Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence

Optimal Cost Almost-Sure Reachability in POMDPs*

Krishnendu Chatterjee
IST Austria
Klosterneuburg, Austria
kchatterjee@ist.ac.at

Martin Chmelík
IST Austria
Klosterneuburg, Austria
mchmelik@ist.ac.at

Raghav Gupta
IIT Bombay
India
raghavgupta93@gmail.com

Ayush Kanodia
IIT Bombay
India
kanodiaayush@gmail.com

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 present efficient stopping criteria for the algorithm and show experimentally that it performs well in many examples.

1 Introduction

POMDPs. *Markov decision processes (MDPs)* are standard models for probabilistic systems that exhibit both probabilistic as well as nondeterministic behavior (Howard 1960). MDPs are widely used to model and solve control problems for stochastic systems (Filar and Vrieze 1997; Puterman 1994): 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* 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 (Papadimitriou and Tsitsiklis 1987). POMDPs provide the appropriate model to study a wide variety of applications such as in computational biology (Durbin et al. 1998), speech processing (Mohri 1997), image processing (Culik and Kari 1997), robot planning (Kress-Gazit, Fainekos, and Pappas 2009; Kaelbling, Littman, and Cassandra 1998), reinforcement learning (Kaelbling, Littman, and Moore 1996), to name a few. POMDPs also subsume many other powerful computational models such as probabilistic finite automata (PFA) (Rabin 1963; Paz 1971) (since PFA 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 (Filar and Vrieze 1997; Puterman 1994; Papadimitriou and Tsitsiklis 1987). 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

*The research was partly supported by Austrian Science Fund (FWF) Grant No P23499-N23, FWF NFN Grant No S11407-N23 (RiSE), ERC Start grant (279307: Graph Games), and Microsoft faculty fellows award.

reached, while ensuring that the target set is reached almost-surely (probability 1). Our results are as follows:

1. *(Integer costs).* We first show that if the 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.

Comparison with Goal-POMDPs. While there are several works for discounted POMDPs (Kurniawati, Hsu, and Lee 2008; Smith and Simmons 2004; Pineau et al. 2003), as mentioned above the problem we consider is different from discounted POMDPs. The most closely related works are Goal-MDPs and POMDPs (Bonet and Geffner 2009; Kolobov et al. 2011). 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 applies to a strict subclass of POMDPs (see Remark 5 in (Chatterjee et al. 2014)); and (b) we present asymptotically tight (double exponential) theoretical bounds on the expected optimal costs. Full proofs are available in (Chatterjee et al. 2014).

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 (Puterman 1994; Littman 1996).

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 $\mathrm{Supp}(f) = \{x \in X \mid f(x) > 0\}$ its support.

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. For simplicity of presentation we consider without loss of generality a deterministic function, see Remark 1 in (Chatterjee et al. 2014) for probabilistic observations; 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, cones, and belief-supports. 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. 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., $\mathcal{B}(\rho) = \{s_n = \mathsf{Last}(w) \mid w = (s_0, a_0, s_1, a_1, \ldots, s_n), w \text{ is a prefix of a play, and for all } 0 \leq i \leq n.\ \mathcal{O}(s_i) = z_i\}$.

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 $\sigma(w) = \sigma(w')$.

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 (Billingsley 1995). We de-

note 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 the following 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 optCost and compute strategies $\sigma \in \mathsf{Almost}_G(T)$ such that the value $\mathsf{Val}(\sigma)$ approximates the optimal cost 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 $\mathsf{Val}(\sigma) \leq \mathsf{optCost}\cdot(1+\epsilon)$.

3 Approximation for Integer Costs

In this section we will show that the problem of approximating the optimal cost optCost is undecidable. We will show that deciding whether the optimal cost 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 problem is undecidable (Paz 1971).

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. 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. We show that if the answer to the strict emptiness problem is yes, then the optimal cost is $-\infty$, and conversely, if the answer is no, then the optimal cost is 1.

