Goal-HSVI: Heuristic Search Value Iteration for Goal-POMDPs *

Karel Horák[1], **Branislav Bošanský**[1], **Krishnendu Chatterjee**[2]
[1] Department of Computer Science, FEE, Czech Technical University in Prague
[2] IST Austria, Klosterneuburg, Austria
horak@agents.fel.cvut.cz, bosansky@agents.fel.cvut.cz, krishnendu.chatterjee@ist.ac.at

Abstract

Partially observable Markov decision processes (POMDPs) are the standard models for planning under uncertainty with both finite and infinite horizon. Besides the well-known discounted-sum objective, indefinite-horizon objective (aka Goal-POMDPs) is another classical objective for POMDPs. In this case, given a set of target states and a positive cost for each transition, the optimization objective is to minimize the expected total cost until a target state is reached. In the literature, RTDP-Bel or heuristic search value iteration (HSVI) have been used for solving Goal-POMDPs. Neither of these algorithms has theoretical convergence guarantees, and HSVI may even fail to terminate its trials. We give the following contributions: (1) We discuss the challenges introduced in Goal-POMDPs and illustrate how they prevent the original HSVI from converging. (2) We present a novel algorithm inspired by HSVI, termed *Goal-HSVI*, and show that our algorithm has convergence guarantees. (3) We show that Goal-HSVI outperforms RTDP-Bel on a set of well-known examples.

1 Introduction

POMDPs. The standard model for analysis of probabilistic systems with both nondeterministic as well as probabilistic behaviors are *Markov decision processes (MDPs)* [Howard, 1960], which are widely used to solve control and planning problems [Filar and Vrieze, 1997; Puterman, 1994]. The nondeterministic choices represent the freedom of the controller to execute control actions, and the probabilistic behavior describes the stochastic system response to these actions. In the presence of uncertainty about the environment, MDPs are extended to *partially observable MDPs (POMDPs)* where the controller does not have a perfect view of the system [Papadimitriou and Tsitsiklis, 1987; Littman, 1996]. (PO)MDPs provide a model to study a wide variety of applications, ranging from computational biology [Durbin *et al.*, 1998], to speech processing [Mohri, 1997], image processing [Culik and Kari, 1997], software verification [Černý *et al.*, 2011], reinforcement learning [Kaelbling *et al.*, 1996], and others.

Classical optimization objectives. In the standard optimization problems related to POMDPs, the transitions are associated with integer costs. Two classical objectives are the *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.

Goal-POMDPs. Besides the above, another classical and fundamental objective is the *indefinite-horizon* objective (which is similar to stochastic shortest path problem) that has been widely studied for MDPs and POMDPs [Bertsekas and Tsitsiklis, 1996; Patek, 2001; Bertsekas, 2005; Bonet and Geffner, 2009; Kolobov *et al.*, 2011; Chatterjee *et al.*, 2016], often under the name of Goal-POMDPs. In this case, there is a set of target states, all positive costs, and the goal is to minimize the expected total cost till the target set is reached. Note that the objective is not discounted-sum, but a total sum without discounts. The objective is also not finite-horizon, as there is no a priori bound to reach the target set (a target can be reached at different times along different paths). As an example application, consider robot planning, where a target state or goal must be reached, and each transition has an energy consumption requirement. The objective is to reach the goal while spending as little energy as possible. Since the energy consumption is a sum rather than a discounted sum, this problem (and similarly many other applications [Chatterjee *et al.*, 2016]) is naturally modeled as a Goal-POMDP.

Previous results. In contrast to POMDPs with discounted-sum objectives, to the best of our knowledge, there does

*This work has been supported by Vienna Science and Technology Fund (WWTF) Project ICT15-003, Austrian Science Fund (FWF) NFN Grant No S11407-N23 (RiSE/SHiNE), and ERC Starting grant (279307: Graph Games). This research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-13-2-0045 (ARL Cyber Security CRA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

not exist a practical method for solving Goal-POMDPs that provably converges to a near-optimal policy. The algorithms used previously in practice are RTDP-Bel (real-time dynamic programming) [Bonet, 1998; Bonet and Geffner, 2009] and also heuristic search value iteration (HSVI) algorithm [Smith and Simmons, 2004; 2005] that was used for solving Goal-POMDPs in [Warnquist *et al.*, 2013]. While RTDP-Bel does not guarantee convergence, it performs well in practice on examples such as RockSample [Bonet and Geffner, 2009]. Similarly, HSVI performs well on some Goal-POMDP examples [Warnquist *et al.*, 2013], but it need not work for Goal-POMDPs in general [Smith, 2007, Theorem 6.9].

Our contributions. We extend the discussion on the convergence of HSVI algorithm when applied to Goal-POMDPs, and we illuminate the key issues of HSVI on counter-examples. We address these issues and, based on our insights, we present a novel *Goal-HSVI* algorithm for solving Goal-POMDPs. Goal-HSVI is an advancement over previous approaches from the theoretical as well as practical perspective: (1) From the theoretical perspective, Goal-HSVI provides upper and lower bounds on the optimal value (and the quality of the currently considered policy) at all points of time and these bounds converge. Thus we provide the first algorithm with a theoretical guarantee of convergence for Goal-POMDPs, and our algorithm provides an *anytime* approximation. (2) From the practical perspective, we present an implementation of our algorithm and experimental results on several classical POMDP examples from the literature. While Goal-HSVI is comparable to RTDP-Bel on the RockSample domain, it dramatically outperforms RTDP-Bel on several other domains.

