Point-based value iteration: An anytime algorithm for POMDPs

Joelle Pineau, Geoff Gordon and Sebastian Thrun
Carnegie Mellon University
Robotics Institute
5000 Forbes Avenue
Pittsburgh, PA 15213
{jpineau,ggordon,thrun}@cs.cmu.edu

Abstract

This paper introduces the *Point-Based Value Iteration* (PBVI) algorithm for POMDP planning. PBVI approximates an exact value iteration solution by selecting a small set of representative belief points and then tracking the value and its derivative for those points only. By using stochastic trajectories to choose belief points, and by maintaining only one value hyper-plane per point, PBVI successfully solves large problems: we present results on a robotic laser tag problem as well as three test domains from the literature.

1 Introduction

The value iteration algorithm for planning in partially observable Markov decision processes (POMDPs) was introduced in the 1970s [Sondik, 1971]. Since its introduction numerous authors have refined it [Cassandra *et al.*, 1997; Kaelbling *et al.*, 1998; Zhang and Zhang, 2001] so that it can solve harder problems. But, as the situation currently stands, POMDP value iteration algorithms are widely believed not to be able to scale to real-world-sized problems.

There are two distinct but interdependent reasons for the limited scalability of POMDP value iteration algorithms. The more widely-known reason is the so-called curse of dimensionality [Kaelbling *et al.*, 1998]: in a problem with n physical states, POMDP planners must reason about belief states in an $(n-1)$-dimensional continuous space. So, naive approaches like discretizing the belief space scale exponentially with the number of states.

The less-well-known reason for poor scaling behavior is what we will call the curse of history: POMDP value iteration is in many ways like breadth-first search in the space of belief states. Starting from the empty history, it grows a set of histories (each corresponding to a reachable belief) by simulating the POMDP. So, the number of distinct action-observation histories considered grows exponentially with the planning horizon. Various clever pruning strategies [Littman *et al.*, 1995; Cassandra *et al.*, 1997] have been proposed to whittle down the set of histories considered, but the pruning steps are usually expensive and seem to make a difference only in the constant factors rather than the order of growth.

The two curses, history and dimensionality, are related: the higher the dimension of a belief space, the more room it has for distinct histories. But, they can act independently: planning complexity can grow exponentially with horizon even in problems with only a few states, and problems with a large number of physical states may still only have a small number of relevant histories. In most domains, the curse of history affects POMDP value iteration far more strongly than the curse of dimensionality [Kaelbling *et al.*, 1998; Zhou and Hansen, 2001]. That is, the number of distinct histories which the algorithm maintains is a far better predictor of running time than is the number of states. The main claim of this paper is that, if we can avoid the curse of history, there are many real-world POMDPs where the curse of dimensionality is *not* a problem.

Building on this insight, we present *Point-Based Value Iteration* (PBVI), a new approximate POMDP planning algorithm. PBVI selects a small set of representative belief points and iteratively applies value updates to those points. The point-based update is significantly more efficient than an exact update (quadratic vs. exponential), and because it updates both value and value gradient, it generalizes better to unexplored beliefs than interpolation-type grid-based approximations which only update the value [Lovejoy, 1991; Brafman, 1997; Hauskrecht, 2000; Zhou and Hansen, 2001; Bonet, 2002]). In addition, exploiting an insight from policy search methods and MDP exploration [Ng and Jordan, 2000; Thrun, 1992], PBVI uses explorative stochastic trajectories to select belief points, thus reducing the number of belief points necessary to find a good solution compared to earlier approaches. Finally, the theoretical analysis of PBVI included in this paper shows that it is guaranteed to have bounded error.

This paper presents empirical results demonstrating the successful performance of the algorithm on a large (870 states) robot domain called *Tag*, inspired by the game of lasertag. This is an order of magnitude larger than other problems commonly used to test scalable POMDP algorithms. In addition, we include results for three well-known POMDPs, where PBVI is able to match (in control quality, but with fewer belief points) the performance of earlier algorithms.

2 An overview of POMDPs

The POMDP framework is a generalized model for planning under uncertainty [Kaelbling *et al.*, 1998; Sondik, 1971]. A POMDP can be represented using the following n-tuple: $\{S, A, O, b_0, T, \Omega, R, \gamma\}$, where S is a (finite) set of discrete states, A is a set of discrete actions, and O is a set of discrete observations providing incomplete and/or noisy state information. The POMDP model is parameterized by: $b_0(s)$, the initial belief distribution; $T(s,a,s') := Pr(s_{t+1} = s \mid a_t = a, s_t = s)$, the distribution describing the probability of transitioning from state s to state s' when taking action a; $\Omega(o,s,a) := Pr(o_{t+1} = o \mid a_t = a, s_{t+1} = s)$, the distribution describing the probability of observing o from state s after taking action a; $R(s,a)$, the reward signal received when executing action a in state s; and γ, the discount factor.