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

4 Approximation for Positive Costs

In this section we consider 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. Theorem 1 established undecidability for integer costs, and we now show that for positive cost functions the approximation problem is decidable.

4.1 Lower Bound on optCost

We present a double-exponential lower bound on 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. Assigning cost 1 to every transition we get the 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 $q^i_1, q^i_2, \ldots q^i_{p(i)}$, 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 $S = Q_1 \cup Q_2 \cup \ldots Q_n \cup \{s_0, \mathsf{bad}, \mathsf{target}\}$ with initial state s_0.
- 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 $\mathcal{L}_i$ have observation z. The remaining states s_0, bad, and target are visible.

The cost function c is 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 1, where we omit the losing absorbing state bad and the transitions to bad for simplicity.

Lemma 1. *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 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 $2^{2^{q(n)}}$.*

4.2 Upper Bound on optCost

Almost-sure winning belief-supports. Let $\mathsf{Belief}(G)$ denote the set of all belief-supports in a POMDP G, i.e., $\mathsf{Belief}(G) = \{U \subseteq S \mid \exists z \in \mathcal{Z} : U \subseteq \mathcal{O}^{-1}(z)\}$. Since we will only consider belief-supports, for brevity we call them beliefs in the sequel of this section. Let $\mathsf{Belief}_{\mathsf{Win}}(G,T)$ denote the set of almost-sure winning beliefs, 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 uniform distribution λ_U over U.

Restricting to $\mathsf{Belief}_{\mathsf{Win}}(G,T)$. In the sequel w.l.o.g. we will restrict ourselves to beliefs in $\mathsf{Belief}_{\mathsf{Win}}(G,T)$: since from beliefs outside $\mathsf{Belief}_{\mathsf{Win}}(G,T)$ there exists no almost-sure winning strategy, all almost-sure winning strategies with starting beliefs in $\mathsf{Belief}_{\mathsf{Win}}(G,T)$ will ensure that beliefs not in $\mathsf{Belief}_{\mathsf{Win}}(G,T)$ are never reached.

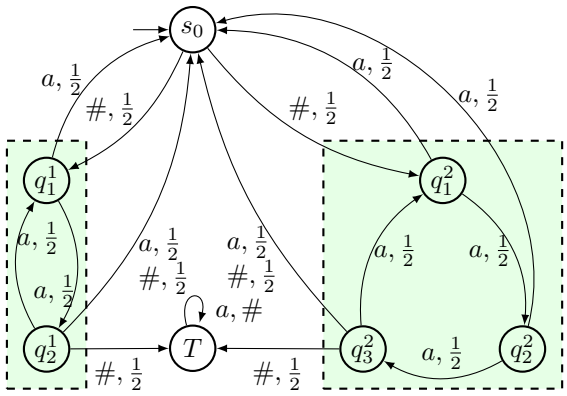

Figure 1: POMDP $\mathcal{F}(2)$

Belief updates. Given a belief $U \in \mathsf{Belief}(G)$, an action $a \in \mathcal{A}$, and an observation $z \in \mathcal{Z}$ we denote by $\mathsf{Update}(U,z,a)$ the updated belief. 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 beliefs 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 $U \in \mathsf{Belief}_{\mathsf{Win}}(G,T)$, we consider the set of actions that are guaranteed to keep the next belief U' in $\mathsf{Belief}_{\mathsf{Win}}(G,T)$ and refer these actions as *allowed or safe*. A framework that restricts playable actions was also considered in (Carvalho and Teichteil-Königsbuch 2013), however, the allowed actions need to be specified as a part of the input. In comparison we show how to compute with respect to the reachability objective the set of allowed actions for the optimal reachability. Formally we consider the set of allowed actions as follows: Given a belief $U \in \mathsf{Belief}_{\mathsf{Win}}(G,T)$ 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 show that almost-sure winning strategies must only play allowed actions. An easy consequence of the lemma is that for all beliefs U in $\mathsf{Belief}_{\mathsf{Win}}(G,T)$, there is always an allowed action. For an illustrative example see Example 1 in (Chatterjee et al. 2014).

