a, s') a transition and write $s \xrightarrow{a} s'$.

¹<https://www.stormchecker.org/>

POMDPs A *partially observable MDP (POMDP)* is a tuple $\mathcal{M} = \langle M, Z, \mathbf{O} \rangle$, where M is the underlying MDP with a finite set S of states, Z is a finite set of observations, and $\mathbf{O}: S \times Act \times S \rightarrow Dist(Z)$ is a probabilistic observation function.

In a POMDP, the agent does not have complete access to the current state of the system to base its decisions on. Instead, upon taking a transition $s \xrightarrow{a} s'$ the agent receives an *observation* z with probability $\mathbf{O}(s, a, s')(z)$.

A (*finite, initial*) *path* in an MDP or POMDP is a sequence $\hat{\pi} = s_0 a_1 s_1 \dots a_n s_n$ with $s_0 = s_{init}$ such that for all $0 < i \leq n$ we have $s_i \in \mathbf{P}(s_{i-1}, a_i)$. We denote by $|\hat{\pi}| := n$ the length, by $\hat{\pi}[i] := s_i$ the i -th state, and by $last(\hat{\pi}) := s_n$ the last state of $\hat{\pi}$. An *observation trace* of a POMDP is a sequence of observations $\tau = z_1 \dots z_n \in Z^*$. Given a path $\hat{\pi} = s_0 a_1 s_1 \dots a_n s_n$, the probability of observing trace $\tau = z_1 \dots z_n$ in $\mathcal{M}$ is $P^{\mathcal{M}}(\tau | \hat{\pi}) = \prod_{i=0}^{n-1} \mathbf{O}(s_i, a_{i+1}, s_{i+1})(z_{i+1})$.

Strategies resolve the non-determinism of POMDPs by determining the next action to play after observing an observation trace τ . Formally, a *strategy* for $\mathcal{M}$ is a function $\sigma: Z^* \rightarrow Act$. We denote the set of strategies for POMDP $\mathcal{M}$ by $\Sigma^{\mathcal{M}}$.

Belief MDP A *belief* is a tuple $b = \langle z, \mu_b \rangle$ consisting of an observation $z \in Z \uplus \{z_{init}\}$, where $z_{init} \notin Z$ is a dedicated initial observation, and a probability distribution $\mu_b \in Dist(S)$ over POMDP states.² The distribution captures the evolution of an agent's information about its current state given histories of actions and observations, while the observation represents the last observation made in such a history. An agent starts with the initial belief $b_{init} = \langle z_{init}, \mu_{b_{init}} \rangle$, with $\mu_{b_{init}}(s_{init}) = 1$. Beliefs are updated when an action is played and a new observation is received. The probability to observe $z \in Z$ after playing action a in belief $b = \langle \hat{z}, \mu_b \rangle$ is

$$P(z | b, a) = \sum_{s \in \mu_b} \mu_b(s) \cdot \sum_{s' \in S} \mathbf{P}(s, a)(s') \cdot \mathbf{O}(s, a, s')(z).$$

The successor belief of b after playing a and observing z is $\text{succ}(b, a, z) = \langle z, \mu_{b,a,z}^{\text{succ}} \rangle$ where $\mu_{b,a,z}^{\text{succ}}$ is given by

$$\begin{aligned} \mu_{b,a,z}^{\text{succ}}(s') &:= P(s' | b, a, z) \\ &= \frac{\sum_{s \in \mu_b} \mu_b(s) \cdot \mathbf{P}(s, a)(s') \cdot \mathbf{O}(s, a, s')(z)}{P(z | b, a)} \end{aligned}$$

if $P(z | b, a) > 0$ and *undefined* otherwise. Successive computation of successor beliefs yields an infinite-state fully observable MDP capturing the POMDP dynamics. This *belief MDP* is the basis for many solution methods for analysis problems on POMDPs. Let $\mathcal{B}_{\mathcal{M}}^n$ be the set of beliefs *reachable in n steps*, given by $\mathcal{B}^0 := \{b_{init}\}$ and $\mathcal{B}_{\mathcal{M}}^{n+1} := \mathcal{B}_{\mathcal{M}}^n \cup \{\text{succ}(b, a, z) | b \in \mathcal{B}_{\mathcal{M}}^n, a \in Act, z \in Z\}$. $\mathcal{B}_{\mathcal{M}} := \bigcup_{n \in \mathbb{N}} \mathcal{B}_{\mathcal{M}}^n$ is the set of reachable beliefs.

The belief MDP of POMDP $\mathcal{M}$ is $bel(\mathcal{M}) = \langle \mathcal{B}_{\mathcal{M}}, Act, \mathbf{P}^B, b_{init} \rangle$ where $\mathbf{P}^B(b, a)(b') := P(z | b, a)$ if $b' = \text{succ}(b, a, z)$ and $\mathbf{P}^B(b, a)(b') := 0$ otherwise.

Costs We annotate POMDPs with *costs* (also referred to as *rewards*). Let $k \in \mathbb{N}$. A (k -dimensional) *cost structure* for $\mathcal{M}$ is a function $\mathbf{C}: S \times Act \times S \rightarrow \mathbb{N}^k$. When taking transition $s \xrightarrow{a} s'$, the cost values $\mathbf{C}(s, a, s') = \langle c_1, \dots, c_k \rangle$ are collected. A cost structure allows the encoding of different, independent cost measures in its dimensions. For example, one dimension can model the expired time, while another models energy consumption. The *cumulative cost of a finite path* $\hat{\pi}$ in $\mathcal{M}$ with respect to $\mathbf{C}$ is $\text{cost}_{\mathbf{C}}(\hat{\pi}) := \sum_{i=1}^{|\hat{\pi}|} \mathbf{C}(s_{i-1}, a_i, s_i)$. The set of all distinct cost vectors occurring in $\mathbf{C}$ is $\Gamma_{\mathbf{C}} := \{\mathbf{C}(s, a, s') | s, s' \in S, a \in Act\}$.

²In the literature, beliefs are typically considered to only be the distribution μ_b . We extend this definition by the explicit inclusion of the observation that yields the belief to simplify later definitions.

4 Problem Statement

The cost-bounded reachability (CBR) problem asks for the maximal probability to reach a set of states in the POMDP via paths that respect (multi-dimensional) bounds on the cumulative costs. Formally, for a k -dimensional cost structure $\mathbf{C}$, relations $\bowtie \in \{\leq, >\}^k$ and threshold $\mathbf{t} \in \mathbb{N}^k$, we call $(\mathbf{C} \bowtie \mathbf{t})$ a (k -dimensional) *cost bound* over $\mathbf{C}$. Cost bounds represent constraints on the cumulative cost of paths with respect to the corresponding cost structure. For a finite path $\hat{\pi}$, bound $(\mathbf{C} \bowtie \mathbf{t})$ is *active in dimension i* iff $\text{cost}_{\mathbf{C}}(\hat{\pi})[i] \bowtie[i] \mathbf{t}[i]$. Moreover, $(\mathbf{C} \bowtie \mathbf{t})$ is *active* for $\hat{\pi}$ iff it is active in all dimensions $1 \leq i \leq k$, i.e., $\text{cost}_{\mathbf{C}}(\hat{\pi}) \bowtie \mathbf{t}$, where the relations in $\bowtie$ are applied element-wise. As we consider only natural values for costs, relations $\geq$ and $<$ are supported by adapting the thresholds in $\mathbf{t}$. Non-negative rational costs are supported by suitable scaling of $\mathbf{C}$ and $\mathbf{t}$.