A key assumption of POMDPs is that the state is only partially observable. Therefore we rely on the concept of a belief state, denoted b, to represent a probability distribution over states. The belief is a sufficient statistic for a given history:

$$b_t := Pr(s_t \mid b_0, a_0, o_1, ..., o_{t-1}, a_{t-1}, o_t) \qquad (1)$$

and is updated at each time-step to incorporate the latest action, observation pair:

$$b_t(s') := \eta\Omega(o,s',a)\sum_{s\in S} T(s,a,s')b_{t-1}(s) \qquad (2)$$

where η is the normalizing constant.

The goal of POMDP planning is to find a sequence of actions $\{a_0, \ldots, a_t\}$ maximizing the expected sum of rewards $E[\sum_t \gamma^t R(s_t,a_t)]$. Given that the state is not necessarily fully observable, the goal is to maximize expected reward for each belief. The value function can be formulated as:

$$V(b) = \max_{a\in A}\left[R(b,a) + \gamma\sum_{b'\in B} T(b,a,b')V(b')\right] \qquad (3)$$

When optimized exactly, this value function is always piecewise linear and convex in the belief [Sondik, 1971] (see Fig. 1, left side). After n consecutive iterations, the solution consists of a set of α-vectors: $V_n = \{\alpha_0, \alpha_1, ..., \alpha_m\}$. Each α-vector represents an $|S|$-dimensional hyper-plane, and defines the value function over a bounded region of the belief: $V_n(b) = \max_{\alpha\in V_n}\sum_{s\in S}\alpha(s)b(s)$. In addition, each α-vector is associated with an action, defining the best immediate policy assuming optimal behavior for the following $(n-1)$ steps (as defined respectively by the sets $\{V_{n-1}, ..., V_0\}$).

The n-th horizon value function can be built from the previous solution V_{n-1} using the *Backup* operator, H. We use notation $V = HV'$ to denote an exact value backup:

$$\begin{aligned} V(b) \;=\; & \max_{a\in A}\Bigg[\sum_{s\in S} R(s,a)b(s) + \qquad (4) \\ & \gamma\sum_{o\in O}\max_{\alpha'\in V'}\sum_{s\in S}\sum_{s'\in S} T(s,a,s')\Omega(o,s',a)\alpha'(s')b(s)\Bigg] \end{aligned}$$

A number of algorithms have been proposed to implement this backup by directly manipulating α-vectors, using a combination of set projection and pruning operations [Sondik, 1971; Cassandra *et al.*, 1997; Zhang and Zhang, 2001]. We now describe the most straight-forward version of exact POMDP value iteration.

To implement the exact update $V = HV'$, we first generate intermediate sets $\Gamma^{a,*}$ and $\Gamma^{a,o}, \forall a \in A, \forall o \in O$ *(Step 1)*:

$$\Gamma^{a,*} \;\leftarrow\; \alpha^{a,*}(s) = R(s,a) \qquad (5)$$

$$\Gamma^{a,o} \;\leftarrow\; \alpha_i^{a,o}(s) = \gamma\sum_{s'\in S} T(s,a,s')\Omega(o,s',a)\alpha_i'(s'), \forall\alpha_i' \in V'$$

Next we create Γ^a ($\forall a \in A$), the cross-sum over observations, which includes one $\alpha^{a,o}$ from each $\Gamma^{a,o}$ *(Step 2)*:

$$\Gamma^a \;=\; \Gamma^{a,*} \oplus \Gamma^{a,o_1} \oplus \Gamma^{a,o_2} \oplus \ldots \qquad (6)$$

Finally we take the union of Γ^a sets *(Step 3)*:

$$V \;=\; \cup_{a\in A}\Gamma^a \qquad (7)$$

In practice, many of the vectors in the final set V may be completely dominated by another vector ($\alpha_i \cdot b < \alpha_j \cdot b, \forall b$), or by a combination of other vectors. Those vectors can be pruned away without affecting the solution. Finding dominated vectors can be expensive (checking whether a single vector is dominated requires solving a linear program), but is usually worthwhile to avoid an explosion of the solution size.

