# Towards Efficient Computation of Error Bounded Solutions in POMDPs: Expected Value Approximation and Dynamic Disjunctive Beliefs

**Pradeep Varakantham[†], Rajiv T. Maheswaran[‡], Tapana Gupta[†] and Milind Tambe[†]**
[†] {varakant, tapanagu, tambe}@usc.edu, Institute for Robotic and Intelligent Systems, University of Southern California
[‡] {maheswar}@isi.edu, Information Sciences Institute, University of Southern California

## Abstract

While POMDPs (partially observable markov decision problems) are a popular computational model with wide-ranging applications, the computational cost for optimal policy generation is prohibitive. Researchers are investigating ever-more efficient algorithms, yet many applications demand such algorithms bound any loss in policy quality when chasing efficiency. To address this challenge, we present two new techniques. The first approximates in the value space to obtain solutions efficiently for a pre-specified error bound. Unlike existing techniques, our technique guarantees the resulting policy will meet this bound. Furthermore, it does not require costly computations to determine the quality loss of the policy. Our second technique prunes large tracts of belief space that are unreachable, allowing faster policy computation without any sacrifice in optimality. The combination of the two techniques, which are complementary to existing optimal policy generation algorithms, provides solutions with tight error bounds efficiently in domains where competing algorithms fail to provide such tight bounds.[1]

## 1 Introduction

Partially observable markov decision problems (POMDPs) are a popular model with potential applications in health care, disaster rescue and space exploration [Hauskrecht and Fraser, 2000; Pineau *et al.*, 2003; Poupart and Boutilier, 2004; Nair *et al.*, 2003]. Unfortunately, the computational cost for optimal policy generation in POMDPs is prohibitive, requiring increasingly efficient algorithms to address problems in these large-scale domains. Furthermore, as we apply POMDPs in domains where property and human safety are of concern, e.g. disaster rescue, patient care and space exploration, error bounds (limiting POMDP policy quality loss) become critical.

[1]This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA), through the Department of the Interior, NBC, Acquisition Services Division, under Contract No. NBCHD030010.

Previous work in POMDPs has made encouraging progress using two approaches: (1) Exact algorithms compute the optimal solution and thus avoid problems with error bounds [Cassandra *et al.*, 1997; Feng and Zilberstein, 2004; 2005], however, they do not scale. (2) Approximate algorithms, a currently popular approach, address the significant computational complexity in POMDP policy generation by sacrificing solution quality for speed [Pineau *et al.*, 2003; Smith and Simmons, 2005; Poupart and Boutilier, 2004]. While some approximate algorithms (point-based approaches [Pineau *et al.*, 2003; Smith and Simmons, 2005]) provide expressions for error bounds, they have many drawbacks: (a) The bounds in these point-based approximation algorithms are based on maximum and minimum possible single-stage reward, which can take extreme values, leading to a very loose bound which is not useful in many domains, especially in those that have penalties (e.g., $R_{min} \ll 0$). (b) The computation of the bound for a particular policy is itself a potentially significant expense (e.g., requiring a non-linear program). (c) The algorithms cannot guarantee that they will yield a policy that can achieve a pre-specified error bound.

To address these challenges, we present two new techniques that are complementary to existing optimal policy generation algorithms. In particular, both are presented as wrappers to Generalized Incremental Pruning (GIP) [Cassandra *et al.*, 1997] and Region-Based Incremental Pruning (RBIP) [Feng and Zilberstein, 2004; 2005]. Our first technique EVA (Expected Value Approximation) is an approximate one that sacrifices quality for speed, but operates within a pre-specified error bound. EVA differs from existing approaches (primarily point-based approaches [Pineau *et al.*, 2003; Smith and Simmons, 2005]), by approximating directly in the value space as opposed to sampling in the belief space for an indirect value approximation. EVA alleviates the three key problems mentioned earlier: (a) EVA provides an error bound that does not depend on the maximum and minimum possible reward. (b) The computation of the error bound in itself imposes no overhead when computing policies in EVA. (c) Given any pre-specified error bound, EVA will generate a policy whose quality meets or exceeds that bound.

While results from EVA are promising, in domains that require tight error bounds, EVA alone may not be sufficiently fast. Our second technique, DDB (Dynamic Disjunctive Beliefs), complements EVA. DDB exploits domain structure to

prune large tracts of belief space that are unreachable. Because DDB does not sacrifice quality, combining it with EVA yields the same error bounds as the original EVA. We illustrate how DDB combined with EVA provides significant performance improvements over EVA alone, and over earlier techniques that prune belief spaces, such as [Varakantham *et al.*, 2005].

