Under-Approximating Expected Total Rewards in POMDPs*

Alexander Bork[1](✉), Joost-Pieter Katoen[1], and Tim Quatmann[1]

RWTH Aachen University, Aachen, Germany
alexander.bork@cs.rwth-aachen.de

Abstract We consider the problem: is the optimal expected total reward to reach a goal state in a partially observable Markov decision process (POMDP) below a given threshold? We tackle this—generally undecidable—problem by computing under-approximations on these total expected rewards. This is done by abstracting finite unfoldings of the infinite belief MDP of the POMDP. The key issue is to find a suitable under-approximation of the value function. We provide two techniques: a simple (cut-off) technique that uses a good policy on the POMDP, and a more advanced technique (belief clipping) that uses minimal shifts of probabilities between beliefs. We use mixed-integer linear programming (MILP) to find such minimal probability shifts and experimentally show that our techniques scale quite well while providing tight lower bounds on the expected total reward.

1 Introduction

The relevance of POMDPs. Partially observable Markov decision processes (POMDPs) originated in operations research and nowadays are a pivotal model for planning in AI [40]. They inherit all features of classical MDPs: each state has a set of discrete probability distributions over the states and rewards are earned when taking transitions. However, states are *not* fully observable. Intuitively, certain aspects of the states can be identified, such as a state's colour, but states themselves cannot be observed. This partial observability reflects, for example, a robot's view of its environment while only having the limited perspective of its sensors at its disposal. The main goal is to obtain a policy—a plan how to resolve the non-determinism in the model—for a given objective. The key problem here is that POMDP policies must base their decisions *only* on the observable aspects (e.g. colours) of states. This stands in contrast to policies for MDPs which can make decisions dependent on the entire history of *full* state information.

Analysing POMDPs. Typical POMDP planning problems consider either finite-horizon objectives or infinite-horizon objectives under discounting. Finite-horizon objectives focus on reaching a certain goal state (such as *"the robot has collected all items"*) within a given number of steps. For infinite horizons, no step bound

* This work is funded by the DFG RTG 2236 "UnRAVeL".

D. Fisman and G. Rosu (Eds.): TACAS 2022, LNCS 13244, pp. 22–40, 2022.
https://doi.org/10.1007/978-3-030-99527-0_2

is provided and typically rewards along a run are weighted by a discounting factor that indicates how much immediate rewards are favoured over more distant ones. Existing techniques to treat these objectives include variations of value iteration [46,36,20,18,52,53] and policy trees [29]. Point-based techniques [38,42] approximate a POMDP's value function using a finite subset of beliefs which is iteratively updated. Algorithms include *PBVI* [38], *Perseus* [48], *SARSOP* [30] and *HSVI* [45]. Point-based methods can treat large POMDPs for both finite- and discounted infinite-horizon objectives [42].

Problem statement. In this paper we consider the problem: *is the maximal expected total reward to reach a given goal state in a POMDP below a given threshold?* We thus consider an infinite-horizon objective *without* discounting—also called an *indefinite-horizon* objective. A specific instance of the considered problem is the reachability probability to eventually reach a given goal state in a POMDP. This problem is undecidable [33,34] in general. Intuitively, this is due to the fact that POMDP policies need to consider the entire (infinite) observation history to make optimal decisions. For a POMDP, this notion is captured by an infinite, fully observable MDP, its *belief MDP*. This MDP is obtained from observation sequences inducing probabilities of being in certain states of the POMDP.

Previously proposed methods to solve the problem are e.g. to use approximate value iteration [22], optimisation and search techniques [1,12], dynamic programming [6], Monte Carlo simulation [43], game-based abstraction [51], and machine learning [13,14,19]. Other approaches restrict the memory size of the policies [35]. The synthesis of (possibly randomised) finite-memory policies is ETR-complete[1] [28]. Techniques to obtain finite-memory policies use e.g. parameter synthesis [28] or satisfiability checking and SMT solving [15,50].

Our approach. We tackle the aforementioned problem by computing under-approximations on maximal total expected rewards. This is done by considering finite unfoldings of the infinite belief MDP of the POMDP, and then applying abstraction. The key issue here is to find a suitable under-approximation of the POMDP's value function. We provide two techniques: a simple (cut-off) technique that uses a good policy on the POMDP, and a more advanced technique (belief clipping) that uses minimal shifts of probabilities between beliefs and can be applied on top of the simple approach. We use mixed-integer linear programming (MILP) to find such minimal probability shifts. Cut-off techniques for indefinite-horizon objectives have been used on computation trees—rather than on the belief MDP as used here—in *Goal-HSVI* [24]. Belief clipping amends the probabilities in a belief to be in a state of the POMDP yielding discretised values, i.e. an abstraction of the probability range $[0, 1]$ is applied. Such grid-based approximations are inspired by Lovejoy's grid-based belief MDP discretisation method [32]. They have also been used in [7] in the context of dynamic programming for POMDPs, and to over-approximate the value function in model checking of POMDPs [8]. In fact, this paper on determining lower bounds for

[1] A decision problem is ETR-complete if it can be reduced to a polynomial-length sentence in the Existential Theory of the Reals (for which the satisfiability problem is decidable) in polynomial time, and there is such a reduction in the reverse direction.

indefinite-horizon objectives can be seen as the dual counterpart of [8]. Our key challenge—compared to the approach of [8]—is that the value at a certain belief cannot easily be under-approximated with a convex combination of values of nearby beliefs. On the other hand, an under-approximation can benefit from a "good" guess of some initial POMDP policy. In the context of [8], such a guessed policy is of limited use for over-approximating values in the POMDP induced by an *optimal* policy. Although our approach is applicable to all thresholds, the focus of our work is on determining under-approximations for *quantitative* objectives. Dedicated verification techniques for the qualitative setting—almost-sure reachability—are presented in [17,16,27].

Experimental results. We have implemented our cut-off and belief clipping approaches on top of the probabilistic model checker Storm [23] and applied it to a range of various benchmarks. We provide a comparison with the model checking approach in [37], and determine the tightness of our under-approximations by comparing them to over-approximations obtained using the algorithm from [8]. Our main findings from the experimental validation are:

- Cut-offs often generate tight bounds while being computationally inexpensive.
- The clipping approach may further improve the accuracy of the approximation.
- Our implementation can deal with POMDPs with tens of thousands of states.
- Mostly, the obtained under-approximations are less than 10% off.

2 Preliminaries and Problem Statement

Let $Dist(A) := \{\mu : A \to [0,1] \mid \sum_{a \in A} \mu(a) = 1\}$ denote the set of probability distributions over a finite set A. The set $supp(\mu) := \{a \in A \mid \mu(a) > 0\}$ is the *support* of $\mu \in Dist(A)$. Let $\mathbb{R}^\infty := \mathbb{R} \cup \{\infty, -\infty\}$. We use Iverson bracket notation, where $[x] = 1$ if the Boolean expression x is *true* and $[x] = 0$ otherwise.

2.1 Partially Observable MDPs

Definition 1 (MDP). *A* Markov decision process (MDP) *is a tuple* $M = \langle S, Act, \mathbf{P}, s_{init} \rangle$ *with a (finite or infinite) set of states* S*, a finite set of actions* Act*, a transition function* $\mathbf{P} \colon S \times Act \times S \to [0,1]$ *with* $\sum_{s' \in S} \mathbf{P}(s, \alpha, s') \in \{0, 1\}$ *for all* $s \in S$ *and* $\alpha \in Act$*, and an initial state* s_{init}*.*

We fix an MDP $M := \langle S, Act, \mathbf{P}, s_{init} \rangle$. For $s \in S$ and $\alpha \in Act$, let $post^M(s, \alpha) := \{s' \in S \mid \mathbf{P}(s, \alpha, s') > 0\}$ denote the set of α-successors of s in M. The set of *enabled actions* in $s \in S$ is given by $Act(s) := \{\alpha \in Act \mid post^M(s, \alpha) \neq \emptyset\}$.

Definition 2 (POMDP). *A* partially observable MDP (POMDP) *is a tuple* $\mathcal{M} = \langle M, Z, O \rangle$*, where* M *is the underlying MDP with* $|S| \in \mathbb{N}$*, i.e.* S *is finite,* Z *is a finite set of observations, and* $O \colon S \to Z$ *is an observation function such that* $O(s) = O(s') \implies Act(s) = Act(s')$ *for all* $s, s' \in S$*.*

We fix a POMDP $\mathcal{M} := \langle M, Z, O \rangle$ with underlying MDP M. We lift the notion of enabled actions to observations $z \in Z$ by setting $Act(z) := Act(s)$ for some

$s \in S$ with $O(s) = z$ which is valid since states with the same observations are required to have the same enabled actions. The notions defined for MDPs below also straightforwardly apply to POMDPs.

Remark 1. More general observation functions of the form $O : S \times Act \to Dist(Z)$ can be encoded in this formalism by using a polynomially larger state space [16].

