Compact, Convex Upper Bound Iteration for Approximate POMDP Planning

Tao Wang
University of Alberta
trysi@cs.ualberta.ca

Pascal Poupart
University of Waterloo
ppoupart@cs.uwaterloo.ca

Michael Bowling and **Dale Schuurmans**
University of Alberta
{bowling,dale}@cs.ualberta.ca

Abstract

Partially observable Markov decision processes (POMDPs) are an intuitive and general way to model sequential decision making problems under uncertainty. Unfortunately, even approximate planning in POMDPs is known to be hard, and developing heuristic planners that can deliver reasonable results in practice has proved to be a significant challenge. In this paper, we present a new approach to approximate value-iteration for POMDP planning that is based on quadratic rather than piecewise linear function approximators. Specifically, we approximate the optimal value function by a convex upper bound composed of a fixed number of quadratics, and optimize it at each stage by semidefinite programming. We demonstrate that our approach can achieve competitive approximation quality to current techniques while still maintaining a bounded size representation of the function approximator. Moreover, an upper bound on the optimal value function can be preserved if required. Overall, the technique requires computation time and space that is only linear in the number of iterations (horizon time).

Introduction

Partially observable Markov decision processes (POMDPs) are a general model of an agent acting in an environment, where the effects of the agent's actions and the observations it can make about the current state of the environment are both subject to uncertainty. The agent's goals are specified by rewards it receives (as a function of the states it visits and actions it executes), and an optimal behavior strategy in this context chooses actions, based on the history of observations, that maximizes the long term reward of the agent.

POMDPs have become an important modeling formalisms in robotics and autonomous agent design (Thrun, Burgard, & Fox 2005; Pineau *et al.* 2003). Much of the current work on robot navigation and mapping, for example, is now based on stochastic transition and observation models (Thrun, Burgard, & Fox 2005; Roy, Gordon, & Thrun 2005). Moreover, POMDP representations have also been used to design autonomous agents for real world applications, including nursing (Pineau *et al.* 2003) and elderly assistance (Boger *et al.* 2005).

Despite their convenience as a modeling framework however, POMDPs pose difficult computational problems. It is well known that solving for optimal behavior strategies or even just approximating optimal strategies in a POMDP is intractable (Madani, Hanks, & Condon 2003; Mundhenk *et al.* 2000). Therefore, a lot of work has focused on developing heuristics for computing reasonable behavior strategies for POMDPs. These approaches have generally followed three broad strategies: value function approximation (Hauskrecht 2000; Spaan & Vlassis 2005; Pineau, Gordon, & Thrun 2003; Parr & Russell 1995), policy based optimization (Ng & Jordan 2000; Poupart & Boutilier 2003; 2004; Amato, Bernstein, & Zilberstein 2006), and stochastic sampling (Kearns, Mansour, & Ng 2002; Thrun 2000). In this paper, we focus on the value function approximation approach and contribute a new perspective to this strategy.

Most previous work on value function approximation for POMDPs has focused on representations that explicitly maintain a set of α-vectors or belief states. This is motivated by the fact that the optimal value function, considered as a function of the belief state, is determined by the maximum of a set of linear functions—specified by α-vectors—where each α-vector is associated with a specific behavior policy. Since the optimal value function is given by the maximum of a (large) set of α-vectors, it is natural to consider approximating it by a subset of α-vectors, or at least a small set of linear functions. In fact, even an exact representation of the optimal value function need not keep every α-vector, but only those that are maximal for at least some "witness" belief state. Motivated by this characterization, most value function approximation strategies attempt to maintain a smaller subset of α-vectors by focusing on a reduced set of belief states (Spaan & Vlassis 2005; Hauskrecht 2000; Pineau, Gordon, & Thrun 2003). Although much recent progress has been made on α-vector based approximations, a drawback of this approach is that the number of α-vectors stored generally has to grow with the number of value iterations to maintain an adequate approximation (Pineau, Gordon, & Thrun 2003).