Lemma 2. *Given a POMDP with a reachability objective $\mathsf{Reach}(T)$, consider a strategy σ and a starting belief in $\mathsf{Belief}_{\mathsf{Win}}(G,T)$. Given σ, if for a reachable belief $U \in \mathsf{Belief}_{\mathsf{Win}}(G,T)$ the strategy σ plays an action not in $\mathsf{Allow}(U)$ with positive probability, then σ is not almost-sure winning for the reachability objective.*

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

The strategy σ_{Allow}. We consider a *belief-based stationary* (for brevity belief-based) strategy σ_{Allow} as follows: for all beliefs U in $\mathsf{Belief}_{\mathsf{Win}}(G,T)$, the strategy plays uniformly at random all actions from $\mathsf{Allow}(U)$.

Lemma 3. *The belief-based strategy σ_{Allow} is an almost-sure winning strategy for all beliefs $U \in \mathsf{Belief}_{\mathsf{Win}}(G,T)$ for the objective $\mathsf{Reach}(T)$.*

Remark 1 (Computation of σ_{Allow}). *It follows from Lemma 3 that the strategy σ_{Allow} can be computed by computing the set of almost-sure winning states in the* belief MDP. *The belief MDP is a perfect-observation MDP where states are beliefs of the original POMDP, and given an action, the next state is obtained according to the belief updates. The strategy σ_{Allow} can be obtained by computing the set of almost-sure winning states in the belief MDP, and for discrete graph algorithms to compute almost-sure winning states in perfect-observation MDPs see (Courcoubetis and Yannakakis 1995; Chatterjee and Henzinger 2011).*

Upper bound. We now establish a double-exponential upper bound on $\mathsf{optCost}$, matching our lower bound from Lemma 1. 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 that the expected hitting time to the target set is at most double exponential.

Lemma 4. *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 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 (Sondik 1971). 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 denote by $\mathcal{H}$ the set of information states. Given an information state b, an action a, an observation z, computing the resulting information state b' is straightforward, see (Cassandra 1998).

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

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 beliefs in the set $\mathsf{Belief}_{\mathsf{Win}}(G,T)$. The equation for the value-iteration algorithm is defined as follows: $V_0^*(b) = 0$ and $V_n^*(b) = \min_{a \in \mathsf{Allow}(\mathrm{Supp}(b))} \left[\mathsf{c}'(b,a) + \sum_{b' \in \mathcal{H}} \psi(b,a)(b') V_{n-1}^*(b')\right]$. We obtain a strategy σ_k^{FO} that is finite-horizon optimal for length k (here FO stands for finite-horizon optimal) from the above equation. 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 5. *For all $k \in \mathbb{N}$ the strategy σ_k^* is almost-sure winning for the reachability objective $\mathsf{Reach}(T)$.*

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 2) we have the following result. Note that since in the first k steps σ_k^* plays as σ_k^{FO} we have Lemma 6 and Proposition 1.

Lemma 6. *For all $k \in \mathbb{N}$, $\mathbb{E}_{\lambda_0}^{\sigma_k^*}[\mathsf{Total}_k] = \inf_{\sigma \in \mathsf{Almost}_G(T)} \mathbb{E}_{\lambda_0}^{\sigma}[\mathsf{Total}_k]$.*

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

4.4 Approximation algorithm

We now 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 $U \in \mathsf{Belief}_{\mathsf{Win}}(G,T)$. Given a belief $U \in \mathsf{Belief}_{\mathsf{Win}}(G,T)$ 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 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 $\mathsf{Almost}_G(T)$, the value $\mathcal{U}_{\mathsf{Allow}}$ is also an upper bound for the optimal cost $\mathsf{optCost}$. By Lemma 4, $\mathcal{U}_{\mathsf{Allow}}$ is at most double exponential in the size of the POMDP.