We fix a POMDP $\mathcal{M}$ and a k -dimensional cost bound $(\mathbf{C} \bowtie \mathbf{t})$. Given a strategy σ , the *cost-bounded reachability probability* for a set $T \subseteq S$ of target states is

$$\Pr_{\sigma}^{\mathcal{M}}(\Diamond_{\mathbf{C} \bowtie \mathbf{t}} T) := \Pr_{\sigma}^{\mathcal{M}}\{\pi \in \text{Cyl}(\hat{\pi}) \mid \text{last}(\hat{\pi}) \in T \text{ and } (\mathbf{C} \bowtie \mathbf{t}) \text{ is active for } \hat{\pi}\},$$

where $\text{Cyl}(\hat{\pi})$ is the set of infinite extensions of finite path $\hat{\pi}$ and $\Pr_{\sigma}^{\mathcal{M}}$ denotes the standard probability measure for $\mathcal{M}$ under strategy σ [31]. We call $\Pr_{\sigma}^{\mathcal{M}}(\Diamond_{\mathbf{C} \bowtie \mathbf{t}} T)$ the *value* of strategy σ . The *maximal* cost-bounded reachability probability is $\Pr_{\max}^{\mathcal{M}}(\Diamond_{\mathbf{C} \bowtie \mathbf{t}} T) := \sup_{\sigma \in \Sigma^{\mathcal{M}}} \Pr_{\sigma}^{\mathcal{M}}(\Diamond_{\mathbf{C} \bowtie \mathbf{t}} T)$. Similar to related problems in POMDPs, a strategy realising the maximal cost-bounded reachability probability is not guaranteed to exist. We formulate our problem in terms of a two-sided ϵ -approximation.

Problem 1 ((Multi-)Cost-Bounded Reachability (CBR)) For POMDP $\mathcal{M}$, $T \subseteq S$, cost bound $(\mathbf{C} \bowtie \mathbf{t})$, and $\epsilon \in [0, 1]$, compute an upper bound $V^U \in [0, 1]$ and a strategy $\tilde{\sigma} \in \Sigma^{\mathcal{M}}$, such that

$$V^U - \epsilon \leq \Pr_{\tilde{\sigma}}^{\mathcal{M}}(\Diamond_{\mathbf{C} \bowtie \mathbf{t}} T) \leq \Pr_{\max}^{\mathcal{M}}(\Diamond_{\mathbf{C} \bowtie \mathbf{t}} T) \leq V^U.$$

Theorem 1 The decision variant of CBR (where the upper bound V^U is given) is undecidable.

A proof of this theorem, together with proofs of the remaining formal results given in this report, can be found in [9].

We write $\Pr_{\max}^{\mathcal{M}}(\Diamond T)$ for the unbounded problem as a special case of CBR with 0-dimensional costs. Methods to tackle unbounded reachability using two-sided approximations have been described in the literature (see Section 2).

Decidability of several subclasses of cost-bounded reachability can be established. For example, *finite-horizon* reachability probabilities—which can be computed exactly [36]—are a special instance of cost-bounded reachability probabilities, where *each* transition induces exactly a cost of one and the costs are bounded upwards by the horizon. Lifting this to arbitrary, but strictly positive costs yields the *risk sensitive* setup from [19], which—as mentioned by the authors—is still decidable. Our setting is more general since we allow transitions with zero costs in any dimension as well as queries with only lower bounds.

5 From Cost-Bounded To Unbounded Reachability

We present an extension of the unfolding approach [1, 29] for the cost-bounded reachability problem in MDPs to the partially observable domain.

Definition 1 (Cost Epoch) A (cost) epoch of dimension k is a tuple $\mathbf{e} = \langle e_1, \dots, e_k \rangle \in (\mathbb{N} \cup \{\perp\})^k$. We denote the domain of all k -dimensional epochs by $\mathbf{E}_k := (\mathbb{N} \cup \{\perp\})^k$.

Each entry $e[i]$ of an epoch keeps track of the costs that can be accumulated until the bound $(C \bowtie t)$ changes its status in dimension i (active to inactive or vice versa).

The initial epoch is defined by the threshold vector t . To evolve cost epochs, we subtract the costs collected in each dimension. For values below zero, we use the dedicated symbol $\perp$ to indicate that the bound changed its status from the initial one. Formally, this is captured by the *monus* operation.

Definition 2 (Monus for Epochs) The monus operator for cost epochs $\ominus : E_k \times \mathbb{N}^k \rightarrow E_k$ is given component-wise as

$$(e \ominus c)[i] := \begin{cases} e[i] - c[i] & \text{if } \perp \neq e[i] \wedge e[i] \geq c[i], \\ \perp & \text{otherwise.} \end{cases}$$

We lift the notion of being active to epochs. The indicator function $\text{actv}_{C \bowtie t} : E_k \rightarrow \{0, 1\}$ is 1 iff bound $(C \bowtie t)$ is active in a given epoch, i.e., $\text{actv}_{C \bowtie t}(e) := 1$ if for all $1 \leq i \leq k$, $\bowtie[i] = \leq$ implies $e[i] \neq \perp$ and $\bowtie[i] = >$ implies $e[i] = \perp$, and $\text{actv}_{C \bowtie t}(e) := 0$ otherwise.

Using the idea of epochs, we construct a POMDP that enables us to reason about the activation of a bound on the level of states instead of the path level. We first recap the construction for an MDP as it is described in the literature, e.g. in [15]. Let the (finite) set of reachable epochs from an epoch e be given by:

$$E_k(e) := \{e' \in E_k \mid \exists c \in \mathbb{N}^k : e' = e \ominus c\}.$$

Definition 3 (Bound Unfolding MDP) For MDP $M = \langle S, Act, P, s_{init} \rangle$ and cost bound $(C \bowtie t)$, the bound unfolding MDP is $\text{un}_{C \bowtie t}(M) := \langle S \times E_k(t), Act, P_{un}, \langle s_{init}, t \rangle \rangle$, and for $s_e := \langle s, e \rangle$, $s'_{e'} := \langle s', e' \rangle$, and $a \in Act$:

$$P_{un}(s_e, a)(s'_{e'}) := \begin{cases} P(s, a)(s') & \text{if } e' = e \ominus C(s, a, s'), \\ 0 & \text{otherwise.} \end{cases}$$

Definition 4 (Bound Unfolding POMDP) Given a POMDP $\mathcal{M} = \langle M, Z, O \rangle$ with underlying MDP $M = \langle S, Act, P, s_{init} \rangle$ and cost bound $(C \bowtie t)$, the bound unfolding POMDP is

$$\text{un}_{C \bowtie t}(\mathcal{M}) := \langle \text{un}_{C \bowtie t}(M), Z, O_{un} \rangle$$

where for $s_e := \langle s, e \rangle$, $s'_{e'} := \langle s', e' \rangle$, and $a \in Act$: $O_{un}(s_e, a, s'_{e'}) := O(s, a, s')$.

Finally, we lift the notion of an active bound to states of the unfolding POMDP which enables us to reason about conformance to the bound on states instead of paths.

Definition 5 (Active States) A state of the unfolding $\langle s, e \rangle \in S \times E_k(t)$ is active iff $\text{actv}_{C \bowtie t}(e) = 1$. Given $T \subseteq S$, the set of active T -states is $\text{actv}_{C \bowtie t}(T) = \{\langle s_T, e \rangle \in S \times E_k(t) \mid s_T \in T \wedge \text{actv}_{C \bowtie t}(e) = 1\}$.

As $\mathcal{M}$ and $\text{un}_{C \bowtie t}(\mathcal{M})$ share the set Z of observations, their possible strategies coincide, i.e., $\Sigma^{\mathcal{M}} = \Sigma^{\text{un}_{C \bowtie t}(\mathcal{M})}$. We formalise the relationship between the original POMDP and its unfolding POMDP regarding cost-bounded reachability.