2 Goal-POMDPs and Previous Algorithm

We use the notation from [Bonet and Geffner, 2009] and define a Goal-POMDP as a tuple $\langle \mathcal{S}, G, \mathcal{A}, \mathcal{O}, P, Q, c, b_0 \rangle$, where $\mathcal{S}$ is a finite non-empty set of states, G is a non-empty set of target (or goal) states ($G \subseteq \mathcal{S}$), $\mathcal{A}$ is a finite non-empty set of actions, $\mathcal{O}$ is a finite non-empty set of observations ($o_g \in \mathcal{O}$ notifies the agent about reaching the goal), $P_a(s, s')$ is the probability to transition from s to s' by using action a, $Q_a(o|s')$ is the probability to observe o when entering state s' by using action a, $c(s, a) > 0$ is the cost for taking action a in a non-target state s and $b_0 \in \Delta(\mathcal{S})$ is the initial belief ($\Delta(\mathcal{S})$ denotes the set of all probability distributions over $\mathcal{S}$). Without loss of generality, we assume $G = \{g\}$.

We assume that the agent does not incur any cost after reaching the target state g, i.e. $c(g, a) = 0$ for every $a \in \mathcal{A}$. Moreover, state g is absorbing, i.e., $P_a(g, g) = 1$ for every $a \in \mathcal{A}$, and the agent is always certain about reaching g (i.e., $Q_a(o_g|g) = 1$ for every $a \in \mathcal{A}$ and $Q_a(o_g|s) = 0$ for every $s \neq g$). We also assume that the goal state is reachable from every non-target state, i.e., the agent can never enter a dead-end. This requirement can, however, be overcome by precomputing a set of allowed actions for each belief support [Chatterjee *et al.*, 2016]. The algorithms for Goal-POMDPs can, therefore, be easily extended to the problems with dead-ends by considering only actions that are allowed in the current belief (and thus avoiding dead-ends). Note that the assumption of positive costs is, however, essential for the approximability of the problem as allowing negative costs renders any approximation undecidable [Chatterjee *et al.*, 2016, Theorem 2].

$b \leftarrow b_0$; $s \sim b$
while $b(g) < 1$ **do**
  $Q(b, a) \leftarrow \sum_{s \in \mathcal{S}} b(s)c(s, a) + \sum_{o \in \mathcal{O}} O_a(o|b)\widehat{V}(\lfloor K \cdot b^o_a \rfloor)$
  $a^* \leftarrow \arg\min_{a \in \mathcal{A}} Q(b, a)$
  $\widehat{V}(\lfloor K \cdot b \rfloor) \leftarrow Q(b, a^*)$
  $s' \sim P_{a^*}(s'|s)$; $o \sim Q_a(o|s')$; $b \leftarrow b^o_{a^*}$; $s \leftarrow s'$

Algorithm 1: A single trial of the RTDP-Bel algorithm.

POMDPs are solved by transformation to MDPs using the notion of belief states $b \in \Delta(\mathcal{S})$ where $b(s)$ corresponds to the probability that the system is in state s. The probability of being in state s' after playing action a in belief state b is

$$P_a(s'|b) = \sum_{s \in \mathcal{S}} b(s) \cdot P_a(s, s') \,. \tag{1}$$

Since the probability of observing o depends only on the action a and the state that is reached after playing a, the probability of observing o after playing a in belief state b is

$$O_a(o|b) = \sum_{s' \in \mathcal{S}} P_a(s'|b) \cdot Q_a(o|s') \,. \tag{2}$$

The probability of being in state s' given that action a was played and observation o received in belief b is

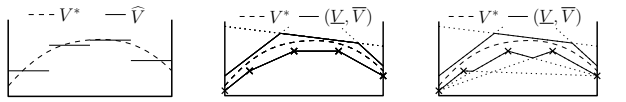
$$b^o_a(s') = [Q_a(o|s') \cdot P_a(s'|b)]/O_a(o|b) \quad \text{when } O_a(o|b) > 0 \,. \tag{3}$$

This transformation creates a perfect information MDP with a continuous state space of beliefs with transition function $P_a(b^o_a|b) = O_a(o|b)$. The Bellman equation for POMDPs is

$$V^*(b) = \min_{a \in \mathcal{A}} \left[\sum_s b(s) \cdot c(s, a) + \sum_o O_a(o|b) \cdot V^*(b^o_a) \right] \,. \tag{4}$$

The value function V^* is a concave function mapping beliefs to expected total costs, it is *unique*, and allows us to control a POMDP by always playing the minimizing action according to Equation (4). A common approach to solve POMDPs is thus to approximate V^*.

RTDP-Bel. The RTDP-Bel algorithm [Bonet, 1998; Bonet and Geffner, 2009] is based on RTDP [Barto *et al.*, 1995] for perfect-information Goal-MDPs. RTDP-Bel adapts RTDP to partially-observable domains by using a grid-based approximation of V^* and using a hash-table to store the values, where $V^*(b) \sim \widehat{V}(\lfloor K \cdot b \rfloor)$ for some fixed parameter $K \in \mathbb{N}$. This approximation, however, loses the theoretical properties of RTDP. The algorithm need not converge as the values of the discretized value function may oscillate. Moreover, there is no guarantee that the values stored in the hash-table will form a bound on the values of V^* [Bonet and Geffner, 2009, p. 3, last paragraph of Section 3]. Despite the lack of theoretical properties, RTDP-Bel has been shown to perform well in practice both on indefinite-horizon problems [Bonet, 1998] and discounted-sum problems after transforming them into Goal-POMDPs [Bonet and Geffner, 2009]. The RTDP-Bel algorithm performs a sequence of trials (see Algorithm 1).