In this paper, we consider an alternative approach that drops the notion of an α-vector entirely from the approximation strategy. Instead we exploit the other fundamental observation about the nature of the optimal value function: since it is determined by a belief-state-wise maxi-

mum over linear functions, the optimal value function must be a convex function of the belief state (Sondik 1978; Boyd & Vandenberghe 2004). Our strategy, then, is to compute a convex approximation to the optimal value function that is based on quadratic rather than linear functions of the belief state. The advantage of using a quadratic basis for value function approximation is several-fold: First, the size of the representation does not have to grow merely to model an increasing number of facets in the optimal solution; thus we can keep a bounded size representation at each horizon. Second, a quadratic representation allows one to conveniently maintain a provable upper bound on the optimal values in an explicit compact representation without requiring auxiliary linear programming to be used to retrieve the bound, as in current grid based approaches (Hauskrecht 2000; Smith & Simmons 2005). Third, the computational cost of updating the approximation does not change with iteration number (either in time or space), so the overall computation time is only linear in the horizon. Finally, as we demonstrate below, despite a significant reduction in representation size, convex quadratics are still able to achieve competitive approximation quality on benchmark POMDP problems.

Background

We begin with Markov decision processes (MDP) since we will need to exploit some basic concepts from MDPs in our approach below. An MDP is defined by a set of states S, a set of actions A, a state transition model $p(s'|s,a)$, and a reward model $r(s,a)$. In this setting, a deterministic policy is specified by a function from states to actions, $\pi : S \rightarrow A$, and the *value function* for a policy is defined as the expected future discounted reward the policy obtains from each state

$$V^{\pi}(s) \quad = \quad \mathrm{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, \pi(s_t)) \;\middle|\; s_0 = s\right]$$

Here the discount factor, $0 \leq \gamma < 1$, expresses a tradeoff between short term and long term reward. It is known that there exists a deterministic optimal policy whose value function dominates all other policy values in every state (Bertsekas 1995). This optimal value function also satisfies the Bellman equation

$$V^*(s) \quad = \quad \max_a r(s,a) + \gamma \sum_{s'} p(s'|s,a) V^*(s') \quad (1)$$

Computing the optimal value function for a given MDP can be accomplished in several ways. The two ways we consider below are *value iteration* and *linear programming*. Value iteration is based on repeatedly applying the Bellman backup operator, $V_{n+1} = HV_n$, specified by

$$V_{n+1}(s) \quad = \quad \max_a r(s,a) + \gamma \sum_{s'} p(s'|s,a) V_n(s') \quad (2)$$

It can be shown that $V_n \rightarrow V^*$ in the L_∞ norm, and thus V^* is a fixed point of (2) (Bertsekas 1995). V^* is also the solution to the linear program

$$\min_V \sum_s V(s) \text{ s.t. } V(s) \geq r(s,a) + \gamma \sum_{s'} p(s'|s,a) V(s') \quad (3)$$

for all $s \in S$ and $a \in A$. It turns out that for *continuous* state spaces, the Bellman equation (1) still characterizes the optimal value function, replacing the transition probabilities with conditional densities and the sums with Lebesgue integrals. However, computationally, the situation is not so simple for continuous state spaces, since the integrals must now somehow be solved in place of the sums, and (3) is no longer finitely defined. Nevertheless, continuous state spaces are unavoidable when one considers POMDP planning.

POMDPs extend MDPs by introducing an observation model $p(o'|a,s')$ that governs how a noisy observation $o' \in O$ is related to the underlying state s' and the action a. Having access to only noisy observations of the state complicates the problem of choosing optimal actions significantly. The agent now never knows the exact state of the environment, but instead must infer a distribution over possible states, $b(s)$, from the history of observations and actions. Nevertheless, given an action a and observation o' the agent's *belief state* can be easily updated by Bayes rule

$$b'_{(b,a,o')}(s') \quad = \quad p(o'|a,s') \sum_s p(s'|s,a) b(s) / Z \quad (4)$$

where $Z = p(o'|b,a) = \sum_{s'} p(o'|a,s') \sum_s p(s'|s,a) b(s)$.

