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

Karel Horák1, Branislav Bošanský1, Krishnendu Chatterjee2
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 (HSVII) 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 $\gamma^j$, 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; Patik, 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

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 (HSV) 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 implementation. (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 \((S, G, A, \mathcal{O}, P, Q, c, b_0)\), where \(S\) is a finite non-empty set of states, \(G\) is a non-empty set of target (or goal) states \((G \subseteq S)\), \(A\) is a finite non-empty set of actions, \(\mathcal{O}\) is a finite non-empty set of observations \((o_a \in \mathcal{O}\) notifies the agent about reaching the goal), \(P_o(s, s') = \text{the probability to transition from } s \text{ to } s' \text{ by using action } a\), \(Q_o(a|s') = \text{the probability to observe } o \text{ when entering state } s' \text{ by using action } a\), \(c(s, a) > 0\) is the cost for taking action \(a\) in a non-target state \(s\) and \(b_0 \in \Delta(S)\) is the initial belief. The Bellman equation for POMDPs is

\[
V^*(b) = \min_{a \in A} \left[ \sum_s b(s) \cdot c(s, a) + \sum_o Q_o(a|s) \cdot V^*(b_o) \right] .
\]

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 \(V^*(b) \sim \tilde{V}((K \cdot b))\) 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).
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; Kurniai 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 (HSVII) [Smith and Simmons, 2004], the algorithm keeps two piecewise-linear and concave (PWLC) functions $V$ and $\hat{V}$ 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 $\Gamma$, while the lower bound is formed as an upper convex hull of a set of points $\Omega = \{(b_i, y_i) \mid i = 1, \ldots, m \}$ where $b_i \in \Delta(S)$.

$$V^*(b) = \min_{a \in A} V^*(a, b)$$ (5)

$$\hat{V}(b) = \max_{\lambda \in \mathbb{R}^m : \sum_{i=1}^{m} \lambda_i b_i = b} \left\{ \lambda \sum_{i=1}^{m} \lambda_i g_i \right\}.$$ (6)

Computing $V(b)$ according to Equation (6) requires solving a linear program. In the second version of the algorithm (HSVII2, [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 than the set of points, the computation of $V(b)$ can be done in linear time in the size of $\Omega$. In this work, we consider exactly this HSVII2 version of the algorithm (and we refer to it as vanilla-HSVII).

HSVII2 initializes the value function $V$ by considering policies `always play action a` and constructing one $a$-vector for each action $a \in A$ 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 $V$ and $\hat{V}$ is done by adding new elements to the sets $\Gamma$ and $\Omega$. 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 $\sum_{i=1}^{m} \gamma^{-i} c_i$ of costs $c_i$ incurred during play.

1. $\alpha^*_a \leftarrow \arg \min_{a \in A} \gamma^{-i} c_i$ for all $a \in A, c_i \in \mathcal{O}$
2. $\alpha_a(s) \leftarrow c(s, a) + \gamma \sum_{s', \alpha} Pr_{\tau}(s', \alpha | s) \alpha_a(s') \quad \forall s, a$
3. $\Gamma' \leftarrow \Gamma \cup \{\alpha_{a} \}$
4. $\Omega' \leftarrow \Omega \cup \{\sum_{i=1}^{m} \lambda_i b_i = b \}$

Algorithm 2: Point-based update of $V$.

Similarly to RTDP-Bel, HSVII2 selects beliefs where the update should be performed based on the simulated play (selecting actions according to $V$). 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 $\hat{V}(b_G) - \hat{V}(b_a) - c_{\gamma^{-t+1}}$ in $b_a$ weighted by the probability $O_a(o|b)$ is the greatest. This heuristic choice attempts to target beliefs where the update will have the most significant impact on $\hat{V}(b_a) - \hat{V}(b_G)$.

The HSVII2 algorithm for discounted-sum POMDPs ($0 < \gamma < 1$) is shown in Algorithm 3. This algorithm provably converges to an $\epsilon$-approximation of $V^*$ using values $\hat{V}(b_a)$ and $\hat{V}(b_G)$, see [Smith and Simmons, 2004].