(a) RTDP-Bel (b) HSVI (PWLC) (c) HSVI2 (sawtooth)

Figure 1: Comparison of value function approximation schemes

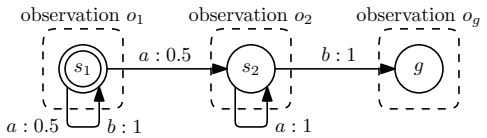


Figure 2: Blind policies have infinite values, $b_0(s_1) = 1$.

3 HSVI for Discounted-Sum POMDPs

Unlike RTDP-Bel, most of the point-based approaches approximate V^* using piecewise-linear functions [Smith and Simmons, 2004; 2005; Kurniawati *et al.*, 2008]. We illustrate the difference between a grid-based approximation used in RTDP-Bel and a piecewise-linear approximation in Figures 1a and 1b. Observe that unlike the grid-based approximation, piecewise-linear approximation can yield a close approximation of V^* even in regions with rapid change of value.

In the original version of the *heuristic-search value iteration* algorithm (HSVI) [Smith and Simmons, 2004], the algorithm keeps two piecewise-linear and concave (PWLC) functions $\underline{V}$ and $\overline{V}$ to approximate V^* (see Figure 1b) and refines them over time. The upper bound on the cost is represented in the vector-set representation using a set of vectors Γ, while the lower bound is formed as an upper convex hull of a set of points $\Upsilon = \{(b_i, y_i) \mid i = 1, \ldots, m\}$ where $b_i \in \Delta(\mathcal{S})$.

$$\overline{V}(b) = \min_{\alpha \in \Gamma} \langle \alpha, b \rangle \tag{5}$$

$$\underline{V}(b) = \max\{\textstyle\sum_{i=1}^{m} \lambda_i y_i \mid \lambda \in \mathbb{R}_+^m : \sum_{i=1}^{m} \lambda_i b_i = b\} \,. \tag{6}$$

Computing $\underline{V}(b)$ according to Equation (6) requires solving a linear program. In the second version of the algorithm (HSVI2, [Smith and Simmons, 2005]), the PWLC representation of lower bound has been replaced by a sawtooth-shaped approximation [Hauskrecht, 2000] (see Figure 1c). While the sawtooth approximation is less tight with the same set of points, the computation of $\underline{V}(b)$ can be done in linear time in the size of Υ. In this work, we consider exactly this HSVI2 version of the algorithm (and we refer to it as *vanilla-HSVI*).

HSVI2 initializes the value function $\overline{V}$ by considering policies 'always play action a' and constructing one α-vector for each action $a \in \mathcal{A}$ corresponding to the expected cost for playing such policy. For the initialization of the lower bound, the fast-informed bound is used [Hauskrecht, 2000].

The refinement of $\underline{V}$ and $\overline{V}$ is done by adding new elements to the sets Γ and Υ. Each of these updates is meant to improve the approximation quality in a selected belief b as much as possible, hence termed *point-based update* (see Algorithm 2). Recall that in discounted-sum POMDPs, $\gamma \in [0, 1)$ is the discount factor and the objective is to minimize the expected discounted sum $\sum_{i=1}^{\infty} \gamma^{i-1} c_i$ of costs c_i incurred during play.

1 $\alpha_a^o \leftarrow \arg\min_{\alpha \in \Gamma} \langle \alpha, b_a^o \rangle$ for all $a \in \mathcal{A}, o \in \mathcal{O}$
2 $\alpha_a(s) \leftarrow c(s,a) + \gamma \sum_{o,s'} \Pr_a(o, s' | s) \alpha_a^o(s') \qquad \forall s, a$
3 $\Gamma \leftarrow \Gamma \cup \{\arg\min_{\alpha_a} \langle \alpha_a, b \rangle\}$
4 $\Upsilon \leftarrow \Upsilon \cup \{(b, \min_a \left[\sum_s b(s) c(s,a) + \gamma \sum_o O_a(o|b) \underline{V}(b_a^o)\right])\}$

Algorithm 2: Point-based `update`(b) procedure of $(\underline{V}, \overline{V})$.

Similarly to RTDP-Bel, HSVI2 selects beliefs where the update should be performed based on the simulated play (selecting actions according to $\underline{V}$). Unlike RTDP-Bel, however, observations are not selected randomly. Instead, it selects an observation with the highest *weighted excess gap*, i.e. the excess approximation error $\overline{V}(b_a^o) - \underline{V}(b_a^o) - \epsilon\gamma^{-(t+1)}$ in b_a^o weighted by the probability $O_a(o|b)$ is the greatest. This heuristical choice attempts to target beliefs where the update will have the most significant impact on $\overline{V}(b_0) - \underline{V}(b_0)$.

The HSVI2 algorithm for discounted-sum POMDPs ($0 < \gamma < 1$) is shown in Algorithm 3. This algorithm provably converges to an ϵ-approximation of $V^*(b_0)$ using values $\underline{V}(b_0)$ and $\overline{V}(b_0)$, see [Smith and Simmons, 2004].

4 Vanilla-HSVI and Goal-POMDPs

While HSVI2 (termed *vanilla-HSVI* for our purposes) was applied in Goal-POMDPs [Warnquist *et al.*, 2013], it loses its desirable theoretical guarantees as shown already by [Smith, 2007]. We discuss three key issues related to the algorithm in the Goal-POMDP setting and illustrate them on examples.