An *infinite path* through an MDP (and a POMDP) is a sequence $\tilde{\pi} = s_0\alpha_1 s_1\alpha_2 \ldots$ such that $\alpha_{i+1} \in Act(s_i)$ and $s_{i+1} \in post^M(s_i, \alpha_{i+1})$ for all $i \in \mathbb{N}$. A *finite path* is a finite prefix $\hat{\pi} = s_0\alpha_1 \ldots \alpha_n s_n$ of an infinite path $\tilde{\pi}$. For finite $\hat{\pi}$ let $last(\hat{\pi}) := s_n$ and $|\hat{\pi}| := n$. For infinite $\tilde{\pi}$ set $|\tilde{\pi}| := \infty$ and let $\tilde{\pi}[i]$ denote the finite prefix of length $i \in \mathbb{N}$. We denote the set of finite and infinite paths in M by $Paths^M_{\text{fin}}$ and $Paths^M_{\text{inf}}$, respectively. Let $Paths^M := Paths^M_{\text{fin}} \cup Paths^M_{\text{inf}}$. Paths are lifted to the observation level by *observation traces*. The observation trace of a (finite or infinite) path $\pi = s_0\alpha_1 s_1\alpha_2 \ldots \in Paths^M$ is $O(\pi) := O(s_0)\alpha_1 O(s_1)\alpha_2 \ldots$. Two paths $\pi, \pi' \in Paths^M$ are *observation-equivalent* if $O(\pi) = O(\pi')$.

Policies resolve the non-determinism present in MDPs (and POMDPs). Given a finite path $\hat{\pi}$, a policy determines the action to take at $last(\hat{\pi})$.

Definition 3 (Policy). *A* policy *for M is a function $\sigma : Paths^M_{\text{fin}} \to Dist(Act)$ such that for each path $\hat{\pi} \in Paths^M_{\text{fin}}$, $supp(\sigma(\hat{\pi})) \subseteq Act(last(\hat{\pi}))$.*

A policy σ is *deterministic* if $|supp(\sigma(\hat{\pi}))| = 1$ for all $\hat{\pi} \in Paths^M_{\text{fin}}$. Otherwise it is *randomised*. σ is *memoryless* if for all $\hat{\pi}, \hat{\pi}' \in Paths^M_{\text{fin}}$ we have $last(\hat{\pi}) = last(\hat{\pi}') \implies \sigma(\hat{\pi}) = \sigma(\hat{\pi}')$. σ is *observation-based* if for all $\hat{\pi}, \hat{\pi}' \in Paths^M_{\text{fin}}$ it holds that $O(\hat{\pi}) = O(\hat{\pi}') \implies \sigma(\hat{\pi}) = \sigma(\hat{\pi}')$. We denote the set of policies for M by Σ^M and the set of observation-based policies for $\mathcal{M}$ by $\Sigma^{\mathcal{M}}_{\text{obs}}$. A *finite-memory* policy (fm-policy) can be represented by a finite automaton where the current memory state and the state of the MDP determine the actions to take [4].

The *probability measure* $\mu^{\sigma,s}_M$ for paths in M under policy σ and initial state s is the probability measure of the Markov chain induced by M, σ, and s [4].

We use *reward structures* to model quantities like time, or energy consumption.

Definition 4 (Reward Structure). *A* reward structure *for M is a function $\mathbf{R}\colon S \times Act \times S \to \mathbb{R}$ such that either for all $s, s' \in S$, $\alpha \in Act$, $\mathbf{R}(s, \alpha, s') \geq 0$ or for all $s, s' \in S$, $\alpha \in Act$, $\mathbf{R}(s, \alpha, s') \leq 0$ holds. In the former case, we call $\mathbf{R}$* positive*, otherwise* negative*.*

We fix a reward structure $\mathbf{R}$ for M. The *total reward* along a path π is defined as $\mathsf{rew}_{M,\mathbf{R}}(\pi) := \sum_{i=1}^{|\pi|} \mathbf{R}(s_{i-1}, \alpha_i, s_i)$. The total reward is always well-defined—even if π is infinite—since all rewards are assumed to be either non-negative or non-positive. For an infinite path $\tilde{\pi}$ we define the *total reward* until reaching a set of goal states $G \subseteq S$ by

$$\mathsf{rew}_{M,\mathbf{R},G}(\tilde{\pi}) := \begin{cases} \mathsf{rew}_{M,\mathbf{R}}(\hat{\pi}) & \text{if } \exists i \in \mathbb{N} : \hat{\pi} = \tilde{\pi}[i] \ \wedge \ last(\hat{\pi}) \in G \ \wedge \\ & \quad \forall j < i : last(\tilde{\pi}[j]) \notin G, \\ \mathsf{rew}_{M,\mathbf{R}}(\tilde{\pi}) & \text{otherwise.} \end{cases}$$

Intuitively, $\mathsf{rew}_{M,\mathbf{R},G}(\tilde{\pi})$ accumulates reward along $\tilde{\pi}$ until the first visit of a goal state $s \in G$. If no goal state is reached, reward is accumulated along the infinite path. The *expected* total reward until reaching G for policy σ and state s is

$$\mathsf{ER}^{\sigma}_{M,\mathbf{R}}(s \models \Diamond G) := \int_{\tilde{\pi} \in Paths^{M}_{\text{inf}}} \mathsf{rew}_{M,\mathbf{R},G}(\tilde{\pi}) \cdot \mu^{\sigma,s}_{M}(d\tilde{\pi}).$$

Observation-based policies capture the notion that a decision procedure for a POMDP only accesses the observations and their history and not the entire state of the system. We are interested in reasoning about *minimal* and *maximal* values over *all* observation-based policies. For our explanations we focus on maximising (non-negative or non-positive) expected rewards. Minimisation can be achieved by negating all rewards.

Definition 5 (Maximal Expected Total Reward). *The* maximal *expected total reward until reaching G from s in POMDP $\mathcal{M}$ is*

$$\mathsf{ER}^{\max}_{\mathcal{M},\mathbf{R}}(s \models \Diamond G) := \sup_{\sigma \in \Sigma^{\mathcal{M}}_{\text{obs}}} \mathsf{ER}^{\sigma}_{\mathcal{M},\mathbf{R}}(s \models \Diamond G).$$

We define $\mathsf{ER}^{\max}_{\mathcal{M},\mathbf{R}}(\Diamond G) := \mathsf{ER}^{\max}_{\mathcal{M},\mathbf{R}}(s_{init} \models \Diamond G)$.

The central problem of our work, the *indefinite-horizon total reward problem*, asks the question whether the maximal expected total reward until reaching a goal exceeds a given threshold.

Problem 1. Given a POMDP $\mathcal{M}$, reward structure $\mathbf{R}$, set of goal states $G \subseteq S$, and threshold $\lambda \in \mathbb{R}$, decide whether $\mathsf{ER}^{\max}_{\mathcal{M},\mathbf{R}}(\Diamond G) \leq \lambda$.

Example 1. Fig. 1 shows a POMDP $\mathcal{M}$ with three states and two observations: $O(s_0) = O(s_1) =$ □ and $O(s_2) =$ ●. A reward of 1 is collected when transitioning from s_1 to s_2 via the β-action. All other rewards are zero.

The policy that always selects α at s_0 and β at s_1 maximizes the expected total reward to reach $G = \{s_2\}$ but is not observation-based. The observation-based policy that for the first $n \in \mathbb{N}$ transition steps selects α and then selects β afterwards yields an expected total reward of $1 - (1/2)^n$. With $n \to \infty$ we obtain $\mathsf{ER}^{\max}_{\mathcal{M},\mathbf{R}}(\Diamond\{s_2\}) = 1$.

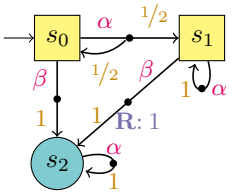

Figure 1. POMDP $\mathcal{M}$

As computing maximal expected rewards exactly in POMDPs is undecidable [34], we aim at under-approximating the actual value $\mathsf{ER}^{\max}_{\mathcal{M},\mathbf{R}}(\Diamond G)$. This allows us to answer our problem negatively if the computed lower bound exceeds λ.

Remark 2. Expected rewards can be used to describe *reachability probabilities* by assigning reward 1 to all transitions entering G and assigning reward 0 to all other transitions. Our approach can thus be used to obtain lower bounds on reachability probabilities in POMDPs. This also holds for almost-sure reachability (i.e. *"is the reachability probabilty one?"*), though dedicated methods like those presented in [17,16,27] are better suited for that setting.

2.2 Beliefs

The semantics of a POMDP $\mathcal{M}$ are captured by its (fully observable) *belief MDP*. The infinite state space of this MDP consists of *beliefs* [3,44]. A belief is a distribution over the states of the POMDP where each component describes the likelihood to be in a POMDP state given a history of observations. We denote the set of all beliefs for $\mathcal{M}$ by $\mathcal{B}_\mathcal{M} := \{b \in \mathit{Dist}(S) \mid \forall s, s' \in \mathit{supp}(b) : O(s) = O(s')\}$ and write $O(b) \in Z$ for the unique observation $O(s)$ of all $s \in \mathit{supp}(b)$.