Lemma 7. *We have $\mathsf{optCost} \leq \mathcal{U}_{\mathsf{Allow}}$.*

Let $\mathcal{E}_k$ denote the event of reaching the target set within k steps, i.e., $\mathcal{E}_k = \{(s_0, a_0, s_1, \ldots) \in \Omega \mid \exists i \leq k : s_i \in T\}$; and $\overline{\mathcal{E}}_k$ the complement of the event $\mathcal{E}_k$.

Lemma 8. *For $k \in \mathbb{N}$ consider the strategy σ_k^* that is obtained by playing an optimal finite-horizon strategy σ_k^{FO} for k steps, followed by strategy σ_{Allow}. Let $\alpha_k = \mathbb{P}_{\lambda_0}^{\sigma_k^*}(\overline{\mathcal{E}}_k)$ denote the probability that the target set is not reached within the first k steps, then $\mathbb{E}_{\lambda_0}^{\sigma_k^*}[\mathsf{Total}] \leq \mathbb{E}_{\lambda_0}^{\sigma_k^*}[\mathsf{Total}_k] + \alpha_k \cdot \mathcal{U}_{\mathsf{Allow}}$.*

Lemma 9. *For $k \in \mathbb{N}$ consider the strategy σ_k^* and α_k (as defined in Lemma 8). The following assertions hold:*

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

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

Correctness and bound on iterations. The correctness follows from Lemma 8 and Lemma 9; and it also follows that $k \geq \frac{\mathcal{U}_{\mathsf{Allow}}^2}{\epsilon}$ ensures that the algorithm stops both for additive and multiplicative approximation.

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	Val.	Time	Trials	Val.	Time	Trials	Val.
Cheese maze - small	$\{1\}$	12, 4, 8	$0.27 \cdot 10^{-3}$s	7	0.54s	4.6	0.06s	12k	4.6		×	
	$\{1,2\}$			8	0.62s	7.2	0.06s	12k	7.2		×	
Cheese maze - large	$\{1\}$	16, 4, 8	$0.57 \cdot 10^{-3}$s	9	12.18s	6.4	0.29s	12k	6.4		×	
	$\{1,2\}$			12	16.55s	10.8	0.3s	12k	10.8		×	
Grid	$\{1\}$	11, 4, 6	$0.47 \cdot 10^{-3}$s	6	0.33s	3.18	0.2s	12k	3.68		×	
	$\{1,2\}$			10	4.21s	5.37	0.21s	12k	5.99		×	
Robot movement - det.	$\{1\}$	15, 3, 11	$0.43 \cdot 10^{-3}$s	9	5.67s	7.0	0.08s	12k	7.0		×	
	$\{1,2\}$			8	5.01s	10.0	0.08s	12k	10.0		×	
Robot movement - ran.	$\{1\}$	15, 3, 11	$0.52 \cdot 10^{-3}$s	10	6.64s	7.25	0.08s	12k	7.25		×	
	$\{1,2\}$			10	6.65s	10.35	0.04s	12k	10.38		×	
Hallway	$\{1\}$	61, 5, 22	$0.32 \cdot 10^{-1}$s	Timeout 20m.			283.88s	12k	6.09	282.47s	12k	6.26
Hallway 2	$\{1\}$	94, 5, 17	$0.58 \cdot 10^{-1}$s	Timeout 20m.			414.29s	14k	4.69	413.21s	14k	4.46
RockSample[4,4]	$\{1, 50, 100\}$	257, 9, 2	0.05s	Timeout 20m.			61.23s	20k	542.49	61.29s	20k	546.73
RockSample[5,5]	$\{1, 50, 100\}$	801, 10, 2	0.26s	Timeout 20m.			99.13s	20k	159.39	98.44s	20k	161.07
RockSample[5,7]	$\{1, 50, 100\}$	3201, 12, 2	4.44s	Timeout 20m.			427.94s	20k	6.02	422.61s	20k	6.14
RockSample[7,8]	$\{1, 50, 100\}$	12545, 13, 2	78.83s	Timeout 20m.			1106.2s	20k	6.31	1104.53s	20k	6.39