Theorem 2 Given a POMDP $\mathcal{M}$, set $T \subseteq S$ and cost bound $(C \bowtie t)$, it holds that for all strategies $\sigma \in \Sigma^{\mathcal{M}}$: $\Pr_{\sigma}^{\mathcal{M}}(\Diamond_{C \bowtie t} T) = \Pr_{\sigma}^{\text{un}_{C \bowtie t}(\mathcal{M})}(\Diamond \text{actv}_{C \bowtie t}(T))$.

We get the following result about maximal probabilities.

Corollary 1 For POMDP $\mathcal{M}$, $T \subseteq S$ and $(C \bowtie t)$: $\Pr_{\max}^{\mathcal{M}}(\Diamond_{C \bowtie t} T) = \Pr_{\max}^{\text{un}_{C \bowtie t}(\mathcal{M})}(\Diamond \text{actv}_{C \bowtie t}(T))$.

Thus, to tackle CBR as in Problem 1, we can consider the unbounded indefinite-horizon reachability problem on the unfolding POMDP. Solution methods for this problem include smart exploration of the belief space [28, 8, 17] or the strategy space [3].

6 Cost-(Level-)Aware POMDPs

The general cost bound framework assumes that an agent’s decisions are solely based on environmental observations the agent receives. However, costs might reflect quantities—such as the level of a battery—that the agent observes. We refine our cost-bounded analysis for the special case where the observation model captures the additional information provided by costs. This notion of *cost-awareness* is related to the reward-based belief updates in [21].

Definition 6 (Cost-Aware POMDP) A POMDP $\mathcal{M} = \langle M, Z, \mathbf{O} \rangle$ with $M = \langle S, Act, \mathbf{P}, s_{init} \rangle$ is cost-aware with respect to a k -dimensional cost structure $\mathbf{C}$ if for all $z \in Z$ there is $c_z \in \mathbb{N}^k$ such that for any transition $s \xrightarrow{a} s'$ with $z \in \text{supp}(\mathbf{O}(s, a, s'))$ we have $\mathbf{C}(s, a, s') = c_z$.

In a cost-aware POMDP, all observations z can be assigned a cost vector c_z . An observation z only occurs at transitions that yield costs equal to c_z , effectively guaranteeing that the collected costs are observable. Cost-awareness implies that, given an observation trace $z_1 \dots z_n \in Z^*$, an agent can derive the costs $\sum_{i=1}^n c_{z_i}$ that have been accumulated so far. Thus, in the bound unfolding POMDP, an agent can always be certain about the cost epoch it is currently in as it has access to the history. This is reflected in the belief space: if the POMDP $\mathcal{M}$ is cost-aware, all reachable beliefs in the belief MDP of its unfolding $\text{bel}(\text{un}_{\mathbf{C} \bowtie t}(\mathcal{M}))$ only contain states that belong to the same epoch. Formally, if $\mathcal{M}$ is cost-aware, then for every reachable belief $b = \langle z, \mu \rangle$ of $\text{bel}(\text{un}_{\mathbf{C} \bowtie t}(\mathcal{M}))$ there is an epoch $e \in E_k$ such that $\text{supp}(\mu_b) \subseteq S \times \{e\}$. This enables an optimised analysis of $\text{bel}(\text{un}_{\mathbf{C} \bowtie t}(\mathcal{M}))$ as discussed in the sequel.

6.1 Sequential Epoch Analysis

Our approach so far is to analyse unbounded reachability for the unfolding POMDP $\text{un}_{\mathbf{C} \bowtie t}(\mathcal{M})$, e.g. via (abstractions of) its belief MDP $\text{bel}(\text{un}_{\mathbf{C} \bowtie t}(\mathcal{M}))$. However, such an analysis operates on the potentially large unfolding POMDP.

Recent works on fully observable MDPs avoid the construction of a large unfolding MDP by considering epochs one after another in a dynamic programming fashion [13, 15, 23]. This *sequential epoch analysis* is based on the epoch dependency graph $\langle E_k(t), \{ \langle e, e \ominus c \rangle \mid c \in \Gamma_{\mathbf{C}} \} \rangle$ which has an edge from epoch e to epoch e' iff a transition of the form $\langle s, e \rangle \xrightarrow{a} \langle s', e' \rangle$ exists in the unfolding POMDP. Since costs are non-negative, the epoch graph is acyclic (except for self-loops). The idea is to process epochs in a reversed topological order $e_0, e_1, \dots, e_n$ with $e_0 = \perp^k$ and $e_n = t$. For each considered epoch e_i , an *epoch MDP*—which is essentially the restriction of the bound unfolding MDP to states with epoch e —is constructed and analysed while propagating results from previous epochs $e_0, \dots, e_{i-1}$. Implementations can exploit similarities between different epoch MDPs. This way, the approach efficiently analyses properties of the large unfolding MDP without an explicit construction.

We lift sequential epoch analysis to POMDPs. In the general case, the POMDP dynamics do not allow a clear separation of epochs. In particular, when considering belief-based solution methods, beliefs may have states representing several different epochs in their support. We therefore focus on cost-aware POMDPs. Our approach is to perform sequential epoch analysis on the belief MDP—or a finite abstraction thereof. To this end, we lift cost bounds to the belief space. We fix a POMDP $\mathcal{M}$ and cost bound $(\mathbf{C} \bowtie t)$ such that $\mathcal{M}$ is cost-aware with respect to $\mathbf{C}$.

Definition 7 (Cost-Aware Belief Cost Bound) The belief cost structure for $\text{bel}(\mathcal{M})$ is $\mathbf{C}^B$ where for $s \in \mu_b, s' \in \mu_{b'}$, $\mathbf{C}^B((z, \mu_b), a, (z', \mu_{b'})) := \mathbf{C}(s, a, s')$. The belief cost bound is $\text{bel}(\mathbf{C} \bowtie t) = (\mathbf{C}^B \bowtie t)$.

$\mathbf{C}^B$ is well-defined as cost-awareness guarantees that $\mathbf{C}(s, a, s') = \mathbf{C}(q, a, q')$ for all $s, q \in \mu_b$ and $s', q' \in \mu_{b'}$. For cost-aware POMDPs, applying the cost unfolding and then constructing the belief MDP

is equivalent to *first* constructing the belief MDP and *then* applying cost unfolding. For MDPs M_1 and M_2 we write $M_1 \cong M_2$ iff the reachable fragments are isomorphic, i.e., equal up to renaming.

Theorem 3 $bel(un_{C \bowtie t}(\mathcal{M})) \cong un_{bel(C \bowtie t)}(bel(\mathcal{M}))$.

The sequential epoch analysis for MDPs outlined above can readily be applied to $bel(\mathcal{M})$ to show properties for its unfolding $un_{bel(C \bowtie t)}(bel(\mathcal{M}))$. Due to Theorem 3, analysis results immediately carry over to $bel(un_{C \bowtie t}(\mathcal{M}))$. Approaches like the ones described in [7, 8, 28] handle large or even infinite belief MDPs through abstraction, yielding a finite MDP $abstr(bel(\mathcal{M}))$ which over- or under-approximates the behaviour of $bel(\mathcal{M})$. If bound unfolding retains the abstraction—i.e., $abstr(bel(un_{C \bowtie t}(\mathcal{M}))) \cong un_{bel(C \bowtie t)}(abstr(bel(\mathcal{M})))$ —the sequential epoch analysis is compatible with such techniques. In our experiments, we use this observation for the cut-off abstraction of [8] and the discretisation of [7].

6.2 Cost Level Awareness

In many problem instances, neither full cost-awareness nor absolute unawareness are realistic. For example, a robot can have a rough estimate of its energy level (high, medium, low, empty) while not being aware of exactly how much energy it has spent already. We capture this notion of *cost levels* using functions that assign in each dimension a *level* according to the collected cost. We assume uniform levels, i.e., we change the level in dimension i whenever an additional cost of some fixed $d[i] > 0$ is collected. Moreover, zero cost constitutes its own level.