## 2 Background : POMDPs

A POMDP can be represented using the tuple $\langle S, A, P, \Omega, O, R\rangle$, where $S$ is the set of states; $A$ is the set of actions; $\Omega$ is the set of observations; $P(s, a, s')$ is the transition function; $O(s', a, \omega)$ denotes the observation model; $R(s, a)$ is the reward function. As the state is partially observable, POMDP policies are expressed as belief-state-to-action mappings, where a *belief state* $b$ is a probability distribution over the set of states $S$. The goal is to compute the policy with maximum expected reward.

Currently, the most efficient exact algorithms for POMDPs are value-iteration algorithms, specifically GIP [Cassandra *et al.*, 1997] and RBIP [Feng and Zilberstein, 2004]. These are dynamic-programming algorithms, where at each iteration the value function and optimal policies are represented by a minimal set of vectors called the *parsimonious set*. Determining the parsimonious set involves pruning out dominated vectors. This requires solving a linear program (LP) for all vectors in the original set. Thus, pruning is recognized to be computationally expensive.

**DB** : The performance of these dynamic programming algorithms can be improved considerably by exploiting the dynamics of the domain [Varakantham *et al.*, 2005]. Dynamic Beliefs, DB (implemented as a wrapper to GIP) eliminates unreachable belief spaces, given information about the starting belief space, $B_0$. It computes belief bounds over states at future time steps by solving maximization and minimization problems on the equation for belief probability updates. Given an action $a$ and observation $\omega$, the maximization problem for belief probability of state $s_t$ is:

$$\max_{b_{t-1}\in B_{t-1}} \frac{O_t(s_t, a, \omega)\Sigma_{s_{t-1}} P_{t-1}(s_{t-1}, a, s_t) b_{t-1}(s_{t-1})}{\sum_{s_t} O_t(s_t, a, \omega)\Sigma_{s_{t-1}} P_{t-1}(s_{t-1}, a, s_t) b_{t-1}(s_{t-1})}$$

Lagrangian techniques are used to solve these maximization problems efficiently in polynomial time. These bounds can significantly improve LP run-times, as the pruning via LPs happens over a restricted belief space. The restricted belief space also leads to fewer vectors in the parsimonious set for a particular interation. This cascades through the steps of the dynamic programming leading to fewer LP calls and consequently, significant speedups. DB can lead to orders of magnitude improvement over existing algorithms [Varakantham *et al.*, 2005].

**Point-Based Approaches** : Two leading approximate algorithms for POMDPs that provide error bounds are Point-Based Value Iteration (PBVI) [Pineau *et al.*, 2003] and Heuristic-Search Value Iteration (HSVI2) [Smith and Simmons, 2005]. In these algorithms, a policy computed for a sampled set of belief points is extrapolated to the entire belief space. PBVI/HSVI2 are anytime algorithms where the sampled set of belief points is expanded over time. The expansion ensures that the belief points are uniformly distributed over the entire belief space, and the heuristics employed for belief set expansion differentiate PBVI and HSVI2. We compare EVA against PBVI, as it is known to be one of the fastest approximate algorithms. The error bound provided by PBVI is: $(R_{max} - R_{min}) * \epsilon_b/(1-\gamma)^2$, with $\epsilon_b$ = $max_{b'\in\Delta} min_{b\in\mathcal{B}} \|b - b'\|_1$, where $\Delta$ is the entire belief space; $\mathcal{B}$ is the set of expanded belief points; $\gamma$ is the discount factor; $R_{max}$ and $R_{min}$ are the extremal single stage rewards. Computing $\epsilon_b$ requires solving a Non-Linear Program, NLP[2](shown in Algorithm 1). While we provide experimental comparisons with only PBVI, the drawbacks of PBVI with regard to calculating and achieving bounds are equally applicable to HSVI2 due to their similarity.

**Algorithm 1** Non-Linear Program to obtain $\epsilon_b$

**Maximize** $\epsilon_b$
*subject to the constraints*
$\Sigma_{1\le i\le|S|} b[i] = 1$
$b[i] \ge 0$ and $b[i] \le 1, \forall i \in \{1, \ldots, |S|\}$
$\epsilon_b < \Sigma_{1\le i\le|S|} \left|b[i] - b_k[i]\right|, \forall b_k \in \mathcal{B}$

## 3 Expected Value Approximation (EVA)

The value function in a POMDP is piecewise linear and can be expressed by a set of vectors. Approximate algorithms generate fewer vector sets than the optimal algorithms. Existing approaches generate these reduced vector sets by sampling the belief space and finding the vectors that apply only at these points. In our approach, Expected Value Approximation (EVA), we choose a reduced set of vectors by approximating the value space with a subset of vectors whose expected reward will be within a desired bound of the optimal reward.