4 Vanilla-HSVI and Goal-POMDPs

While HSVII2 (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 $V$ 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, $V(s_1)$ remains infinite forever.

Solution: Instead of blind policies, we initialize $V$ using the uniform policy. Since the goal state $q$ is reachable from every state, the uniform policy reaches $q$ 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 $\gamma^{-t}$ is strictly increasing and unbounded (since $0 < \gamma < 1$). Its value therefore eventually exceeds the gap $\hat{V}(b) - \hat{V}(b)$ (which is guaranteed by the initialization to be bounded) and the recursive explore procedure of
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 $y$ can only be reached by playing action $r_1$ or $r_2$ at some point. We show, however, that the way the value function $V$ 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 $T \in Z_0^+$ and

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

and that the lower bound on the cost of playing $r_1$ in $b^T$, $V^*(b^T)$, is independent of $T$ and constant

$$V^*(b^T) = \sum_{r_1} (b^T(r_1) c(r, r_1) + b^T(s_1) c(s_1, r_1) + b^T(s_2, r_1) c(s_2, r_1)) = 0 + 0.8^T \cdot 50 + (1 - 0.8^T)/2 \cdot 1 + (1 - 0.8^T)/2 \cdot 0.99. \quad (8)$$

This means that the explore procedure selects $r_2$ in $b^T$ only if $V^m(b^T) = V^m(b^T) + c(s, m) \geq V^*(b^T) = 50$.

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

$$V^m(b^0) = V(b^1) + c(s, m) = 5 + 1 = 6 = y^0 \quad (9)$$

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

At the $T$-th level of explore recursion, set $T$ 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

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

At this point, $V(b^{T+1})$ is a convex combination of values $y^{T-1} = V(b^{T-1})$ and $V(b^\infty), 0.8^2 y^{T-1} + 0.36 \cdot 1$ (see Figure 4 for illustration). If $m$ was always optimal, the sequence of $y^T$ values can be characterized by a difference equation $y^T = 1 + 0.8^2 y^{T-1} + 0.36 - 1$. 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 $V(b^T) \geq V^*(b^T) = 50$ for every $T \in Z_0^+$ (we cannot avoid playing $r_1$ eventually) and thus for sufficiently small $\epsilon$ 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 $\epsilon$-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 $V$ and $\bar{V}$ are never changed during the trial. Hence, after termination, a new trial operates on the same $V$ and $\bar{V}$. 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 $\bar{V}$ (line 1).**
2. **The search depth is bounded (line 5).** We terminate trials longer than $\frac{C}{c_{\min}} \frac{1}{\epsilon}$ steps where $C$ 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 $\frac{\bar{V}(b) - V^*(b)}{\eta} \leq \epsilon$
(for $\eta < 1$) guarantees that an $\epsilon$-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 $(\mathbf{h}, \mathbf{o})$ 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 $a^* \in O'$ (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 $V$ and $\nabla$.

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

**Proof.** Let $(\emptyset, \emptyset) \in CL$. Let $\sigma$ be an action-observation history and consider a policy $\pi(\sigma)$ where the agent plays action $a^*$ chosen in explore when $\sigma$ (represented by $(\mathbf{h}, \mathbf{o})$ in the algorithm) was closed on line 10. Since a history is closed when the horizon $T$ or precision $\eta$ is reached, or when all action-observation histories reached by playing $a^*$ are closed, it is clear that all plays according to $\pi$ eventually reach a terminal action-observation history (closed on line 6). Let $b_0$ be the belief after playing $\sigma$ and let us assign values $v_{\pi}(\sigma)$ and $v_{\pi}(\sigma)$ to each terminal action-observation history $\sigma$ corresponding to values $V(b_0)$ and $\nabla(b_0)$ at the time $\sigma$ has been closed. Let us propagate values $V(b_0)$ and $\nabla(b_0)$ using Bellman equation in a bottom-up manner, i.e. for $i \in \{lb, ub\}$,

\[
v_i(\sigma) = c(\sigma, \pi(\sigma)) + \sum_o O_\pi(\sigma, o) \cdot v_i(\sigma \pi(\sigma) o)
\]

