Optimal Cost Almost-sure Reachability in POMDPs

1 Anonymous and 2 Anonymous and 3 Anonymous and 4 Anonymous

Technical Report No. IST-2014-307-v1+1
Deposited at 09 Sep 2014 11:39
http://repository.ist.ac.at/307/1/main.pdf

IST Austria (Institute of Science and Technology Austria)
Am Campus 1
A-3400 Klosterneuburg, Austria

Optimal Cost Almost-sure Reachability in POMDPs (Full Version)

Anonymous1, Anonymous2, Anonymous3, and Anonymous4

Abstract. We consider partially observable Markov decision processes (POMDPs) with a set of target states and every transition is associated with an integer cost. The optimization objective we study asks to minimize the expected total cost till the target set is reached, while ensuring that the target set is reached almost-surely (with probability 1). We show that for integer costs approximating the optimal cost is undecidable. For positive costs, our results are as follows: (i) we establish matching lower and upper bounds for the optimal cost and the bound is double exponential; (ii) we show that the problem of approximating the optimal cost is decidable and present approximation algorithms developing on the existing algorithms for POMDPs with finite-horizon objectives. While the worst-case running time of our algorithm is double exponential, we also present efficient stopping criteria for the algorithm and show experimentally that it performs well in many examples of interest.

1 Introduction

Partially observable Markov decision processes (POMDPs). *Markov decision processes (MDPs)* are standard models for probabilistic systems that exhibit both probabilistic as well as nondeterministic behavior [11]. MDPs are widely used to model and solve control problems for stochastic systems [10, 26]: nondeterminism represents the freedom of the controller to choose a control action, while the probabilistic component of the behavior describes the system response to control actions. In *perfect-observation (or perfect-information) MDPs (PIMDPs)* the controller observes the current state of the system precisely to choose the next control actions, whereas in *partially observable MDPs (POMDPs)* the state space is partitioned according to observations that the controller can observe, i.e., given the current state, the controller can only view the observation of the state (the partition the state belongs to), but not the precise state [22]. POMDPs provide the appropriate model to study a wide variety of applications such as in computational biology [8], speech processing [21], image processing [7], robot planning [15, 12], reinforcement learning [13], to name a few. POMDPs also subsume many other powerful computational models such as probabilistic finite automata (PFA) [27, 24] (since probabilistic finite automata (aka blind POMDPs) are a special case of POMDPs with a single observation).

Classical optimization objectives. In stochastic optimization problems related to POMDPs, the transitions in the POMDPs are associated with integer costs, and the two classical objectives that have been widely studied are *finite-horizon* and *discounted-sum* objectives [10, 26, 22]. For finite-horizon objectives, a finite length k is given and the goal is to minimize the expected total cost for k steps. In discounted-sum objectives, the cost in the j-th step is multiplied by γ^j, for $0 < \gamma < 1$, and the goal is to minimize the expected total discounted cost over the infinite horizon.

Reachability and total-cost. In this work we consider a different optimization objective for POMDPs. We consider POMDPs with a set of target states, and the optimization objective is to minimize the expected total cost till the target set is reached. First, note that the objective is not the discounted sum, but the total sum without discounts. Second, the objective is not a finite-horizon objective, as there is no bound apriori known to reach the target set, and along different paths the target set can be reached at different time points. The objective we consider is very relevant in many control applications such as in robot planning: for example, the robot has a target or goal; and the objective is to minimize the number of steps to reach the target, or every transition is associated with energy consumption and the objective is to reach the target with minimal energy consumption.

Our contributions. In this work we study POMDPs with a set of target states, and costs in every transition, and the goal is to minimize the expected total cost till the target set is reached, while ensuring that the target set is reached almost-surely (with probability 1). Our results are as follows:

1. *(Integer costs).* We first show that if the transition costs are integers, then approximating the optimal cost is undecidable.
2. *(Positive integer costs).* Since the problem is undecidable for integer costs, we next consider that costs are positive integers. We first remark that if the costs are positive, and there is a positive probability not to reach the target set, then the expected total cost is infinite. Hence the expected total cost is not infinite only by ensuring that the target is reached almost-surely. First we establish a double-exponential lower and upper bound for the expected optimal cost. We show that the approximation problem is decidable, and present approximation algorithms using the well-known algorithms for finite-horizon objectives.
3. *(Implementation).* Though we establish that in the worst-case the algorithm requires double-exponential time, we also present efficient stopping criteria for the algorithm, and experimentally show that the algorithm is efficient in several practical examples. We have implemented our approximation algorithms developing on the existing implementations for finite-horizon objectives, and present experimental results on a number of well-known examples of POMDPs.

Comparison with Goal-POMDPs. While there are several works for discounted POMDPs [16, 29, 25], as mentioned above the problem we consider is different from discounted POMDPs. The most closely related work is Goal-POMDPs [2]. The key differences are as follows: (a) our results for approximation apply to all POMDPs with positive integer costs, whereas the solution for Goal-POMDPs apply to a strict subclass of POMDPs (see Remark 5); and (b) we present asymptotically tight (double exponential) theoretical bounds on the expected optimal costs.

2 Definitions

We present the definitions of POMDPs, strategies, objectives, and other basic notions required for our results. Throughout this work, we follow standard notations from [26, 17].

Notations. Given a finite set X, we denote by $\mathcal{P}(X)$ the set of subsets of X, i.e., $\mathcal{P}(X)$ is the power set of X. A probability distribution f on X is a function $f : X \to [0, 1]$ such that $\sum_{x \in X} f(x) = 1$, and we denote by $\mathcal{D}(X)$ the set of all probability distributions on X. For $f \in \mathcal{D}(X)$ we denote by $\mathrm{Supp}(f) = \{x \in X \mid f(x) > 0\}$ the support of f.

POMDPs. A *Partially Observable Markov Decision Process (POMDP)* is a tuple $G = (S, \mathcal{A}, \delta, \mathcal{Z}, \mathcal{O}, \lambda_0)$ where: (i) S is a finite set of states; (ii) $\mathcal{A}$ is a finite alphabet of *actions*; (iii) $\delta : S \times \mathcal{A} \to \mathcal{D}(S)$ is a *probabilistic transition function* that given a state s and an action $a \in \mathcal{A}$ gives the probability distribution over the successor states, i.e., $\delta(s, a)(s')$ denotes the transition probability from s to s' given action a; (iv) $\mathcal{Z}$ is a finite set of *observations*; (v) $\mathcal{O} : S \to \mathcal{Z}$ is an *observation function* that maps every state to an observation; and (vi) λ_0 is a probability distribution for the initial state, and for all $s, s' \in \mathrm{Supp}(\lambda_0)$ we require that $\mathcal{O}(s) = \mathcal{O}(s')$. If the initial distribution is Dirac, we often write λ_0 as s_0 where s_0 is the unique starting (or initial) state. Given $s, s' \in S$ and $a \in \mathcal{A}$, we also write $\delta(s'|s, a)$ for $\delta(s, a)(s')$. A state s is *absorbing* if for all actions a we have $\delta(s, a)(s) = 1$ (i.e., s is never left from s). For an observation z, we denote by $\mathcal{O}^{-1}(z) = \{s \in S \mid \mathcal{O}(s) = z\}$ the set of states with observation z. For a set $U \subseteq S$ of states and $Z \subseteq \mathcal{Z}$ of observations we denote $\mathcal{O}(U) = \{z \in \mathcal{Z} \mid \exists s \in U.\ \mathcal{O}(s) = z\}$ and $\mathcal{O}^{-1}(Z) = \bigcup_{z \in Z} \mathcal{O}^{-1}(z)$. A POMDP is a *perfect-observation (or perfect-information) MDP* if each state has a unique observation.

Plays and cones. A *play* (or a path) in a POMDP is an infinite sequence $(s_0, a_0, s_1, a_1, s_2, a_2, \ldots)$ of states and actions such that for all $i \geq 0$ we have $\delta(s_i, a_i)(s_{i+1}) > 0$ and $s_0 \in \mathrm{Supp}(\lambda_0)$. We write Ω for the set of all plays. For a finite prefix $w \in (S \cdot A)^* \cdot S$ of a play, we denote by $\mathsf{Cone}(w)$ the set of plays with w as the prefix (i.e., the cone or cylinder of the prefix w), and denote by $\mathsf{Last}(w)$ the last state of w.

Belief-support and belief-support updates. For a finite prefix $w = (s_0, a_0, s_1, a_1, \ldots, s_n)$ we denote by $\mathcal{O}(w) = (\mathcal{O}(s_0), a_0, \mathcal{O}(s_1), a_1, \ldots, \mathcal{O}(s_n))$ the observation and action sequence associated with w. For a finite sequence $\rho = (z_0, a_0, z_1, a_1, \ldots, z_n)$ of observations and actions, the *belief-support* $\mathcal{B}(\rho)$ after the prefix ρ is the set of states in which a finite prefix of a play can be after the sequence ρ of observations and actions, i.e.,

$$\mathcal{B}(\rho) = \{s_n = \mathsf{Last}(w) \mid w = (s_0, a_0, s_1, a_1, \ldots, s_n), w \text{ is a prefix of a play, and for all } 0 \leq i \leq n.\ \mathcal{O}(s_i) = z_i\}.$$

The belief-support updates associated with finite-prefixes are as follows: for prefixes w and $w' = w \cdot a \cdot s$ the belief-support update is defined inductively as $\mathcal{B}(\mathcal{O}(w')) = \left(\bigcup_{s_1 \in \mathcal{B}(\mathcal{O}(w))} \mathrm{Supp}(\delta(s_1, a))\right) \cap \mathcal{O}^{-1}(\mathcal{O}(s))$, i.e., the set $\left(\bigcup_{s_1 \in \mathcal{B}(\mathcal{O}(w))} \mathrm{Supp}(\delta(s_1, a))\right)$ denotes the possible successors given the belief-supprt $\mathcal{B}(\mathcal{O}(w))$ and action a, and then the intersection with the set of states with the current observation $\mathcal{O}(s)$ gives the new belief-support set.

Strategies (or policies). A *strategy (or a policy)* is a recipe to extend prefixes of plays and is a function $\sigma : (S \cdot A)^* \cdot S \to \mathcal{D}(A)$ that given a finite history (i.e., a finite prefix of a play) selects a probability distribution over the actions. Since we consider POMDPs, strategies are *observation-based*, i.e., for all histories $w = (s_0, a_0, s_1, a_1, \ldots, a_{n-1}, s_n)$ and $w' = (s'_0, a_0, s'_1, a_1, \ldots, a_{n-1}, s'_n)$ such that for all $0 \leq i \leq n$ we have $\mathcal{O}(s_i) = \mathcal{O}(s'_i)$ (i.e., $\mathcal{O}(w) = \mathcal{O}(w')$), we must have $\sigma(w) = \sigma(w')$. In other words, if the observation sequence is the same, then the strategy cannot distinguish between the prefixes and must play the same. A strategy σ is *belief-support based stationary* if it depends only on the current belief-support, i.e., whenever for two histories w and w', we have $\mathcal{B}(\mathcal{O}(w)) = \mathcal{B}(\mathcal{O}(w'))$, then $\sigma(w) = \sigma(w')$.