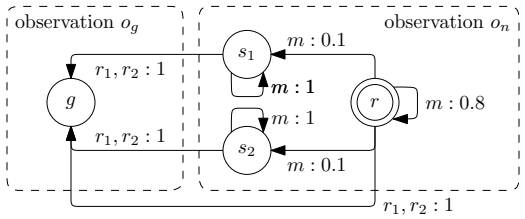
Initial values can be infinite. Vanilla-HSVI initializes value function $\overline{V}$ by considering the values of a blind policy (i.e., a policy prescribing the agent to use a fixed action a forever). Such policies, however, need not reach the goal with probability 1. Observe that in the example from Figure 2 the policies 'play a forever' and 'play b forever' *never* reach the goal and their cost is thus infinite. Moreover, since the play stays in s_1 with positive probability, $\overline{V}(s_1)$ remains infinite forever.

Solution: Instead of blind policies, we initialize $\overline{V}$ using the uniform policy. Since the goal state g is reachable from *every* state, the uniform policy reaches g with probability 1 and thus it has finite values [Chatterjee *et al.*, 2016, Lemma 5 and 6].

Exploration need not terminate. In discounted-sum problems, the sequence $\epsilon\gamma^{-t}$ is strictly increasing and unbounded (since $0 < \gamma < 1$). Its value therefore eventually exceeds the gap $\overline{V}(b) - \underline{V}(b)$ (which is guaranteed by the initialization to be bounded) and the recursive `explore` procedure of

1 Initialize $\underline{V}$ and $\overline{V}$
2 **while** $\overline{V}(b_0) - \underline{V}(b_0) > \epsilon$ **do** `explore`$(b_0, \epsilon, 0)$
3 **procedure** `explore`(b, ϵ, t)
4 **if** $\overline{V}(b) - \underline{V}(b) \leq \epsilon\gamma^{-t}$ **then return**
5 $a^* \leftarrow \arg\min_a \left[\sum_s b(s) c(s,a) + \gamma \sum_o O_a(o|b) \underline{V}(b_a^o)\right]$
6 `update`(b)
7 $o^* \leftarrow \arg\max_o O_{a^*}(o|b) \cdot [\overline{V}(b_{a^*}^o) - \underline{V}(b_{a^*}^o) - \epsilon\gamma^{-(t+1)}]$
8 `explore`$(b_{a^*}^{o^*}, \epsilon, t+1)$
9 `update`(b)

Algorithm 3: HSVI2 for discounted POMDPs. The pseudocode follows the ZMDP implementation and includes `update` on line 6.



$c(r, r_i) = 50$, $c(s_1, r_2) = c(s_2, r_1) = 100$, other costs in non-goals 1

Figure 3: Goal-POMDP where the `explore` procedure of vanilla-HSVI does not terminate, r_i must be played to reach g, $b_0(r) = 1$.

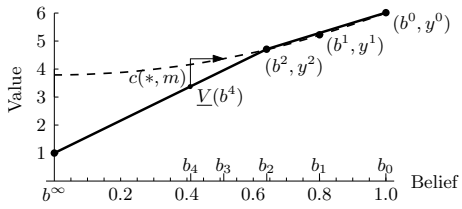


Figure 4: Value function $\underline{V}$ for the example from Figure 3 in the third level of **explore** recursion.

vanilla-HSVI terminates. This is clearly not the case in Goal-POMDPs where $\gamma = 1$. In [Smith, 2007, Theorem 6.9], it has been shown that the observation-selection heuristic of vanilla-HSVI (HSVI2) may cause the algorithm to enter an infinite loop. We show that even the action selection (line 5 of Algorithm 3) is susceptible to this behavior (and thus modifying the observation-selection heuristic does not fix the algorithm).

Consider the Goal-POMDP shown in Figure 3 where r is the initial state. The target state g can only be reached by playing action r_1 or r_2 at some point. We show, however, that the way the value function $\underline{V}$ is updated and beliefs are changed during the **explore** recursion prevents these actions to be ever considered. First, observe that the only reachable non-goal beliefs in this POMDP are b^T, where $T \in \mathbb{Z}_0^+$ and

$$b^T(r) = 0.8^T \qquad b^T(s_1) = b^T(s_2) = (1 - 0.8^T)/2\,, \quad (7)$$

and that the lower bound on the cost of playing r_i in b^T, $\underline{V}^{r_i}(b^T)$, is independent of T and constant

$$\begin{aligned}\underline{V}^{r_i}(b^T) &= \underline{V}(t) + b^T(r)c(r, r_i) + \quad (8)\\ &\quad + b^T(s_i)c(s_i, r_i) + b^T(s_{-i}, r_i)c(s_{-i}, r_i)\\ &= 0 + 0.8^T \cdot 50 + (1 - 0.8^T)/2 \cdot 1 + (1 - 0.8^T)/2 \cdot 99\,.\end{aligned}$$

This means that the **explore** procedure selects r_i in b^T only if $\underline{V}^m(b^T) = \underline{V}(b^{T+1}) + c(*, m) \geq \underline{V}^{r_i}(b^T) = 50$.

The recursion starts with a linear $\underline{V}$ where $\underline{V}(b^\infty) = 1$ and $\underline{V}(b^0) = 6$ (as initialized by the fast informed bound). In this situation, $\underline{V}(b^1)$ is a convex combination $0.8\underline{V}(b^0) + 0.2\underline{V}(b^\infty)$ and action m is optimal in b^0, its value in b^0 is

$$\underline{V}^m(b^0) = \underline{V}(b^1) + c(*, m) = 5 + 1 = 6 = y^0 \quad (9)$$

and the addition of the point (b^0, y^0) does not change $\underline{V}$.