Using an approximation (subset) of the optimal parsimonious set will lead to lower expected quality at some set of belief points. Let $\epsilon$ denote the maximum loss in quality we will allow at any belief point. We henceforth refer to any vector set that is at most $\epsilon$ away from the optimal value at all points in the belief space (as illustated in Fig 1) as an $\epsilon$-parsimonious set. The key problem in EVA is to determine this $\epsilon$-parsimonious set efficiently.

To that end, we employ a heuristic that extends the pruning strategies presented in GIP. In GIP, a parsimonious set $\mathcal{V}$ corresponding to a set of vectors, $\mathcal{U}$ is obtained in three steps:

1. Initialize the parsimonious set $\mathcal{V}$ with the dominant vectors at the simplex points.
2. For some chosen vector $u \in \mathcal{U}$, execute a LP to compute the belief point $b$ where $u$ dominates the current parsimonious set $\mathcal{V}$.
3. Compute the vector $u'$ with highest expected value in the set $\mathcal{U}$ at the belief point, $b$; remove vector $u'$ from $\mathcal{U}$ and add it to $\mathcal{V}$.

[2]An LP formulation with non-convex feasible region is possible, but it will have O($2^{|S|}$) number of linear constraints. The smallest of our problems has 60 states ($\sim 10^{18}$ constraints), making it infeasible to solve such an LP.

EVA modifies the first two steps, to obtain the $\epsilon$-parsimonious set:

1. Since we are interested in representing the optimal parsimonious set with as few vectors as possible, the initialization process only selects one vector over the beliefs in the simplex extrema. We choose a vector with the highest expected value at the most number of simplex belief points, choosing randomly to break ties.
2. The LP is modified to check for $\epsilon$-dominance, i.e., dominating all other vectors by $\epsilon$ at some belief point. Algorithm 2 provides a modified LP with $b_t^{max}$ and $b_t^{min}$.

**Algorithm 2** LP-DOMINATE($w, U, b_t^{max}, b_t^{min}, \epsilon$)

1: **variables:** $d, b(s_t)\ \forall s_t \in S_t$
2: **maximize** $d$
3: **subject to the constraints**
4: $\quad b \cdot (w - u) \geq d + \epsilon, \forall u \in U$
5: $\quad \Sigma_{s_t \in S_t} b(s_t) = 1, b(s_t) \leq b_t^{max}(s_t), b(s_t) \geq b_t^{min}(s_t)$
6: **if** $d \geq 0$ return $b$ **else** return nil

The key difference between the LP used in GIP and the one in Algorithm 2 is the $\epsilon$ in RHS of line 4 which checks for expected value dominance of the given vector $w$ over a vector $u \in U$. Including $\epsilon$ as part of the RHS constrains $w$ to dominate other vectors by at least $\epsilon$. In the following propositions, we prove the correctness of the EVA algorithm and the error bound provided by EVA. Let $\mathcal{V}^\epsilon$ and $\mathcal{V}^*$ denote the $\epsilon$-parsimonious and optimal parsimonious set, respectively.

**Proposition 1** *$\forall b \in \Delta$, the entire belief space, if $v_b^\epsilon = \arg\max_{v^\epsilon \in \mathcal{V}^\epsilon} v^\epsilon \cdot b$ and $v_b^* = \arg\max_{v^* \in \mathcal{V}^*} v^* \cdot b$, then $v_b^\epsilon \cdot b + \epsilon \geq v_b^* \cdot b$.*

**Proof.** We prove this by contradiction. Assume $\exists b \in \Delta$ such that $v_b^\epsilon \cdot b + \epsilon < v_b^* \cdot b$. This implies $v_b^\epsilon \neq v_b^*$, and $v_b^* \notin \mathcal{V}^\epsilon$. We now consider the situation(s) when $v_b^*$ is considered by EVA. At these instants, there will be a current parsimonious set $\mathcal{V}$ and a set of vectors still to be considered $\mathcal{U}$. Let

$$\hat{b} = \arg\max_{\tilde{b} \in \Delta} \{\min_{v \in \mathcal{V}} (v_b^* \cdot \tilde{b} - v \cdot \tilde{b})\}$$

be the belief point at which $v_b^*$ is best w.r.t. $\mathcal{V}$. Let

$$\hat{v}_{\hat{b}} = \arg\max_{v \in \mathcal{V}} v \cdot \hat{b}$$

be the vector in $\mathcal{V}$ which is best at $\hat{b}$. Let

$$\hat{u}_{\hat{b}} = \arg\max_{u \in \mathcal{U}} u \cdot \hat{b}$$

be the vector in $\mathcal{U}$ which is best at $\hat{b}$. There are three possibilities:

1. $v_b^* \cdot \hat{b} < \hat{v}_{\hat{b}} \cdot \hat{b} + \epsilon$: This implies $v_b^* \cdot \hat{b} - \hat{v}_{\hat{b}} \cdot \hat{b} < \epsilon$. By the definition of $\hat{b}$, we have
$$v_b^* \cdot b - \hat{v}_b \cdot b < v_b^* \cdot \hat{b} - \hat{v}_{\hat{b}} \cdot \hat{b} < \epsilon$$
where $\hat{v}_b = \arg\max_{v \in \mathcal{V}} v \cdot b$. This implies
$$v_b^* \cdot b < \hat{v}_b \cdot b + \epsilon \leq v_b^\epsilon \cdot b + \epsilon,$$
which is a contradiction.
2. $v_b^* \cdot \hat{b} \geq \hat{v}_{\hat{b}} \cdot \hat{b} + \epsilon$ and $v_b^* \cdot \hat{b} \geq \hat{u} \cdot \hat{b}$: This means $v_b^*$ would have been included in the $\epsilon$-parsimonious set, $v_b^* \in \mathcal{V}^\epsilon$, which is a contradiction.
3. $v_b^* \cdot \hat{b} \geq \hat{v}_{\hat{b}} \cdot \hat{b} + \epsilon$ and $v_b^* \cdot \hat{b} < \hat{u} \cdot \hat{b}$: $\hat{u}$ will be included in $\mathcal{V}$ and $v_b^*$ is returned to $\mathcal{U}$ to be considered again until one of previous two terminal conditions occur. ■

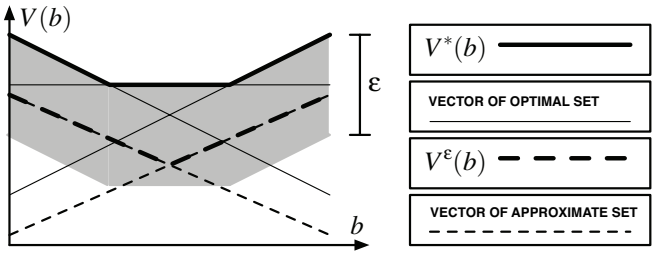


Figure 1: EVA: An example of an $\epsilon$-parsimonious set

**Proposition 2** *The error introduced by EVA at each stage of the policy computation, is bounded by $2\epsilon|\Omega|$ for GIP-type cross-sum pruning.*

**Proof**. The EVA algorithm introduces an error of $\epsilon$ in a parsimonious set whenever a pruning operation (PRUNE) is performed, due to Proposition 1. In GIP, there are three pruning steps at each stage of policy computation.

1. $\mathcal{V}^{a,o} = PRUNE(\mathcal{V}^{a,o,i})$: After this step, each $\mathcal{V}^{a,o}$ is at most $\epsilon$ away from optimal $\forall a, \forall o$.
2. $\mathcal{V}^a = PRUNE(\cdots(PRUNE(\mathcal{V}^{a,o_1} \oplus \mathcal{V}^{a,o_2}) \cdots \oplus \mathcal{V}^{a,o_{|\Omega|}})$: We begin with $\mathcal{V}^{a,o_1}$ which is away from optimal by at most $\epsilon$. Each pruning operation adds $2\epsilon$ to the bound ($\epsilon$ for the new term $\mathcal{V}^{a,o_i}$ and $\epsilon$ for the $PRUNE$). There are $|\Omega| - 1$ prune operations. Thus, each $\mathcal{V}^{a,o}$ is away from the optimal by at most $2\epsilon(|\Omega| - 1) + \epsilon$.
3. $\mathcal{V}' = PRUNE(\bigcup_{a \in A} \mathcal{V}^a)$: The error of $\bigcup_{a \in A} \mathcal{V}^a$ is bounded by the error of $\mathcal{V}^a$. The $PRUNE$ adds $\epsilon$, leading to a total one-stage error bound of $2\epsilon|\Omega|$. ■

**Proposition 3** *The total error introduced by EVA (for GIP-type cross-sum pruning) is bounded by $2\epsilon|\Omega|T$ for a $T$-horizon problem.*