Strategies with memory and finite-memory strategies A *strategy* with memory is a tuple $\sigma = (\sigma_u, \sigma_n, M, m_0)$ where:(i) *(Memory set).* M is a denumerable set (finite or infinite) of memory elements (or memory states). (ii) *(Action selection function).* The function $\sigma_n : M \to \mathcal{D}(\mathcal{A})$ is the *action selection function* that given the current memory state gives the probability distribution over actions. (iii) *(Memory update function).* The function $\sigma_u : M \times \mathcal{Z} \times \mathcal{A} \to \mathcal{D}(M)$ is the *memory update function* that given the current memory state, the current observation and action, updates the memory state probabilistically. (iv) *(Initial memory).* The memory state $m_0 \in M$ is the initial memory state. A strategy is a *finite-memory* strategy if the set M of memory elements is finite. A strategy is *pure (or deterministic)* if the memory update function and the action selection function are deterministic, i.e., $\sigma_u : M \times \mathcal{Z} \times \mathcal{A} \to M$ and $\sigma_n : M \to \mathcal{A}$. The general class of strategies is sometimes referred to as the class of *randomized infinite-memory* strategies.

Probability and expectation measures. Given a strategy σ and a starting state s, the unique probability measure obtained given σ is denoted as $\mathbb{P}^{\sigma}_{s}(\cdot)$. We first define the measure $\mu^{\sigma}_{s}(\cdot)$ on cones. For $w = s$ we have $\mu^{\sigma}_{s}(\mathsf{Cone}(w)) = 1$, and for $w = s'$ where $s \neq s'$ we have $\mu^{\sigma}_{s}(\mathsf{Cone}(w)) = 0$; and for $w' = w \cdot a \cdot s$ we have $\mu^{\sigma}_{s}(\mathsf{Cone}(w')) = \mu^{\sigma}_{s}(\mathsf{Cone}(w)) \cdot \sigma(w)(a) \cdot \delta(\mathsf{Last}(w), a)(s)$. By Carathéodory's extension theorem, the function $\mu^{\sigma}_{s}(\cdot)$ can be uniquely extended to a probability measure $\mathbb{P}^{\sigma}_{s}(\cdot)$ over Borel sets of infinite plays [1]. We denote by $\mathbb{E}^{\sigma}_{s}[\cdot]$ the expectation measure associated with the strategy σ. For an initial distribution λ_0 we have $\mathbb{P}^{\sigma}_{\lambda_0}(\cdot) = \sum_{s \in S} \lambda_0(s) \cdot \mathbb{P}^{\sigma}_{s}(\cdot)$ and $\mathbb{E}^{\sigma}_{\lambda_0}[\cdot] = \sum_{s \in S} \lambda_0(s) \cdot \mathbb{E}^{\sigma}_{s}[\cdot]$.

Objectives. We consider reachability and total-cost objectives.

- *Reachability objectives.* A *reachability objective* in a POMDP G is a measurable set $\varphi \subseteq \Omega$ of plays and is defined as follows: given a set $T \subseteq S$ of *target* states, the *reachability* objective $\mathsf{Reach}(T) = \{(s_0, a_0, s_1, a_1, s_2 \ldots) \in \Omega \mid \exists i \geq 0 : s_i \in T\}$ requires that a target state in T is visited at least once.
- *Total-cost and finite-length total-cost objectives.* A *total-cost* objective is defined as follows: Let G be a POMDP with a set of absorbing target states T and a *cost* function $\mathsf{c} : S \times \mathcal{A} \to \mathbb{Z}$ that assigns integer-valued weights to all states and actions such that for all states $t \in T$ and all actions $a \in \mathcal{A}$ we have $\mathsf{c}(t, a) = 0$. The total-cost of a play $\rho = (s_0, a_0, s_1, a_1, s_2, a_2, \ldots)$ is $\mathsf{Total}(\rho) = \sum_{i=0}^{\infty} \mathsf{c}(s_i, a_i)$ the sum of the costs of the play. To analyze total-cost objectives we will also require finite-length total-cost objectives, that for a given length k sum the total costs upto length k; i.e., $\mathsf{Total}_k(\rho) = \sum_{i=0}^{k} \mathsf{c}(s_i, a_i)$.

Almost-sure winning. Given a POMDP G with a reachability objective $\mathsf{Reach}(T)$ a strategy σ is *almost-sure winning* iff $\mathbb{P}^{\sigma}_{\lambda_0}(\mathsf{Reach}(T)) = 1$. We will denote by $\mathsf{Almost}_G(T)$ the set of almost-sure winning strategies in POMDP G for the objective $\mathsf{Reach}(T)$. Given a set U such that all states in U have the same observation, a strategy is almost-sure winning from U, if given the uniform probability distribution λ_U over U we have $\mathbb{P}^{\sigma}_{\lambda_U}(\mathsf{Reach}(T)) = 1$; i.e., the strategy ensures almost-sure winning if the starting belief-support is U.

Optimal cost under almost-sure winning and approximations. Given a POMDP G with a reachability objective $\mathsf{Reach}(T)$ and a cost function c we are interested in minimizing the expected total cost before reaching the target set T, while ensuring that the target set is reached almost-surely. Formally, the value of an almost-sure winning strategy $\sigma \in \mathsf{Almost}_G(T)$ is the expectation $\mathsf{Val}(\sigma) = \mathbb{E}^{\sigma}_{\lambda_0}[\mathsf{Total}]$. The *optimal cost* is defined as the infimum of expected costs among all almost-sure winning strategies: $\mathsf{optCost} = \inf_{\sigma \in \mathsf{Almost}_G(T)} \mathsf{Val}(\sigma)$. We consider the computational problems of approximating $\mathsf{optCost}$ and compute strategies $\sigma \in \mathsf{Almost}_G(T)$ such that the value $\mathsf{Val}(\sigma)$ approximates the optimal cost $\mathsf{optCost}$. Formally, given $\epsilon > 0$, the *additive* approximation problem asks to compute a strategy $\sigma \in \mathsf{Almost}_G(T)$ such that $\mathsf{Val}(\sigma) \leq \mathsf{optCost} + \epsilon$; and the *multiplicative* approximation asks to compute a strategy $\sigma \in \mathsf{Almost}_G(T)$ such that $\mathsf{Val}(\sigma) \leq \mathsf{optCost} \cdot (1 + \epsilon)$.

Remark 1. We remark about some of our notations.

1. *Rational costs:* We consider integer costs as compared rational costs, and given a POMDP with rational costs one can obtain a POMDP with integer costs by multiplying the costs with the least common multiple of the denominators. The transformation is polynomial given binary representation of numbers.
2. *Probabilistic observations:* Given a POMDP $G = (S, \mathcal{A}, \delta, \mathcal{Z}, \mathcal{O}, \lambda_0)$, the most general type of the observation function $\mathcal{O}$ considered in the literature is of type $S \times \mathcal{A} \to \mathcal{D}(\mathcal{Z})$, i.e., the state and the action gives a probability distribution over the set of observations $\mathcal{Z}$. We show how to transform the POMDP G into one where the observation function is deterministic and defined on states, i.e., of type $S \to \mathcal{Z}$ as in our definitions. We construct an equivalent POMDP $G' = (S', \mathcal{A}, \delta', \mathcal{Z}, \mathcal{O}', \lambda'_0)$ as follows: (i) the new state space is $S' = S \times \mathcal{Z}$; (ii) the transition function δ' given a state $(s, z) \in S'$ and an action a is as follows $\delta'((s, z), a)(s', z') = \delta(s, a)(s') \cdot \mathcal{O}(s', a)(z')$; and (iii) the deterministic observation function for a state $(s, z) \in S'$ is defined as $\mathcal{O}'((s, z)) = z$. Informally, the probabilistic aspect of the observation function is captured in the transition function, and by enlarging the state space with the product with the observations, we obtain an observation function only on states. Thus we consider observation on states which greatly simplifies the notation.
3. *Strategies:* Note that in our definition of strategies, the strategies operate on state action sequences, rather than observation action sequences. However, since we restrict strategies to be observation based, in effect they operate on observation action sequences.

3 Approximating optCost for Integer Costs

In this section we will show that the problem of approximating the optimal cost $\mathsf{optCost}$ is undecidable. We will show that deciding whether the optimal cost $\mathsf{optCost}$ is $-\infty$ or not is undecidable in POMDPs with integer costs. We present a reduction from the standard undecidable problem for probabilistic finite automata (PFA). A PFA $\mathsf{P} = (S, \mathcal{A}, \delta, F, s_0)$ is a special case of a POMDP $G = (S, \mathcal{A}, \delta, \mathcal{Z}, \mathcal{O}, s_0)$ with a single observation $\mathcal{Z} = \{z\}$ such that for all states $s \in S$ we have $\mathcal{O}(s) = z$. Moreover, the PFA proceeds for only finitely many steps, and has a set F of desired final states. The *strict emptiness problem* asks for the existence of a strategy w (a finite word over the alphabet $\mathcal{A}$) such that the measure of the runs ending in the desired final states F is strictly greater than $\frac{1}{2}$; and the strict emptiness problem for PFA is undecidable [24].

Reduction. Given a PFA $\mathsf{P} = (S, \mathcal{A}, \delta, F, s_0)$ we construct a POMDP $G = (S', \mathcal{A}', \delta', \mathcal{Z}, \mathcal{O}, s'_0)$ with a cost function c and a target set T such that there exists a word $w \in \mathcal{A}^*$ accepted with probability strictly greater than $\frac{1}{2}$ in PFA P iff the optimal cost in the POMDP G is $-\infty$. Intuitively, the construction of the POMDP G is as follows: for every state $s \in S$ of P we construct a pair of states $(s, 1)$ and $(s, -1)$ in S' with the property that $(s, -1)$ can only be reached with a new action \$ (not in $\mathcal{A}$) played in state $(s, 1)$. The transition function δ' from the state $(s, -1)$ mimics the transition function δ, i.e., $\delta'((s, -1), a)((s', 1)) = \delta(s, a)(s')$. The cost c of $(s, 1)$ (resp. $(s, -1)$) is 1 (resp. -1), ensuring the sum of the pair to be 0. We add a new available action # that when played in a final state reaches a newly added state $\mathsf{good} \in S'$, and when played in a non-final state reaches a newly added state $\mathsf{bad} \in S'$. For states good and bad given action # the next state is the initial state; with negative cost -1 for good and positive cost 1 for bad. We introduce a single absorbing target state $T = \{\mathsf{target}\}$ and give full power to the player to decide when to reach the target state from the initial state, i.e., we introduce a new action $\surd$ that when played in the initial state deterministically reaches the target state target.

An illustration of the construction on an example is depicted on Figure 1. Whenever an action is played in a state where it is not available, the POMDP reaches a losing absorbing state, i.e., an absorbing state with cost 1 on all actions, and for brevity we omit transitions to the losing absorbing state. The formal construction of the POMDP G is as follows:

- $S' = (S \times \{-1, 1\}) \cup \{\mathsf{good}, \mathsf{bad}, \mathsf{target}\}$,
- $s'_0 = (s_0, 1)$,
- $\mathcal{A}' = \mathcal{A} \cup \{\#, \$, \surd\}$,
- The actions $a \in \mathcal{A} \cup \{\#\}$ in states $(s, 1)$ (for $s \in S$) lead to the losing absorbing state; the action \$ in states $(s, -1)$ (for $s \in S$) leads to the losing absorbing state; and the actions $a \in \mathcal{A} \cup \{\$\}$ in states good and bad lead to the losing absorbing state. The action $\surd$ played in any state other than the initial state s'_0 also leads to the losing absorbing state. The other transitions are as follows: For all $s \in S$: (i) $\delta'((s, 1), \$)((s, -1)) = 1$, (ii) for all $a \in \mathcal{A}$ we have $\delta'((s, -1), a)((s', 1)) = \delta(s, a)(s')$, and (iii) for action $\#$ and $\surd$ we have

$$\delta'((s,-1),\#)(\mathsf{good}) = \begin{cases} 1 & \text{if } s \in F; \\ 0 & \text{otherwise;} \end{cases} \qquad \delta'((s,-1),\#)(\mathsf{bad}) = \begin{cases} 1 & \text{if } s \notin F; \\ 0 & \text{otherwise;} \end{cases}$$

$$\delta'(\mathsf{good}, \#)(s'_0) = 1; \qquad \delta'(\mathsf{bad}, \#)(s'_0) = 1; \qquad \delta'(s'_0, \surd)(\mathsf{target}) = 1;$$

- there is a single observation $\mathcal{Z} = \{o\}$, and all the states $s \in S'$ have $\mathcal{O}(s) = o$.