To better understand the complexity of the exact update, let $|V'|$ be the number of α-vectors in the previous solution set. Step 1 creates $|A|\,|O|\,|V'|$ projections and Step 2 generates $|A|\,|V'|^{|O|}$ cross-sums. So, in the worst case, the new solution $|V|$ has size $|A|\,|V'|^{|O|}$ (time $|S|^2|A|\,|V'|^{|O|}$). Given that this exponential growth occurs for every iteration, the importance of pruning away unnecessary vectors is clear. It also highlights the impetus for approximate solutions.

3 Point-based value iteration

It is a well understood fact that most POMDP problems, even given arbitrary action and observation sequences of infinite length, are unlikely to reach most of the points in the belief simplex. Thus it seems unnecessary to plan equally for all beliefs, as exact algorithms do, and preferable to concentrate planning on most probable beliefs.

The *point-based value iteration* (PBVI) algorithm solves a POMDP for a finite set of belief points $B = \{b_0, b_1, ..., b_q\}$. It initializes a separate α-vector for each selected point, and repeatedly updates (via value backups) the value of that α-vector. As shown in Figure 1, by maintaining a full α-vector for each belief point, PBVI preserves the piece-wise linearity and convexity of the value function, and defines a value function over the entire belief simplex. This is in contrast to grid-based approaches [Lovejoy, 1991; Brafman, 1997; Hauskrecht, 2000; Zhou and Hansen, 2001; Bonet, 2002], which update only the value at each belief grid point.

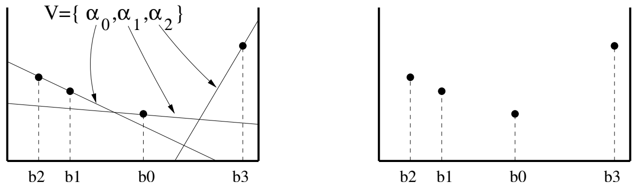


Figure 1: POMDP value function representation using PBVI (on the left) and a grid (on the right).

The complete PBVI algorithm is designed as an *anytime* algorithm, interleaving steps of value iteration and steps of belief set expansion. It starts with an initial set of belief points for which it applies a first series of backup operations. It then grows the set of belief points, and finds a new solution for the expanded set. By interleaving value backup iterations with expansions of the belief set, PBVI offers a range of solutions, gradually trading off computation time and solution quality. We now describe how we can efficiently perform point-based value backups and how we select belief points.

3.1 Point-based value backup

To plan for a finite set of belief points B, we modify the backup operator (Eqn 4) such that only one α-vector per belief point is maintained. For a point-based update $V = \tilde{H}V'$, we start by creating projections (exactly as in Eqn 5) $\forall a \in A, \forall o \in O$ *(Step 1)*:

$$\Gamma^{a,*} \leftarrow \alpha^{a,*}(s) = R(s,a) \tag{8}$$

$$\Gamma^{a,o} \leftarrow \alpha_i^{a,o}(s) = \gamma \sum_{s' \in S} T(s,a,s')\Omega(o,s',a)\alpha_i'(s'), \forall \alpha_i' \in V'$$

Next, the cross-sum operation (Eqn 6) is much simplified by the fact that we are now operating over a finite set of points. We construct $\forall b \in B, \forall a \in A$ *(Step 2)*:

$$\Gamma_b^a = \Gamma^{a,*} + \sum_{o \in O} \underset{\alpha \in \Gamma^{a,o}}{\operatorname{argmax}}(\alpha \cdot b) \tag{9}$$

Finally, we find the best action for each belief point *(Step 3)*:

$$V \leftarrow \underset{\Gamma_b^a, \forall a \in A}{\operatorname{argmax}}(\Gamma_b^a \cdot b), \quad \forall b \in B \tag{10}$$

When performing point-based updates, the backup creates $|A|\,|O|\,|V'|$ projections as in exact VI. However the final solution V is limited to containing only $|B|$ components (in time $|S|\,|A|\,|V'|\,|O|\,|B|$). Thus a full point-based value update takes only polynomial time, and even more crucial, the size of the solution set V remains constant. As a result, the pruning of α vectors (and solving of linear programs), so crucial in exact POMDP algorithms, is now unnecessary. The only pruning step is to refrain from adding to V any vector already included, which arises when two nearby belief points support the same vector (e.g. b_1, b_2 in Fig. 1).

In problems with a finite horizon h, we run h value backups before expanding the set of belief points. In infinite-horizon problems, we select the horizon so that $(R_{max} - R_{min})\gamma^h < \epsilon$.