Table 1: Experimental results

Algorithm 1 APPROXALGO **Input:** POMDP, $\epsilon > 0$

1: $k \leftarrow 1$
2: $\sigma_{\mathsf{Allow}}, \mathcal{U}_{\mathsf{Allow}} \leftarrow$ Compute $\sigma_{\mathsf{Allow}}, \mathcal{U}_{\mathsf{Allow}}$ ▷ Rem. 1
3: $\sigma_k^{\mathsf{FO}} \leftarrow$ Fin.-hor. val. it. on allowed act. for k steps
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: $\mathbb{P}_{\lambda_0}^{\sigma_k^*}(\overline{\mathcal{E}}_k) = \mathbb{P}_{\lambda_0}^{\sigma_k^{\mathsf{FO}}}(\overline{\mathcal{E}}_k)$
6: Add. app.: **if** $\alpha_k \cdot \mathcal{U}_{\mathsf{Allow}} \leq \epsilon$ **then goto line:** 10
7: Mult. app.: **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** σ_k^*, i.e., strategy σ_k^{FO} followed by σ_{Allow}

Theorem 2. *In POMDPs with positive costs, the additive and multiplicative approximation problems for the optimal cost* 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 2. *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 could be satisfied in fewer iterations.*

5 Experimental Results

We have implemented Algorithm 1: our algorithm first implements σ_{Allow} computation; and for the finite-horizon value iteration (Step 3 of Algorithm 1) we implement two approaches. The first is the exact finite-horizon value iteration using a modified version of POMDP-Solve (Cassandra 2005); and the second is an approximate finite-horizon value iteration using a modified version of RTDP-Bel (Bonet and Geffner 2009); and in both cases our straightforward modification is that the computation of the finite-horizon value iteration is restricted to allowed actions and almost-sure winning beliefs. 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 studied in (McCallum 1992; Dutech 2000; Littman, Cassandra, and Kaelbling 1995; McCracken and Bowling 2005). Along with the standard example, we considered a larger maze version; and two cost functions: one that assigns cost 1 to all transitions and the other where the movement on the baseline is assigned cost 2. (B) We considered the *Grid* POMDP studied in (Russell et al. 1995; Littman, Cassandra, and Kaelbling 1995; Parr and Russell 1995; McCracken and Bowling 2005). 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 (Littman, Cassandra, and Kaelbling 1995), where we considered both deterministic transitions and a randomized version. We also considered two cost functions: one where all costs are 1 and the other where turning is assigned cost 2. (D) We consider the Hallway example from (Littman, Cassandra, and Kaelbling 1995; Spaan 2004; Smith and Simmons 2004; Bonet and Geffner 2009). (E) We consider the RockSample example from (Bonet and Geffner 2009; Smith and Simmons 2004).

Discussion on Experimental results. Our experimental results are shown in Table 1, where we compare our approach to RTDP-Bel (Bonet and Geffner 2009). Other approaches such as SARSOP (Kurniawati, Hsu, and Lee 2008), anytime POMDP (Pineau et al. 2003), ZMDP (Smith and Simmons 2004) are for discounted setting, and hence are different from our approach. The RTDP-Bel approach works only for Goal-POMDPs where from every state the goal states are reachable, and our first five examples do not fall into this category. For the five examples, both of our exact and approximate implementation work very efficiently. For the other 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 itera-

tions, the time required by the exact computation, and the computed value. For the approximate computation, we report the time required by the number of trials specified for the computation of the finite-horizon value iteration and the computed value. Further details in (Chatterjee et al. 2014).

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

References

Billingsley, P., ed. 1995. *Probability and Measure*. Wiley-Interscience.

Bonet, B., and Geffner, H. 2009. Solving POMDPs: RTDP-Bel vs. point-based algorithms. In *IJCAI*, 1641–1646.

Carvalho, C., and Teichteil-Königsbuch, F. 2013. Properly Acting under Partial Observability with Action Feasibility Constraints. volume 8188 of *LNCS*. Springer. 145–161.