At the T-th level of **explore** recursion, set Υ contains $(b^0, y^0), \ldots, (b^{T-1}, y^{T-1})$ and $(b^\infty, 1)$. A point-based update is performed in b^T (generating point (b^T, y^T)). It holds

$$\underline{V}^m(b^T) = c(*, m) + \underline{V}(b^{T+1}) = 1 + \underline{V}(b^{T+1})\,. \quad (10)$$

At this point, $\underline{V}(b^{T+1})$ is a convex combination of values $y^{T-1} = \underline{V}(b^{T-1})$ and $\underline{V}(b^\infty)$, $0.8^2 y^{T-1} + 0.36 \cdot 1$ (see Figure 4 for illustration). If m was always optimal, the sequence of y^T values can be characterized by a difference equation $y^T = 1 + 0.8^2 y^{T-1} + 0.36 \cdot 1$. For $y^0 = 6$, this sequence (generating points on the dashed line in Figure 4) is decreasing and it never exceeds value 50. Hence m is always optimal and actions r_1 and r_2 are never used during the trial. Note that $\overline{V}(b^T) \geq V^*(b^T) = 50$ for every $T \in \mathbb{Z}_0^+$ (we cannot avoid playing r_i eventually) and thus for sufficiently small ϵ the excess gap is always positive, and the trial never terminates.

Solution: In our Goal-HSVI algorithm, we cut off excessively long samples while guaranteeing that an ϵ-optimal solution is found within this limit. Such an approach has been used previously in practice without studying its impact on the solution quality (e.g., in POMDPSolver for discounted-sum POMDPs used in [Warnquist *et al.*, 2013] where the depth limit is set to 200). However, using any fixed depth limit for all indefinite-horizon POMDPs is not sound (consider a POMDP where more than 200 precisely timed steps have to be taken to reach the goal). We address this and consider a depth limit that provides theoretical guarantees on the solution quality.

Observation-selection heuristic may suppress exploration. Finally, terminating excessively long trials alone is not sufficient. Consider the example from [Smith, 2007, Theorem 6.9]. Value functions $\underline{V}$ and $\overline{V}$ are never changed during the trial. Hence, after termination, a new trial operates on the same $\underline{V}$ and $\overline{V}$. Thus it remains in states s_0 and s_1 – and the state s_2 that is necessary for convergence is never considered.

Solution: There are multiple possible solutions to this problem (e.g., changing the observation-selection heuristic). To remain consistent with vanilla-HSVI (and use the same heuristic), our Goal-HSVI algorithm keeps a data structure (a *closed-list*, denoted CL) tracking the action-observation sequences that have been already considered. When selecting an observation according to the observation-selection heuristic, the algorithm avoids choosing one that would lead to an action-observation sequence fully explored earlier.

5 Goal-HSVI

In this section, we present our novel Goal-HSVI approach to solve Goal-POMDPs. In Section 5.1, we present the basic algorithm and its theoretical guarantees, and in Section 5.2, we extend this algorithm to obtain a practical approach.

5.1 Basic Algorithm

The changes we presented in the previous section constitute the basis for our Goal-HSVI algorithm (see Algorithm 4). Our algorithm extends the vanilla-HSVI (Algorithm 3) with the three following key modifications:

(1) *The uniform policy is used for initialization of $\overline{V}$ (line 1).*

(2) *The search depth is bounded (line 5).* We terminate trials longer than $\frac{\overline{C}}{c_{\min}} \cdot \frac{\overline{C} - \eta\epsilon}{(1-\eta)\epsilon}$ steps where $\overline{C}$ is the upper bound on cost of playing the uniform policy and $c_{\min}$ is the minimum per-step cost. We prove that this choice together with a stricter termination condition $\overline{V}(b) - \underline{V}(b) \leq \eta\epsilon$

(for $\eta < 1$) guarantees that an ϵ-optimal solution is found by the Goal-HSVI algorithm (see Theorem 1).

(3) *Exploring the same history more than once is avoided.* We keep track of the action-observation history $(\overline{\mathbf{a}}, \overline{\mathbf{o}})$ during the exploration. A history is marked as *closed* by adding it to the closed list CL in case the history is terminal (line 6) or all histories reachable when using action a^* are already closed (lines 9-10). The set of observations O' denotes all observations that lead to an action-observation history that has not yet been closed. Together with the choice of $o^* \in O'$ (line 11), this guarantees that no action-observation history is considered twice.

Note that the closed list CL can be efficiently represented using a prefix tree. Each action-observation history is represented by a path in the tree (where each node corresponds to playing the given action or seeing the given observation). Furthermore, each node is attributed a binary flag indicating whether the given action-observation history is closed. The memory efficiency of this representation comes from the fact that multiple histories share the same prefix (and thus the same part of the path). This claim is supported by the experimental evaluation (see Table 1) as the number of nodes in the prefix tree is typically comparable with the number of elements representing value functions $\underline{V}$ and $\overline{V}$.

Theorem 1. *Assume $\eta \in [0, 1)$ and let $\overline{C}$ be the maximum cost of playing the uniform policy ($\overline{C} = \max_s \overline{V}(b^s)$ for the initial $\overline{V}$). Let $c_{\min} = \min_{s,a} c(s, a)$. Then the Goal-HSVI algorithm (Algorithm 4) with the cutoff after $\overline{T} = \frac{\overline{C}}{c_{\min}} \cdot \frac{\overline{C}-\eta\epsilon}{(1-\eta)\epsilon}$ steps terminates and yields an ϵ-approximation of $V^*(b_0)$.*