Definition 8 (Level Function) For $d \in (\mathbb{N} \setminus \{0\})^k$, the level function $lvl_d : \mathbb{N}^k \rightarrow \mathbb{N}^k$ is given by

$$lvl_d(c)[i] := \left\lceil \frac{c[i]}{d[i]} \right\rceil.$$

We fix a POMDP $\mathcal{M}$, cost bound $(C \bowtie t)$, target states T , and a level function lvl_d , and define a *level-aware* instance of CBR where in addition to the observations in $\mathcal{M}$, the agent can use information about the current cost level to make decisions. This level-aware instance is in general not equivalent to the CBR instance $\mathcal{M}$ with cost bounds $(C \bowtie t)$. It introduces new observations that provide additional information which is not observable in the original model. However, the level-aware instance may more accurately capture the scenario we are interested in as an agent may have access to such information. Our definition of a level-aware instance decouples the modelling of observations arising from the environment and those arising from possible cost observation. This simplifies the modelling of such instances and enables, for example, the comparison of different degrees of cost levels. By choosing $d = \langle 1, \dots, 1 \rangle$, we can define a variant of the original POMDP with *full* cost-awareness which we call the *cost-aware variant*. Selecting $d = t$ means that only activeness of bounds and the first collection of a non-zero cost can be observed for each dimension.

To incorporate cost level awareness into our framework, we present a transformation of $\mathcal{M}$, C and lvl_d into a new cost-aware POMDP (cf. Definition 6). The transformation encodes the cost that can still be collected until a new level is reached in the state space of the POMDP and introduces fresh observations to mark transitions in which one or more *level changes* occur. This way, an observation trace suffices to deduce the current level.

Definition 9 (Level Unfolding POMDP) The level unfolding with respect to lvl_d is the POMDP $lvl_d(\mathcal{M}) = \langle M_{lvl_d}, Z_{lvl_d}, O_{lvl_d} \rangle$ and the cost structure C_{lvl_d} with

- $M_{lvl_d} = \langle S_{lvl_d}, Act, P_{lvl_d}, \langle s_{init}, \langle 0, \dots, 0 \rangle \rangle \rangle$,
- $S_{lvl_d} = S \times \{ \ell \in \mathbb{N}^k \mid \forall i : \ell[i] < d[i] \}$,

- $Z_{lvl_d} = Z \times \{c \in \mathbb{N}^k \mid \forall i : c[i] \leq \lceil c_i^{\max}/d[i] \rceil\}$, where $c_i^{\max} = \max_{s,s' \in S, a \in Act} C(s, a, s')[i]$,
- $P_{lvl_d}(\langle s, \ell \rangle, a, \langle s', \ell' \rangle) = P(s, a, s')$ if for all i : $\ell'[i] = \ell[i] - C(s, a, s')[i] \bmod d$,
- $O_{lvl_d}(\langle s, \ell \rangle, a, \langle s', \ell' \rangle)(\langle z, c \rangle) = O(s, a, s')(z)$ if for all i : $c[i] = \lceil (C(s, a, s')[i] - \ell[i])/d[i] \rceil$,
- $C_{lvl_d}(\langle s, \ell \rangle, a, \langle s', \ell' \rangle) = C(s, a, s')$,

and P_{lvl_d} and O_{lvl_d} are zero in all other cases.

The CBR instance with POMDP $lvl_d(\mathcal{M})$, bounds $(C_{lvl_d} \bowtie t)$ and target states $T_d = (T \times \mathbb{N}^k) \cap S_{lvl_d}$ is the level-aware variant (w.r.t. lvl_d) for $\mathcal{M}$, $(C \bowtie t)$, and $T \subseteq S$.

When assuming that a level function captures when a bound becomes (in-)active—e.g. if an energy limit is exceeded—the level-aware instance can be reduced to an equivalent CBR instance on a *fully cost-aware* POMDP which can then be solved using the sequential approach. Mathematically, this is the case if for all i , $d[i]$ divides $t[i]$ (written $d[i] \mid t[i]$).

Theorem 4 Let $lvl_d(\mathcal{M})$ such that $\forall i : d[i] \mid t[i]$ and let $(L \bowtie t_d)$ be a cost bound for $lvl_d(\mathcal{M})$ with $\forall i$: $L(\langle s, \ell \rangle, a, \langle s', \ell' \rangle)[i] := \lceil (C(s, a, s')[i] - \ell[i])/d[i] \rceil$ and $t_d[i] := t[i]/d[i]$. Then, $lvl_d(\mathcal{M})$ is cost-aware w.r.t. L and

$$Pr_{\max}^{lvl_d(\mathcal{M})}(\Diamond_{C_{lvl_d} \bowtie t} T_d) = Pr_{\max}^{lvl_d(\mathcal{M})}(\Diamond_{L \bowtie t_d} T_d).$$

7 Empirical Evaluation

We evaluate the practicality of cost-bounded reachability (CBR) analysis in POMDPs to answer the following questions:

- (Q1) Are the presented approaches suitable for solving cost-bounded reachability problems in practice?
- (Q2) How is this influenced by cost (level) awareness (as discussed in Section 6.2)?
- (Q3) Does applying the sequential approach to cost-aware belief MDPs (cf. Section 6.1) improve performance compared to reducing the problem to unbounded reachability?

7.1 Implementation

Our evaluation employs a prototypical extension of the model-checking tool **STORM**³ [16], which supports the analysis of systems involving random or probabilistic phenomena. Given an input model and a quantitative specification, it can determine whether the model conforms to the specification. Here, models comprise both discrete- and continuous-time as well as deterministic and non-deterministic Markov models, such as Discrete-Time Markov Chains, Markov Decision Processes, or Markov Automata. **STORM** is publicly available for download⁴ and offers a python API named **Stormpy**⁵ to facilitate interfacing with other code bases.

The current official release can analyse unbounded reachability in POMDPs via finite abstractions of the belief MDP [7, 8] as well as cost-bounded reachability in (fully observable) MDPs via the sequential approach from [15].

The extension to handle CBR queries for POMDPs, as required for our evaluation, supports the construction of bound unfolding and level unfolding of POMDPs, along with the exploration of cost-aware

³<https://www.stormchecker.org/>

⁴<https://www.stormchecker.org/documentation/obtain-storm/build.html>

⁵<https://moves-rwth.github.io/stormpy/>

belief MDPs with belief cost bounds. The corresponding implementations are integrated into the existing POMDP verification framework of STORM and will become part of the official release in the course of 2026, enabling us to tackle CBR problems in three different *configurations*:

- **UNFOLD**: Transform to an unbounded reachability problem on the unfolding POMDP (see Section 5), then verify (finite abstractions of) its belief MDP.
- **CA-UNFOLD**: Construct a cost-aware variant of the POMDP (see Section 6.2), then analyse it as in UNFOLD.
- **CA-BEL-SEQ**: Construct a cost-aware variant of the POMDP, then construct (a finite abstraction of) its cost-aware belief MDP and analyse CBR on this fully observable MDP using the sequential epoch approach (see Section 6.1).

CA-UNFOLD and CA-BEL-SEQ both first construct a cost-aware variant based on the input POMDP, extending it with additional observations which in general changes the optimal achievable value. We opt for this method of defining cost-aware instances to decouple the modelling of environmental observations and those observations stemming from costs. CA-UNFOLD and CA-BEL-SEQ differ in the underlying solution approach. Instead of full cost-awareness, they can also be used with cost level awareness by applying a level unfolding first. Moreover, cost-awareness can also be induced for only a subset of the dimensions $I \subseteq \{1, \dots, k\}$. STORM uses a state-based observation model $\mathbf{O} : S \rightarrow Z$. We transform our observation model into such a state-based one by encoding observations in the state space [12]. For CA-UNFOLD and CA-BEL-SEQ, this results in larger state spaces compared to the original POMDP due to the additional observations.