Cassandra, A. 1998. *Exact and approximate algorithms for partially observable Markov decision processes*. Brown University.

Cassandra, A. 2005. Pomdp-solve [software, version 5.3]. http://www.pomdp.org/.

Chatterjee, K., and Henzinger, M. 2011. Faster and dynamic algorithms for maximal end-component decomposition and related graph problems in probabilistic verification. In *SODA*. ACM-SIAM.

Chatterjee, K.; Chmelik, M.; Gupta, R.; and Kanodia, A. 2014. Optimal Cost Almost-sure Reachability in POMDPs. *CoRR* abs/1411.3880.

Courcoubetis, C., and Yannakakis, M. 1995. The complexity of probabilistic verification. *JACM* 42(4):857–907.

Culik, K., and Kari, J. 1997. Digital images and formal languages. *Handbook of formal languages* 599–616.

Durbin, R.; Eddy, S.; Krogh, A.; and Mitchison, G. 1998. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge Univ. Press.

Dutech, A. 2000. Solving POMDPs using selected past events. In *ECAI*, 281–285.

Filar, J., and Vrieze, K. 1997. *Competitive Markov Decision Processes*. Springer-Verlag.

Howard, H. 1960. *Dynamic Programming and Markov Processes*. MIT Press.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artificial intelligence* 101(1):99–134.

Kaelbling, L. P.; Littman, M. L.; and Moore, A. W. 1996. Reinforcement learning: A survey. *JAIR* 4:237–285.

Kolobov, A.; Mausam; Weld, D.; and Geffner, H. 2011. Heuristic search for generalized stochastic shortest path MDPs. In *ICAPS*.

Kress-Gazit, H.; Fainekos, G. E.; and Pappas, G. J. 2009. Temporal-logic-based reactive mission and motion planning. *IEEE Transactions on Robotics* 25(6):1370–1381.

Kurniawati, H.; Hsu, D.; and Lee, W. 2008. SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces. In *Robotics: Science and Systems*, 65–72.

Littman, M. L.; Cassandra, A. R.; and Kaelbling, L. P. 1995. Learning policies for partially observable environments: Scaling up. In *ICML*, 362–370.

Littman, M. L. 1996. *Algorithms for Sequential Decision Making*. Ph.D. Dissertation, Brown University.

McCallum, R. A. 1992. First results with utile distinction memory for reinforcement learning.

McCracken, P., and Bowling, M. H. 2005. Online discovery and learning of predictive state representations. In *NIPS*.

Mohri, M. 1997. Finite-state transducers in language and speech processing. *Comp. Linguistics* 23(2):269–311.

Papadimitriou, C. H., and Tsitsiklis, J. N. 1987. The complexity of Markov decision processes. *Mathematics of Operations Research* 12:441–450.

Parr, R., and Russell, S. J. 1995. Approximating optimal policies for partially observable stochastic domains. In *IJCAI*, 1088–1095.

Paz, A. 1971. *Introduction to probabilistic automata (Computer science and applied mathematics)*. Academic Press.

Pineau, J.; Gordon, G.; Thrun, S.; et al. 2003. Point-based value iteration: An anytime algorithm for POMDPs. In *IJCAI*, volume 3, 1025–1032.

Puterman, M. L. 1994. *Markov Decision Processes*. John Wiley and Sons.

Rabin, M. 1963. Probabilistic automata. *Information and Control* 6:230–245.

Russell, S. J.; Norvig, P.; Canny, J. F.; Malik, J. M.; and Edwards, D. D. 1995. *Artificial intelligence: a modern approach*, volume 74. Prentice hall Englewood Cliffs.

Smith, T., and Simmons, R. 2004. Heuristic search value iteration for POMDPs. In *UAI*, 520–527. AUAI Press.

Sondik, E. J. 1971. *The Optimal Control of Partially Observable Markov Processes.* Stanford University.

Spaan, M. 2004. A point-based POMDP algorithm for robot planning. In *ICRA*, volume 3, 2399–2404. IEEE.