**Proof**. Let $V_t^\epsilon(b)$ and $V_t^*(b)$ denote the EVA-policy and optimal value function, respectively, at time $t$. If $A_t^\epsilon \subseteq A$, is the set of actions at the roots of all policy-trees associated with $\mathcal{V}_t^\epsilon$, the EVA vector set for time $t$ and $e_t = \max_{b \in B}\{V_t^*(b) - V_t^\epsilon(b)\}$, then, $V_{t-1}^\epsilon(b) =$

$$\begin{aligned}
&= max_{a \in A_t^\epsilon}\{R(b,a) + \Sigma_{b' \in B} P(b'|b,a) V_t^\epsilon(b')\} \\
&\geq max_{a \in A_t^\epsilon}\{R(b,a) + \Sigma_{b' \in B} P(b'|b,a) V_t^*(b')\} \\
&\quad - \Sigma_{b' \in B} P(b'|b,a) e_t \\
&= max_{a \in A_t^\epsilon}\{R(b,a) + \Sigma_{b' \in B} P(b'|b,a) V_t^*(b')\} - e_t \\
&\geq max_{a \in A}\{R(b,a) + \Sigma_{b' \in B} P(b'|b,a) V_t^*(b')\} - 2\epsilon|\Omega| - e_t \\
&= V_{t-1}^*(b) - 2\epsilon|\Omega| - e_t
\end{aligned}$$

The last inequality is due to Proposition 2, and the other relations are by definition. The above implies that $e_{t-1} = e_t + 2\epsilon|\Omega|$ and with $e_T = 2\epsilon|\Omega|$, we have the error for the EVA-policy at the beginning of execution, $e_1 = 2\epsilon|\Omega|T$ (the total error bound for EVA). ■

Similarly, it can be proved that for $\gamma$-discounted infinite horizon problems, the total error bound is $\frac{2\epsilon|\Omega|}{1-\gamma}$.

## 4 Dynamic Disjunctive Beliefs (DDB)

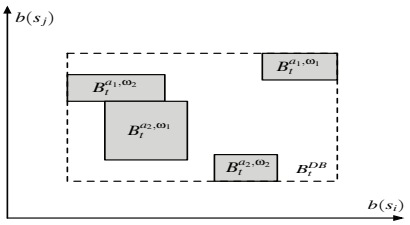


Figure 2: DDB: An example of DDB vs. DB

By eliminating reasoning about optimal policies at unreachable beliefs, it is possible to improve policy computation time greatly without sacrificing any solution quality. Our key insight is that the reachable belief space is composed of small belief polytopes; hence we introduce an efficient technique for exploiting these smaller polytopes. This contrasts with the previous work on Dynamic Beliefs, DB by [Varakantham *et al.*, 2005], which assumed a single large belief polytope. In particular, DB obtains the maximum belief probability for a state given an action and observation, $b_t^{a,\omega,\max}(s)$ by solving a constrained optimization problem using Lagrangian techniques. Similarly minimum possible belief probability is also computed. The dynamic belief polytope at epoch $t$ was created as follows:

1. Find the maximum and minimum possible belief for each state over all actions and observations: $b_t^{\max}(s) = \max_{a,\omega} b_t^{a,\omega,\max}(s)$, $b_t^{\min}(s) = \max_{a,\omega} b_t^{a,\omega,\min}(s)$.
2. Create a belief polytope that combines these bounds over all states: $B_t^{DB} =$

$$[b_t^{\min}(s_1)\ b_t^{\max}(s_1)] \times \cdots \times [b_t^{\min}(s_{|S|})\ b_t^{\min}(s_{|S|})].$$

While this is an appropriate bound in that any belief outside $B_t^{DB}$ is not possible at time $t$, in many domains there is not a single belief polytope, but a collection of smaller belief polytopes - because future beliefs depend on particular action-observation pairs. Unfortunately, DB insists on a single belief polytope, thus including many unreachable beliefs. Figure 2 illustrates this for a two-observation two-action system over a two-dimensional support. While $B_t^{DB}$ is smaller than the entire space of potential beliefs, it is larger than necessary; it is not possible to believe anything outside of $\cup_{i=1,2;j=1,2} B_t^{a_i,\omega_j}$. In contrast, our new method for expressing feasible beliefs is referred to as Dynamic Disjunctive Belief (DDB) bounds. The DDB method for computing feasible belief spaces is to eliminate conditioning on action and observations (eliminate taking max over $a, \omega$ in (1) above):