We define the cost function c assigning only two different costs and only as a function of the state, i.e., $\mathsf{c} : S' \to \{-1, 1\}$ and show the undecidability even for this special case of cost functions. For all $s \in S$ the cost is $\mathsf{c}((s, -1)) = -1$, and similarly $\mathsf{c}((s, 1)) = 1$, and the remaining states have costs $\mathsf{c}(\mathsf{good}) = -1$ and $\mathsf{c}(\mathsf{bad}) = 1$. The absorbing target state has cost 0; i.e., $\mathsf{c}(\mathsf{target}) = 0$. Note that though the costs are assigned as function of states, the costs appear on the out-going transitions of the respective states.

Intuitive proof idea. The basic idea of the proof is as follows: Consider a word w accepted by the PFA with probability at least $\nu > \frac{1}{2}$. Let the length of the word be $|w| = n$, and $w[i]$ denote the i^{th} letter in w. Consider a strategy in the POMDP $u = (\$\ w[1]\ \$\ w[2] \ldots\ \$w[n]\ \#\ \#)^k \surd$ for some constant $k \geq 0$; that plays alternately the letters in w and \$, then two #'s, repeat the above k times, and finally plays $\surd$. For any $\tau < 0$, for $k \geq \frac{\tau - 1}{1 - 2 \cdot \nu}$, the expected total cost is below τ. Hence if the answer to the strict emptiness problem is yes, then the optimal cost is $-\infty$. Conversely, if there is no word accepted with probability strictly greater than $\frac{1}{2}$, then the expected total cost between consecutive visits to the starting state is positive, and hence the optimal cost is at least 1. We now formalize the intuitive proof idea.

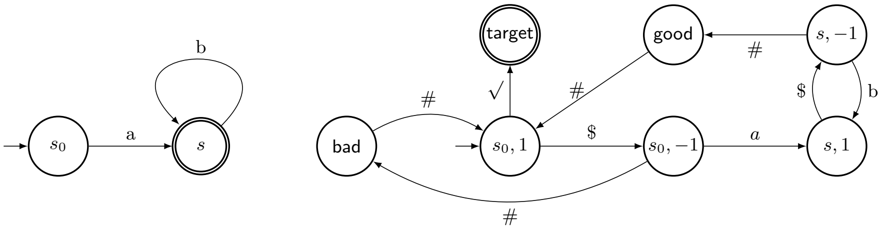

Fig. 1. PFA P to a POMDP G

Lemma 1. *If there exists a word $w \in \mathcal{A}^*$ that is accepted with probability strictly greater than $\frac{1}{2}$ in P, then the optimal cost* $\mathsf{optCost}$ *in the POMDP G is $-\infty$.*

Proof. Let $w \in \mathcal{A}^*$ be a word that is accepted in P with probability $\nu > \frac{1}{2}$ and let $\tau \in \mathbb{R}$ be any negative real-number threshold. We will construct a strategy in POMDP G ensuring that the target state target is reached almost-surely and the value of the strategy is below τ. As this is will be true for every $\tau < 0$ it will follow that the optimal cost $\mathsf{optCost}$ is $-\infty$.

Let the length of the word be $|w| = n$. We construct a pure finite-memory strategy in the POMDP G as follows: We denote by $w[i]$ the i^{th} action in the word w. The finite-memory strategy we construct is specified as a word $u = (\$\, w[1]\, \$\, w[2] \dots \$w[n]\; \#\; \#)^k \surd$ for some constant $k \geq 0$, i.e., the strategy plays alternately the letters in w and \$, then two #'s, repeat the above k times, and finally plays $\surd$. Observe that by the construction of the POMDP G, the sequence of costs (that appear on the transitions) is $(1, -1)^n$ followed by (i) -1 with probability ν (when F is reached), and (ii) $+1$ otherwise; and the whole sequence is repeated k times.

Let $r_1, r_2, r_3, \dots r_m$ be the finite sequence of costs and $s_j = \sum_{i=1}^{j} r_i$. The sequence of costs can be partitioned into blocks of length $2 \cdot n + 1$, intuitively corresponding to the transitions of a single run on the word $(\$\, w[1]\, \$\, w[2] \dots \$w[n]\; \#\, \#)$. We define a random variable X_i denoting the sum of costs of the i^{th} block in the sequence, i.e., with probability ν for all i the value of X_i is -1 and with probability $1 - \nu$ the value is 1. The expected value of X_i is therefore equal to $\mathbb{E}[X_i] = 1 - 2 \cdot \nu$, and as we have that $\nu > \frac{1}{2}$ it follows that $\mathbb{E}[X_i] < 0$. The fact that after the ## the initial state is reached implies that the random variable sequence $(X_i)_{0 \leq i \leq k}$ is a finite sequence of i.i.d's. By linearity of expectation we have that the expected total cost of the word u is $k \cdot \mathbb{E}[X_i]$ plus an additional 1 for the last $\surd$ action. Therefore, by choosing an appropriately large k (in particular for $k \geq \frac{\tau - 1}{1 - 2 \cdot \nu}$) we have the expected total cost is below τ. As playing the $\surd$ action from the initial state reaches the target state target almost-surely, and after ## the initial state is reached almost-surely, we have that by playing according to the strategy u the target state target is reached almost-surely. The desired result follows. ◻

We now show that pure finite-memory strategies are sufficient for the POMDP we constructed from the probabilistic automata, and then prove a lemma that proves the converse of Lemma 1.

Lemma 2. *Given the POMDP G of our reduction from the PFA, if there is a randomized (possibly infinite-memory) strategy σ with $\mathsf{Val}(\sigma) < 1$, then there exists a pure finite-memory strategy σ' with $\mathsf{Val}(\sigma') < 1$.*

Proof. Let σ be a randomized (possibly infinite-memory) strategy with the expected total cost strictly less than 1. As there is a single observation in the POMDP G constructed in our reduction, the strategy does not receive any useful feedback from the play, i.e., the memory update function σ_u always receives as one of the parameters the unique observation. Note that with probability 1 the resolving of the probabilities in the strategy σ leads to finite words of the form $\rho = w_1\#\#w_2\#\# \dots \#\#w_n\#\#\surd$, as otherwise the target state target is not reached with probability 1. From each such word ρ we extract the finite words $w_1, w_2, \dots, w_n$ that occur in ρ, and consider the union of all such words as $W(\rho)$, and then consider the union W over all such words ρ. We consider two cases:

1. If there exists a word v in W such that the expected total cost after playing the word $v\#\#$ is strictly less than 0, then the pure finite-memory strategy $v\#\#\surd$ ensures that the expected total cost strictly less than 1.
2. Assume towards contradiction that for all the words v in W the expected total cost of $v\#\#$ is at least 0. Then with probability 1 resolving the probabilities in the strategy σ leads to finite words of the form $\overline{w} = w_1\#\#w_2\#\# \dots w_n\#\#\surd$, where each word w_i belongs to W, that is played on the POMDP G. Let us define a random variable X_i denoting the sum between i and $(i+1)$-th occurrence of ##. The expected total cost of $w_i\#\#$ is $\mathbb{E}[X_i]$ and is at least 0 for all i. Therefore the expected cost of the sequence $\overline{w}$ (which has $\surd$ in the end with cost 1) is at least 1. Thus we arrive at a contradiction. Hence, there must exist a word v in W such that $v\#\#$ has an expected total cost strictly less than 0.

This concludes the proof. ◻

Lemma 3. *If there exists no word $w \in \mathcal{A}^*$ that is accepted with probability strictly greater than $\frac{1}{2}$ in P, then the optimal cost $\mathsf{optCost} = 1$.*

Proof. We will show that playing $\surd$ is an optimal strategy. It reaches the target state target almost-surely with an expected total cost of 1. Assume towards contradiction that there exists a strategy (and by Lemma 2, a pure finite-memory strategy) σ with the expected total cost strictly less than 1. Observe that as there is only a single observation in the POMDP G the pure finite-memory strategy σ can be viewed as a finite word of the form $w_1\#\#w_2\#\#\dots w_n\#\#\surd$. We extract the set of words $W = \{w_1, w_2, \dots, w_n\}$ from the strategy σ. By the condition of the lemma, there exists no word accepted in the PFA P with probability strictly greater that $\frac{1}{2}$. As in Lemma 1 we define a random variable X_i denoting the sum of costs after reading $w_i\#\#$. It follows that the expected value of $\mathbb{E}[X_i] \geq 0$ for all $0 \leq i \leq n$. By using the linearity of expectation we have the expected total cost of $w_1\#\#w_2\#\#\dots w_n\#\#$ is at least 0, and hence the expected total cost of the strategy is at least 1 due to the cost of the last $\surd$ action. Thus we have a contradiction to the assumption that the expected total cost of strategy σ is strictly less than 1. □

The above lemmas establish that if the answer to the strict emptiness problem for PFA is yes, then the optimal cost in the POMDP is $-\infty$; and otherwise the optimal cost is 1. Hence in POMDPs with integer costs determining whether the optimal cost $\mathsf{optCost}$ is $-\infty$ or 1 is undecidable, and thus the problem of approximation is also undecidable.

Theorem 1. *The problem of approximating the optimal cost* $\mathsf{optCost}$ *in POMDPs with integer costs is undecidable for all $\epsilon > 0$ both for additive and multiplicative approximation.*

4 Approximating optCost for Positive Costs

In this section we consider POMDPs with positive cost functions, i.e., $\mathsf{c} : S \times \mathcal{A} \to \mathbb{N}$ instead of $\mathsf{c} : S \times \mathcal{A} \to \mathbb{Z}$. Note that the transitions from the absorbing target states have cost 0 as the goal is to minimize the cost till the target set is reached, and also note that all other transitions have cost at least 1. We established (in Theorem 1) that for integer costs the problem of approximating the optimal cost is undecidable, and in this section we show that for positive cost functions the approximation problem is decidable. We first start with a lower bound on the optimal cost.

4.1 Lower Bound on optCost

We present a double-exponential lower bound on $\mathsf{optCost}$ with respect to the number of states of the POMDP. We define a family of POMDPs $\mathcal{F}(n)$, for every n, with a single target state, such that there exists an almost-sure winning strategy, and for every almost-sure winning strategy the expected number of steps to reach the target state is double-exponential in the number of states of the POMDP. Thus assigning cost 1 to every transition we obtain the double-exponential lower bound.

Preliminary. The action set we consider consists of two symbols $\mathcal{A} = \{a, \#\}$. The state space consists of an initial state s_0, a target state target, a losing absorbing state bad and a set of n sub-POMDPs $\mathcal{L}_i$ for $1 \leq i \leq n$. Every sub-POMDP $\mathcal{L}_i$ consists of states Q_i that form a loop of $p(i)$ states $q^i_1, q^i_2, \dots q^i_{p(i)}$, where $p(i)$ denotes the i-th prime number and q^i_1 is the initial state of the sub-POMDP. For every state q^i_j (for $1 \leq j \leq p(i)$) the transition function under action a moves the POMDP to the state $q^i_{(j \mod p(i))+1}$ with probability $\frac{1}{2}$ and to the initial state s_0 with the remaining probability $\frac{1}{2}$. The action $\#$ played in the state $q^i_{p(i)}$ moves the POMDP to the target state target with probability $\frac{1}{2}$ and to the initial state s_0 with the remaining probability $\frac{1}{2}$. For every other state in the loop q^i_j such that $1 \leq j < p(i)$ the POMDP moves under action $\#$ to the losing absorbing state bad with probability 1. The losing state bad and the target state target are absorbing and have a self-loop under both actions with probability 1.