3.2 Belief point set expansion

As explained above, PBVI focuses its planning on relevant beliefs. More specifically, our error bound below suggests that PBVI performs best when its belief set is uniformly dense in the set of reachable beliefs. So, we initialize the set B to contain the initial belief b_0 and expand B by greedily choosing new reachable beliefs that improve the worst-case density as rapidly as possible.

For a given $b \in B$, PBVI stochastically simulates a single-step forward trajectory using each action to produce new beliefs $\{b_{a_0}, b_{a_1}, ...\}$.[1] It then measures the L_1 distance from b_{a_i} to B, and throws away b_{a_i} if $b_{a_i} \in B$. Finally, it keeps only the new belief b_{a_i} which is farthest away from any point already in B.[2] PBVI tries to generate one new belief from each previous belief; so, B at most doubles in size on each expansion.[3] Since expansion phases are interleaved with value iteration, PBVI offers an *anytime* solution.

3.3 Convergence and error bounds

For any belief set B and horizon n, PBVI produces an estimate V_n^B. The error between V_n^B and the true value function V_n^* is bounded. The bound depends on how densely B samples the belief simplex Δ; with denser sampling, V_n^B converges to V_n^*, the true value function.[4] As $n \to \infty$, V_n^B does not necessarily converge; but, our error bound still holds. Cutting off the PBVI iteration at *any* sufficiently large horizon, we know that the difference between V_n^B and the optimal infinite-horizon V^* is not too large. (The overall error is bounded by $\|V_n^B - V_n^*\|_\infty + \|V_n^* - V^*\|_\infty$. The first term is bounded by our theorem below; the second is bounded by $\gamma^n \|V_0^* - V^*\|$.) The remainder of this section states and proves our error bound.

Define the density ϵ_B of a set of belief points B to be the maximum distance from any legal belief to B. More precisely, $\epsilon_B = \max_{b' \in \Delta} \min_{b \in B} \|b - b'\|_1$. Then, we can prove:

Lemma 1 *The error introduced by PBVI's pruning step is at most* $\eta_{\text{prune}} = \frac{(R_{\max} - R_{\min})\epsilon_B}{1-\gamma}$.

Proof: Let $b' \in \Delta$ be the point where PBVI makes its worst pruning error, and $b \in B$ be the closest (1-norm) sampled belief to b'. Let α be the vector which is maximal at b, and α' be maximal at b'. By erroneously pruning α', PBVI makes an error of at most $\alpha' \cdot b' - \alpha \cdot b'$. On the other hand, since α is maximal at b, then $\alpha' \cdot b \le \alpha \cdot b$. So,

$$\begin{aligned}
\eta_{\text{prune}} &\le \alpha' \cdot b' - \alpha \cdot b' \\
&= \alpha' \cdot b' - \alpha \cdot b' + (\alpha' \cdot b - \alpha' \cdot b) && \text{add zero} \\
&\le \alpha' \cdot b' - \alpha \cdot b' + \alpha \cdot b - \alpha' \cdot b && \alpha \text{ opt. at } b \\
&= (\alpha' - \alpha) \cdot (b' - b) && \text{collect terms} \\
&\le \|\alpha' - \alpha\|_\infty \|b' - b\|_1 && \text{Hölder} \\
&\le \|\alpha' - \alpha\|_\infty \epsilon_B && \text{def'n of } \epsilon_B \\
&\le \tfrac{R_{\max} - R_{\min}}{1-\gamma} \epsilon_B && \text{see text}
\end{aligned}$$

The last inequality holds because each α-vector represents the reward achievable starting from some state and following some sequence of actions and observations. ∎

Theorem 1 *For any belief set B and any horizon n, the error of the PBVI algorithm $\eta_n = \|V_n^B - V_n^*\|_\infty$ is bounded by*

$$\eta_n \le \frac{(R_{max} - R_{min})\epsilon_B}{(1-\gamma)^2}$$

[1] To simulate an action a, first sample a state s from the distribution b. Then sample an observation according to $\Omega(s,a,o)$ for the given s and a. Finally compute b_a using a Bayesian update (Eqn 2).

[2] The actual choice of norm doesn't appear to matter in practice; some of our experiments below used Euclidean distance (instead of L_1) and the results appear identical.

[3] We experimented with other strategies such as adding a fixed number of new beliefs, but since value iteration is much more expensive than belief computation the above algorithm worked best. If desired, we can impose a maximum size on B based on time constraints or performance requirements.