Since the bounds might have improved since the histories were closed, it holds $V_{lb}(\emptyset) \leq \nabla(b_0) \leq V_{ub}(\emptyset)$. Observe that the probability that any history consistent with $\pi$ reaches the depth limit $T$ (i.e., the sum of probabilities of all plays reaching the depth $T$ when following $\pi$) is at most $p_{\text{cut-off}} \leq (1 - \eta)c/(\overline{C} - \eta \epsilon)$. Otherwise, the contribution of those plays to the expected cost $v_{lb}(\emptyset)$ 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 $v_{lb}(\sigma) - v_{lb}(\emptyset)$ in terminal histories that were cut off is less than $\overline{C}$ (reached with probability $p_{\text{cut-off}}$), and $v_{ub}(\sigma) - v_{ub}(\emptyset) \leq \eta \epsilon$ otherwise, the difference $v_{lb}(\emptyset) - v_{lb}(\emptyset)$ in the root is a weighted average of these values,

\[
v_{lb}(\emptyset) - v_{lb}(\emptyset) \leq p_{\text{cut-off}} \overline{C} + (1 - p_{\text{cut-off}}) \eta \epsilon \leq \epsilon.
\]

As $v_{lb}(\emptyset) - v_{lb}(\emptyset) \geq \nabla(b_0) - V(b_0)$, 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 $T = \frac{C}{c_{\min}} \cdot \frac{1}{\epsilon} - \eta \epsilon$, a cut-off depth $\hat{T}$ is introduced, starting with $\hat{T} := 1$ and increasing it when one of the following situations occur:

1. All action-observation histories within the current depth limit $\hat{T}$ are explored. If no $\epsilon$-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 $V(b) - V(b)$ before and after a call to explore (denoted $\delta$ and $\delta'$) weighted by the probability of seeing the observation sequence $\mathbf{o}$ when playing actions $\mathbf{a}$ (denoted $O_{\mathbf{o}}(\mathbf{a})$) is greater than $\rho(\hat{T})$, i.e. $O_{\mathbf{o}}(\mathbf{a}) \cdot (\delta - \delta') \geq \rho(\hat{T})$. Our implementation uses $\rho(\hat{T}) = p^{\hat{T}}$ with $p = 0.95$.

Whenever $\hat{T}$ 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 $\hat{T}$ 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.
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 $\bar{V}(b_0)$ 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.

<table>
<thead>
<tr>
<th></th>
<th>Cost (RTDP-Bel (150k trials))</th>
<th>Cost (Goal-HSVI (15min time limit))</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$</td>
<td>S</td>
</tr>
<tr>
<td></td>
<td>$\text{Cost}$</td>
<td>$%\text{Goal}$</td>
</tr>
<tr>
<td>Hallway</td>
<td>60</td>
<td>5</td>
</tr>
<tr>
<td>Hallway 2</td>
<td>92</td>
<td>5</td>
</tr>
<tr>
<td>RS[4,4]</td>
<td>257</td>
<td>9</td>
</tr>
<tr>
<td>RS[5,5]</td>
<td>801</td>
<td>10</td>
</tr>
<tr>
<td>RS[5,7]</td>
<td>320</td>
<td>12</td>
</tr>
<tr>
<td>Seq[5,5,2]</td>
<td>121</td>
<td>5</td>
</tr>
<tr>
<td>Seq[5,5,3]</td>
<td>281</td>
<td>5</td>
</tr>
<tr>
<td>Seq[5,5,4]</td>
<td>601</td>
<td>5</td>
</tr>
</tbody>
</table>

Goal-HSVI. Our implementation is based on the ZMDP\footnote{https://github.com/trey0/zmdp} 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 $\epsilon$ between iterations (fixed values $\epsilon = 2$ and $\eta = 0.8$ are used). (2) We do not use $\alpha$-vector masking as the implementation of this technique in ZMDP is incompatible with Goal-POMDPs. We terminate the algorithm after 900s if an $\alpha$-optimal solution is not found.

RTDP-Bel. GPT solver\footnote{https://github.com/bonetblai/gpt-rewards} 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[$n,k,d$]. 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 100$t$ 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.
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