By the Markov assumption, the belief state is a sufficient representation upon which an optimal behavior strategy can be defined (Sondik 1978). Therefore, a policy is naturally specified in this setting by a function from belief states to actions, $\pi : B \rightarrow A$, where B is the set of all possible distributions over the underlying state space S (an $|S|-1$ dimensional simplex). Obviously for any environment with two or more states there are an infinite number of belief states, and not every policy can be finitely represented. Nevertheless, one can still define the value function of a policy as the expected future discounted reward obtained from each belief state

$$V^{\pi}(b) \quad = \quad \mathrm{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(b_t, \pi(b_t)) \;\middle|\; b_0 = b\right]$$

where $r(b,a) = \sum_s r(s,a) b(s)$. Thus, a POMDP can be treated as an MDP over belief states; that is, a continuous state MDP. As before, an optimal policy obtains the maximum value for each belief state, and its value function satisfies the Bellman equation (Sondik 1978)

$$\begin{aligned} V^*(b) \quad &= \quad \max_a r(b,a) + \gamma \sum_{b'} p(b'|b,a) V^*(b') \\ &= \quad \max_a r(b,a) + \gamma \sum_{o'} p(o'|b,a) V^*(b'_{(b,a,o')}) \quad (5) \end{aligned}$$

Unfortunately, solving the functional equation (5) for V^* is hard. Known techniques for computing the optimal value function are generally based on *value iteration* (Cassandra, Littman, & Zhang 1997; Zhang & Zhang 2001); although policy based approaches are also possible (Sondik 1978; Poupart & Boutilier 2003; 2004). As above, value iteration is based on repeatedly applying a Bellman backup operator, $V_{n+1} = HV_n$, to a current value function approximation. In this case, a current lower bound, V_n, is represented by a

finite set of α-vectors, $\Gamma_n = \{\alpha_{\pi'} : \pi' \in \Pi_n\}$, where each α-vector is associated with an n-step behavior strategy π'. Given Γ_n, the value function is represented by

$$V_n(b) = \max_{\alpha_{\pi'} \in \Gamma_n} b \cdot \alpha_{\pi'}$$

At each stage of value iteration, the current lower bound is updated according to the Bellman backup, $V_{n+1} = HV_n$, such that

$$V_{n+1}(b) = \max_a r(b,a) + \gamma \sum_{o'} p(o'|b,a) V_n(b'_{(b,a,o')}) \quad (6)$$

$$= \max_a b \cdot r_a + \gamma \sum_{o'} b \cdot \arg_{g_{(\pi',a,o')}} \max_{\pi' \in \Pi_n} b \cdot g_{(\pi',a,o')}$$

$$= \max_{a, \{o' \to \pi'\}} b \cdot \alpha_{a,\{o' \to \pi'\}} \quad (7)$$

where we use the quantities

$$g_{(\pi',a,o')}(s) = \sum_{s'} p(o'|a,s') p(s'|a,s) \alpha_{\pi'}(s')$$

$$\alpha_{a,\{o' \to \pi'\}} = r_a + \gamma \sum_{o'} g_{(\pi',a,o')}$$

Once again it is known that $V_n \to V^*$ in the L_∞ norm, and thus V^* is a fixed point of (6) (Sondik 1978).

Although the size of the representation for V_{n+1} remains finite, it can be exponentially larger than V_n in the worst case, since enumerating every possibility for $a, \{o' \to \pi'\}$ over $a \in A$, $o \in O$, $\pi' \in \Pi_n$, yields $|\Pi_{n+1}| \leq |A||\Pi_n|^{|O|}$ combinations. Many of these α-vectors are not maximal for any belief state, and can be pruned by running a linear program for each that verifies whether there is a witness belief state for which it is maximal (Cassandra, Littman, & Zhang 1997). Thus, the set of α-vectors, Γ_n, action strategies, Π_n, and witness belief states, B_n, are all associated 1 to 1. However, even with pruning, exact value iteration cannot be run for many steps, even on small problems.