The belief MDP of $\mathcal{M}$ is constructed by starting in the belief corresponding to the initial state and computing successor beliefs to unfold the MDP. Let $\mathbf{P}(s, \alpha, z) := \sum_{s' \in S} [O(s') = z] \cdot \mathbf{P}(s, \alpha, s')$ be the probability to observe $z \in Z$ after taking action α in POMDP state s. Then, the probability to observe z after taking action α in belief b is $\mathbf{P}(b, \alpha, z) := \sum_{s \in S} b(s) \cdot \mathbf{P}(s, \alpha, z)$. We refer to $[\![b|\alpha, z]\!] \in \mathcal{B}_\mathcal{M}$—the belief after taking α in b, conditioned on observing z—as the *α-z-successor of b*. If $\mathbf{P}(b, \alpha, z) > 0$, it is defined component-wise as

$$[\![b|\alpha, z]\!](s) := \frac{[O(s) = z] \cdot \sum_{s' \in S} b(s') \cdot \mathbf{P}(s', \alpha, s)}{\mathbf{P}(b, \alpha, z)}$$

for all $s \in S$. Otherwise $[\![b|\alpha, z]\!]$ is *undefined*.

Definition 6 (Belief MDP). *The* belief MDP *of $\mathcal{M}$ is the MDP $\mathit{bel}(\mathcal{M}) = \langle \mathcal{B}_\mathcal{M}, \mathit{Act}, \mathbf{P}^B, b_{\mathit{init}} \rangle$, where $\mathcal{B}_\mathcal{M}$ is the set of all beliefs in $\mathcal{M}$, Act is as for $\mathcal{M}$, $b_{\mathit{init}} := \{s_{\mathit{init}} \mapsto 1\}$ is the initial belief, and $\mathbf{P}^B \colon \mathcal{B}_\mathcal{M} \times \mathit{Act} \times \mathcal{B}_\mathcal{M} \to [0, 1]$ is the belief transition function with*

$$\mathbf{P}^B(b, \alpha, b') := \begin{cases} \mathbf{P}(b, \alpha, z) & \text{if } b' = [\![b|\alpha, z]\!], \\ 0 & \text{otherwise.} \end{cases}$$

We lift a POMDP reward structure $\mathbf{R}$ to the belief MDP [25].

Definition 7 (Belief Reward Structure). *For beliefs $b, b' \in \mathcal{B}_\mathcal{M}$ and action $\alpha \in \mathit{Act}$, the* belief reward structure $\mathbf{R}^B$ *based on $\mathbf{R}$ associated with $\mathit{bel}(\mathcal{M})$ is given by*

$$\mathbf{R}^B(b, \alpha, b') := \frac{\sum_{s \in S} b(s) \cdot \sum_{s' \in S} [O(s') = O(b')] \cdot \mathbf{R}(s, \alpha, s') \cdot \mathbf{P}(s, \alpha, s')}{\mathbf{P}(b, \alpha, O(b'))}.$$

Given a set of goal states $G \subseteq S$, we assume—for simplicity—that there is a set of observations $Z' \subseteq Z$ such that $s \in G$ iff $O(s) \in Z'$. This assumption can always be ensured by transforming the POMDP $\mathcal{M}$. See the full technical report [10] for details. The set of *goal beliefs* for G is given by $G_\mathcal{B} := \{b \in \mathcal{B}_\mathcal{M} \mid \mathit{supp}(b) \subseteq G\}$.

We now lift the computation of expected rewards to the belief level. Based on the well-known Bellman equations [5], the belief MDP induces a function that maps every belief to the expected total reward accumulated from that belief.

Definition 8 (POMDP Value Function). *For $b \in \mathcal{B}_\mathcal{M}$, the* n-step value function $V_n : \mathcal{B}_\mathcal{M} \to \mathbb{R}$ *of $\mathcal{M}$ is defined recursively as $V_0(b) := 0$ and*

$$V_n(b) := [b \notin G_\mathcal{B}] \cdot \max_{\alpha \in \mathit{Act}} \sum_{b' \in \mathit{post}^{\mathit{bel}(\mathcal{M})}(b, \alpha)} \mathbf{P}^B(b, \alpha, b') \cdot \left(\mathbf{R}^B(b, \alpha, b') + V_{n-1}(b')\right).$$

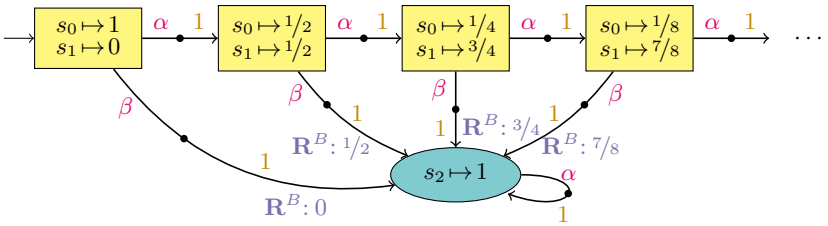

Figure 2. Belief MDP $bel(\mathcal{M})$ of POMDP $\mathcal{M}$ from Fig. 1

The (optimal) value function $V^* : \mathcal{B}_\mathcal{M} \to \mathbb{R}^\infty$ *is given by* $V^*(b) := \lim_{n\to\infty} V_n(b)$.

The n-step value function is piecewise linear and convex [44]. Thus, the optimal value function can be approximated arbitrarily close by a piecewise linear convex function [47]. The value function yields expected total rewards in $\mathcal{M}$ and $bel(\mathcal{M})$:

$$\mathsf{ER}^{\max}_{\mathcal{M},\mathbf{R}}(s \models \Diamond G) \;=\; \mathsf{ER}^{\max}_{bel(\mathcal{M}),\mathbf{R}^B}(\{s \mapsto 1\} \models \Diamond G_\mathcal{B}) \;=\; V^*(\{s \mapsto 1\}).$$

Example 2. Fig. 2 shows a fragment of the belief MDP of the POMDP from Fig. 1. Observe $\mathsf{ER}^{\max}_{bel(\mathcal{M}),\mathbf{R}^B}(\Diamond\, \{s_2 \mapsto 1\}) = 1$.

We reformulate our problem statement to focus on the belief MDP.

> *Problem 2 (equivalent to Problem 1).* For a POMDP $\mathcal{M}$, reward structure $\mathbf{R}$, goal states $G \subseteq S$, and threshold $\lambda \in \mathbb{R}$, decide whether $V^*(\{s_{init} \mapsto 1\}) \leq \lambda$.

As the belief MDP is fully observable, standard results for MDPs apply. However, an exhaustive analysis of $bel(\mathcal{M})$ is intractable since the belief MDP is—in general—infinitely large[2].

3 Finite Exploration Under-Approximation

Instead of approximating values directly on the POMDP, we consider approximations of the corresponding belief MDP. The basic idea is to construct a finite abstraction of the belief MDP by unfolding parts of it and approximate values at beliefs where we decide not to explore. In the resulting finite MDP, under-approximative expected reward values can be computed by standard model checking techniques. We present two approaches for abstraction: *belief cut-offs* and *belief clipping*. We incorporate those techniques into an algorithmic framework that yields arbitrarily tight under-approximations.

The technical report [10] contains formal proofs of our claims.

[2] The set of all beliefs—i.e. the state space of $bel(\mathcal{M})$—is uncountable. The reachable fragment is countable, though, since each belief has at most $|Z|$ many successors.

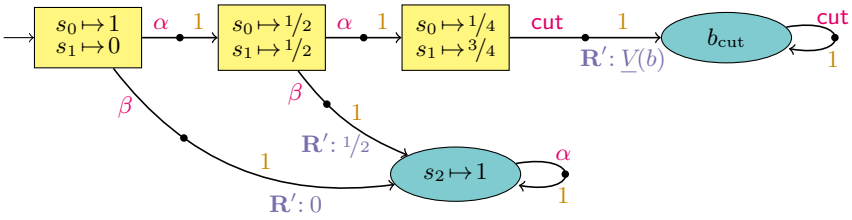

Figure 3. Applying belief cut-offs to the belief MDP from Fig. 2

3.1 Belief Cut-Offs

The general idea of *belief cut-offs* is to stop exploring the belief MDP at certain beliefs—the *cut-off beliefs*—and assume that a goal state is immediately reached while sub-optimal reward is collected. Similar techniques have been discussed in the context of fully observable MDPs and other model types [11,26,49,2]. Our work adapts the idea of cut-offs for POMDP *over-approximations* described in [8] to under-approximations. The main idea of belief cut-offs shares similarities with the *SARSOP* [30] and *Goal-HSVI* [24] approaches. While they apply cut-offs on the level of the computation tree, our approach directly manipulates the belief MDP to yield a finite model.

Let $\underline{V} : \mathcal{B}_{\mathcal{M}} \to \mathbb{R}^{\infty}$ with $\underline{V}(b) \leq V^*(b)$ for all $b \in \mathcal{B}_{\mathcal{M}}$. We call $\underline{V}$ an *under-approximative value function* and $\underline{V}(b)$ the *cut-off value* of b. In each of the cut-off beliefs b, instead of adding the regular transitions to its successors, we add a transition with probability 1 to a dedicated goal state b_{cut}. In the modified reward structure $\mathbf{R}'$, this *cut-off transition* is assigned a reward[3] of $\underline{V}(b)$, causing the value for a cut-off belief b in the modified MDP to coincide with $\underline{V}(b)$. Hence, the exact value of the cut-off belief—and thus the value of all other explored beliefs—is under-approximated.

Example 3. Fig. 3 shows the resulting *finite* MDP obtained when considering the belief MDP from Fig. 2 with single cut-off belief $b = \{s_0 \mapsto 1/4,\ s_1 \mapsto 3/4\}$.