POMDP family $\mathcal{F}(n)$. Given an $n \in \mathbb{N}$ we define the POMDP $\mathcal{F}(n)$ as follows:

- The state space $S = Q_1 \cup Q_2 \cup \dots Q_n \cup \{s_0, \mathsf{bad}, \mathsf{target}\}$.
- There are two available actions $\mathcal{A} = \{a, \#\}$.

- The transition function is defined as follows: action a in the initial state leads to bad with probability 1 and action # in the initial state leads with probability $\frac{1}{n}$ to the initial state of the sub-POMDP $\mathcal{L}_i$ for every $1 \leq i \leq n$. The transitions for the states in the sub-POMDPs are described in the previous paragraph.
- All the states in the sub-POMDPs $\mathcal{L}_i$ do have the same observation z. The remaining states s_0, bad, and target are visible, i.e., each of these three states has its own observation.
- The initial state is s_0.

The cost function c is defined as follows: the self-loop transitions at target have cost 0 and all other transitions have cost 1. An example of the construction for $n = 2$ is depicted in Figure 2, where we omit the losing absorbing state bad and the transitions leading to bad for simplicity.

Intuitive proof idea. For a given $n \in \mathbb{N}$ let $p_n^* = \prod_{i=1}^{n} p(i)$ and $p_n^+ = \sum_{i=1}^{n} p(i)$ denote the product and the sum of the first n prime numbers, respectively. Note that p_n^* is exponential is p_n^+. An almost-sure winning strategy must play as follows: in the initial state s_0 it plays #, and then if it observes the observation z for at least p_n^* consecutive steps, then for each step it must play action a, and at the p_n^* step it can play action #. Hence the probability to reach the target state in p_n^* steps is at most $(\frac{1}{2} \cdot (\frac{1}{2})^{p_n^*})$; and hence the expected number of steps to reach the target state is at least $p_n^* \cdot 2 \cdot 2^{p_n^*}$. The size of the POMDP is polynomial in p_n^+ and thus the expected total cost is double exponential.

Lemma 4. *There exists a family $(\mathcal{F}(n))_{n\in\mathbb{N}}$ of POMDPs of size $\mathcal{O}(p(n))$ for a polynomial p with a reachability objective, such that the following assertion holds: There exists a polynomial q such that for every almost-sure winning strategy the expected total cost to reach the target state is at least $2^{2^{q(n)}}$.*

Proof. For $n \in \mathbb{N}$, consider the POMDP $\mathcal{F}(n)$, and an almost-sure winning strategy in the POMDP. In the first step the strategy needs to play the # action from s_0, as otherwise the losing absorbing state is reached. The POMDP reaches the initial state of the sub-POMDPs $\mathcal{L}_i$, for all i, with positive probability. As all the states in the sub-POMDPs have the same observation z, the strategy cannot base its decision on the current sub-POMDP. The strategy has to play the action a until the observation z is observed for p_n^* steps in a row before playing action #. If the strategy plays the action # before observing the sequence of z observations p_n^* times, then it reaches the losing absorbing state with positive probability (and would not have been an almost-sure winning strategy). This follows from the fact that there is a positive probability of being in a sub-POMDP, where the current state is not the last one of the loop. Hence an almost-sure winning strategy must play a as long as the length of the sequence of the observation z is less than p_n^* consecutive steps. Note that in between the POMDP can move to the initial state s_0, and the strategy restarts. In every step of the sub-POMDPs, with probability $\frac{1}{2}$ the initial state is reached, and the next state is in the sub-POMDP with probability $\frac{1}{2}$. After observing the z observation for p_n^* consecutive steps, the strategy can play the action # that moves the POMDP to the target state target with probability $\frac{1}{2}$ and restarts the POMDP with the remaining probability $\frac{1}{2}$. Therefore the probability of reaching the target state in p_n^* steps is at most $(\frac{1}{2} \cdot (\frac{1}{2})^{p_n^*})$; and hence the expected number of steps to reach the target state is at least $p_n^* \cdot 2 \cdot 2^{p_n^*}$. The size of the POMDP is polynomial in p_n^+ and hence it follows that the expected total cost to reach the target state is at least double exponential in the size of the POMDP. □

4.2 Upper Bound on optCost

In this section we will present a double-exponential upper bound on $\mathsf{optCost}$.

Almost-sure winning belief-supports. Let $\mathsf{Belief}(G)$ denote the set of all belief-supports in a POMDP G, i.e., $\mathsf{Belief}(G) = \{U \subseteq S \mid \exists z \in \mathcal{Z} : U \subseteq \mathcal{O}^{-1}(z)\}$. Let $\mathsf{Belief}_{\mathsf{Win}}(G,T)$ denote the set of almost-sure winning belief-supports, i.e., $\mathsf{Belief}_{\mathsf{Win}}(G,T) = \{U \in \mathsf{Belief}(G) \mid$ there exists an almost-sure winning strategy from U $\}$, i.e., there exists an almost-sure winning strategy with initial distribution λ_U that is the uniform distribution over U.

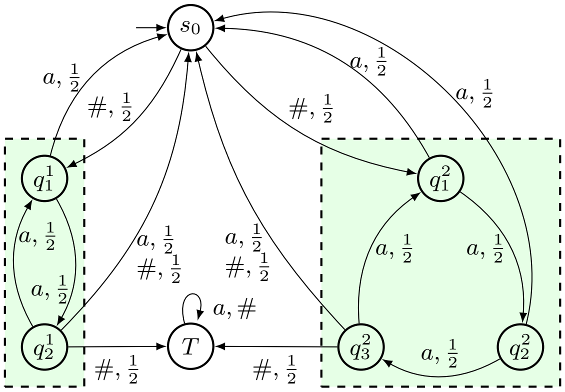

Fig. 2. POMDP $\mathcal{F}(2)$

Restricting to $\mathsf{Belief}_{\mathsf{Win}}(G, T)$**.** In the sequel without loss of generality we will restrict ourselves to belief-supports in $\mathsf{Belief}_{\mathsf{Win}}(G, T)$: since from belief-supports outside $\mathsf{Belief}_{\mathsf{Win}}(G, T)$ there exists no almost-sure winning strategy, all almost-sure winning strategies with starting belief-support in $\mathsf{Belief}_{\mathsf{Win}}(G, T)$ will ensure that belief-supports not in $\mathsf{Belief}_{\mathsf{Win}}(G, T)$ are never reached.

Belief updates. Given a belief-support $U \in \mathsf{Belief}(G)$, an action $a \in \mathcal{A}$, and an observation $z \in \mathcal{Z}$ we denote by $\mathsf{Update}(U, z, a)$ the updated belief-support. Formally, the set $\mathsf{Update}(U, z, a)$ is defined as follows: $\mathsf{Update}(U, z, a) = \bigcup_{s' \in U} \mathrm{Supp}(\delta(s', a)) \cap \mathcal{O}^{-1}(z)$. The set of belief-supports reachable from U by playing an action $a \in \mathcal{A}$ is denoted by $\mathsf{Update}(U, a)$. Formally, $\mathsf{Update}(U, a) = \{U' \subseteq S \mid \exists z \in \mathcal{Z} : U' = \mathsf{Update}(U, z, a)) \wedge U' \neq \emptyset\}$.

Allowed actions. Given a POMDP G and a belief-support $U \in \mathsf{Belief}_{\mathsf{Win}}(G, T)$, we consider the set of actions that are guaranteed to keep the next belief-support U' in $\mathsf{Belief}_{\mathsf{Win}}(G, T)$ and refer these actions as *allowed or safe*. Formally we consider the set of allowed actions as follows: Given a belief-support $U \in \mathsf{Belief}_{\mathsf{Win}}(G, T)$ we define $\mathsf{Allow}(U) = \{a \in \mathcal{A} \mid \forall U' \in \mathsf{Update}(U, a) : U' \in \mathsf{Belief}_{\mathsf{Win}}(G, T)\}$.

We now show that almost-sure winning strategies must only play allowed actions. An easy consequence of the lemma is that for all belief-supports U in $\mathsf{Belief}_{\mathsf{Win}}(G, T)$, there is always an allowed action.

Lemma 5. *Given a POMDP with a reachability objective* $\mathsf{Reach}(T)$*, consider a strategy* σ *and a starting belief-support in* $\mathsf{Belief}_{\mathsf{Win}}(G, T)$*. Given* σ*, if for a reachable belief-support* $U \in \mathsf{Belief}_{\mathsf{Win}}(G, T)$ *the strategy* σ *plays an action* a *not in* $\mathsf{Allow}(U)$ *with positive probability, then* σ *is not almost-sure winning for the reachability objective.*

Proof. Assume the strategy σ reaches the belief-support U and plays an action $a \notin \mathsf{Allow}(U)$. Since the belief-support U is reachable, it follows that given the strategy σ when the belief-support is U, all states in U are reached with positive probability, i.e., given the strategy the belief-support U is reached with positive probability. It follows from the definition of Allow that there exists a belief-support $U' \in \mathsf{Update}(U, a)$ that is not in $\mathsf{Belief}_{\mathsf{Win}}(G, T)$. By the definition of Update there exists an observation $z \in \mathcal{Z}$ such that $U' = \mathsf{Update}(U, z, a)$ and $U' \neq \emptyset$. It follows that by playing a in belief-support U, there is a positive probability of observing observation z and reaching belief-support U' that does not belong to $\mathsf{Belief}_{\mathsf{Win}}(G, T)$. It follows that under action a, given the current belief-support is U, the next belief-support is U' with positive probability. By definition, for all belief-supports U' that does not belong to $\mathsf{Belief}_{\mathsf{Win}}(G, T)$, if the starting belief-support is U', then for all strategies the probability to reach T is strictly less than 1. Hence if U' is reached with positive probability from U under action a, then σ is not almost-sure winning. The desired result follows. □

Corollary 1. *For all* $U \in \mathsf{Belief}_{\mathsf{Win}}(G, T)$ *we have* $\mathsf{Allow}(U) \neq \emptyset$*.*

Markov chains and reachability. A Markov chain $\overline{G} = (\overline{S}, \overline{\delta})$ consists of a finite set $\overline{S}$ of states and a probabilistic transition function $\overline{\delta} : \overline{S} \to \mathcal{D}(\overline{S})$. Given the Markov chain, we consider the directed graph $(\overline{S}, \overline{E})$ where $\overline{E} = \{(\overline{s}, \overline{s}') \mid \delta(\overline{s}' \mid \overline{s}) > 0\}$. The following standard properties of reachability in Markov chains will be used in our proofs [14]:

1. *Property 1 of Markov chains.* For a set $\overline{T} \subseteq \overline{S}$, if for all states $\overline{s} \in \overline{S}$ there is a path to $\overline{T}$ (i.e., for all states there is a positive probability to reach $\overline{T}$), then from all states the set $\overline{T}$ is reached with probability 1.
2. *Property 2 of Markov chains.* In a Markov chain if a set $\overline{T}$ is reached almost-surely from $\overline{s}$, then the expected hitting time from $\overline{s}$ to $\overline{T}$ is at most exponential in the number of the states of the Markov chain.