1. Obtain $b_t^{a,\omega,\max}(s)$ and $b_t^{a,\omega,\min}(s)$, $\forall a \in A, \omega \in \Omega$
2. Create multiple belief polytopes, one for each action-observation pair: $B_t^{a_i,\omega_i} =$

$$[b_t^{a_i,\omega_i,\min}(s_1)\ b_t^{a_i,\omega_i,\max}(s_1)] \times \cdots$$
$$\cdots \times [b_t^{a_i,\omega_i,\min}(s_{|S|})\ b_t^{a_i,\omega_i,\max}(s_{|S|})].$$

The feasible belief space is $B_t^{DDB} = \cup_{a,\omega} B_t^{a,\omega}$. However, this is disjunctive and cannot be expressed in the LP that is used for pruning. Instead, the prune occurs over each $B_t^{a,\omega}$ and we take the union of dominant vectors for these belief spaces. This increases the number of LP calls for a fixed epoch but the LPs cover smaller spaces and will yield fewer vectors at the end. The cascading effects of the reduced vector sets (as in EVA) over the dynamic programming process can lead to fewer policies at the next stage, resulting in significant computational savings.

## 5 Experiments

Our experiments[3] show how EVA and DDB address key shortcomings of current POMDP algorithms when a policy that meets a pre-specified error bound is desired. Three key results provided in this section relevant to EVA are: (1) EVA can guarantee the generation of a policy that meets any pre-specified error bound where point-based algorithms cannot (Section 5.1). (2) The cost of calculating the error bound of a policy, which EVA avoids, can significantly impact performance of point based approaches(Section 5.2). (3) There are domains, where point-based algorithms cannot provide functional bounds (Section 5.3). For DDB, we show an order-of-magnitude speedup over the previously best technique for belief space pruning (Section 5.4).

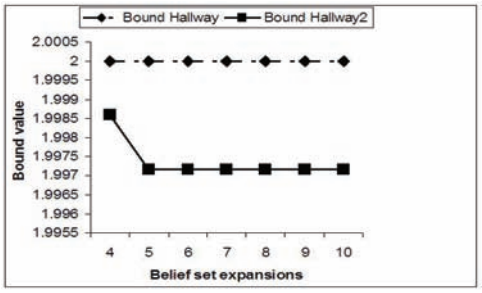


Figure 3: Error bounds for Hallway and Hallway2

### 5.1 Bound Achievability

In the general case, point-based approaches can always encounter problems where they cannot generate a policy that meets a pre-specified error bound. If the span of belief support, *chosen independently from the value function*, is smaller

[3]Hallway, Hallway2, Tiger-Grid, and Aircraft are benchmark problems available from *Anthony Cassandra's POMDP website*. Scotland Yard is a 216-state, 16-act, 6-obs problem motivated by the Ravensburger Scotland Yard game. Tag is a 870-state, 5-act, 30-obs problem from [Pineau *et al.*, 2003].

than the feasible belief space, there exists a value function for which the algorithm will have a minimum error, i.e., the solution quality cannot exceed some threshold. If the pre-specified error bound is smaller than this minimum error, the algorithm will never be able to provide a satisfactory policy. Current point-based approaches exacerbate this problem by (a) starting with one belief point and (b) only considering points that are reachable from this initial point. Thus, they potentially restrict the span of their belief support severely which implies a significant minimum error. Figure 3 presents the PBVI error bound for the Hallway and Hallway2 problems as the belief set is expanded over time. The $x$-axis represents the number of belief set expansions, and the $y$-axis represents the error bound (by solving an NLP for each belief set expansion). The error bound remains constant for both problems after the fifth expansion (around 30 belief points in the set). After the tenth expansion (around 1000 belief points) the NLP becomes too expensive to calculate. Thus, for instance, for Hallway2, it is not verifiable if PBVI would be able to reach an error bound of 1.8. However with EVA, by using an $\epsilon$ (in Algorithm 2) corresponding to the pre-specified error bound (obtained by equating error bound to $2 * \epsilon * |\Omega| * T$), any error bound is achievable.

## 5.2 Bound Calculation Cost

Now let us consider the case where a given point-based approach can generate a policy whose quality is within a pre-specified error bound. These algorithms still do not know the quality loss in their policy as they expand their belief support. As indicated earlier, they require an expensive NLP to calculate the error bound [4]. We ran both EVA and PBVI for the Tiger-Grid, Hallway, Hallway2, Aircraft, Scotland Yard and Tag problems. As PBVI depends on the starting belief point, for each problem we obtained the bound for 10 randomly selected initial belief points. For each given initial belief point, we ran PBVI for a fixed number of belief set expansions and computed the bound of the resulting policy with an NLP. The deviations in the obtained bounds were extremely small. This gives an advantage to PBVI by assuming that it has the correct belief support for the obtained bound immediately. Additionally, we only counted the average run-time of the final expansion.