Computing cut-off values. The question of finding a suitable under-approximative value function $\underline{V}$ is central to the cut-off approach. For an effective approximation, such a function should be easy to compute while still providing values close to the optimum. If we assume a positive reward structure, the constant value 0 is always a valid under-approximation. A more sophisticated approach is to compute suboptimal expected reward values for the states of the POMDP using *some* arbitrary, fixed observation-based policy $\sigma \in \Sigma^{\mathcal{M}}_{\text{obs}}$. Let $U^{\sigma} : S \to \mathbb{R}^{\infty}$ such that for all $s \in S$, $U^{\sigma}(s) = \mathsf{ER}^{\sigma}_{\mathcal{M},\mathbf{R}}(s \models \Diamond G)$. Then, we define the function $\mathfrak{U}^{\sigma} : \mathcal{B}_{\mathcal{M}} \to \mathbb{R}^{\infty}$ as $\mathfrak{U}^{\sigma}(b) := \sum_{s \in supp(b)} b(s) \cdot U^{\sigma}(s)$.

[3] We slightly deviate from Def. 4 by allowing transition rewards to be $-\infty$ or $+\infty$. Alternatively, we could introduce new sink states with a non-zero self-loop reward.

Lemma 1. *$\mathfrak{U}^\sigma$ is an under-approximative value function, i.e. for all $b \in \mathcal{B}_\mathcal{M}$:*

$$\mathfrak{U}^\sigma(b) := \sum_{s \in supp(b)} b(s) \cdot U^\sigma(s) \leq V^*(b).$$

Thus, finding a suitable under-approximative value function reduces to finding "good" policies for $\mathcal{M}$, e.g. by using randomly guessed fm-policies, machine learning methods [13], or a transformation to a parametric model [28].

3.2 Belief Clipping

The cut-off approach provides a universal way to construct an MDP which under-approximates the expected total reward value for a given POMDP. The quality of the approximation, however, is highly dependent on the under-approximative value function used. Furthermore, regions where the belief MDP slowly converges towards a belief may pose problems in practice.

As a potential remedy for these problems, we propose a different concept called *belief clipping*. Intuitively, the procedure shifts some of the probability mass of a belief b in order to transform b to another belief $\tilde{b}$. We then connect b to $\tilde{b}$ in a way that the accuracy of our approximation of the value $V^*(b)$ depends only on the approximation of $V^*(\tilde{b})$ and the so-called *clipping value*—some notion of distance between b and $\tilde{b}$ that we discuss below. We can thus focus on exploring the successors of $\tilde{b}$ to obtain good approximations for both beliefs b and $\tilde{b}$.

Definition 9 (Belief Clip). *For $b \in \mathcal{B}_\mathcal{M}$, we call $\mu\colon supp(b) \to [0,1]$ a* belief clip *if $\forall s \in supp(b)\colon \mu(s) \leq b(s)$ and $\sum(\mu) := \sum_{s \in supp(b)} \mu(s) < 1$. The belief $(b \ominus \mu) \in \mathcal{B}_M$ induced by μ is defined by*

$$\forall s \in supp(b)\colon \ (b \ominus \mu)(s) \ := \ \frac{b(s) - \mu(s)}{1 - \sum(\mu)}.$$

Intuitively, a belief clip μ for b describes for each $s \in supp(b)$ the probability mass that is removed ("clipped away") from $b(s)$. The induced belief is obtained when normalising the resulting values so that they sum up to one.

Example 4. For belief $b = \{s_0 \mapsto 1/4, s_1 \mapsto 3/4\}$, consider the two belief clips $\mu_1 = \{s_0 \mapsto 1/4, s_1 \mapsto 1/4\}$ and $\mu_2 = \{s_0 \mapsto 1/4, s_1 \mapsto 0\}$. Both induce the same belief: $(b \ominus \mu_1) = (b \ominus \mu_2) = \{s_0 \mapsto 0, s_1 \mapsto 1\}$.

We have $supp((b \ominus \mu)) \subseteq supp(b)$, which also implies $O((b \ominus \mu)) = O(b)$. Given some candidate belief $\tilde{b}$, consider the set of inducing belief clips:

$$\mathcal{C}(b, \tilde{b}) := \left\{ \mu\colon supp(b) \to [0,1] \mid \mu \text{ is a belief clip for } b \text{ with } \tilde{b} = (b \ominus \mu) \right\}.$$

Belief $\tilde{b}$ is called an adequate clipping candidate for b iff $\mathcal{C}(b, \tilde{b}) \neq \emptyset$.

Definition 10 (Clipping Value). *For $b \in \mathcal{B}_\mathcal{M}$ and adequate clipping candidate $\tilde{b}$, the* clipping value *is $\Delta_{b \to \tilde{b}} := \sum(\delta_{b \to \tilde{b}})$, where $\delta_{b \to \tilde{b}} := \arg\min_{\mu \in \mathcal{C}(b,\tilde{b})} \sum(\mu)$. The values $\delta_{b \to \tilde{b}}(s)$ for $s \in supp(b)$ are the* state clipping values.

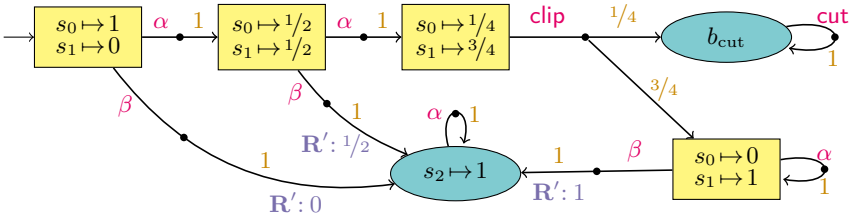

Figure 4. Applying belief clipping to the belief MDP from Fig. 2

Given a belief b and an adequate clipping candidate $\tilde{b}$, we outline how the notion of belief clipping is used to obtain valid under-approximations. We assume $b \neq \tilde{b}$, implying $0 < \Delta_{b \to \tilde{b}} < 1$. Instead of exploring all successors of b in $bel(\mathcal{M})$, the approach is to add a transition from b to $\tilde{b}$. The newly added transition has probability $1 - \Delta_{b \to \tilde{b}}$ and gets assigned a reward of 0. The remaining probability mass (i.e. $\Delta_{b \to \tilde{b}}$) leads to a designated goal state b_{cut}. To guarantee that—in general—the clipping procedure yields a valid under-approximation, we need to add a corrective reward value to the transition from b to b_{cut}. Let $\mathfrak{L} : S \to \mathbb{R}^{\infty}$ which maps each POMDP state to its *minimum* expected reward in the underlying, fully observable MDP M of $\mathcal{M}$[4], i.e. $\mathfrak{L}(s) = \mathsf{ER}^{\min}_{M,\mathbf{R}}(s \models \Diamond G)$. This function soundly under-approximates the state values which can be achieved by *any* observation-based policy. It can be generated using standard MDP analysis. Given state clipping values $\delta_{b \to \tilde{b}}(s)$ for $s \in supp(b)$, the reward for the transition from b to b_{cut} is $\sum_{s \in supp(b)} (\delta_{b \to \tilde{b}}(s) / \Delta_{b \to \tilde{b}}) \cdot \mathfrak{L}(s)$.

Example 5. For the belief MDP from Fig. 2, belief $b = \{s_0 \mapsto 1/4,\ s_1 \mapsto 3/4\}$, and clipping candidate $\tilde{b} = \{s_0 \mapsto 0,\ s_1 \mapsto 1\}$ we get $\Delta_{b \to \tilde{b}} = 1/4$, as $\delta_{b \to \tilde{b}} = \mu_2 = \{s_0 \mapsto 1/4,\ s_1 \mapsto 0\}$ with the belief clip μ_2 as in Example 4. Furthermore, $\mathfrak{L}(s_0) = 0$. The resulting MDP following our construction above is given in Fig. 4.

The following lemma shows that the construction yields an under-approximation.

Lemma 2. $(1 - \Delta_{b \to \tilde{b}}) \cdot V^*(\tilde{b}) \ + \ \Delta_{b \to \tilde{b}} \cdot \sum_{s \in supp(b)} \frac{\delta_{b \to \tilde{b}}(s)}{\Delta_{b \to \tilde{b}}} \cdot \mathfrak{L}(s) \ \leq \ V^*(b).$

Proof (sketch). To gain some intuition, consider the special case, where $\Delta_{b \to \tilde{b}} = \delta_{b \to \tilde{b}}(s) = b(s)$ for some $s \in supp(b)$. The clipping candidate $\tilde{b}$ can be interpreted as the conditional probability distribution arising from distribution b given that s is *not* the current state. The value $V^*(b)$ can be split into the sum of (i) the probability that s is *not* the current state times the reward accumulated from belief $\tilde{b}$ and (ii) the probability that s *is* the current state times the reward accumulated from s, i.e. from the belief $\{s \mapsto 1\}$. However, for the two summands

[4] When rewards are negative, we might have $\mathfrak{L}(s) = -\infty$ for many $s \in S \setminus G$ in which case the applicability of the clipping approach is very limited.

we must consider a policy that does not distinguish between the beliefs b, $\tilde{b}$, and $\{s \mapsto 1\}$ as well as their observation-equivalent successors. In other words, the same sequence of actions must be executed when the same observations are made.