The strategy σ_{Allow}. We consider a *belief-support-based stationary* (for brevity belief-based)[1] strategy σ_{Allow} as follows: for all belief-supports U in $\mathsf{Belief}_{\mathsf{Win}}(G, T)$, the strategy plays uniformly at random all actions from $\mathsf{Allow}(U)$. Note that as the strategy σ_{Allow} is belief-based, it can be viewed as a finite-memory strategy $\sigma_{\mathsf{Allow}} = (\sigma_u, \sigma_n, \mathsf{Belief}_{\mathsf{Win}}(G, T), m_0)$, where the components are defined as follows: (i) The set of memory elements are the winning belief-supports $\mathsf{Belief}_{\mathsf{Win}}(G, T)$; (ii) the belief-support $m_0 \in \mathsf{Belief}_{\mathsf{Win}}(G, T)$ is the initial belief (i.e., $\mathrm{Supp}(\lambda_0)$); (iii) the action selection function given memory $U \in \mathsf{Belief}_{\mathsf{Win}}(G, T)$ is a uniform distribution over the set $\mathsf{Allow}(U)$ of actions, i.e., $\sigma_n(U) = \mathsf{unif}(\mathsf{Allow}(U))$ where unif denotes the uniform distribution; and (iv) the memory update function given memory $U \in \mathsf{Belief}_{\mathsf{Win}}(G, T)$, observation $z \in \mathcal{Z}$, and action $a \in \mathsf{Allow}(U)$ is defined as the belief-support update U' from belief-support U under action a and observation z, i.e., $\sigma_u(U, z, a) = \mathsf{Update}(U, z, a)$.

The Markov chain $G \upharpoonright \sigma_{\mathsf{Allow}}$. Given a POMDP $G = (S, \mathcal{A}, \delta, \mathcal{Z}, \mathcal{O}, s_0)$ and the strategy $\sigma_{\mathsf{Allow}} = (\sigma_u, \sigma_n, \mathsf{Belief}_{\mathsf{Win}}(G, T), m_0)$ the Markov chain $G \upharpoonright \sigma_{\mathsf{Allow}} = (\overline{S}, \overline{\delta})$ obtained by playing strategy σ_{Allow} in G is defined as follows:

- The set of states $\overline{S}$ is defined as follows: $\overline{S} = \{(s, U) \mid U \in \mathsf{Belief}_{\mathsf{Win}}(G, T), s \in U\}$, i.e., the second components are the almost-sure winning belief-supports and the first component is a state in the belief-support.
- The probability that the next state is (s', U') from a state (s, U) is $\overline{\delta}((s, U))((s', U')) = \sum_{a \in \mathcal{A}} \sigma_n(U)(a) \cdot \delta(s, a)(s') \cdot \sigma_u(U, \mathcal{O}(s'), a)(U')$.

The probability of transition can be decomposed as follows: (i) First an action $a \in \mathcal{A}$ is sampled according to the distribution $\sigma_n(U)$; (ii) then the next state s' is sampled according to the distribution $\delta(s, a)$; and (iii) finally the new memory U' is sampled according to the distribution $\sigma_u(U, \mathcal{O}(s'), a)$.

Remark 2. Note that due to the definition of the strategy σ_{Allow} (that only plays allowed actions) all states (s', U') of the Markov chain $G \upharpoonright \sigma_{\mathsf{Allow}}$ that are reachable from a state (s, U) where $s \in U$ and $U \in \mathsf{Belief}_{\mathsf{Win}}(G, T)$ satisfy that $s' \in U'$ and $U' \in \mathsf{Belief}_{\mathsf{Win}}(G, T)$.

Lemma 6. *The belief-based strategy σ_{Allow} is an almost-sure winning strategy for all belief-supports $U \in$ $\mathsf{Belief}_{\mathsf{Win}}(G, T)$ for the objective $\mathsf{Reach}(T)$.*

Proof. Consider the Markov chain $G \upharpoonright \sigma_{\mathsf{Allow}}$ and a state (s, U) of the Markov chain. As $U \in \mathsf{Belief}_{\mathsf{Win}}(G, T)$ by the definition of almost-sure winning belief-supports, there exists a strategy σ that is almost-sure winning for the reachability objective $\mathsf{Reach}(T)$ starting with belief-support U.

Reachability under σ. Note that by Lemma 5 the strategy must only play allowed actions. The strategy must ensure that from s a target state t is reached with positive probability by playing according to σ (given the initial belief-support is U). It follows that there exists a finite prefix of a play $(s_1, a_1, s_2, a_2, \ldots a_{n-1}, s_n)$ induced by σ where $s_1 = s$ and $s_n = t$ and for all $1 \leq i < n$ we have that $a_i \in \mathrm{Supp}(\sigma(\mathcal{O}((s_1, a_1, s_2, a_2, \ldots, s_i))))$. We define a sequence of belief-supports $U_1, U_2, \ldots U_n$, where $U_1 = U$ and $U_{i+1} = \mathsf{Update}(U_i, \mathcal{O}(s_{i+1}), a_i)$. As σ is an almost-sure winning strategy, it follows from Lemma 5 that $a_i \in \mathsf{Allow}(U_i)$ for all $1 \leq i < n$.

[1] recall, for a belief-support-based stationary strategy, the probability distribution only depends on the current belief-support

Reachability in the Markov chain. Recall that the strategy σ_{Allow} plays all the allowed actions uniformly at random. Hence it follows from the definition of the Markov chain $G\restriction \sigma_{\mathsf{Allow}}$ that for all $0 \leq i < n$ we have $\overline{\delta}((s_i, U_i))((s_{i+1}, U_{i+1})) > 0$, i.e, there is a positive probability to reach (t, U_n) from (s, U) in the Markov chain $G\restriction \sigma_{\mathsf{Allow}}$. It follows that for an arbitrary state (s, U) of the Markov chain $G\restriction \sigma_{\mathsf{Allow}}$ there exists a state (t', U') with $t' \in T$ that is reached with positive probability. In other words, in the graph of the Markov chain, there is a path from all states (s, U) to a state (t', U') where $t' \in T$. Thus by Property 1 of Markov chains it follows that the target set T is reached with probability 1. It follows that σ_{Allow} is an almost-sure winning strategy for all belief-supports in $\mathsf{Belief}_{\mathsf{Win}}(G, T)$. The desired result follows. □

Remark 3 (Computation of σ_{Allow}). It follows from Lemma 6 that the strategy σ_{Allow} can be computed by computing the set of almost-sure winning states in the *belief-support MDP*. The belief-support MDP is a perfect-observation MDP where each state is a belief-support of the original POMDP, and given an action, the next state is obtained according to the belief-support updates. The strategy σ_{Allow} can be obtained by computing the set of almost-sure winning states in the belief-support MDP, and for discrete graph-based algorithms to compute almost-sure winning states in perfect-observation MDPs see [6, 5].

Upper bound. We now establish a double-exponential upper bound on $\mathsf{optCost}$, matching our lower bound from Lemma 4. We have that $\sigma_{\mathsf{Allow}} \in \mathsf{Almost}_G(T)$. Hence we have $\mathsf{Val}(\sigma_{\mathsf{Allow}}) \geq \inf_{\sigma \in \mathsf{Almost}_G(T)} \mathsf{Val}(\sigma) = \mathsf{optCost}$. Once σ_{Allow} is fixed, since the strategy is belief-based (i.e., depends on the subset of states) we obtain an exponential size Markov chain. It follows from Property 2 of Markov chains that given σ_{Allow} the expected hitting time to the target set is at most double exponential. If $\mathsf{c}_{\max}$ denotes the maximal cost of transitions, then $\mathsf{optCost}$ is bounded by $\mathsf{c}_{\max}$ times the expected hitting time. Thus we obtain the following lemma.

Lemma 7. *Given a POMDP G with n states, let $\mathsf{c}_{\max}$ denote the maximal value of the cost of all transitions. There is a polynomial function q such that $\mathsf{optCost} \leq 2^{2^{q(n)}} \cdot \mathsf{c}_{\max}$.*

4.3 Optimal finite-horizon strategies

Our algorithm for approximation of $\mathsf{optCost}$ will use algorithms for optimizing the finite-horizon costs. We first recall the well-known construction of the optimal finite-horizon strategies that minimizes the expected total cost in POMDPs for length k.

Information state. For minimizing the expected total cost, strategies based on information states are sufficient [30]. An *information state* b is defined as a probability distribution over the set of states, where for $s \in S$ the value $b(s)$ denotes the probability of being in state s. We will denote by $\mathcal{H}$ the set of all information states. Given an information state b, an action a, and an observation z, computing the resulting information state b' can be done in a straight forward way, see [4].

Value-iteration algorithm. The standard finite-horizon value-iteration algorithm for expected total cost in the setting of perfect-information MDPs can be formulated by the following equation:

$$V_0^*(s) = 0;$$
$$V_n^*(s) = \min_{a \in \mathcal{A}} \left[\mathsf{c}(s, a) + \sum_{s' \in S} \delta(s, a)(s') V_{n-1}^*(s') \right];$$

where $V_n^*(s)$ represents the value of an optimal policy, when the starting state is s and there are n decision steps remaining. For a POMDP the finite-horizon value-iteration algorithm works on the information states. Let $\psi(b, a)$ denote the probability distribution over the information states given that action a was played in the information state b. The cost function $\mathsf{c}' : \mathcal{H} \times \mathcal{A} \to \mathbb{N}$ that maps every pair of an information state and an action to a positive real-valued cost is defined as follows: $\mathsf{c}'(b, a) = \sum_{s \in S} b(s) \cdot \mathsf{c}(s, a)$. The resulting equation for finite-horizon value-iteration algorithm for POMDPs is as follows:

$$V_0^*(b) = 0;$$
$$V_n^*(b) = \min_{a \in \mathcal{A}} \left[\mathsf{c}'(b, a) + \sum_{b' \in \mathcal{H}} \psi(b, a)(b') V_{n-1}^*(b') \right].$$

The optimal strategy σ_k^{FO} and σ_k^*. In our setting we modify the standard finite-horizon value-iteration algorithm by restricting the optimal strategy to play only allowed actions and restrict it only to belief-supports in the set $\mathsf{Belief}_{\mathsf{Win}}(G,T)$. The equation for the value-iteration algorithm is defined as follows:

$$
\begin{aligned}
&V_0^*(b) = 0;\\
&V_n^*(b) = \min_{a \in \mathsf{Allow}(\mathrm{Supp}(b))} \left[\mathsf{c}'(b,a) + \sum_{b' \in \mathcal{H}} \psi(b,a)(b') V_{n-1}^*(b') \right].
\end{aligned}
$$

We obtain a strategy $\sigma_k^{\mathsf{FO}} = (\sigma_u, \sigma_n, M, m_0)$ that is finite-horizon optimal for length k (here FO stands for finite-horizon optimal) from the above equation as follows: (i) the set of memory elements M is defined as $\mathcal{H} \times \mathbb{N}$; (ii) the initial memory state is (λ_0, k); (iii) for all $1 \le n \le k$, the action selection function $\sigma_n((b,n))$ selects an arbitrary action a such that $a = \arg\min_{a \in \mathsf{Allow}(\mathrm{Supp}(b))} \left[\mathsf{c}'(b,a) + \sum\limits_{b' \in \mathcal{H}} \psi(b,a)(b') V_{n-1}^*(b') \right]$; and (iv) the memory update function given a memory state (b,n), action a, and an observation o updates to a memory state $(b', n-1)$, where b' is the unique information state update from information state b under action a and observation z. As the target states T in the POMDP G are absorbing and the costs on all outgoing edges from the target states are the only edges with cost 0, it follows that for sufficiently large n the strategy σ_{FO}^k minimizes the expected total cost to reach the target set T. Given σ_k^{FO}, we define a strategy σ_k^* as follows: for the first k steps, the strategy σ_k^* plays as the strategy σ_k^{FO}, and after the first k steps the strategy plays as the strategy σ_{Allow}.