For each of the three configurations, we can either use the *cut-off* approach of [8] or the *discretisation* approach of [7] to obtain a finite abstraction of the belief MDP yielding sound lower or upper bounds for the optimal cost-bounded reachability probabilities. Both abstractions—respectively denoted below by CUT and DISCR—have a hyper-parameter that controls the size of the obtained belief MDP abstractions. We conducted experiments on Intel Xeon 8468 Sapphire systems (2.1 GHz) with memory limited to 64 GB. STORM runs on a single core. For each combination of benchmark instance, configuration, and abstraction type, we considered 25 different hyper-parameter assignments to capture different trade-offs between approximation accuracy and computational tractability. A time limit of 1800 seconds (walltime) was applied for the individual runs.

7.2 Benchmarks

Since there is no established benchmark set for CBR problems, we use partially observable variants of some cost-bounded reachability problems from [15] (resrc, rover, serv). In addition, we consider three variants of grid world examples where reaching a goal is made difficult by either an incline (incline), obstacles (obstcl), water levels (water), or uncertain movements (walk40, walk120). Finally, we incorporate our motivating example (clean6) and a version with twelve tiles (clean12). The benchmarks are given in the guarded command language of PRISM [25].

Table 1 contains more information about the specific instances we consider, in particular the bounds on the different cost structures and the resulting number of epochs $|E|$. The bounds are chosen such that the resulting instances are challenging for the implementation while still resulting in non-trivial values for most configuration.

clean This model is a generalised version of the cleaning robot scenario described in Section 2. A robot is placed in position 0 of a hallway consisting of N tiles, all initially dirty. In each step, it can decide between moving to the next tile, increasing its position by one, or attempting to clean the tile at the current position. A cleaning attempt is successful with probability 0.8. The robot can repeatedly

Model	$ S $	Bounds	$ E $
clean6	37	$C[1] \leq 60, C[2] > 5$	413
clean12	73	$C[1] \leq 120, C[2] > 11$	1508
incline	25	$C[1] \leq 75, C[2] \leq 21$	497
obstcl	25	$C[1] \leq 25, C[2] \leq 7$	83
resrc	721	$C[1] > 4, C[2] > 4, C[3] \leq 60$	2107
	721	$C[1] > 14, C[2] > 14, C[3] \leq 180$	$4 \cdot 10^4$
rover	16	$C[1] > 199, C[2] \leq 360, C[3] \leq 200$	$7 \cdot 10^5$
		$C[1] > 599, C[2] \leq 1080, C[3] \leq 600$	$2 \cdot 10^7$
serv	$8 \cdot 10^4$	$C[1] \leq 570$	40
		$C[1] \leq 1000$	68
walk40	84	$C[1] \leq 80$	82
walk120	244	$C[1] \leq 80$	82
water	34	$C[1] \leq 590, C[2] > 49$	$3 \cdot 10^4$
		$C[1] \leq 1790, C[2] > 149$	$3 \cdot 10^5$

Table 1: Overview of Benchmarking Instances

attempt to clean the same tile. Moving always consumes one unit of energy, while a cleaning attempt requires either two or four units, each with probability 0.5. If the robot moves to position N , i.e., out of the hallway, it has reached its target position. The robot can observe its position, but generally does not observe how many tiles it has already cleaned.

We use two cost dimensions to model the energy consumption of the robot and the amount of tiles it has successfully cleaned. The objective is to reach the target position with a specified bound on the energy while successfully cleaning at least a specified amount of tiles. We consider two sizes, $N = 6$ and $N = 12$.

incline The agent has to reach a target in a 5×5 grid of cells. The agent can move in any of the four cardinal directions. Each move is either *uphill*, *downhill* or neither (*flat*). With probability 0.5, an attempted move fails due to slipping. If the attempted move is uphill, the agent is staying in place if it slips. Similarly, slipping downhill causes the agent to overshoot and move a cell further in the chosen direction. With a flat move, slipping has no effect. In case a move would lead out of bounds, the agent moves as far as possible in the direction and then stays in place. The agent does not observe its current position, but knows its starting position in the south-west corner of the grid. We consider a map where the incline is such that all moves north and east are uphill and all moves south and west are downhill. We consider a cost model where downhill steps consume one unit of energy, flat moves two units and uphill moves three units. The objective is to reach the target in the north-east corner of the grid. We are interested in reaching the goal within the energy budget (modelled in the first dimension) and a bounded number of steps (modelled in the second dimension).

obstcl Similar to incline, the agent is supposed to reach a target in a 5×5 grid of cells. Only the outermost ring of cells can be traversed freely, the inner ring contains light forest and the center cell is dense forest. The agent cannot observe its position and it slips, i.e., stays in place when executing a move from light forest or dense forest with probability 0.25 or 0.5 respectively. Moves from free tiles, light forest tiles and dense forest tiles consume one, two or three units of energy each. Like before, the objective is to reach the goal within the energy budget and a bounded number of steps (modelled by a

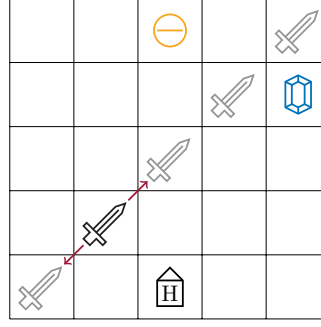


Figure 2: Resource Gathering Scenario

Task	Time	Energy	Sci. Value	Success Prob.
1	10	{3, 5}	10	$1/2$
2	5	{5, 7}	10	$3/5$
3	5	3	2	$4/5$
4	10	7	30	$1/10$

Table 2: Values of the Tasks in the rover Problem

bound on $C[1]$ and $C[2]$ respectively).

resrc This benchmark is a variant of the resource gathering model from [6]. The scenario is similar to a problem arising in many strategy games. We provide a depiction of the setting in Figure 2. An agent, starting in a home base marked by H , is tasked with collecting two kinds of resources – *gold* (depicted by a coin) and *gems* (depicted by a gem) – in a grid environment. In every step, the agent can move in any of the four cardinal directions. When it reaches either the gold or the gem location, it picks up one unit of the respective resource. To collect it, the agent needs to return the resource to the base. The agent can hold at most one unit of each resource.

An *enemy* patrols the diagonal of the grid. It starts in the south-west corner and changes its position to one of the adjacent locations on the diagonal with each step of the agent. If the enemy is in one of the corners, it will certainly move to the adjacent location. When the agent and the enemy enter the same location at the same time, the agent loses all currently held resources and is teleported home without collecting anything. The objective is to maximise the probability to collect a minimum amount of each resource (lower bounds on $C[1]$ for gold and $C[2]$ for gems) within a given step bound (upper bound on $C[3]$).

rover A partially observable version of the Mars rover task scheduling problem described in [15] based on [10]. The problem models the scheduling of a variety of experiments on Mars. Experiments have differing time and energy consumption and success probabilities. Upon success of an experiment, a certain scientific value is collected. Energy consumption for some tasks is subject to uncertainty; the consumed energy has a high or low value with probability 0.5 each. The specific parameters for each task are given in Table 2.