We consider such a policy that in addition is optimal at $\tilde{b}$, i.e. the reward accumulated from $\tilde{b}$ is equal to $V^*(\tilde{b})$. For the reward accumulated from $\{s \mapsto 1\}$, $\mathfrak{L}(s)$ provides a lower bound. Hence, $(1 - b(s)) \cdot V^*(b) + b(s) \cdot \mathfrak{L}(s)$ is a lower bound for the reward accumulated from b. A formal proof is given in [10]. □

To find a suitable clipping candidate for a given belief b, we consider a finite *candidate set* $\mathfrak{B} \subseteq \mathcal{B}_{\mathcal{M}}$ consisting of beliefs with observation $O(b)$. These beliefs do not need to be reachable in the belief MDP. The set can be constructed, e.g. by taking already explored beliefs or by using a fixed, discretised set of beliefs.

We are interested in minimising the clipping value $\Delta_{b \to b'}$ over all candidate beliefs $b' \in \mathfrak{B}$. A naive approach is to explicitly compute all clipping values for all candidates. We are using *mixed-integer linear programming (MILP)* [41] instead. An MILP is a system of linear inequalities (*constraints*) and a linear *objective function* considering real-valued and integer variables. A *feasible solution* of the MILP is a variable assignment that satisfies all constraints. An *optimal solution* is a feasible solution that minimises the objective function.

Definition 11 (Belief Clipping MILP). *The belief clipping MILP for belief $b \in \mathcal{B}_{\mathcal{M}}$ and finite set of candidates $\mathfrak{B} \subseteq \{b' \in \mathcal{B}_{\mathcal{M}} \mid O(b') = O(b)\}$ is given by:*

minimise Δ such that:

$$\sum_{b' \in \mathfrak{B}} a_{b'} = 1 \qquad \triangleright \textit{Select exactly one candidate } b' \tag{1}$$

$$\forall b' \in \mathfrak{B}\colon \quad a_{b'} \in \{0, 1\} \tag{2}$$

$$\sum_{s \in \mathit{supp}(b)} \delta_s = \Delta \qquad \triangleright \textit{Compute clipping value for selected } b' \tag{3}$$

$$\forall s \in \mathit{supp}(b)\colon \quad \delta_s \in [0,\, b(s)] \tag{4}$$

$$\forall b' \in \mathfrak{B}\colon \quad \delta_s \geq b(s) - (1 - \Delta) \cdot b'(s) - (1 - a_{b'}) \tag{5}$$

The MILP consists of $\mathcal{O}(|\mathit{supp}(b)| + |\mathfrak{B}|)$ variables and $\mathcal{O}(|\mathit{supp}(b)| \cdot |\mathfrak{B}|)$ constraints. For $b' \in \mathfrak{B}$, the binary variable $a_{b'}$ indicates whether b' has been chosen as the clipping candidate. Moreover, we have variables δ_s for $s \in \mathit{supp}(b)$ and a variable Δ to represent the (state) clipping values for b and the chosen candidate b'. Constraints 1 and 2 enforce that exactly one of the $a_{b'}$ variables is one, i.e. exactly one belief is chosen. Constraint 3 forces Δ to be the sum of all state clipping values. δ_s variables get a value between zero and $b(s)$ (Constraint 4). Constraint 5 only affects δ_s if the corresponding belief is chosen. Otherwise, $a_{b'}$ is set to 0 and the value on the right-hand side becomes negative. If a belief b' is chosen, the minimisation forces Constraint 5 to hold with equality as the right-hand side is greater or equal to 0. Assuming Δ is set to a value below 1, we obtain a valid clipping values as

$$\forall s \in \mathit{supp}(b)\colon \quad \delta_s = b(s) - (1 - \Delta) \cdot b'(s) \quad \iff \quad b'(s) = \frac{b(s) - \delta_s}{1 - \Delta}.$$

Input : POMDP $\mathcal{M} = \langle M, Z, O\rangle$ with $M = \langle S, Act, \mathbf{P}, s_{init}\rangle$, reward structure $\mathbf{R}$, goal states $G \subseteq S$, under-approx. value function $\underline{V}$, function $\mathfrak{L} : S \to \mathbb{R}^{\infty}$ with $\mathfrak{L}(s) = \mathsf{ER}^{\min}_{M,\mathbf{R}}(s \models \Diamond G)$

Output : Clipping belief MDP $\mathcal{K}_{\mathcal{M}}$ and reward structure $\mathbf{R}^{\mathcal{K}}$

1 $S^{\mathcal{K}} \leftarrow \{b_{init}, b_{\text{cut}}\}$ with $b_{init} = \{s_{init} \mapsto 1\}$ and a new belief state b_{cut}
2 $\mathbf{P}^{\mathcal{K}}(b_{\text{cut}}, \mathsf{cut}, b_{\text{cut}}) \leftarrow 1$, $\mathbf{R}^{\mathcal{K}}(b_{\text{cut}}, \mathsf{cut}, b_{\text{cut}}) \leftarrow 0$ *// add self-loop*
3 $Q \leftarrow \{b_{init}\}$ *// initialize exploration set*
4 **while** $Q \neq \emptyset$ **do**
5 $\quad b \leftarrow \texttt{chooseBelief}(Q)$, $Q \leftarrow Q \setminus \{b\}$ *// pop next belief to explore from Q*
6 $\quad$ **if** $supp(b) \subseteq G$ **then** $\mathbf{P}^{\mathcal{K}}(b, \mathsf{goal}, b) \leftarrow 1$, $\mathbf{R}^{\mathcal{K}}(b, \mathsf{goal}, b) \leftarrow 0$ *// add self-loop*
7 $\quad$ **else if** $\texttt{exploreBelief}(b)$ **then** *// expand b*
8 $\quad\quad$ **foreach** $\alpha \in Act(b)$ **do** *// Using bel(M) and $\mathbf{R}^B$ as in Defs. 6 and 7*
9 $\quad\quad\quad$ **foreach** $b' \in post^{bel(\mathcal{M})}(b, \alpha)$ **do**
10 $\quad\quad\quad\quad \mathbf{P}^{\mathcal{K}}(b, \alpha, b') \leftarrow \mathbf{P}^{B}(b, \alpha, b')$, $\mathbf{R}^{\mathcal{K}}(b, \alpha, b') \leftarrow \mathbf{R}^{B}(b, \alpha, b')$
11 $\quad\quad\quad\quad$ **if** $b' \notin S^{\mathcal{K}}$ **then** $S^{\mathcal{K}} \leftarrow S^{\mathcal{K}} \cup \{b'\}$, $Q \leftarrow Q \cup \{b'\}$
12 $\quad$ **else** *// apply cut-off and clipping to b*
13 $\quad\quad \mathbf{P}^{\mathcal{K}}(b, \mathsf{cut}, b_{\text{cut}}) \leftarrow 1$, $\mathbf{R}^{\mathcal{K}}(b, \mathsf{cut}, b_{\text{cut}}) \leftarrow \underline{V}(b)$ *// add cut-off transition*
14 $\quad\quad$ choose a finite set $\mathfrak{B} \subseteq \mathcal{B}_{\mathcal{M}}$ of clipping candidates for b
15 $\quad\quad \tilde{b}, \Delta_{b\to\tilde{b}}, \delta_{b\to\tilde{b}} \leftarrow \texttt{solveClippingMILP}(b, \mathfrak{B})$
16 $\quad\quad$ **if** $\tilde{b} \neq b$ and $\tilde{b}$ is adequate **then** *// Clip b using $\tilde{b}$*
17 $\quad\quad\quad \mathbf{P}^{\mathcal{K}}(b, \mathsf{clip}, \tilde{b}) \leftarrow (1 - \Delta_{b\to\tilde{b}})$, $\mathbf{P}^{\mathcal{K}}(b, \mathsf{clip}, b_{\text{cut}}) \leftarrow \Delta_{b\to\tilde{b}}$
18 $\quad\quad\quad \mathbf{R}^{\mathcal{K}}(b, \mathsf{clip}, \tilde{b}) \leftarrow 0$, $\mathbf{R}^{\mathcal{K}}(b, \mathsf{clip}, b_{\text{cut}}) \leftarrow \sum_{s \in supp(b)} \frac{\delta_{b\to\tilde{b}}(s)}{\Delta_{b\to\tilde{b}}} \cdot \mathfrak{L}(s)$
19 $\quad\quad\quad$ **if** $\tilde{b} \notin S^{\mathcal{K}}$ **then** $S^{\mathcal{K}} \leftarrow S^{\mathcal{K}} \cup \{\tilde{b}\}$, $Q \leftarrow Q \cup \{\tilde{b}\}$
20 **return** $\mathcal{K}_{\mathcal{M}} = \left\langle S^{\mathcal{K}}, Act \uplus \{\mathsf{goal}, \mathsf{cut}, \mathsf{clip}\}, \mathbf{P}^{\mathcal{K}}, b_{init}\right\rangle$ and $\mathbf{R}^{\mathcal{K}}$

Algorithm 1: Belief exploration algorithm with cut-offs and clipping

A trivial solution of the MILP is always obtained by setting $a_{b'}$ and Δ to 1 and δ_s to $b(s)$ for all s and an arbitrary $b' \in \mathfrak{B}$. This corresponds to an invalid belief clip. However, as we minimise the value for Δ, we can conclude that *no* belief in the candidate set is adequate for clipping if Δ is 1 in an optimal solution.