Lemma 8. *For all $k \in \mathbb{N}$ the strategy σ_k^* is almost-sure winning for the reachability objective* $\mathsf{Reach}(T)$.

Proof. By definition the strategy σ_k^{FO} (and hence the strategy σ_k^*) plays only allowed actions in the first k steps. Hence it follows that every reachable belief-support in the first k steps belongs to $\mathsf{Belief}_{\mathsf{Win}}(G,T)$. After the first k steps, the strategy plays as σ_{Allow}, and by Lemma 6, the strategy σ_{Allow} is almost-sure winning for the reachability objective $\mathsf{Reach}(T)$ from every belief-support in $\mathsf{Belief}_{\mathsf{Win}}(G,T)$. The result follows. □

Note that the only restriction in the construction of the strategy σ_k^{FO} is that it must play only allowed actions, and since almost-sure winning strategies only play allowed actions (by Lemma 5) it follows that σ_k^{FO} (and hence σ_k^*) is optimal for the finite-horizon of length k (i.e., for the objective Total_k) among all almost-sure winning strategies.

Lemma 9. *For all $k \in \mathbb{N}$ we have $\mathbb{E}_{\lambda_0}^{\sigma_k^*}[\mathsf{Total}_k] = \inf_{\sigma \in \mathsf{Almost}_G(T)} \mathbb{E}_{\lambda_0}^{\sigma}[\mathsf{Total}_k]$.*

Note that since in the first k steps σ_k^* plays as σ_k^{FO} we have the following proposition.

Proposition 1. *For all $k \in \mathbb{N}$ we have $\mathbb{E}_{\lambda_0}^{\sigma_k^*}[\mathsf{Total}_k] = \mathbb{E}_{\lambda_0}^{\sigma_k^{\mathsf{FO}}}[\mathsf{Total}_k]$.*

4.4 Approximation algorithm

In this section we will show that for all $\epsilon > 0$ there exists a bound k such that the strategy σ_k^* approximates $\mathsf{optCost}$ within ϵ. First we consider an upper bound on $\mathsf{optCost}$.

Bound $\mathcal{U}_{\mathsf{Allow}}$. We consider an upper bound $\mathcal{U}_{\mathsf{Allow}}$ on the expected total cost of the strategy σ_{Allow} starting in an arbitrary state $s \in U$ with the initial belief-support $U \in \mathsf{Belief}_{\mathsf{Win}}(G,T)$. Given a belief-support $U \in \mathsf{Belief}_{\mathsf{Win}}(G,T)$ and a state $s \in U$ let $T_{\mathsf{Allow}}(s,U)$ denote the expected total cost of the strategy σ_{Allow} starting in the state s with the initial belief-support U. Then the upper bound is defined as $\mathcal{U}_{\mathsf{Allow}} = \max_{U \in \mathsf{Belief}_{\mathsf{Win}}(G,T), s \in U} T_{\mathsf{Allow}}(s,U)$. As the strategy σ_{Allow} is in $\mathsf{Almost}_G(T)$ it follows that the value $\mathcal{U}_{\mathsf{Allow}}$ is also an upper bound for the optimal cost $\mathsf{optCost}$. Observe that by Lemma 7 it follows that $\mathcal{U}_{\mathsf{Allow}}$ is at most double exponential in the size of the POMDP.

Lemma 10. *We have $\mathsf{optCost} \le \mathcal{U}_{\mathsf{Allow}}$.*

Key lemma. We will now present our key lemma to obtain the bound on k depending on ϵ. We start with a few notations. Given $k \in \mathbb{N}$, let $\mathcal{E}_k$ denote the event of reaching the target set within k steps, i.e., $\mathcal{E}_k = \{(s_0, a_0, s_1, a_1, s_2 \ldots) \in \Omega \mid \exists i \le k : s_i \in T\}$; and $\overline{\mathcal{E}}_k$ the complement of the event $\mathcal{E}_k$ that denotes the target set is not reached within the first k steps. Recall that for plays $\rho = (s_0, a_0, s_1, a_1, s_2, a_2, \ldots)$ we have $\mathsf{Total}_k = \sum_{i=0}^{k} \mathsf{c}(s_i, a_i)$ and we consider $\overline{\mathsf{Total}}_k = \sum_{i=k+1}^{\infty} \mathsf{c}(s_i, a_i)$ the sum of the costs after k steps. Note that we have $\mathsf{Total} = \mathsf{Total}_k + \overline{\mathsf{Total}}_k$.

Lemma 11. *For $k \in \mathbb{N}$ consider the strategy σ^*_k that is obtained by playing an optimal finite-horizon strategy σ^{FO}_k for k steps, followed by strategy σ_{Allow}. Let $\alpha_k = \mathbb{P}^{\sigma^*_k}_{\lambda_0}(\overline{\mathcal{E}}_k)$ denote the probability that the target set is not reached within the first k steps. We have*

$$\mathbb{E}^{\sigma^*_k}_{\lambda_0}[\mathsf{Total}] \le \mathbb{E}^{\sigma^*_k}_{\lambda_0}[\mathsf{Total}_k] + \alpha_k \cdot \mathcal{U}_{\mathit{Allow}}$$

Proof. We have

$$\begin{aligned}
\mathbb{E}^{\sigma^*_k}_{\lambda_0}[\mathsf{Total}] &= \mathbb{P}^{\sigma^*_k}_{\lambda_0}(\mathcal{E}_k) \cdot \mathbb{E}^{\sigma^*_k}_{\lambda_0}[\mathsf{Total} \mid \mathcal{E}_k] + \mathbb{P}^{\sigma^*_k}_{\lambda_0}(\overline{\mathcal{E}}_k) \cdot \mathbb{E}^{\sigma^*_k}_{\lambda_0}[\mathsf{Total} \mid \overline{\mathcal{E}}_k] \\
&= \mathbb{P}^{\sigma^*_k}_{\lambda_0}(\mathcal{E}_k) \cdot \mathbb{E}^{\sigma^*_k}_{\lambda_0}[\mathsf{Total} \mid \mathcal{E}_k] + \mathbb{P}^{\sigma^*_k}_{\lambda_0}(\overline{\mathcal{E}}_k) \cdot \mathbb{E}^{\sigma^*_k}_{\lambda_0}[(\mathsf{Total}_k + \overline{\mathsf{Total}}_k) \mid \overline{\mathcal{E}}_k] \\
&= \mathbb{P}^{\sigma^*_k}_{\lambda_0}(\mathcal{E}_k) \cdot \mathbb{E}^{\sigma^*_k}_{\lambda_0}[\mathsf{Total} \mid \mathcal{E}_k] + \mathbb{P}^{\sigma^*_k}_{\lambda_0}(\overline{\mathcal{E}}_k) \cdot \mathbb{E}^{\sigma^*_k}_{\lambda_0}[\mathsf{Total}_k \mid \overline{\mathcal{E}}_k] + \mathbb{P}^{\sigma^*_k}_{\lambda_0}(\overline{\mathcal{E}}_k) \cdot \mathbb{E}^{\sigma^*_k}_{\lambda_0}[\overline{\mathsf{Total}}_k \mid \overline{\mathcal{E}}_k] \\
&= \mathbb{P}^{\sigma^*_k}_{\lambda_0}(\mathcal{E}_k) \cdot \mathbb{E}^{\sigma^*_k}_{\lambda_0}[\mathsf{Total}_k \mid \mathcal{E}_k] + \mathbb{P}^{\sigma^*_k}_{\lambda_0}(\overline{\mathcal{E}}_k) \cdot \mathbb{E}^{\sigma^*_k}_{\lambda_0}[\mathsf{Total}_k \mid \overline{\mathcal{E}}_k] + \mathbb{P}^{\sigma^*_k}_{\lambda_0}(\overline{\mathcal{E}}_k) \cdot \mathbb{E}^{\sigma^*_k}_{\lambda_0}[\overline{\mathsf{Total}}_k \mid \overline{\mathcal{E}}_k] \\
&= \mathbb{E}^{\sigma^*_k}_{\lambda_0}[\mathsf{Total}_k] + \mathbb{P}^{\sigma^*_k}_{\lambda_0}(\overline{\mathcal{E}}_k) \cdot \mathbb{E}^{\sigma^*_k}_{\lambda_0}[\overline{\mathsf{Total}}_k \mid \overline{\mathcal{E}}_k] \le \mathbb{E}^{\sigma^*_k}_{\lambda_0}[\mathsf{Total}_k] + \alpha_k \cdot \mathcal{U}_{\mathsf{Allow}}.
\end{aligned}$$

The first equality is obtained by splitting with respect to the complementary events $\mathcal{E}_k$ and $\overline{\mathcal{E}}_k$; the second equality is obtained by writing $\mathsf{Total} = \mathsf{Total}_k + \overline{\mathsf{Total}}_k$; and the third equality is by linearity of expectation. The fourth equality is obtained as follows: since all outgoing transitions from target states have cost zero, it follows that given the event $\mathcal{E}_k$ we have $\mathsf{Total} = \mathsf{Total}_k$. The fifth equality is obtained by combining the first two terms. The final inequality is obtained as follows: from the $(k+1)$-th step the strategy plays as σ_{Allow} and the expected total cost given σ_{Allow} is bounded by $\mathcal{U}_{\mathsf{Allow}}$. The result follows. □

Lemma 12. *For $k \in \mathbb{N}$ consider the strategy σ^*_k and α_k (as defined in Lemma 11). The following assertions hold:*

$$(1)\ \mathbb{E}^{\sigma^*_k}_{\lambda_0}[\mathsf{Total}_k] \le \mathsf{optCost}; \quad \textit{and} \quad (2)\ \alpha_k \le \frac{\mathsf{optCost}}{k}.$$

Proof. We prove both the inequalities below.

1. For $k \in \mathbb{N}$ we have that

$$\mathbb{E}^{\sigma^*_k}_{\lambda_0}[\mathsf{Total}_k] \le \inf_{\sigma \in \mathsf{Almost}_G(T)} \mathbb{E}^{\sigma}_{\lambda_0}[\mathsf{Total}_k] \le \inf_{\sigma \in \mathsf{Almost}_G(T)} \mathbb{E}^{\sigma}_{\lambda_0}[\mathsf{Total}] = \mathsf{optCost}$$

   The first inequality is due to Lemma 9 and the second inequality follows from the fact that $\mathsf{Total}_k \le \mathsf{Total}$ for non-negative weights.
2. Note that α_k denotes the probability that the target state is not reached within the first k steps. Since the costs are positive integers (for all transitions other than the target state transitions), given the event $\overline{\mathcal{E}}_k$ the total cost for k steps is at least k. Hence it follows that $\mathbb{E}^{\sigma^*_k}_{\lambda_0}[\mathsf{Total}_k] \ge k \cdot \alpha_k$. Thus it follows from the first inequality that we have $\alpha_k \le \frac{\mathsf{optCost}}{k}$.