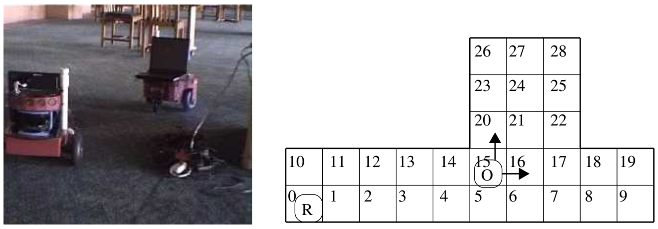
[4] If not all beliefs are reachable, we don't need to sample all of Δ densely, but replace Δ by the set of reachable beliefs $\bar{\Delta}$ below. The error bounds and convergence results hold on $\bar{\Delta}$.

Proof:

$$\begin{aligned}
\eta_n &= ||V_n^B - V_n^*||_\infty && \text{def'n of } \eta_n \\
&= ||\tilde{H}V_{n-1}^B - HV_{n-1}^*||_\infty && \text{def'n of } H, \tilde{H} \\
&\leq ||\tilde{H}V_{n-1}^B - HV_{n-1}^B||_\infty + \\
&\quad\ ||HV_{n-1}^B - HV_{n-1}^*||_\infty && \text{triangle ineq.} \\
&\leq \eta_{\text{prune}} + ||HV_{n-1}^B - HV_{n-1}^*||_\infty && \text{def'n of } \eta_{\text{prune}} \\
&\leq \eta_{\text{prune}} + \gamma||V_{n-1}^B - V_{n-1}^*||_\infty && \text{contraction} \\
&= \eta_{\text{prune}} + \gamma\eta_{n-1} && \text{def'n of } \eta_{n-1} \\
&\leq \frac{(R_{max}-R_{min})\epsilon_B}{1-\gamma} + \gamma\eta_{n-1} && \text{lemma 1} \\
&\leq \frac{(R_{max}-R_{min})\epsilon_B}{(1-\gamma)^2} && \text{series sum} \quad \blacksquare
\end{aligned}$$

4 Experimental results

The domain of *Tag* is based on the popular game of lasertag. The goal is to search for and tag a moving opponent [Rosencrantz *et al.*, 2003]. Figure 2a shows the live robot as it moves in to capture an opponent. In our POMDP formulation, the opponent moves stochastically according to a fixed policy. The spatial configuration of the domain used for planning is illustrated in Figure 2b. This domain is an order of magnitude larger (870 states) than most other POMDP problems considered thus far in the literature [Cassandra, 1999], and is proposed as a new challenge for fast, scalable, POMDP algorithms. A single iteration of optimal value iteration on a problem of this size could produce over 10^{20} α-vectors before pruning.



a. Robots playing Tag b. Tag configuration

Figure 2: Tag domain (870 states, 5 actions, 30 observations)

The state space is described by the cross-product of two features, *Robot* $= \{s_0, \ldots, s_{29}\}$ and *Opponent* $= \{s_0, \ldots, s_{29}, s_{tagged}\}$. Both agents start in independently-selected random positions, and the game finishes when *Opponent* $= s_{tagged}$. The robot can select from five actions: $\{$*North, South, East, West, Tag*$\}$. A reward of -1 is imposed for each motion action; the *Tag* action results in a $+10$ reward if *Robot* $=$ *Opponent*, or -10 otherwise. Throughout the game, the Robot's position is fully observable, and the effect of a *Move* action has the predictable deterministic effect, e.g.:

$$Pr(\textit{Robot} = s_{10} \mid \textit{Robot} = s_0, North) = 1$$

The position of the opponent is completely unobservable unless both agents are in the same cell. At each step, the opponent (with omniscient knowledge) moves away from the robot with $Pr = 0.8$ and stays in place with $Pr = 0.2$, e.g.:

$$Pr(\textit{Opponent} = s_{16} \mid \textit{Opponent} = s_{15} \& \textit{Robot} = s_0) = 0.4$$
$$Pr(\textit{Opponent} = s_{20} \mid \textit{Opponent} = s_{15} \& \textit{Robot} = s_0) = 0.4$$
$$Pr(\textit{Opponent} = s_{15} \mid \textit{Opponent} = s_{15} \& \textit{Robot} = s_0) = 0.2$$

Figure 3 shows the performance of PBVI on the Tag domain. Results are averaged over 10 runs of the algorithm, times 100 different (randomly chosen) start positions for each run. It shows the gradual improvement in performance as samples are added (each shown data point represents a new expansion of the belief set with value backups). In addition to PBVI, we also apply the QMDP approximation as a baseline comparison [Littman *et al.*, 1995]. The QMDP approximation is calculated by solving a POMDP as though it were fully observable: $Q(s,a) = R(s,a) + \gamma \sum_{s' \in S} T(s,a,s')V(s')$, and linearizing across Q-values to obtain the value at a belief: $V(b) = max_{a \in A} \sum_{s \in S} b(s)Q(s,a)$. This approximation is quick to compute, and is remarkably effective in some domains. In the Tag domain, however, it lacks the representational power to compute a good policy.