Theorem 1. *An optimal solution to the belief clipping MILP for belief b and candidate set $\mathfrak{B}$ sets $a_{\tilde{b}}$ to 1 and Δ to a value below 1 iff $\tilde{b} \in \mathfrak{B}$ is an adequate clipping candidate for b with minimal clipping value.*

3.3 Algorithm

We incorporate belief cut-offs and belief clipping into an algorithmic framework outlined in Algorithm 1. As input, the algorithm takes an instance of Problems 1 and 2, i.e. a POMDP $\mathcal{M}$ with reward structure $\mathbf{R}$ and goal states G. In addition, the algorithm considers an under-approximative value function $\underline{V}$ (Sect. 3.1) and a function $\mathfrak{L}$ for the computation of corrective reward values (Sect. 3.2).

Lines 1 and 2 initialise the state set $S^{\mathcal{K}}$ of the under-approximative MDP $\mathcal{K}_{\mathcal{M}}$ with the initial belief b_{init} and the designated goal state b_{cut} which has only one

transition to itself with reward 0. Furthermore, we initialise the *exploration set* Q by adding b_{init} (Line 3). During the computation, Q is used to keep track of all beliefs we still need to process. We then execute the exploration loop (Lines 4 to 19) until Q becomes empty. In each exploration step, a belief b is selected[5] and removed from Q. There are three cases for the currently processed belief b.

If $supp(b) \subseteq G$, i.e. b is a goal belief, we add a self-loop with reward 0 to b and continue with the next belief (Line 6). b is not expanded as successors of goal beliefs will not influence the result of the computation.

If b is not a goal belief, we use a heuristic function[6] `exploreBelief` to decide if b is expanded in Line 7. Lines 8 to 11 outline the expansion step. The transitions from b to its successor beliefs and the corresponding rewards as in the original belief MDP (see Sect. 2.2) are added. Furthermore, the successor beliefs that have not been encountered before are added to the set of states $S^{\mathcal{K}}$ and the exploration set Q.

If b is *not* expanded, we apply the cut-off approach *and* the clipping approach to b in Lines 12 to 19. In Line 13 we add a cut-off transition from b to b_{cut} with a new action cut. We use the given under-approximative value function $\underline{V}$ to compute the cut-off reward. Towards the clipping approach, a set of candidate beliefs is chosen and the belief clipping MILP for b and the candidate set is constructed as described in Def. 11 (Lines 14 and 15). If an adequate candidate $\tilde{b}$ with clipping values $\Delta_{b\to\tilde{b}}$ and $\delta_{b\to\tilde{b}}(s)$ for $s \in supp(b)$ has been found, we add the transitions from b to b_{cut} and to $\tilde{b}$ using a new action clip and probabilities $\Delta_{b\to\tilde{b}}$ and $1 - \Delta_{b\to\tilde{b}}$, respectively. Furthermore, we equip the transitions with reward values as described in Sect. 3.2 using the given function $\mathfrak{L}$ (Lines 16 to 18). If the clipping candidate $\tilde{b}$ has not been encountered before, we add it to the state space of the MDP and to the exploration set in Line 19.

The result of the algorithm is an MDP $\mathcal{K}_{\mathcal{M}}$ with reward structure $\mathbf{R}^{\mathcal{K}}$. The set of states $S^{\mathcal{K}}$ of $\mathcal{K}_{\mathcal{M}}$ contains all encountered beliefs. To guarantee termination of the algorithm, the decision heuristic `exploreBelief` has to stop exploring further beliefs at some point. Moreover, the handling of clipping candidates in Line 19 should not add new beliefs to Q infinitely often. We therefore fix a finite set of candidate beliefs $\mathcal{B}^{\#} \subseteq \mathcal{B}_{\mathcal{M}}$ and make sure that the candidate sets $\mathfrak{B}$ in Line 14 satisfy $(\mathfrak{B} \setminus S^{\mathcal{K}}) \subseteq \mathcal{B}^{\#}$. To ensure a certain progress in the exploration "clip-cycles"—i.e. paths of the form $b_1\, \mathsf{clip} \ldots \mathsf{clip}\, b_n\, \mathsf{clip}\, b_1$—are avoided in $\mathcal{K}_{\mathcal{M}}$. This can be done, e.g. by always expanding the candidate beliefs $b \in \mathcal{B}^{\#}$.

Expected total rewards until reaching the extended set of goal beliefs $G_{\text{cut}} := G_{\mathcal{B}} \cup \{b_{\text{cut}}\}$ in $\mathcal{K}_{\mathcal{M}}$ under-approximate the values in the belief MDP:

Theorem 2. *For all beliefs $b \in S^{\mathcal{K}} \setminus \{b_{cut}\}$ it holds that*

$$\mathsf{ER}^{\max}_{\mathcal{K}_{\mathcal{M}},\mathbf{R}^{\mathcal{K}}}(b \models \Diamond G_{cut}) \leq V^*(b) = \mathsf{ER}^{\max}_{bel(\mathcal{M}),\mathbf{R}^{B}}(b \models \Diamond G_{\mathcal{B}}).$$

Corollary 1. $\mathsf{ER}^{\max}_{\mathcal{K}_{\mathcal{M}},\mathbf{R}^{\mathcal{K}}}(\Diamond G_{cut}) \leq \mathsf{ER}^{\max}_{\mathcal{M},\mathbf{R}}(\Diamond G)$.

[5] For example, Q can be implemented as a FIFO queue.

[6] The decision can be made for example by considering the size of the already explored state space such that the expansion is stopped if a size threshold has been reached. More involved decision heuristics are subject to further research.

Table 1. Results for benchmark POMDPs with maximisation objective

Benchmark Model	ϕ	Data $S/Act/Z$	Prism	Storm					
				Cut-Off Only	Cut-Off + Clipping η=2	η=3	η=4	η=6	Over-Approx.
Drone 4-1	$P_{\max}$	1226 2954 384	TO / MO	≥ 0.79 < 1s $3{\cdot}10^4$	≥ 0.79 1360s $3{\cdot}10^4$	TO	TO	TO	≤ 0.94
Drone 4-2	$P_{\max}$	1226 2954 761	TO / MO	≥ 0.86 < 1s $2{\cdot}10^4$	≥ 0.91 249s $2{\cdot}10^4$	≥ 0.92 1902s $2{\cdot}10^4$	TO	TO	≤ 0.97
Grid-av 4-0	$P_{\max}$	17 59 4	[**0.21**, 1.0] 5.14s η =6	≥ 0.86 < 1s 238	≥ 0.93 < 1s 312	≥ 0.93 1.77s 472	≥ 0.93 3.63s 663	≥ 0.93 13.9s 1300	≤ 0.98
Grid-av 4-0.1	$P_{\max}$	17 59 4	[**0.21**, 1.0] 1.47s η =3	≥ 0.82 < 1s 238	≥ 0.85 26.1s 317	≥ 0.82 198s 461	≥ 0.85 1913s 759	TO	≤ 0.99
Netw-p 2-8-20	$R_{\max}$	$2{\cdot}10^4$ $3{\cdot}10^4$ 4909	[**557**, 557] 2355s η =10	≥ 537 2.3s $8{\cdot}10^4$	≥ 537 98.5s $1{\cdot}10^5$	≥ 537 320s $1{\cdot}10^5$	≥ 537 651s $1{\cdot}10^5$	≥ 537 2368s $1{\cdot}10^5$	≤ 558
Netw-p 3-8-20	$R_{\max}$	$2{\cdot}10^5$ $3{\cdot}10^5$ $2{\cdot}10^4$	TO / MO	≥ 769 290s $1{\cdot}10^6$	≥ 769 6640s $1{\cdot}10^6$	TO	TO	TO	≤ 819
Refuel 06	$P_{\max}$	208 565 50	[**0.67**, 0.72] 4625s η =3	≥ 0.67 < 1s 4576	≥ 0.67 5.89s 4834	≥ 0.67 24.3s 5204	≥ 0.67 92s 5603	≥ 0.67 2076s 6135	≤ 0.69
Refuel 08	$P_{\max}$	470 1431 66	TO / MO	≥ 0.45 < 1s $2{\cdot}10^4$	≥ 0.45 839s $2{\cdot}10^4$	TO	TO	TO	≤ 0.51

4 Experimental Evaluation

Implementation details. We integrated Algorithm 1 in the probabilistic model checker Storm [23] as an extension of the POMDP verification framework described in [8]. Inputs are a POMDP—encoded either explicitly or using an extension of the Prism language [37]—and a property specification. Internally, POMDPs and MDPs are represented using sparse matrices. The implementation supports minimisation[7] and maximisation of reachability probabilities, reach-avoid probabilities (i.e. the probability to avoid a set of bad state until a set of goal states is reached), and expected total rewards. In a preprocessing step, functions $\underline{V}$ and $\mathfrak{L}$ as considered in Algorithm 1 are generated. For $\underline{V}$, we consider the function $\mathfrak{U}^\sigma$ as in Lemma 1, where σ is a memoryless observation-based policy given by a heuristic[8]. For the function $\mathfrak{L}$, we apply standard MDP analysis on the underlying MDP. When exploring the abstraction MDP $\mathcal{K}_{\mathcal{M}}$, our heuristic expands a belief iff $|S^{\mathcal{K}}| \leq |S| \cdot \max_{z \in z} |O^{-1}(z)|$, where $|S^{\mathcal{K}}|$ is the number of already explored beliefs and $|O^{-1}(z)|$ is the number of POMDP states with observation z. Belief clipping can either be disabled entirely, or we consider candidate sets $\mathfrak{B} \subseteq \mathcal{B}^{\#}_{\eta}$, where $\mathcal{B}^{\#}_{\eta} := \{b \in \mathcal{B} \mid \forall s \in S : b(s) \in \{i/\eta \mid i \in \mathbb{N}, 0 \leq i \leq \eta\}\}$ forms a finite, regular *grid* of beliefs with resolution $\eta \in \mathbb{N} \setminus \{0\}$. Grid beliefs $b \in \mathcal{B}^{\#}_{\eta}$ are always expanded.