The agent can schedule several experiments each day. It does not directly observe whether a task has been successful or not. The objective is to maximise the probability of achieving at least a certain cumulative scientific value without exceeding both time and energy limits. Scientific values are modelled in $C[1]$, while time and energy are modelled in $C[2]$ and $C[3]$ respectively.

serv The *service* domain is a partially observable variant of the care home scenario described in [26]. A robot is navigating a care home environment consisting of a central hallway with rooms adjacent to each side of the hallway. Overall, the map consists of 21 locations. The robot does not observe its

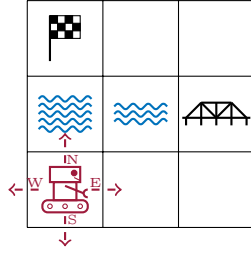
	Energy Cost	
	Tile	High Normal
	Deep Water	10 6
	Shallow Water	6 2
	Bridge	1 1
	Land	1 1

Figure 3: Robot Navigation Task with Water Obstacles

location and each movement to an adjacent location can fail with probability 0.01.

The robot's routine consists of the main task of checking whether three occupants of specified rooms want water and deliver it if it is desired. The robot can collect bottles of water at a central kitchen area. At any point, it can only carry at most two bottles. For delivery, the robot first has to check whether the occupant wants the water. With a probability of 0.2, this is the case and the robot can deliver it. In addition, the robot has the secondary task of interacting with four designated occupants. Each action of the robot requires a certain amount of time. The central area, where the robot starts and which it has to cross to get to the kitchen, is crowded with a probability of 0.2 every time the robot enters it, causing an additional time cost.

We are interested in the probability that the robot delivers water to where it is required within the time limit (bound on $C[1]$).

water The *water* problem considers an amphibious robot starting in the south-west corner of a regular grid consisting of nine cells. In every step, the robot moves in any of the four cardinal directions, where a move out of bounds makes the robot stay in place. The task is to visit the flag in the north-west corner and then return to the initial position multiple times. A river runs through the center row of the map, modelled as one tile of *deep water*, one tile of *shallow water*, and one *bridge* tile. All other tiles are *land* tiles. With a probability of 0.5, there are *high water* conditions, making crossing the river without the bridge more difficult. The robot is not able to observe the water conditions. The energy consumption for each move depends on the current tile and the water conditions as outlined on the right of Figure 3. A move out-of-bounds still consumes the respective energy.

The agent collects one unit of cost/reward (modelled in dimension $C[2]$) every time it reaches the target position. The objective is to compute the probability that a minimum number of trips (bound on $C[2]$) is completed within the given energy budget (bound on dimension $C[2]$).

walk An agent is starting at position 0 of a hallway consisting of $N+1$ positions. In each step, it can decide to either attempt to *observe* its current position i for a cost of one unit, to *move* to the adjacent position $i+1$, or to completely *stop* the process. A move action fails with probability 0.5, resulting in the agent staying at position i . In position N , a move will always make the agent stay in place. The agent is only able to observe its current position if it executes an *observe* action. However, *observe* actions are unreliable and fail with probability 0.9.

The goal of the agent is to stop the process exactly in position $N-1$, leading to a target state. If it stops the process in any other position, it is trapped and cannot fulfil its goal any more.

The cost-bounded objective is to reach the target state, i.e., stopping in position $N-1$, while using at most a specified number of *observe* actions. We consider two sizes, $N = 40$ and $N = 120$.

7.3 Results

Table 3 lists the best value bounds obtained within the time and memory limit as well as the time it took to obtain these bounds. *CUT* yields a *lower bound* on the optimal value, and *DISCR* yields an *upper*

Model	E	S _{un}	UNFOLD: CUT / DISCR		CA-UNFOLD: CUT / DISCR		CA-BEL-SEQ: CUT / DISCR	
clean6	413	3809	0.86 (<1s)	1 (713s)	0.929 (3.9s)*	0.971 (299s)	0.929 (2.2s)	0.949 (<1s)
clean12	1508	3·10 ⁴	0.77 (262s)	1 (61.9s)	0.708 (240s)	1 (128s)	0.88 (381s)	0.957 (4.8s)
incline	497	2094	0.989 (21.5s)	0.989 (<1s)	0.989 (1.1s)	0.989 (<1s)	0.989 (<1s)	0.989 (<1s)
obstcl	83	741	0.87 (<1s)*	0.87 (<1s)	0.87 (<1s)	0.87 (<1s)	0.87 (<1s)	0.87 (<1s)
resrc	2107	2·10 ⁵	7·10 ⁻¹⁵ (360s)	0.0312 (2.7s)	7·10 ⁻¹⁵ (345s)	0.0312 (3.6s)	0.0312 (<1s)	0.0312 (<1s)
resrc	4·10 ⁴	6·10 ⁶	2·10 ⁻⁷³ (400s)	3·10 ⁻⁵ (81.1s)	2·10 ⁻⁷³ (427s)	3·10 ⁻⁵ (101s)	3·10⁻⁵ (16.3s)	3·10⁻⁵ (4.4s)
rover	7·10 ⁵	1·10 ⁷	0.353 (337s)	0.853 (237s)	0.861 (462s)*	0.861 (406s)	0.861 (12.2s)*	0.861 (12.4s)
rover	2·10 ⁷	?	-	-	-	-	0.951 (466s)*	0.951 (456s)
serv	40	10·10 ⁴	0.0474 (111s)	0.378 (1.9s)	0.0474 (107s)	0.378 (6.4s)	-	0.378 (15.4s)
serv	68	3·10 ⁵	0.172 (282s)	0.636 (158s)	0.169 (281s)	0.636 (528s)	-	0.637 (139s)
walk40	82	6847	0.916 (97.3s)*	0.932 (1603s)	0.916 (174s)*	0.935 (1431s)	0.916 (<1s)	0.93 (1295s)
walk120	82	2·10 ⁴	0.867 (846s)	0.931 (542s)	0.869 (1681s)	0.931 (1309s)	0.895 (11.0s)*	0.926 (727s)
water	3·10 ⁴	6·10 ⁵	3·10 ⁻¹²³ (327s)	1 (144s)	0.166 (17.9s)*	0.166 (17.5s)	0.166 (<1s)*	0.166 (<1s)
water	3·10 ⁵	5·10 ⁶	-	-	0.181 (268s)*	0.181 (269s)	0.181 (4.2s)*	0.181 (4.3s)

Table 3: Overview of Obtained Value Bounds and Runtimes

bound V^U (see Problem 1). Table entries are bold-faced when they depict the tightest lower or upper bound obtained within the fastest runtime. This does not include UNFOLD as it computes a different measure. A dash (-) indicates that no non-trivial bound was obtained within the time limit. Column $|S_{un}|$ denotes the number of reachable states in the unfolding POMDP (if known). For CUT, an asterisk (*) indicates that the belief MDP was fully explored.

The plots at the top of Figure 4 show the obtained value bounds over analysis time for two selected instances. A data point $\langle x, y \rangle$ for configuration z means that (lower or upper) value bound y was established within x seconds using configuration z and appropriate hyper-parameters. Similarly, the plots at the bottom show obtained value bounds for our motivating example clean6, on the left when increasing the number of observable energy levels (intuitively providing more information for a strategy) and on the right when increasing the energy budget.