Proof. Let $(\emptyset, \emptyset) \in CL$. Let σ be an action-observation history and consider a policy $\pi(\sigma)$ where the agent plays action a^* chosen in `explore` when σ (represented by $(\overline{\mathbf{a}}, \overline{\mathbf{o}})$ in the algorithm) was closed on line 10. Since a history is closed when the horizon $\overline{T}$ or precision $\eta\epsilon$ is reached, or when all action-observation histories reached by playing a^* are closed, it is clear that all plays according to π eventually reach a terminal action-observation history (closed on line 6). Let b_σ be the belief after playing σ and let us assign values $v_{\mathrm{lb}}(\sigma)$ and $v_{\mathrm{ub}}(\sigma)$ to each terminal action-observation history σ corresponding to values $\underline{V}(b_\sigma)$ and $\overline{V}(b_\sigma)$ at the time σ has been closed. Let us propagate values v_{lb} and v_{ub} using Bellman equation in a bottom-up manner, i.e. for $i \in \{\mathrm{lb}, \mathrm{ub}\}$,

$$v_i(\sigma) = c(\sigma, \pi(\sigma)) + \textstyle\sum_o O_{\pi(\sigma)}(o|\sigma) \cdot v_i(\sigma\pi(\sigma)o)\,. \qquad (11)$$

Since the bounds might have improved since the histories were closed, it holds $v_{\mathrm{lb}}(\emptyset) \le \underline{V}(b_0) \le \overline{V}(b_0) \le v_{\mathrm{ub}}(\emptyset)$. Observe that the probability that any history consistent with π reaches the depth limit $\overline{T}$ (i.e., the sum of probabilities of all plays reaching the depth $\overline{T}$ when following π) is at most $p_{\mathrm{cutoff}} \le (1-\eta)\epsilon/(\overline{C} - \eta\epsilon)$. Otherwise, the contribution of those plays to the expected cost $v_{\mathrm{lb}}(\emptyset)$ would have been greater than $\overline{C}$ (as at least $c_{\min}$ is paid per step) which would have contradicted that $\overline{C}$ is the upper bound. Now, since $v_{\mathrm{ub}}(\sigma) - v_{\mathrm{lb}}(\sigma)$ in terminal histories that were cut off is less than $\overline{C}$ (reached with probability p_{cutoff}), and $v_{\mathrm{ub}}(\sigma) - v_{\mathrm{lb}}(\sigma) \le \eta\epsilon$ otherwise, the difference $v_{\mathrm{ub}}(\emptyset) - v_{\mathrm{lb}}(\emptyset)$ in the root is a weighted average of these values,

$$v_{\mathrm{ub}}(\emptyset) - v_{\mathrm{lb}}(\emptyset) \le p_{\mathrm{cutoff}}\overline{C} + (1 - p_{\mathrm{cutoff}})\eta\epsilon \le \epsilon\,. \qquad (12)$$

As $v_{\mathrm{ub}}(\emptyset) - v_{\mathrm{lb}}(\emptyset) \ge \overline{V}(b_0) - \underline{V}(b_0)$, the result follows. □

```
1  Initialize V̲ and V̄ (V̄ initialized by the uniform policy)
2  CL ← ∅
3  while V̄(b0) − V̲(b0) > ε do explore(b0, ε, 0, ∅, ∅)
4  procedure explore(b, ε, t, ā, ō)
5      if V̄(b) − V̲(b) ≤ ηε or t ≥ C̄/c_min · (C̄−ηε)/((1−η)ε) then
6          CL ← CL ∪ {(ā, ō)} and return
7      a* ← arg min_a [Σ_s b(s)c(s,a) + Σ_o O_a(o|b) V̲(b^o_a)]
8      update(b)
9      O' ← {o | (āa*, ōo) ∉ CL}
10     if O' = ∅ then CL ← CL ∪ {(ā, ō)} and return
11     o* ← arg max_{o∈O'} O_{a*}(o|b) · [V̄(b^o_{a*}) − V̲(b^o_{a*}) − ηε]
12     explore(b^{o*}_{a*}, ε, t + 1, āa*, ōo*)
13     update(b)
```

Algorithm 4: Goal-HSVI, $\eta \in [0, 1)$

As $v_{\mathrm{ub}}(\emptyset) - v_{\mathrm{lb}}(\emptyset) \ge \overline{V}(b_0) - \underline{V}(b_0)$, the result follows. □

5.2 Practical Extension: Iterative Deepening

The bound on the search depth induced by Theorem 1 can be unnecessarily large in practice. To avoid generating excessively long trials, we propose a variant of our Goal-HSVI algorithm enriched by the ideas of iterative deepening. Instead of terminating trials when they exceed $\overline{T} = \frac{\overline{C}}{c_{\min}} \cdot \frac{\overline{C}-\eta\epsilon}{(1-\eta)\epsilon}$, a cut-off depth $\widehat{T}$ is introduced, starting with $\widehat{T} \coloneqq 1$ and increasing it when one of the following situations occur:

(1) *All action-observation histories within the current depth limit $\widehat{T}$ are explored.* If no ϵ-optimal solution was found with the current depth limit, the limit must be increased. This situation is indicated by $(\emptyset, \emptyset) \in CL$.

(2) *The improvement during the current trial was insufficient.* We say that the improvement is sufficient if the difference $\overline{V}(b) - \underline{V}(b)$ *before* and *after* a call to `explore` (denoted δ and δ') weighted by the probability of seeing the observation sequence $\overline{\mathbf{o}}$ when playing actions $\overline{\mathbf{a}}$ (denoted $O_{\overline{\mathbf{a}}}(\overline{\mathbf{o}})$) is greater than $\rho(\widehat{T})$, i.e. $O_{\overline{\mathbf{a}}}(\overline{\mathbf{o}}) \cdot (\delta - \delta') \ge \rho(\widehat{T})$. Our implementation uses $\rho(\widehat{T}) = p^{\widehat{T}}$ with $p = 0.95$.