[7] For minimisation, the under-approximation yields *upper bounds*.

[8] The heuristic uses optimal values obtained on the fully observable underlying MDP.

Table 2. Results for benchmark POMDPs with minimisation objective

Benchmark Model	ϕ	Data $S/Act/Z$	Prism	Storm Cut-Off Only	Storm Cut-Off + Clipping η=2	η=3	η=4	η=6	Storm Over-Approx.
Grid 4-0.1	$R_{\min}$	17	$[4.52, \mathbf{4.7}]$	≤ 4.78	≤ 4.78	≤ 4.78	≤ 4.78		≥ 4.52
		62	649s	< 1s	15.6s	148s	1940s	TO	
		3	$\eta = 10$	258	255	255	255		
Grid 4-0.3	$R_{\min}$	17	$[6.12, \mathbf{6.31}]$	≤ 6.56	≤ 6.56	≤ 6.56	≤ 6.56		≥ 6.08
		62	1077s	< 1s	15.8s	148s	1983s	TO	
		3	$\eta = 10$	255	256	256	256		
Maze2 0.1	$R_{\min}$	15	$[6.32, \mathbf{6.32}]$	≤ 6.34	≤ 6.34	≤ 6.34	≤ 6.34	≤ 6.34	≥ 6.32
		54	1.79s	< 1s	< 1s	< 1s	< 1s	2.02s	
		8	$\eta = 10$	91	90	90	90	90	
Netw 2-8-20	$R_{\min}$	4589	$[3.17, \mathbf{3.2}]$	≤ 6.56	≤ 6.56	≤ 6.56	≤ 6.56	≤ 6.56	≥ 3.14
		6973	211s	< 1s	5.31s	17.2s	42.3s	167s	
		1173	$\eta = 10$	$2{\cdot}10^4$	$2{\cdot}10^4$	$2{\cdot}10^4$	$3{\cdot}10^4$	$3{\cdot}10^4$	
Netw 3-8-20	$R_{\min}$	$2{\cdot}10^4$	$[5.61, \mathbf{6.79}]$	≤ 11.9	≤ 11.9	≤ 11.9	≤ 11.9		≥ 6.13
		$3{\cdot}10^4$	7133s	3.51s	214s	1372s	4910s	TO	
		2205	$\eta = 6$	$1{\cdot}10^5$	$2{\cdot}10^5$	$2{\cdot}10^5$	$2{\cdot}10^5$		
Rocks 12	$R_{\min}$	6553		≤ 38	≤ 38	≤ 38	≤ 20	≤ 21	≥ 20
		$3{\cdot}10^4$	TO / MO	1.39s	61.1s	138s	230s	532s	
		1645		$3{\cdot}10^4$	$3{\cdot}10^4$	$3{\cdot}10^4$	$5{\cdot}10^4$	$6{\cdot}10^4$	
Rocks 16	$R_{\min}$	$1{\cdot}10^4$		≤ 44	≤ 44	≤ 44	≤ 26	≤ 27	≥ 26
		$5{\cdot}10^4$	TO / MO	3.85s	114s	230s	399s	1062s	
		2761		$4{\cdot}10^4$	$4{\cdot}10^4$	$4{\cdot}10^4$	$6{\cdot}10^4$	$1{\cdot}10^5$	

Furthermore, we exclude clipping candidates $\tilde{b}$ with $\delta_{b \to \tilde{b}}(s) > 0$ for s with $\mathfrak{L}(s) = -\infty$; clipping with such candidates is not useful as it induces a value of $-\infty$. Expected total rewards on fully observable MDPs are computed using *Sound Value Iteration* [39] with relative precision 10^{-6}. MILPs are solved using Gurobi [21].

Set-up. We evaluate our under-approximation approach with cut-offs only and with enabled belief clipping procedure using grid resolutions $\eta = 2, 3, 4, 6$. We consider the same POMDP benchmarks[9] as in [37,8]. The POMDPs are scalable versions of case studies stemming from various application domains. To establish an external baseline, we compare with the approach of [37] implemented in Prism [31]. Prism generates an under-approximation based on an optimal policy for an over-approximative MDP which—in contrast to Storm—means that always both, under- and over-approximations, have to be computed. We ran Prism with resolutions $\eta = 2, 3, 4, 6, 8, 10$ and report on the *best* approximation obtained. To provide a further reference for the tightness of our under-approximation, we compute over-approximative bounds as in [8] using the implementation in Storm with a resolution of $\eta = 8$. All experiments were run on an Intel® Xeon® Platinum 8160 CPU using 4 threads[10], 64GB RAM and a time limit of 2 hours.

Results. Tables 1 and 2 show our results for maximising and minimising properties, respectively. The first columns contain for each POMDP the benchmark name,

[9] Instances with a finite belief MDP that would be fully explored by our algorithm are omitted since the exact value can be obtained without approximation techniques.

[10] For our implementation, only Gurobi runs multi-threaded. Prism uses multiple threads for garbage collection.

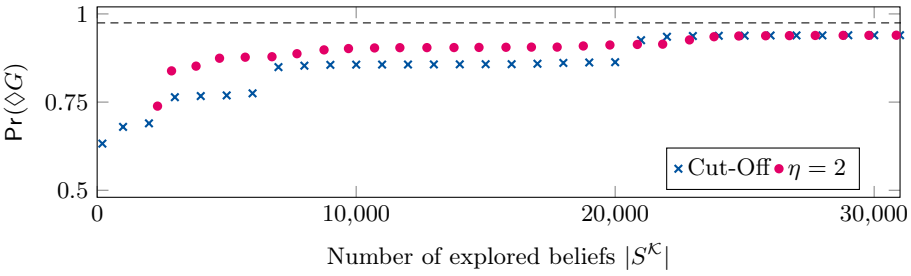

Figure 5. Accuracy for Drone 4-2 with different sizes of approximation MDP $\mathcal{K}_{\mathcal{M}}$

model parameters, property type (probabilities (P) or rewards (R)), and the numbers of states, state-action pairs, and observations. Column PRISM gives the result with the smallest gap between over- and under-approximation computed with the approach of [37]. For maximising (minimising) properties, our approach competes with the lower (upper) bound of the provided interval. The relevant value is marked in bold. We also provide the computation time and the considered resolution η. For our implementation, we give results for the configuration with disabled clipping and for clipping with different resolutions η. In each cell, we give the obtained value, the computation time and the number of states in the abstraction MDP $\mathcal{K}_{\mathcal{M}}$. Time- and memory-outs are indicated by TO and MO. The right-most column indicates the over-approximation value computed via [8].

Discussion. The pure cut-off approach yields valid under-approximations in *all* benchmark instances—often exceeding the accuracy of the approach of [37] while being consistently faster. In some cases, the resulting values improve when clipping is enabled. However, larger candidate sets significantly increase the computation time which stems from the fact that many clipping MILPs have to be solved.

For Drone 4-2, Fig. 5 plots the resulting under-approximation values (y-axis) for varying sizes of the explored MDP $\mathcal{K}_{\mathcal{M}}$ (x-axis). The horizontal, dashed line indicates the computed over-approximation value. The quality of the approximation further improves with an increased number of explored beliefs.

5 Conclusion

We presented techniques to safely under-approximate expected total rewards in POMDPs. The approach scales to large POMDPs and often produces tight lower bounds. Belief clipping generally does not improve on the simpler cut-off approach in terms of results and performance. However, considering—and optimising—the approach for particular classes of POMDPs might prove beneficial. Future work includes integrating the algorithm into a refinement loop that also considers over-approximation techniques from [8]. Furthermore, lifting our approach to partially observable stochastic games is promising.

Data Availability. The artifact [9] accompanying this paper contains source code, benchmark files, and replication scripts for our experiments.