7.4 Discussion

Concerning (Q1), we see that all considered approaches produce non-trivial bounds on the optimal value for the majority of benchmark instances. Towards (Q2), we observe that the true optimal values in cost-(level-)aware variants of POMDPs is always at least as good as in the original POMDP. Our results show that cost-awareness does not affect the obtained upper value bounds for many of our benchmarks. However, the obtained lower bounds are often larger with cost-awareness enabled, resulting in tighter gaps between lower and upper bounds. As we expect the maximal value to increase under cost-awareness, this indicates that the discretisation method works better for obtaining tight approximations when having more information about incurred costs. This is also indicated by the data shown in the bottom left of Figure 4. With an increasing number of observable cost levels within the cost threshold $t[0]$, the obtained upper bounds decrease, while we expect the actual (unknown) maximal probability to increase when more information is available to the strategy, pointing towards tighter upper bounds. In addition, we see that with increasingly finer levels, the obtained values indeed increase, hinting at an increasingly higher true optimal value. An exception to our general observation is clean12 where CA-UNFOLD yields a smaller value than UNFOLD for CUT. This is a result of the state space increase for cost-aware variants caused by STORM’s state-based observation model. The exploration of the cost-aware variant’s belief space appears to be not as thorough as for the original POMDP within our given time limit. This results in a worse approximation (smaller lower bound) of a larger optimal value. Regarding (Q3), CA-BEL-SEQ often yields better approximations in less time compared to CA-UNFOLD. A notable exception is the serv case study. Here, the unfolding POMDP has similar size

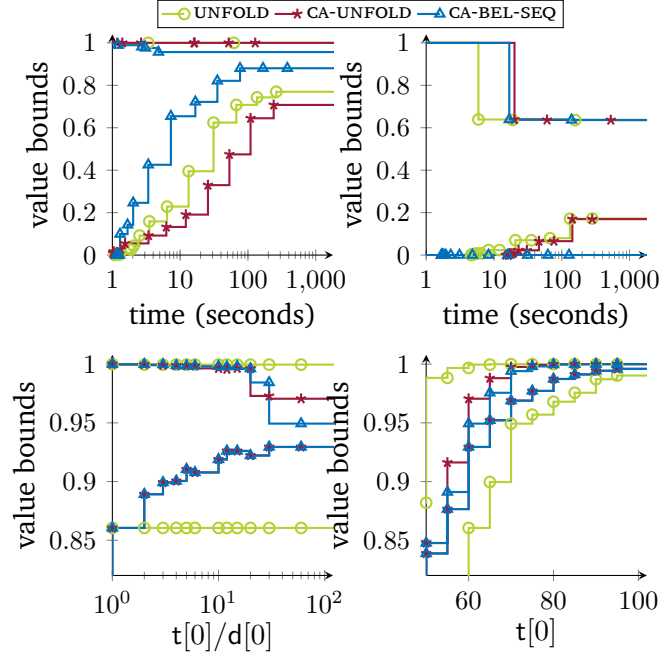


Figure 4: Value bounds obtained over time (top left: clean12, $|E|=1508$; top right: serv, $|E|=40$), for different observation levels (bottom left: clean6, $|E|=413$), and for increasing cost thresholds (bottom right: clean6, $168 \leq |E| \leq 658$).

compared to the original POMDP ($|S_{un}| \approx |S|$) as we have comparably few reachable epochs.

8 Conclusion

This deliverable presents a library of efficient synthesis techniques addressing uncertainty in resource dynamics. This library has been implemented as part of the development of strategy synthesis techniques for partially observable stochastic resource-constrained systems. It is based on a general framework with solid formal foundations. This framework has the cost-bounded reachability (CBR) problem at its core, which asks for the maximal probability to reach a set of states in a partially observable Markov decision process via paths that respect (multi-dimensional) bounds on the cumulative costs. These objectives can be tackled by considering an unbounded objective on an unfolding of the POMDP. Observing incurred costs enables advanced techniques based on a sequential analysis of cost epochs on the belief MDP. Awareness of cost levels provides a more realistic way to model certain scenarios and can often be reduced to the cost-aware setting.

Our experiments using a prototype implementation in the probabilistic model-checking tool STORM demonstrate the suitability of cost-bounded reachability analysis for practical applications. This implementation is currently being integrated into the existing POMDP verification framework of STORM and will become part of the official release in the course of 2026.

Moreover, as future research we propose the extension of our framework towards cost-bounded expected reward objectives and a more flexible notion of cost levels. Additionally, developing more refined approximation methods tailored towards the cost-bounded setting is useful for increasing scalability.

References

- [1] S. Andova, H. Hermanns, and J. Katoen. Discrete-time rewards model-checked. In *Formal Modeling and Analysis of Timed Systems: First International Workshop, FORMATS 2003*, volume 2791 of *Lecture Notes in Computer Science*, pages 88–104. Springer, 2003. doi:[10.1007/978-3-540-40903-8_8](https://doi.org/10.1007/978-3-540-40903-8_8).
- [2] R. Andriushchenko, A. Bork, M. Ceska, S. Junges, J. Katoen, and F. Macák. Search and explore: Symbiotic policy synthesis in POMDPs. In *Computer Aided Verification – 35th International Conference, CAV 2023*, volume 13966 of *Lecture Notes in Computer Science*, pages 113–135. Springer, 2023. doi:[10.1007/978-3-031-37709-9_6](https://doi.org/10.1007/978-3-031-37709-9_6).
- [3] R. Andriushchenko, M. Ceska, S. Junges, and J. Katoen. Inductive synthesis of finite-state controllers for POMDPs. In *Thirty-Eighth Conference on Uncertainty in Artificial Intelligence, UAI 2022*, volume 180 of *Proceedings of Machine Learning Research*, pages 85–95. PMLR, 2022. URL: <https://proceedings.mlr.press/v180/andriushchenko22a.html>.
- [4] C. Baier, M. Daum, C. Dubslaff, J. Klein, and S. Klüppelholz. Energy-utility quantiles. In *NASA Formal Methods - 6th International Symposium, NFM 2014*, volume 8430 of *Lecture Notes in Computer Science*, pages 285–299. Springer, 2014. doi:[10.1007/978-3-319-06200-6_24](https://doi.org/10.1007/978-3-319-06200-6_24).
- [5] C. Baier and J. Katoen. *Principles of Model Checking*. MIT Press, 2008.
- [6] L. Barrett and S. Narayanan. Learning all optimal policies with multiple criteria. In *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008)*, volume 307 of *ACM International Conference Proceeding Series*, pages 41–47. ACM, 2008. doi:[10.1145/1390156.1390162](https://doi.org/10.1145/1390156.1390162).
- [7] A. Bork, S. Junges, J. Katoen, and T. Quatmann. Verification of indefinite-horizon POMDPs. In *Automated Technology for Verification and Analysis - 18th International Symposium, ATVA 2020*, volume 12302 of *Lecture Notes in Computer Science*, pages 288–304. Springer, 2020. doi:[10.1007/978-3-030-59152-6_16](https://doi.org/10.1007/978-3-030-59152-6_16).
- [8] A. Bork, J. Katoen, and T. Quatmann. Under-approximating expected total rewards in POMDPs. In *Tools and Algorithms for the Construction and Analysis of Systems - 28th International Conference, TACAS 2022*, volume 13244 of *Lecture Notes in Computer Science*, pages 22–40. Springer, 2022. doi:[10.1007/978-3-030-99527-0_2](https://doi.org/10.1007/978-3-030-99527-0_2).
- [9] A. Bork, J.-P. Katoen, T. Quatmann, and S. Stein. Multi-cost-bounded reachability analysis of POMDPs. In *The 41st Conference on Uncertainty in Artificial Intelligence*, 2025. URL: <https://openreview.net/forum?id=1005cyJ3qm>.
- [10] J. L. Bresina, A. K. Jónsson, P. H. Morris, and K. Rajan. Activity planning for the Mars exploration rovers. In *Proceedings of the Fifteenth International Conference on Automated Planning and Scheduling (ICAPS 2005)*, pages 40–49. AAAI, 2005. URL: <http://www.aaai.org/Library/ICAPS/2005/icaps05-005.php>.
- [11] I. Chades, J. Carwardine, T. G. Martin, S. Nicol, R. Sabbadin, and O. Buffet. MOMDPs: A solution for modelling adaptive management problems. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pages 267–273. AAAI Press, 2012. doi:[10.1609/AAAI.V26I1.8171](https://doi.org/10.1609/AAAI.V26I1.8171).
- [12] K. Chatterjee, M. Chmelik, R. Gupta, and A. Kanodia. Optimal cost almost-sure reachability in POMDPs. *Artif. Intell.*, 234:26–48, 2016. doi:[10.1016/J.ARTINT.2016.01.007](https://doi.org/10.1016/J.ARTINT.2016.01.007).