We then ran EVA with the equivalent error bound, forcing EVA to obtain a policy as good or better (bound calculation for EVA is trivial, i.e., constant-time computation). Furthermore, EVA was also run with a tighter bound (half the value obtained for PBVI), to address the possibility that an alternate belief expansion heuristic in PBVI led to a better bound. The results are shown in Figure 4. The NLP bar denotes the time taken for computing the error bound in PBVI, while the PBVI bar shows the run-time for computing the policy corresponding to the final expansion (not including the NLP run-time). The bars for EVA (same bound) and EVA (half bound)indicate the time taken by EVA when provided with a bound equal to and half of that used by PBVI respectively.

[4]We employed an industrial optimization software called Lingo 8.0 for solving NLPs. Machine specs for all experiments: Intel Xeon 3.6 GHZ processor, 2GB RAM

We note that: (1) The same-bound EVA run-times are less than NLP run-times, i.e. just the bound computation ovehead of PBVI is a significant burden. (2) Even without the NLP, EVA run-times are better than PBVI run-times in all cases. (3) The half-bound EVA run-times are better than NLP run-times in all but one case. We see that a hampered EVA outperformed an ameliorated PBVI. HSVI2 [Smith and Simmons, 2005] outperforms PBVI in run-time but suffers from the same problems i.e. the time to compute the bound and the unattainability of the bound. Beyond the experimental results, point-based approaches will suffer as their belief supports get larger because the NLP cost increases exponentially.

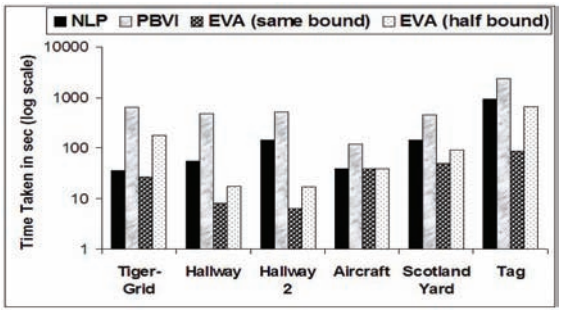


Figure 4: Run time comparison of EVA and PBVI

## 5.3 Bound Functionality

Even if the NLP cost vanished, there are domains where the expression for error bound in point-based approaches yields a value that is not functional. Consider the Task Management Problem (TMP) [Varakantham *et al.*, 2005] in software personal assitant domains. There is a penalty ($R_{min} \ll 0$) associated with certain state-action pairs to control agent autonomy. In this case, the output of the NLP yields an error bound so loose with respect to the actual solution quality that it renders the bound useless. There is no method for a point-based approach to determine when to stop expanding the set of belief sets in TMPs.

## 5.4 DDB

As shown in Figure 5, the TMP domain also illustrates the benefits of DDB . GIP and RBIP did not terminate for TMP within the pre-specified time limit. With a sufficiently large bound, EVA produced a policy within our time limits. To improve performance, one option was to use EVA with a larger bound, but this would have meant an additional sacrifice in solution quality. EVA+DB reduced run-time without sacrificing solution quality. We see that EVA+DDB (combining our two new techniques) provided an *order-of-magnitude* speedup beyond EVA+DB, also without losing solution quality. Runtimes for EVA+DDB and EVA+DB included the overhead for computation of the belief bounds.

## 5.5 Experiment Summary

We demonstrated the drawbacks of current point-based approaches when tasked with generating policies with pre-specified error bounds. These limitations include the inability

to guarantee a policy within the bound, an expensive cost to compute the bound even if achievable, and a non-functional expression of the bound even if computable. We also show that the run-times of EVA are not only shorter than the run-times of PBVI but also to the NLP calculation run-times alone. Finally, we exemplify the significant benefit of DDB in domains with structure in belief evolution as it most tightly bounds the dynamic belief polytope.