The desired result follows. □

```
Algorithm 1 ApproxAlgo Input: POMDP, $\epsilon > 0$
 1: $k \leftarrow 1$
 2: $\sigma_{\mathsf{Allow}}, \mathcal{U}_{\mathsf{Allow}} \leftarrow$ Compute $\sigma_{\mathsf{Allow}}$ and $\mathcal{U}_{\mathsf{Allow}}$        ▷ See Remark 3
 3: $\sigma_k^{\mathsf{FO}} \leftarrow$ Finite-horizon value iteration for horizon $k$ restricted to allowed actions        ▷ See Section 4.3
 4: $T_k \leftarrow \mathbb{E}_{\lambda_0}^{\sigma_k^{\mathsf{FO}}}[\mathsf{Total}_k]$
 5: $\alpha_k \leftarrow \mathbb{P}_{\lambda_0}^{\sigma_k^{\mathsf{FO}}}(\overline{\mathcal{E}}_k)$        ▷ Note that $\mathbb{P}_{\lambda_0}^{\sigma_k^*}(\overline{\mathcal{E}}_k) = \mathbb{P}_{\lambda_0}^{\sigma_k^{\mathsf{FO}}}(\overline{\mathcal{E}}_k)$
 6: Add. approx.: if $\alpha_k \cdot \mathcal{U}_{\mathsf{Allow}} \leq \epsilon$ then goto line: 10
 7: Mult. approx.: if $\alpha_k \cdot \mathcal{U}_{\mathsf{Allow}} \leq T_k \cdot \epsilon$ then goto line: 10
 8: $k \leftarrow k + 1$
 9: goto line: 3
10: return Strategy $\sigma_k^*$ obtained by playing $\sigma_k^{\mathsf{FO}}$ for $k$-steps followed by $\sigma_{\mathsf{Allow}}$
```

Approximation algorithms. Our approximation algorithm is presented as Algorithm 1.

Correctness and bound on iterations. Observe that by Proposition 1 we have $T_k = \mathbb{E}_{\lambda_0}^{\sigma_k^{\mathsf{FO}}}[\mathsf{Total}_k] = \mathbb{E}_{\lambda_0}^{\sigma_k^*}[\mathsf{Total}_k]$. Thus by Lemma 11 and Lemma 12 (first inequality) we have $\mathbb{E}_{\lambda_0}^{\sigma_k^*}[\mathsf{Total}] \leq T_k + \alpha_k \cdot \mathcal{U}_{\mathsf{Allow}} \leq \mathsf{optCost} + \alpha_k \cdot \mathcal{U}_{\mathsf{Allow}}$ (since $T_k \leq \mathsf{optCost}$). Thus if $\alpha_k \cdot \mathcal{U}_{\mathsf{Allow}} \leq \epsilon$ we obtain an additive approximation. If $\alpha_k \cdot \mathcal{U}_{\mathsf{Allow}} \leq T_k \cdot \epsilon$, then we have $\mathbb{E}_{\lambda_0}^{\sigma_k^*}[\mathsf{Total}] \leq T_k + \alpha_k \cdot \mathcal{U}_{\mathsf{Allow}} \leq T_k \cdot (1 + \epsilon) \leq \mathsf{optCost} \cdot (1 + \epsilon)$; and we obtain an multiplicative approximation. This establishes the correctness Algorithm 1. Finally, we present the theoretical upper bound on k that ensures stopping for the algorithm. By Lemma 12 (second inequality) we have $\alpha_k \leq \frac{\mathsf{optCost}}{k} \leq \frac{\mathcal{U}_{\mathsf{Allow}}}{k}$. Thus $k \geq \frac{\mathcal{U}_{\mathsf{Allow}}^2}{\epsilon}$ ensures that $\alpha_k \cdot \mathcal{U}_{\mathsf{Allow}} \leq \epsilon$, and the algorithm stops both for additive and multiplicative approximation. We now summarize the main results of this section.

Theorem 2. *In POMDPs with positive costs, the additive and multiplicative approximation problems for the optimal cost* $\mathsf{optCost}$ *are decidable. Algorithm 1 computes the approximations using finite-horizon optimal strategy computations and requires at most double-exponentially many iterations; and there exists POMDPs where double-exponentially many iterations are required.*

Remark 4. Note that though the theoretical upper bound k on the number of iterations $\frac{\mathcal{U}_{\mathsf{Allow}}^2}{\epsilon}$ is double exponential in the worst case, in practical examples of interest the stopping criteria is expected to be satisfied in much fewer iterations.

Remark 5. We remark that if we consider POMDPs with positive costs, then considering almost-sure strategies is not a restriction. For every strategy that is not almost-sure winning, with positive probability the target set is not reached, and since all costs are positive, the expected cost is infinite. If every strategy is not almost-sure winning (i.e., there exists no almost-sure winning strategy to reach the target set from the starting state), then the expected cost is infinite, and if there exists an almost-sure winning strategy, we approximate the optimal cost. Thus our result is applicable to *all* POMDPs with positive costs. A closely related work to ours is Goal-POMDPs, and the solution of Goal-POMDPs applies to the class of POMDPs where the target state is reachable from every state (see [2, line-3, right column page 1] for the restriction of Goal-MDPs and the solution of Goal-POMDPs is reduced to Goal-MDPs). For example, in the following Section 5, the first three examples for experimental results do not satisfy the restriction of Goal-POMDPs.

5 Experimental Results

Implementation. We have implemented Algorithm 1. In principle our algorithm suggests the following: First, compute the almost-sure winning belief-supports, the set of allowed actions, and σ_{Allow}; and then compute finite-horizon value iteration restricted to allowed actions. An important feature of our algorithm is its flexibility that any finite-horizon value iteration algorithm can be used for our purpose. We have implemented our approach where we first implement the computation of almost-sure winning belief-supports, allowed actions, and σ_{Allow}; and for the finite-horizon value iteration (Step 3 of Algorithm 1) we implement two approaches. The first approach is the exact finite-horizon value iteration using a modified version of POMDP-Solve [3]; and the second approach is an approximate finite-horizon value iteration using a modified version of RTDP-Bel [2]; and in both the cases our straightforward modification is that the computation of the finite-horizon value iteration is restricted to allowed actions and almost-sure winning belief-supports.

Examples for experimental results. We experimented on several well-known examples of POMDPs. The POMDP examples we considered are as follows: (A) We experimented with the *Cheese maze* POMDP example which was introduced in [19] and also studied in [9, 18, 20]. Along with the standard example, we also considered a larger maze version; and considered two cost functions: one that assign cost 1 to all transitions and the other where the cost of movement on the baseline is assigned cost 2. (B) We considered the *Grid* POMDP introduced in [28] and also studied in [18, 23, 20]. We considered two cost functions: one where all costs are 1 and the other where transitions in narrow areas are assigned cost 2. (C) We experimented with the robot navigation problem POMDP introduced in [18], where we considered both deterministic transition and a randomized version. We also considered two cost functions: one where all costs are assigned 1 and the other where costs of turning is assigned cost 2. (D) We consider the Hallway example from [18, 31, 29, 2]. (E) We consider the RockSample example from [2, 29].

Discussion on Experimental results. Our experimental results are shown in Table 1, where we compare our approach to RTDP-Bel [2]. Other approaches such as SARSOP [16], anytime POMDP [25], ZMDP [29] are for discounted setting, and hence are different from our approach. The RTDP-Bel approach works for POMDPs where from every state the goal states are reachable, and our first three examples do not fall into this category. For the first three examples, both of our exact and approximate implementation work very efficiently. For the other two larger examples, the exact method does not work since POMDP-Solve cannot handle large POMDPs, whereas our approximate method gives comparable result to RTDP-Bel. For the exact computation, we consider multiplicative approximation with $\epsilon = 0.1$ and report the number of iterations and the time required by the exact computation. For the approximate computation, we report the time required by the number of trials specified for the computation of the finite-horizon value iteration. For the first three examples, the obtained value of the strategies of our approximate version closely matches the value of the strategy of the exact computation, and for the last two example, the values of the strategies obtained by our approximate version closely matches the values of the strategies obtained by RTDP-Bel.

Details of the POMDP examples. We now present the details of the POMDP examples.

1. *Cheese maze:* The example models a simple maze, where there are four actions `n`, `e`, `s`, `w` that correspond to the movement in the four compass directions. The POMDP examples are shown in Figure 3 and Figure 4. Actions that attempt to move outside of the maze have no effect on the position; otherwise the movement is determined deterministically given the action in all four directions. In the small version, there are 12 states and 8 observations, which correspond to what walls would be seen in all four directions that are immediately adjacent to the current location, i.e., states $5, 6$, and 7 have the same observation. The game starts in a unique initial state that is not depicted in the figure, where all actions lead to the baseline states 0, 1, 2, 3, or 4 with uniform probability. The target state is depicted with a star, and there are also two absorbing trap states depicted with a skull. The initial state, the trap states, and the goal state have their own unique observations. In the larger variant of the POMDP there are four more states and intuitively they add a new leftmost branch to the POMDP with a third absorbing trap state at the end. The new baseline is formed out of states $0, 1, 2, \dots, 6$. In the first setting all the costs are 1, and this represents the number of steps to the target state; and in the second setting the cost of any movement on the baseline is 2 and the movement in the branches costs 1 (which models that baseline exploration is more costly).
2. *Grid* 4×3*:* The Grid POMDP is shown in Figure 5 and models a maze with 11 states: the starting state 0, one target state depicted with a star, and an absorbing trap state that is depicted with a skull. There are four actions `n`, `e`, `s`, `w` that correspond to the movement in the four compass directions. The movement succeeds only with probability 0.96 and with probability 0.02 moves perpendicular to the intended direction. Attempts to move outside of the grid have no effect on the position, i.e., playing action `s` from state 0 will move with probability 0.96 to state 4; with probability 0.02 to state 1, and with

Example	Costs	$\lvert S\rvert$	$\lvert\mathcal{A}\rvert$	$\lvert\mathcal{Z}\rvert$	σ_{Allow} comp.	**Exact** $\epsilon = 0.1$		**Approx.**		**RTDP-Bel**	
						Iter.	Time	Time	Trials	Time	Trials
Cheese maze - small	$\{1\}$	12	4	8	$0.27 \cdot 10^{-3}$s	7	0.54s	0.06s	12k	×	
	$\{1,2\}$					8	0.62s	0.06s	12k	×	
Cheese maze - large	$\{1\}$	16	4	8	$0.57 \cdot 10^{-3}$s	9	12.18s	0.29s	12k	×	
	$\{1,2\}$					12	16.55s	0.3s	12k	×	
Grid	$\{1\}$	11	4	6	$0.47 \cdot 10^{-3}$s	6	0.33s	0.2s	12k	×	
	$\{1,2\}$					10	4.21s	0.21s	12k	×	
Robot movement - det.	$\{1\}$	15	3	11	$0.43 \cdot 10^{-3}$s	9	5.67s	0.08s	12k	×	
	$\{1,2\}$					8	5.01s	0.08s	12k	×	
Robot movement - ran.	$\{1\}$	15	3	11	$0.52 \cdot 10^{-3}$s	10	6.64s	0.08s	12k	×	
	$\{1,2\}$					10	6.65s	0.04s	12k	×	
Hallway	$\{1\}$	61	5	22	$0.32 \cdot 10^{-1}$s	Timeout 20m.		283.88s	12k	282.47s	12k
Hallway 2	$\{1\}$	94	5	17	$0.58 \cdot 10^{-1}$s	Timeout 20m.		414.29s	14k	413.21s	14k
RockSample[4,4]	$\{1, 50, 100\}$	257	9	2	0.05s	Timeout 20m.		61.23s	20k	61.29s	20k
RockSample[5,5]	$\{1, 50, 100\}$	801	10	2	0.26s	Timeout 20m.		99.13s	20k	98.44s	20k
RockSample[5,7]	$\{1, 50, 100\}$	3201	12	2	4.44s	Timeout 20m.		427.94s	20k	422.61s	20k
RockSample[7,8]	$\{1, 50, 100\}$	12545	13	2	78.83s	Timeout 20m.		1106.2s	20k	1104.53s	20k

Table 1. Experimental results

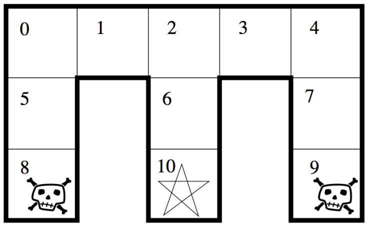

Fig. 3. The Cheese Maze - small POMDP

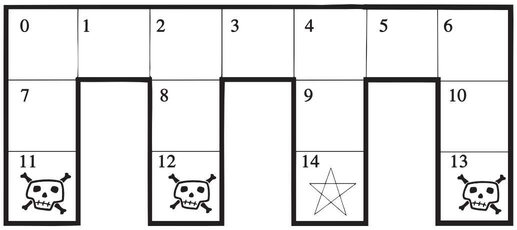

Fig. 4. The Cheese Maze - large POMDP

probability 0.02 to state 0. There are 6 observations that correspond to the information from detectors that can detect whether there are walls immediately adjacent to the east and to the west of the current state. The goal and the absorbing trap state have their own observations. We have again considered the setting where all the costs before reaching the target state are 1. In the second setting we have assigned to movements in the narrow areas of the maze (states $0, 1, 4, 7$, and 8) cost of all actions to 2. Intuitively, the higher costs compensate for the wall bumps that can make the movement in the narrow areas of the maze more predictable.