- [13] E. M. Hahn and A. Hartmanns. A comparison of time- and reward-bounded probabilistic model checking techniques. In *Dependable Software Engineering: Theories, Tools, and Applications - Second International Symposium, SETTA 2016*, volume 9984 of *Lecture Notes in Computer Science*, pages 85–100, 2016. doi:10.1007/978-3-319-47677-3_6.
- [14] A. Harris, T. Valade, T. Teil, and H. Schaub. Generation of spacecraft operations procedures using deep reinforcement learning. *Journal of Spacecraft and Rockets*, 59(2):611–626, 2022. doi:10.2514/1.A35169.
- [15] A. Hartmanns, S. Junges, J. Katoen, and T. Quatmann. Multi-cost bounded tradeoff analysis in MDP. *J. Autom. Reason.*, 64(7):1483–1522, 2020. doi:10.1007/S10817-020-09574-9.
- [16] C. Hensel, S. Junges, J. Katoen, T. Quatmann, and M. Volk. The probabilistic model checker Storm. *Int. J. Softw. Tools Technol. Transf.*, 24(4):589–610, 2022. doi:10.1007/S10009-021-00633-Z.
- [17] Q. H. Ho, M. S. Feather, F. Rossi, Z. N. Sunberg, and M. Lahijanian. Sound heuristic search value iteration for undiscounted POMDPs with reachability objectives. In *40th Conference on Uncertainty in Artificial Intelligence, UAI 2024*, 2024. URL: <https://openreview.net/forum?id=3zSiuXYtqf>.
- [18] K. Horák, B. Bosanský, and K. Chatterjee. Goal-HSVI: Heuristic search value iteration for goal POMDPs. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018*, pages 4764–4770. ijcai.org, 2018. doi:10.24963/IJCAI.2018/662.
- [19] P. Hou, W. Yeoh, and P. Varakantham. Solving risk-sensitive POMDPs with and without cost observations. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 3138–3144. AAAI Press, 2016. doi:10.1609/AAAI.V30I1.10402.
- [20] J. D. Isom, S. P. Meyn, and R. D. Braatz. Piecewise linear dynamic programming for constrained POMDPs. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008*, pages 291–296. AAAI Press, 2008. URL: <http://www.aaai.org/Library/AAAI/2008/aaai08-046.php>.
- [21] M. T. Izadi and D. Precup. Using rewards for belief state updates in partially observable Markov decision processes. In *Machine Learning: ECML 2005, 16th European Conference on Machine Learning*, volume 3720 of *Lecture Notes in Computer Science*, pages 593–600. Springer, 2005. doi:10.1007/11564096_58.
- [22] K. C. Kalagarla, D. Kartik, D. Shen, R. Jain, A. Nayyar, and P. Nuzzo. Optimal control of logically constrained partially observable and multiagent Markov decision processes. *IEEE Trans. Autom. Control.*, 70(1):263–277, 2025. doi:10.1109/TAC.2024.3422213.
- [23] J. Klein, C. Baier, P. Chrszon, M. Daum, C. Dubslaff, S. Klüppelholz, S. Märcker, and D. Müller. Advances in probabilistic model checking with PRISM: variable reordering, quantiles and weak deterministic Büchi automata. *Int. J. Softw. Tools Technol. Transf.*, 20(2):179–194, 2018. doi:10.1007/S10009-017-0456-3.
- [24] H. Kurniawati, D. Hsu, and W. S. Lee. SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In *Robotics: Science and Systems IV*. The MIT Press, 2008. doi:10.15607/RSS.2008.IV.009.
- [25] M. Z. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0: Verification of probabilistic real-time systems. In *Computer Aided Verification – 23rd International Conference, CAV 2011*, volume 6806 of *Lecture Notes in Computer Science*, pages 585–591. Springer, 2011. doi:10.1007/978-3-642-22110-1_47.

- [26] B. Lacerda, D. Parker, and N. Hawes. Multi-objective policy generation for mobile robots under probabilistic time-bounded guarantees. In *Proceedings of the Twenty-Seventh International Conference on Automated Planning and Scheduling, ICAPS 2017*, pages 504–512. AAAI Press, 2017. URL: <https://aaai.org/ocs/index.php/ICAPS/ICAPS17/paper/view/15751>.
- [27] E. Miehling, M. Rasouli, and D. Teneketzis. A POMDP approach to the dynamic defense of large-scale cyber networks. *IEEE Trans. Inf. Forensics Secur.*, 13(10):2490–2505, 2018. doi:10.1109/TIFS.2018.2819967.
- [28] G. Norman, D. Parker, and X. Zou. Verification and control of partially observable probabilistic systems. *Real Time Syst.*, 53(3):354–402, 2017. doi:10.1007/S11241-017-9269-4.
- [29] Y. Ohtsubo. Optimal threshold probability in undiscounted Markov decision processes with a target set. *Appl. Math. Comput.*, 149(2):519–532, 2004. doi:10.1016/S0096-3003(03)00158-9.
- [30] P. Poupart, A. Malhotra, P. Pei, K. Kim, B. Goh, and M. Bowling. Approximate linear programming for constrained partially observable Markov decision processes. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 3342–3348. AAAI Press, 2015. doi:10.1609/AAAI.V29I1.9655.
- [31] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics. Wiley, 1994. doi:10.1002/9780470316887.
- [32] M. Randour, J. Raskin, and O. Sankur. Percentile queries in multi-dimensional Markov decision processes. *Formal Methods Syst. Des.*, 50(2-3):207–248, 2017. doi:10.1007/S10703-016-0262-7.
- [33] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson, 4th edition, 2020. URL: <http://aima.cs.berkeley.edu/>.
- [34] P. Santana, S. Thiébaux, and B. C. Williams. RAO*: An algorithm for chance-constrained POMDPs. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 3308–3314. AAAI Press, 2016. doi:10.1609/AAAI.V30I1.10423.
- [35] G. Shani, J. Pineau, and R. Kaplow. A survey of point-based POMDP solvers. *Autonomous Agents and Multi-Agent Systems*, 27(1):1–51, 2013. doi:10.1007/S10458-012-9200-2.
- [36] R. D. Smallwood and E. J. Sondik. The optimal control of partially observable Markov processes over a finite horizon. *Oper. Res.*, 21(5):1071–1088, 1973. doi:10.1287/OPRE.21.5.1071.
- [37] T. Smith and R. G. Simmons. Heuristic search value iteration for POMDPs. In *20th Conference in Uncertainty in Artificial Intelligence, UAI 2004*, pages 520–527. AUAI Press, 2004.
- [38] M. T. J. Spaan and N. Vlassis. A point-based POMDP algorithm for robot planning. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation, ICRA 2004*, pages 2399–2404. IEEE, 2004. doi:10.1109/ROBOT.2004.1307420.
- [39] G. F. Stock, A. Haberl, J. A. Fraire, and H. Hermanns. POMDP-based routing for DTNs with partial knowledge and dependent failures. In *2025 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*, pages 1–6. IEEE, 2025. doi:10.1109/WiSEE57913.2025.11229850.
- [40] A. Undurti and J. P. How. An online algorithm for constrained POMDPs. In *IEEE International Conference on Robotics and Automation, ICRA 2010*, pages 3966–3973. IEEE, 2010. doi:10.1109/ROBOT.2010.5509743.

- [41] B. Wu, M. Ahmadi, S. Bharadwaj, and U. Topcu. Cost-bounded active classification using partially observable Markov decision processes. In *2019 American Control Conference, ACC 2019*, pages 1216–1223. IEEE, 2019. [doi:10.23919/ACC.2019.8814415](https://doi.org/10.23919/ACC.2019.8814415).