References

1. Amato, C., Bernstein, D.S., Zilberstein, S.: Optimizing fixed-size stochastic controllers for POMDPs and decentralized POMDPs. Auton. Agents Multi Agent Syst. **21**(3), 293–320 (2010)
2. Ashok, P., Butkova, Y., Hermanns, H., Kretínský, J.: Continuous-time Markov decisions based on partial exploration. In: ATVA. Lecture Notes in Computer Science, vol. 11138, pp. 317–334. Springer (2018)
3. Aström, K.J.: Optimal control of Markov processes with incomplete state information. J. of Mathematical Analysis and Applications **10**(1), 174–205 (1965)
4. Baier, C., Katoen, J.P.: Principles of model checking. MIT Press (2008)
5. Bellman, R.: A Markovian decision process. Journal of Mathematics and Mechanics **6**, 679–684 (1957)
6. Bonet, B.: Solving large POMDPs using real time dynamic programming. In: AAAI Fall Symp. on POMDPs (1998)
7. Bonet, B., Geffner, H.: Solving POMDPs: RTDP-Bel vs. Point-based Algorithms. In: IJCAI. pp. 1641–1646 (2009)
8. Bork, A., Junges, S., Katoen, J., Quatmann, T.: Verification of indefinite-horizon POMDPs. In: ATVA. Lecture Notes in Computer Science, vol. 12302, pp. 288–304. Springer (2020)
9. Bork, A., Katoen, J.P., Quatmann, T.: Artifact for Paper: Under-Approximating Expected Total Rewards in POMDPs. Zenodo (2022). https://doi.org/10.5281/zenodo.5643643
10. Bork, A., Katoen, J.P., Quatmann, T.: Under-Approximating Expected Total Rewards in POMDPs. arXiv e-print (2022), https://arxiv.org/abs/2201.08772
11. Brázdil, T., Chatterjee, K., Chmelik, M., Forejt, V., Křetínskỳ, J., Kwiatkowska, M., Parker, D., Ujma, M.: Verification of Markov decision processes using learning algorithms. In: ATVA. Lecture Notes in Computer Science, vol. 8837, pp. 98–114. Springer (2014)
12. Braziunas, D., Boutilier, C.: Stochastic local search for POMDP controllers. In: AAAI. pp. 690–696. AAAI Press / The MIT Press (2004)
13. Carr, S., Jansen, N., Topcu, U.: Verifiable rnn-based policies for POMDPs under temporal logic constraints. In: IJCAI. pp. 4121–4127. ijcai.org (2020)
14. Carr, S., Jansen, N., Wimmer, R., Serban, A.C., Becker, B., Topcu, U.: Counterexample-guided strategy improvement for POMDPs using recurrent neural networks. In: IJCAI. pp. 5532–5539. ijcai.org (2019)
15. Chatterjee, K., Chmelík, M., Davies, J.: A symbolic SAT-based algorithm for almost-sure reachability with small strategies in POMDPs. In: AAAI. pp. 3225–3232 (2016)
16. Chatterjee, K., Chmelík, M., Gupta, R., Kanodia, A.: Optimal cost almost-sure reachability in POMDPs. Artificial Intelligence **234**, 26–48 (2016)
17. Chatterjee, K., Doyen, L., Henzinger, T.A.: Qualitative analysis of partially-observable Markov decision processes. In: MFCS. Lecture Notes in Computer Science, vol. 6281, pp. 258–269. Springer (2010)
18. Cheng, H.T.: Algorithms for partially observable Markov decision processes. Ph.D. thesis, University of British Columbia (1988)
19. Doshi, F., Pineau, J., Roy, N.: Reinforcement learning with limited reinforcement: Using Bayes risk for active learning in POMDPs. In: ICML. pp. 256–263 (2008)
20. Eagle, J.N.: The optimal search for a moving target when the search path is constrained. Operations Research **32**(5), 1107–1115 (1984)

21. Gurobi Optimization, LLC: Gurobi Optimizer Reference Manual (2021), https://www.gurobi.com
22. Hauskrecht, M.: Value-function approximations for partially observable Markov decision processes. J. Artif. Intell. Res. **13**, 33–94 (2000)
23. Hensel, C., Junges, S., Katoen, J., Quatmann, T., Volk, M.: The probabilistic model checker Storm. Int. J. on Software Tools for Technology Transfer (2021). https://doi.org/10.1007/s10009-021-00633-z
24. Horák, K., Bošanský, B., Chatterjee, K.: Goal-HSVI: Heuristic Search Value Iteration for Goal POMDPs. In: IJCAI. pp. 4764–4770. ijcai.org (7 2018)
25. Itoh, H., Nakamura, K.: Partially observable Markov decision processes with imprecise parameters. Artificial Intelligence **171**(8-9), 453–490 (2007)
26. Jansen, N., Dehnert, C., Kaminski, B.L., Katoen, J., Westhofen, L.: Bounded model checking for probabilistic programs. In: ATVA. Lecture Notes in Computer Science, vol. 9938, pp. 68–85 (2016)
27. Junges, S., Jansen, N., Seshia, S.A.: Enforcing almost-sure reachability in POMDPs. In: CAV (2). Lecture Notes in Computer Science, vol. 12760, pp. 602–625. Springer (2021)
28. Junges, S., Jansen, N., Wimmer, R., Quatmann, T., Winterer, L., Katoen, J.P., Becker, B.: Finite-state Controllers of POMDPs via Parameter Synthesis. In: UAI. pp. 519–529. AUAI Press (2018)
29. Kaelbling, L.P., Littman, M.L., Cassandra, A.R.: Planning and acting in partially observable stochastic domains. Artificial Intelligence **101**(1-2), 99–134 (1998)
30. Kurniawati, H., Hsu, D., Lee, W.S.: SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In: Robotics: Science and Systems. vol. 2008 (2008)
31. Kwiatkowska, M., Norman, G., Parker, D.: PRISM 4.0: Verification of probabilistic real-time systems. In: CAV. Lecture Notes in Computer Science, vol. 6806, pp. 585–591. Springer (2011)
32. Lovejoy, W.S.: Computationally feasible bounds for partially observed Markov decision processes. Operations Research **39**(1), 162–175 (1991)
33. Madani, O., Hanks, S., Condon, A.: On the undecidability of probabilistic planning and infinite-horizon partially observable Markov decision problems. In: AAAI/IAAI. pp. 541–548 (1999)
34. Madani, O., Hanks, S., Condon, A.: On the undecidability of probabilistic planning and related stochastic optimization problems. Artificial Intelligence **147**(1-2), 5–34 (2003)
35. Meuleau, N., Kim, K.E., Kaelbling, L.P., Cassandra, A.R.: Solving POMDPs by searching the space of finite policies. In: UAI. pp. 417–426 (1999)
36. Monahan, G.E.: State of the art — a survey of partially observable Markov decision processes: theory, models, and algorithms. Management Science **28**(1), 1–16 (1982)
37. Norman, G., Parker, D., Zou, X.: Verification and Control of Partially Observable Probabilistic Systems. Real-Time Systems **53**(3), 354–402 (2017)
38. Pineau, J., Gordon, G., Thrun, S.: Point-based value iteration: An anytime algorithm for POMDPs. In: IJCAI. vol. 3, pp. 1025–1032 (2003)
39. Quatmann, T., Katoen, J.: Sound value iteration. In: CAV (1). Lecture Notes in Computer Science, vol. 10981, pp. 643–661. Springer (2018)
40. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach (4th Edition). Pearson (2020)
41. Schrijver, A.: Theory of Linear and Integer Programming. John Wiley & Sons (1986)

42. Shani, G., Pineau, J., Kaplow, R.: A survey of point-based POMDP solvers. Autonomous Agents and Multi-Agent Systems **27**(1), 1–51 (2013)
43. Silver, D., Veness, J.: Monte-Carlo planning in large POMDPs. In: NIPS. pp. 2164–2172 (2010)
44. Smallwood, R.D., Sondik, E.J.: The optimal control of partially observable Markov processes over a finite horizon. Operations Research **21**(5), 1071–1088 (1973)
45. Smith, T., Simmons, R.: Heuristic search value iteration for POMDPs. In: UAI. pp. 520–527 (2004)
46. Sondik, E.J.: The Optimal Control of Partially Observable Markov Processes. Ph.D. thesis, Stanford Univ Calif Stanford Electronics Labs (1971)
47. Sondik, E.J.: The optimal control of partially observable Markov processes over the infinite horizon: Discounted costs. Operations research **26**(2), 282–304 (1978)
48. Spaan, M.T., Vlassis, N.: Perseus: Randomized point-based value iteration for POMDPs. J. of Artificial Intelligence Research **24**, 195–220 (2005)
49. Volk, M., Junges, S., Katoen, J.P.: Fast dynamic fault tree analysis by model checking techniques. IEEE Transactions on Industrial Informatics **14**(1), 370–379 (2017)
50. Wang, Y., Chaudhuri, S., Kavraki, L.E.: Bounded Policy Synthesis for POMDPs with Safe-Reachability Objectives. In: AAMAS. pp. 238–246 (2018)
51. Winterer, L., Junges, S., Wimmer, R., Jansen, N., Topcu, U., Katoen, J.P., Becker, B.: Motion planning under partial observability using game-based abstraction. In: CDC. pp. 2201–2208. IEEE (2017)
52. Zhang, N.L., Lee, S.S.: Planning with partially observable Markov decision processes: advances in exact solution method. In: UAI. pp. 523–530 (1998)
53. Zhang, N.L., Zhang, W.: Speeding up the convergence of value iteration in partially observable Markov decision processes. Journal of Artificial Intelligence Research **14**, 29–51 (2001)