5 Additional experiments

5.1 Comparison on well-known problems

To further analyze the performance of PBVI, we applied it to three well-known problems from the POMDP literature. We selected Maze33, Hallway and Hallway2 because they are commonly used to test scalable POMDP algorithms [Littman *et al.*, 1995; Brafman, 1997; Poon, 2001]. Figure 3 presents results for each domain. Replicating earlier experiments, results for Maze33 are averaged over 151 runs (reset after goal, terminate after 500 steps); results for Hallway and Hallway2 are averaged over 251 runs (terminate at goal, max 251 steps). In all cases, PBVI is able to find a good policy. Table 1 compares PBVI's performance with previously published results, comparing goal completion rates, sum of rewards, policy computation time, and number of required belief points. In all domains, PBVI achieves competitive performance, but with fewer samples.

5.2 Validation of the belief set expansion

To further investigate the validity of our approach for generating new belief states (Section 3.2), we compared our approach with three other techniques which might appear promising. In all cases, we assume that the initial belief b_0 (given as part of the model) is the sole point in the initial set, and consider four expansion methods:

1. Random (**RA**)
2. Stochastic Simulation with Random Action (**SSRA**)
3. Stochastic Simulation with Greedy Action (**SSGA**)
4. Stochastic Simulation with Explorative Action (**SSEA**)

The **RA** method consists of sampling a belief point from a uniform distribution over the entire belief simplex. **SSEA** is the standard PBVI expansion heuristic (Section 3.2). **SSRA** similarly uses single-step forward simulation, but rather than try all actions, it randomly selects one and automatically accepts the posterior belief unless it was already in B. Finally, **SSGA** uses the most recent value function solution to pick the greedy action at the given belief b, and performs a single-step simulation to get a new belief $b' \rightarrow B$.

We revisited the Hallway, Hallway2, and Tag problems from sections 4 and 5.1 to compare the performance of these

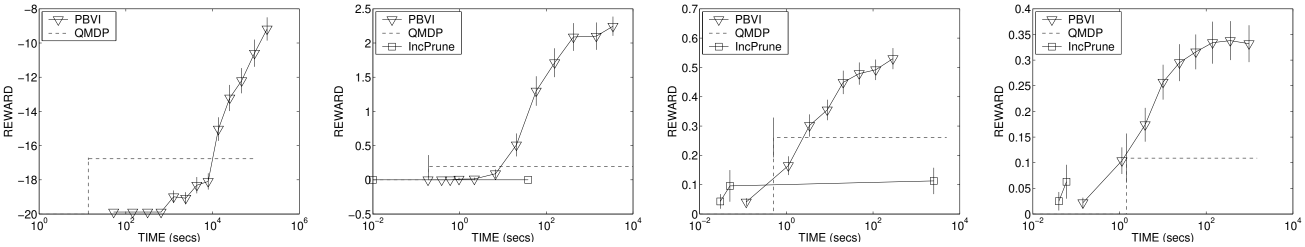


Figure 3: PBVI performance for four problems: Tag(left), Maze33(center-left), Hallway(center-right) and Hallway2(right)

Method	Goal%	Reward	Time(s)	$\|B\|$
Maze33 / Tiger-Grid				
QMDP[*]	n.a.	0.198	0.19	n.a.
Grid [Brafman, 1997]	n.a.	0.94	n.v.	174
PBUA [Poon, 2001]	n.a.	2.30	12116	660
PBVI[*]	n.a.	2.25	3448	470
Hallway				
QMDP[*]	47	0.261	0.51	n.a.
QMDP [Littman *et al.*, 1995]	47.4	n.v.	n.v.	n.a.
PBUA [Poon, 2001]	100	0.53	450	300
PBVI[*]	96	0.53	288	86
Hallway2				
QMDP[*]	22	0.109	1.44	n.a.
QMDP [Littman *et al.*, 1995]	25.9	n.v.	n.v.	n.a.
Grid [Brafman, 1997]	98	n.v.	n.v.	337
PBUA [Poon, 2001]	100	0.35	27898	1840
PBVI[*]	98	0.34	360	95
Tag				
QMDP[*]	17	-16.769	13.55	n.a.
PBVI[*]	59	-9.180	180880	1334
n.a.=not applicable	n.v.=not available			