Whenever $\widehat{T}$ is increased, CL is set to $\emptyset$ to allow the algorithm to reexplore action-observation histories considered previously and search them to a greater depth. In our implementation, we always increase the search depth $\widehat{T}$ by one.

6 Empirical Evaluation

We present an experimental evaluation of Goal-HSVI in comparison with RTDP-Bel [Bonet and Geffner, 2009]. We do not include vanilla-HSVI (HSVI2 [Smith and Simmons, 2005]) in the experiments as the algorithm without modifications is theoretically incorrect in the Goal-POMDP setting, and it indeed crashed due to the recursion limit on some instances

				Goal-HSVI (15min time limit)							RTDP-Bel (150k trials)		
	$\|\mathcal{S}\|$	$\|\mathcal{A}\|$	$\|\mathcal{O}\|$	Cost	%Goal	Bounds	Time	t_{RTDP}	$\|CL\|$	$\|\Gamma \cup \Upsilon\|$	Cost	%Goal	Time
Hallway	60	5	21	14.4±0.04	100.0%	[12.8 .. 15.0]	904s	1s	14277	16788	64.7±1.2	97.5%	397s
Hallway 2	92	5	17	29.2±0.07	100.0%	[13.4 .. 66.4]	909s	2s	14616	12648	356.1±2.8	83.5%	5071s
RS[4,4]	257	9	2	231.0±0.01	100.0%	[229.0 .. 231.0]	23s	1s	16348	6651	230.9±0.3	100.0%	12s
RS[5,5]	801	10	2	306.9±0.04	100.0%	[299.2 .. 306.9]	900s	9s	62926	31661	309.9±0.3	100.0%	23s
RS[5,7]	3201	12	2	336.5±0.01	100.0%	[310.8 .. 336.5]	901s	120s	17137	8078	336.3±0.4	100.0%	62s
Seq[5,5,2]	121	5	2	15.5±0.02	100.0%	[14.0 .. 16.0]	196s	1s	19675	24107	138.9±1.9	93.8%	38s
Seq[5,5,3]	281	5	2	35.4±0.04	100.0%	[27.4 .. 36.7]	901s	1s	46753	37915	1496.7±3.3	25.9%	645s
Seq[5,5,4]	601	5	2	43.4±0.07	100.0%	[32.5 .. 51.0]	901s	1s	31355	30070	1426.5±3.5	29.6%	841s

Table 1: Experimental results (on Intel Core i7-8700K). Cost denotes average cost of the computed policy for the first 2,000 steps taken over 250,000 simulated plays. %Goal is the percentage of the simulated plays that reached a target in less than 2,000 steps. t_{RTDP} denotes the time when the Goal-HSVI upper bound $\overline{V}(b_0)$ reached the confidence interval of the cost of RTDP-Bel. $|CL|$ denotes the size of the closed list as the number of nodes of the representing prefix tree. 95% confidence intervals are reported.

(Hallway 2, Seq[5,5,3], Seq[5,5,4]). We start with the description of the setting of the algorithms considered.

Goal-HSVI. Our implementation is based on the ZMDP[1] implementation of HSVI2. We updated the solver according to Section 5. Few other changes were made: (1) We do not use adaptive precision from ZMDP that changes ϵ between iterations (fixed values $\epsilon = 2$ and $\eta = 0.8$ are used). (2) We do not use α-vector masking as the implementation of this technique in ZMDP is incompatible with Goal-POMDPs. We terminate the algorithm after 900s if an ϵ-optimal solution is not found.

RTDP-Bel. GPT solver[2] is used as a reference implementation of RTDP-Bel. Since there are no guidelines for choosing the parameters of RTDP-Bel, we use the default values used in GPT (most importantly, $K = 15$ as in [Bonet and Geffner, 2009]) except for increasing the cutoff parameter from 250 to 2000. In our experiments we let RTDP-Bel perform 150,000 trials before terminating. As RTDP-Bel is a randomized algorithm, we perform 12 independent runs and report the result with the lowest average cost. We consider the cost of RTDP-Bel policies as a reference, and we report the time when Goal-HSVI finds a policy of the same quality as t_{RTDP}.

Policy evaluation. We evaluate the quality of the policies computed by the algorithms using simulation. We perform 250,000 simulated plays (we cut each of them after 2,000 steps if the goal is not reached by that time) and we report the average total cost. We also report the percentage of simulated plays that did not terminate within the limit.

We evaluate the performance of our Goal-HSVI algorithm on three different domains. The domains are: *Hallway* [Littman *et al.*, 1995] and *RockSample* [Smith and Simmons, 2004] in their Goal-POMDP variants, and a new domain of *Sequencing* (inspired by [Kress-Gazit *et al.*, 2009]).

Hallway [Littman *et al.*, 1995]**.** An agent is navigating in a maze trying to reach the goal location while using unreliable actuators and sensors. In the original version, the agent receives a reward only when the goal is reached, and the discounted-sum objective is considered. For the Goal-POMDP version of the problem, we assume that the goal state is absorbing and each step of the agent costs one unit. [Bonet and Geffner, 2009] observed that RTDP-Bel had been outperformed by HSVI2 in the discounted-sum setting. Our Goal-HSVI algorithm similarly outperforms RTDP-Bel in the Goal-POMDP variant of Hallway (see Table 1). Moreover, the upper bound on cost produced by Goal-HSVI after 2s is lower than the cost of RTDP-Bel, and unlike RTDP-Bel, the policy produced by our algorithm always reached the goal in less than 2000 steps. (RTDP-Bel failed to reach goal in 2.5% and 16.5% of plays on Hallway and Hallway 2, respectively.)