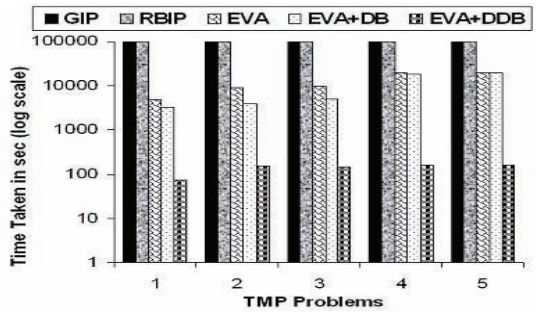


Figure 5: Comparison of GIP, RBIP EVA, EVA+DB, EVA+DDB

## 6 Related Work

As mentioned earlier, techniques for solving POMDPs can be categorized as exact and approximate. GIP [Cassandra *et al.*, 1997] and RBIP [Feng and Zilberstein, 2004; 2005] belong to the class of exact algorithms that provide optimal policies at the expense of enormous computational complexity. DB is a wrapper that exploits the dynamics of the domain to provide improved performance.

There has been much work in the area of approximate algorithms which are faster than exact algorithms at the cost of solution quality. Point-based [Smith and Simmons, 2005; Pineau *et al.*, 2003], policy-search [Braziunas and Boutilier, 2004; Poupart and Boutilier, 2004; Menleau *et al.*, 1999], and grid [Hauskrecht, 2000; Zhou and Hansen, 2001] approaches dominate other algorithms. Point-based approaches are discussed in Section 2. Policy-search approaches typically employ a finite state controller (FSC) to represent the policy. The FSC is updated until convergence to a stable controller. Grid-based methods are similar to point-based approaches. The difference is maintaining "values" at belief points, as opposed to "value gradients" in point-based techniques. Another approach that scales well is the dimensionality-reduction technique in [Roy and Gordon, 2002] which applies E-PCA (an improvement to Principal Component Analysis) on a set of belief vectors to obtain a low-dimensional representation of the original state space. Though all these algorithms significantly improve the run-time performance, they provide loose or no quality guarantees on the solutions.

## 7 Summary

Despite the wide-ranging applicability of POMDPs, obtaining optimal policies are often intractable and current approximate approaches do not sufficiently address solution quality. Indeed, many domains demand tight error bounds on policies generated in POMDPs. To that end, we present two new techniques that are complementary to existing optimal policy generation algorithms: EVA and DDB. EVA is a new type of approximation technique that directly approximates in the value space. Due to this, it is able to obtain solutions efficiently for a pre-specified error bound. Furthermore, it does not require costly computations to compute the bound. DDB prunes out large tracts of belief space that are unreachable in many real-world problems, allowing faster policy computation without any sacrifice in optimality. The combination of the two techniques is shown to provide high quality solutions efficiently in domains with dynamic structure.

## References

[Braziunas and Boutilier, 2004] D. Braziunas and Craig Boutilier. Stochastic local search for POMDP controllers. In *AAAI*, 2004.

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

[Feng and Zilberstein, 2004] Z. Feng and S. Zilberstein. Region based incremental pruning for POMDPs. In *UAI*, 2004.

[Feng and Zilberstein, 2005] Z. Feng and S. Zilberstein. Efficient maximization in solving POMDPs. In *AAAI*, 2005.

[Hauskrecht and Fraser, 2000] M. Hauskrecht and H. Fraser. Planning treatment of ischemic heart disease with partially observable markov decision processes. *AI in Medicine*, 18:221–244, 2000.

[Hauskrecht, 2000] M. Hauskrecht. Value-function approximations for POMDPs. *JAIR*, 13:33–94, 2000.

[Menleau *et al.*, 1999] N. Menleau, K. E. Kim, L. P. Kaelbling, and A. R. Cassandra. Solving POMDPs by searching the space of finite policies. In *UAI*, 1999.

[Nair *et al.*, 2003] R. Nair, M. Tambe, and S. Marsella. Role allocation and reallocation in multiagent teams: Towards a practical analysis. In *AAMAS*, 2003.

[Pineau *et al.*, 2003] J. Pineau, G. Gordon, and S. Thrun. PBVI: An anytime algorithm for POMDPs. In *IJCAI*, 2003.

[Poupart and Boutilier, 2004] P. Poupart and C. Boutilier. Vdcbpi: an approximate scalable algorithm for large scale POMDPs. In *NIPS*, 2004.

[Roy and Gordon, 2002] N. Roy and G. Gordon. Exponential family PCA for belief compression in POMDPs. In *NIPS*, 2002.

[Smith and Simmons, 2005] T. Smith and R. Simmons. Point-based POMDP algorithms: Improved analysis and implementation. In *UAI*, 2005.

[Varakantham *et al.*, 2005] P. Varakantham, R. Maheswaran, and M. Tambe. Exploiting Belief Bounds: Practical POMDPs for Personal Assistant Agents. In *AAMAS*, 2005.

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