Table 1: Results for POMDP domains. Those marked [*] were computed by us; other results were likely computed on different platforms, and therefore time comparisons may be approximate at best. All results assume a standard (not *lookahead*) controller.

four heuristics. For each problem we apply PBVI using each of the belief-point selection heuristics, and include the QMDP approximation as a baseline comparison. Figure 4 shows the computation time versus the reward performance for each domain.

The key result from Figure 4 is the rightmost panel, which shows performance on the largest, most complicated domain. In this domain our SSEA rule clearly performs best. In smaller domains (left two panels) the choice of heuristic matters less: all heuristics except random exploration (RA) perform equivalently well.

6 Related work

Significant work has been done in recent years to improve the tractability of POMDP solutions. A number of increasingly efficient exact value iteration algorithms have been proposed [Cassandra *et al.*, 1997; Kaelbling *et al.*, 1998; Zhang and Zhang, 2001]. They are successful in finding optimal solutions, however are generally limited to very small problems (a dozen states) since they plan optimally for all beliefs. PBVI avoids the exponential growth in plan size by restricting value updates to a finite set of (reachable) beliefs.

There are several approximate value iteration algorithms which are related to PBVI. For example, there are many grid-based methods which iteratively update the values of discrete belief points. These methods differ in how they partition the belief space into a grid [Brafman, 1997; Zhou and Hansen, 2001].

More similar to PBVI are those approaches which update both the value and gradient at each grid point [Lovejoy, 1991; Hauskrecht, 2000; Poon, 2001]. While the actual point-based update is essentially the same between all of these, the overall algorithms differ in a few important aspects. Whereas Poon only accepts updates that increase the value at a grid point (requiring special initialization of the value function), and Hauskrecht always keeps earlier α-vectors (causing the set to grow too quickly), PBVI requires no such assumptions. A more important benefit of PBVI is the theoretical guarantees it provides: our guarantees are more widely applicable and provide stronger error bounds than those for other point-based updates.

In addition, PBVI is significantly smarter than previous algorithms about how it selects belief points. PBVI selects only reachable beliefs; other algorithms use random beliefs, or (like Poon's and Lovejoy's) require the inclusion of a large number of fixed beliefs such as the corners of the probability simplex. Moreover, PBVI selects belief points which improve its error bounds as quickly as possible. In practice, our experiments on the large domain of lasertag demonstrate that PBVI's belief-selection rule handily outperforms several alternate methods. (Both Hauskrecht and Poon did consider using stochastic simulation to generate new points, but neither found simulation to be superior to random point placements. We attribute this result to the smaller size of their test domains. We believe that as more POMDP research moves to larger planning domains, newer and smarter belief selection rules will become more and more important.)

Gradient-based policy search methods have also been used to optimize POMDP solutions [Baxter and Bartlett, 2000; Kearns *et al.*, 1999; Ng and Jordan, 2000], successfully solving multi-dimensional, continuous-state problems. In our view, one of the strengths of these methods lies in the fact that they restrict optimization to reachable beliefs (as does PBVI). Unfortunately, policy search techniques can be hampered by low-gradient plateaus and poor local minima, and typically require the selection of a restricted policy class.

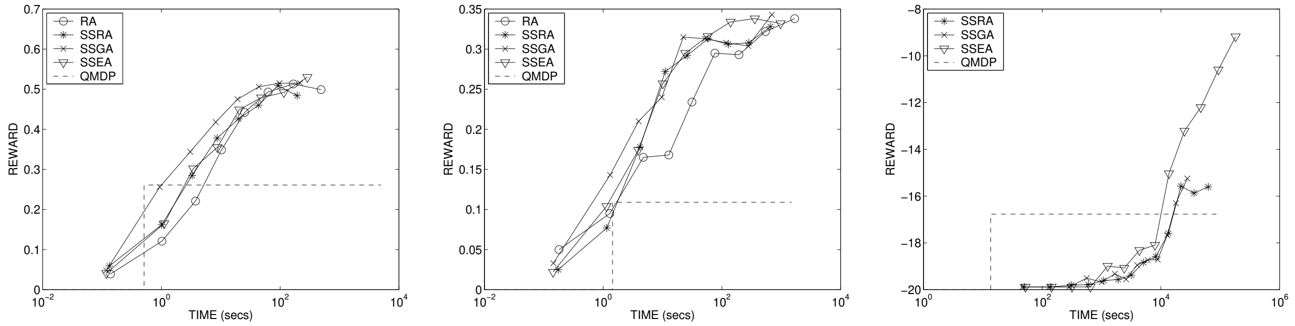


Figure 4: Belief expansion results for three problems: Hallway(left), Hallway2(center) and Tag(right)