RockSample[n,k] [Smith and Simmons, 2004]**.** A robot is operating in an $n \times n$ grid with k rocks. Each of the rocks can be either 'good' or 'bad' (unknown to the agent). The goal is to sample all the good rocks (approach them and perform expensive sampling) and then leave the map. In the Goal-POMDP version [Chatterjee *et al.*, 2016], a cost is associated with each movement. Moreover, the agent pays a penalty for all the 'good' rocks he failed to sample upon leaving the map.

RTDP-Bel works well on discounted RockSample [Bonet and Geffner, 2009] due to the problem structure (e.g., knowledge of the current position), and the same is expected in the Goal-POMDP setting. Although Goal-HSVI does not leverage the problem structure, it is competitive on all RockSample instances we consider, see Table 1. Moreover, it *provably* found solutions of a comparable (or even better) quality as RTDP-Bel by decreasing the upper bound on cost (see t_{RTDP} for the time required). Recall that RTDP-Bel cannot provide any such guarantees on the quality of the computed policy.

Sequencing[n,k,t]. An agent inspects t targets in an $n \times n$ grid with k obstacles (see Figure 5). He is uncertain about his position, and he has 5 actions available – 4 movement actions N, S, W, E and the inspection action. The movement actions are not reliable and may result in a step in any unintended direction with probability 0.15. The inspection action is deterministic and inspects the target (if there is one at the current position). The agent may receive two observations – either the last action succeeded (he stepped on an empty square / inspected a target) or it failed (he hit an obstacle / there is no target to inspect). The agent has to inspect the targets in a prescribed order – otherwise he pays a penalty $100t$ where t is the number of targets he should have inspected earlier. For example, if he inspects targets in Figure 5 in the order $(4, 1, 3, 2)$, he accumulates a penalty 400.

[1] https://github.com/trey0/zmdp

[2] https://github.com/bonetblai/gpt-rewards

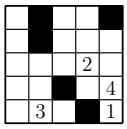


Figure 5: Sequencing[5,5,4]

We observe that RTDP-Bel does not work well for Sequencing and fails to find a policy reaching the goal state reliably, especially on the larger two instances. In contrary, our Goal-HSVI algorithm produces superior policies that always reached the goal (see Table 1). Notice that on Sequencing[5,5,3] and Sequencing[5,5,4], the time to complete 150,000 trials of RTDP-Bel is comparable to the time given to Goal-HSVI, yet still, the policy of RTDP-Bel is inferior.

7 Conclusions

We address the theoretical foundations of point-based methods for Goal-POMDPs. Previous approaches like RTDP-Bel and HSVI2 provide no convergence guarantees. We present the first algorithm, namely Goal-HSVI, with convergence guarantees and experimental results show it also performs well on examples. We believe that our work may spark further interest in analyzing point-based methods for Goal-POMDPs or generalizing Goal-HSVI to a multi-agent setting.

References

[Barto *et al.*, 1995] A. G. Barto, S. J. Bradtke, and S. P. Singh. Learning to act using real-time dynamic programming. *Artificial intelligence*, 72(1-2):81–138, 1995.

[Bertsekas and Tsitsiklis, 1996] D. P. Bertsekas and J. N. Tsitsiklis. Neuro-dynamic programming. *Athena Scientific*, 1996.

[Bertsekas, 2005] D. P. Bertsekas. *Dynamic programming and optimal control, 3rd Edition*. Athena Scientific, 2005.

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

[Bonet, 1998] B. Bonet. Solving large POMDPs using real time dynamic programming. In *In Proc. AAAI Fall Symp. on POMDPs*. Citeseer, 1998.

[Černý *et al.*, 2011] P. Černý, K. Chatterjee, T. A. Henzinger, A. Radhakrishna, and R. Singh. Quantitative synthesis for concurrent programs. In *Proc. of CAV*, LNCS 6806, pages 243–259. Springer, 2011.

[Chatterjee *et al.*, 2016] K. Chatterjee, M. Chmelík, R. Gupta, and A. Kanodia. Optimal cost almost-sure reachability in POMDPs. *Artificial Intelligence*, 234:26–48, 2016.

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

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

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

[Hauskrecht, 2000] M. Hauskrecht. Value-function approximations for partially observable Markov decision processes. *Journal of Artificial Intelligence Research*, 13:33–94, 2000.

[Howard, 1960] R. A. Howard. *Dynamic Programming and Markov Processes*. MIT Press, 1960.

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

[Kolobov *et al.*, 2011] A. Kolobov, Mausam, D.S. Weld, and H. Geffner. Heuristic search for generalized stochastic shortest path MDPs. In *ICAPS*, 2011.

[Kress-Gazit *et al.*, 2009] 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.

[Kurniawati *et al.*, 2008] 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.

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

[Littman, 1996] M. L. Littman. *Algorithms for Sequential Decision Making*. PhD thesis, Brown University, 1996.

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

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

[Patek, 2001] S. D. Patek. On partially observed stochastic shortest path problems. In *Proceedings of the 40th IEEE Conference on Decision and Control*, volume 5, pages 5050–5055. IEEE, 2001.

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

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

[Smith and Simmons, 2005] T. Smith and R. Simmons. Point-based POMDP algorithms: improved analysis and implementation. In *UAI*, pages 542–549. AUAI Press, 2005.

[Smith, 2007] T. Smith. *Probabilistic planning for robotic exploration*. Carnegie Mellon University, 2007.

[Warnquist *et al.*, 2013] H. Warnquist, J. Kvarnström, and P. Doherty. Exploiting Fully Observable and Deterministic Structures in Goal POMDPs. In *ICAPS*, pages 242–250, 2013.