3. *Robot navigation:* The robot navigation POMDP models the movement of a robot in an environment. The robot can be in four possible states: facing north, east, south, and west. The environment has states 1, 2, 3, and a final state depicted with a star. The robot has three available actions: move forward `f`, turn left `l`, and turn right `r`. The original setting of the problem is that all actions are deterministic – *Robot movement - det.* We also consider a variant *Robot movement - ran.*, where the attempt to make an action may fail and with probability 0.04 has no effect, i.e., the action does not change the state. The POMDP starts in a unique initial state that is not depicted in the figure and under all actions reaches the state 1 with the robot facing north, east, south or west with uniform probability. Any bump to the wall results in a damaged immobile robot, modeled by an absorbing state not depicted in the figure. There

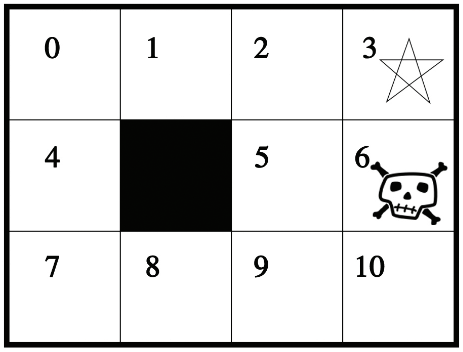

Fig. 5. The Grid 4 × 3 POMDP

are 11 observations that correspond to what would be seen in all four directions that are adjacent to the current location. The initial state, the damaged state, and the target state have their own observations. For both variants we have considered two different cost settings. In the first setting all the costs before reaching the target state are 1. In the second setting we assign cost 1 to the move forward action, and cost 2 to the turn-left and turn-right action (i.e., turning is more costly than moving forward).

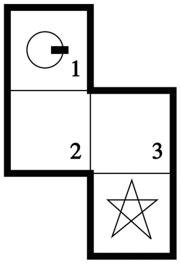

Fig. 6. The Robot POMDP

4. *Hallway.* We consider two versions of the Hallway example introduced in in [18] and used later in [31, 29, 2]. The basic idea behind both of the Hallway problems, is that there is an agent wandering around some office building. It is assumed that the locations have been discretized so there are a finite number of locations where the agent could be. The agent has a small finite set of actions it can take, but these only succeed with some probability. Additionally, the agent is equipped with very short range sensors to provide it only with information about whether it is adjacent to a wall. These sensors also have the property that they are somewhat unreliable and will sometimes miss a wall or see a wall when there is none. It can "see" in four directions: forward, left, right, and backward. It is important to note that these observations are relative to the current orientation of the agent (N, E, S, W). In these problems the location in the building and the agent's current orientation comprise the states. There is a single goal location, denoted by the star. The actions that can be chosen consists of movements: forward, turn-left, turn-right, turn-around, and no-op (stay in place). Action forward succeeds with probability 0.8, leaves the state unchanged with probability 0.05, moves the agent to the left and rotates the agent to the left with probability 0.05, similarly with probability 0.05 the agent moves to the right and is rotated to the right, with probability 0.025 the agent is moved back without changing its orientation, and with

probability 0.025 the agent is moved back and is rotated backwards. The action move-left and move-right succeeds with probability 0.7, and with probability 0.1 each of the three remaining orientation is reached. Action turn-around succeeds with probability 0.6, leaves the state unchanged with probability 0.1, turns the agent to left or right, each with probability 0.15. The last action no-op leaves the state unchanged with probability 1. In states where moving forward is impossible the probability mass for the impossible next state is collapsed into the probability of not changing the state. Every move of the agent has a cost of 1 and the agent starts with uniform probability in all non-goal states. In the smaller Hallway problem there are 61 states and 22 observations. In the Hallway2 POMDP there are 94 states and 17 observations.

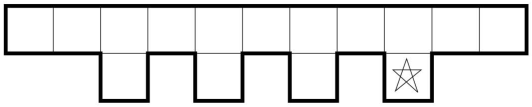

Fig. 7. Hallway

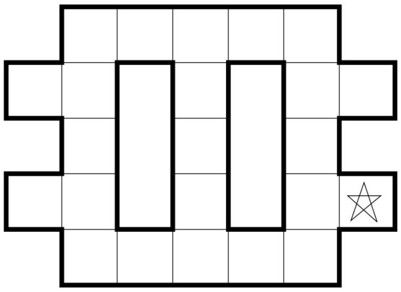

Fig. 8. Hallway 2

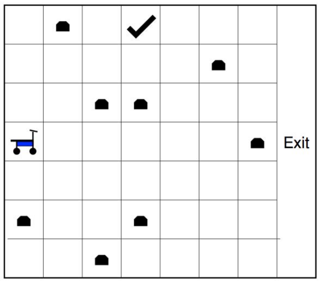

Fig. 9. RockSample[7,8]

5. *RockSample.* The RockSample problem introduced in [29] and used later in [2] is a scalable problem that models rover science exploration (Figure 9). The rover can achieve various costs by sampling rocks in the immediate area, and by continuing its traverse (reaching the exit at the right side of the map). The positions of the rover and the rocks are known, but only some of the rocks have scientific value; we will call these rocks good. Sampling a rock is expensive, so the rover is equipped with a noisy long-range sensor that it can use to help determine whether a rock is good before choosing whether to approach and sample it. An instance of RockSample with map size $n \times n$ and k rocks is described as RockSample[n,k]. The POMDP model of RockSample[n,k] is as follows. The state space is the cross product of $k+1$ features: $Position = \{(1,1),(1,2),...,(n,n)\}$, and k binary features $RockType_i = \{Good, Bad\}$ that indicate which of the rocks are good. There is an additional terminal state, reached when the rover moves off the right-hand edge of the map. The rover can select from $k+5$ actions: $\{N, S, E, W, sample, check_1, ..., check_k\}$.

The first four are deterministic single-step motion actions. The *sample* action samples the rock at the rover's current location. If the rock is good, the rover receives a small cost of 1 and the rock becomes bad (indicating that nothing more can be gained by sampling it). If the rock is bad, it receives a higher cost of 50. The cost of performing a measurement induces a cost of 1, attempt to move outside of the map has a cost of 100. All other moves have a cost of 50. Each $check_i$ action applies the rover's long-range sensor to rock i, returning a noisy observation from $Good, Bad$. The noise in the long-range sensor reading is determined by the efficiency ν, which decreases exponentially as a function of Euclidean distance from the target. At $\nu = 1$, the sensor always returns the correct value. At $\nu = 0$, it has a 50/50 chance of returning *Good* or *Bad*. At intermediate values, these behaviors are combined linearly. The initial belief is that every rock has equal probability of being Good or Bad. All the problems have 2 observations, and RockSample[4,4] has 257 states, RockSample[5,5] has 801 states, RockSample[5,7] has 3201 states, and RockSample[7,8] has 12545 states.

Acknowledgments. We thank Blai Bonet for helping us with RTDP-Bel.

References

1. P. Billingsley, editor. *Probability and Measure.* Wiley-Interscience, 1995.
2. B. Bonet and H. Geffner. Solving POMDPs: RTDP-Bel vs. point-based algorithms. In *IJCAI*, pages 1641–1646, 2009.
3. A. Cassandra. Pomdp-solve [software, version 5.3]. http://www.pomdp.org/, 2005.
4. A.R. Cassandra. *Exact and approximate algorithms for partially observable Markov decision processes.* Brown University, 1998.
5. K. Chatterjee and M. Henzinger. Faster and dynamic algorithms for maximal end-component decomposition and related graph problems in probabilistic verification. In *SODA*. ACM-SIAM, 2011.
6. C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *Journal of the ACM*, 42(4):857–907, 1995.
7. K. Culik and J. Kari. Digital images and formal languages. *Handbook of formal languages*, pages 599–616, 1997.
8. R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acids.* Cambridge Univ. Press, 1998.
9. A. Dutech. Solving POMDPs using selected past events. In *ECAI*, pages 281–285, 2000.
10. J. Filar and K. Vrieze. *Competitive Markov Decision Processes.* Springer-Verlag, 1997.
11. H. Howard. *Dynamic Programming and Markov Processes.* MIT Press, 1960.
12. L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1):99–134, 1998.
13. L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *J. of Artif. Intell. Research*, 4:237–285, 1996.
14. J.G. Kemeny, J.L. Snell, and A.W. Knapp. *Denumerable Markov Chains.* D. Van Nostrand Company, 1966.
15. H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas. Temporal-logic-based reactive mission and motion planning. *IEEE Transactions on Robotics*, 25(6):1370–1381, 2009.
16. 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*, pages 65–72, 2008.
17. M. L. Littman. *Algorithms for Sequential Decision Making.* PhD thesis, Brown University, 1996.
18. M. L. Littman, A. R. Cassandra, and L. P Kaelbling. Learning policies for partially observable environments: Scaling up. In *ICML*, pages 362–370, 1995.
19. R. A. McCallum. First results with utile distinction memory for reinforcement learning. 1992.
20. P. McCracken and M. H. Bowling. Online discovery and learning of predictive state representations. In *NIPS*, 2005.
21. M. Mohri. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23(2):269–311, 1997.
22. C. H. Papadimitriou and J. N. Tsitsiklis. The complexity of Markov decision processes. *Mathematics of Operations Research*, 12:441–450, 1987.
23. R. Parr and S. J. Russell. Approximating optimal policies for partially observable stochastic domains. In *IJCAI*, pages 1088–1095, 1995.

24. A. Paz. *Introduction to probabilistic automata (Computer science and applied mathematics)*. Academic Press, 1971.
25. J. Pineau, G. Gordon, S. Thrun, et al. Point-based value iteration: An anytime algorithm for POMDPs. In *IJCAI*, volume 3, pages 1025–1032, 2003.
26. M. L. Puterman. *Markov Decision Processes*. John Wiley and Sons, 1994.
27. M.O. Rabin. Probabilistic automata. *Information and Control*, 6:230–245, 1963.
28. S. J. Russell, P. Norvig, J. F. Canny, J. M. Malik, and D. D. Edwards. *Artificial intelligence: a modern approach*, volume 74. Prentice hall Englewood Cliffs, 1995.
29. T. Smith and R. Simmons. Heuristic search value iteration for POMDPs. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 520–527. AUAI Press, 2004.
30. E. J. Sondik. *The Optimal Control of Partially Observable Markov Processes*. Stanford University, 1971.
31. M.T.J. Spaan. A point-based POMDP algorithm for robot planning. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 3, pages 2399–2404. IEEE, 2004.