7 Conclusion

This paper presents PBVI, a scalable anytime algorithm for approximately solving POMDPs. We applied PBVI to a robotic version of lasertag, where it successfully developed a policy for capturing a moving opponent. Other POMDP solvers had trouble computing useful policies for this domain. PBVI also compared favorably with other solvers on three well-known smaller test problems. We attribute PBVI's success to two features, both of which directly target the curse of history. First, by using a trajectory-based approach to select belief points, PBVI focuses planning on reachable beliefs. Second, because it uses a fixed set of belief points, it can perform fast value backups.

In experiments, PBVI beats back the curse of history far enough that we can solve POMDPs an order of magnitude larger than most previous algorithms. With this success, we can now identify the next hurdle for POMDP research: contrary to our expectation, it turns out to be the old-fashioned MDP problem of having too many distinct physical states. This problem hits us in the cost of updating the point-based value function vectors. (This cost is quadratic in the number of physical states.) While this problem is not necessarily easy to overcome, we believe that sparse matrix computations, together with other approaches from the existing literature [Poupart and Boutilier, 2003; Roy and Gordon, 2003], will allow us to to scale PBVI to problems which are at least another order of magnitude larger. So, PBVI represents a considerable step towards making POMDPs usable for real-world problems.

References

[Baxter and Bartlett, 2000] J. Baxter and P. L. Bartlett. Reinforcement learning on POMDPs via direct gradient ascent. In *ICML*, 2000.

[Bonet, 2002] B. Bonet. An e-optimal grid-based algorithm for partially obserable Markov decision processes. In *ICML*, 2002.

[Brafman, 1997] R. I. Brafman. A heuristic variable grid solution method for POMDPs. In *AAAI*, 1997.

[Cassandra *et al.*, 1997] A. Cassandra, M. Littman, and N. Zhang. Incremental pruning: A simple, fast, exact method for partially observable Markov decision processes. In *UAI*, 1997.

[Cassandra, 1999] A. Cassandra. Tony's POMDP page. http://www.cs.brown.edu/research/ai/pomdp/code/index.html, 1999.

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

[Kaelbling *et al.*, 1998] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134, 1998.

[Kearns *et al.*, 1999] M. Kearns, Y. Mansour, and A. Y. Ng. Approximate planning in large POMDPs via reusable trajectories. *NIPS 12*, 1999.

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

[Lovejoy, 1991] W. S. Lovejoy. Computationally feasible bounds for partially observed Markov decision processes. *Operations Research*, 39(1):162–175, 1991.

[Ng and Jordan, 2000] A. Y. Ng and M. Jordan. PEGASUS: A policy search method for large MDPs and POMDPs. In *UAI*, 2000.

[Poon, 2001] K.-M. Poon. A fast heuristic algorithm for decision-theoretic planning. Master's thesis, The Hong-Kong University of Science and Technology, 2001.

[Poupart and Boutilier, 2003] P. Poupart and C. Boutilier. Value-directed compression of POMDPs. In *NIPS 15*, 2003.

[Rosencrantz *et al.*, 2003] M. Rosencrantz, G. Gordon, and S. Thrun. Locating moving entities in dynamic indoor environments with teams of mobile robots. In *AAMAS*, 2003.

[Roy and Gordon, 2003] N. Roy and G. Gordon. Exponential family PCA for belief compression in POMDPs. In *NIPS 15*, 2003.

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

[Thrun, 1992] S. Thrun. *Handbook for Intelligent Control: Neural, Fuzzy and Adaptive Approaches*, chapter The Role of Exploration in Learning Control. Van Nostrand Reinhold, 1992.

[Zhang and Zhang, 2001] N. L. Zhang and W. Zhang. Speeding up the convergence of value iteration in partially observable Markov decision processes. *Journal of Artificial Intelligence Research*, 14:29–51, 2001.

[Zhou and Hansen, 2001] R. Zhou and E. A. Hansen. An improved grid-based approximation algorithm for POMDPs. In *IJCAI*, 2001.