Value function approximation strategies

Much research has focused on approximating the optimal value function, aimed for the most part at reducing the time and space complexity of the value iteration update. Work in this area has considered various strategies (Hauskrecht 2000), including direct MDP approximations and variants, and using function approximation to fit V_{n+1} over sampled belief states (Parr & Russell 1995; Littman, Cassandra, & Kaelbling 1995). However, two approaches have recently become the most dominant: grid based and belief point approximations.

The grid based approach (Gordon 1995; Hauskrecht 2000; Zhou & Hansen 2001; Bonet 2002) maintains a finite collection of belief states along with associated value estimates $\{\langle b, \bar{V}_n(b)\rangle : b \in B_{grid}\}$. These value estimates are updated by applying the Bellman update on $b \in B_{grid}$. An important advantage of this approach is that it can maintain an upper bound on the optimal value function. Unfortunately, maintaining a tight bound entails significant computational expense (Hauskrecht 2000): First, B_{grid} must contain all corners of the simplex so that its convex closure spans B. Second, each successor belief state b' in (6) must have its interpolated value estimate minimized by a linear program (Zhou & Hansen 2001). Below we show that this large number of linear programs can be replaced with a single convex optimization.

Unlike the grid based approach, which takes a current belief state in B_{grid} and projects it forward to belief states outside of B_{grid}, the belief point approach only considers belief states in a witness set B_{wit} (Pineau, Gordon, & Thrun 2003; Smith & Simmons 2005). Specifically, the belief point approximation maintains a lower bound by keeping a subset of α-vectors associated with these witness belief states. To further explain this approach, let $\Gamma_n = \{\alpha_{\pi'} : \pi' \in \hat{\Pi}_n\}$, so that there is a 1 to 1 correspondence between α-vectors in Γ_n, action strategies in $\hat{\Pi}_n$ and belief states in B_{wit}. Then the set of α-vectors is updated by applying the Bellman backup, but restricting the choices in (7) to $\pi' \in \hat{\Pi}_n$, and only computing (7) for $b \in B_{wit}$. Thus, the number of α-vectors in each iteration remains bounded and associated with $b \in B_{wit}$.

The quality of both these approaches is strongly determined by the sets of belief points, B_{grid} and B_{wit}, they maintain. For the belief point approach, one generally has to grow the number of belief points at each iteration to maintain an adequate bound on the optimal value function. Pineau *et al*. (2003) suggested doubling the size at each iteration, but recently a more refined approach was suggested by (Smith & Simmons 2005).

Convex quadratic upper bounds

The key observation behind our approach is that one does not need to be confined to piecewise linear approximations. Our intuition is that convex quadratic approximations are particularly well suited for value function approximation in POMDPs. This is motivated by the fact that each value iteration step produces a maximum over a set of convex functions, yielding a result that is always convex. Thus, one can plausibly use a convex quadratic function to upper bound the maximum over α-vectors, and more generally to upper bound the maximum over any set of back-projected convex value approximations from iteration n. Our basic goal then is to retain a compact representation of the value approximation by exploiting the fact that quadratics can be more efficient at approximating a convex upper bound than a set of linear functions; see Figure 1. As with piecewise linear approximations, the quality of the approximation can be improved by taking a maximum over a set of convex quadratics, which would yield a convex piecewise quadratic rather than piecewise linear approximation. In this paper, however, we will focus on the most naive choice, and approximate the value function with a *single* quadratic in each step of value iteration. The subsequent extension to multiple quadratics is discussed below.

An important advantage the quadratic form has over other function approximation representations is that it permits a convex minimization of the upper bound, as we demonstrate below. Such a convenient formulation is not readily achievable for other function representations. Also, since we are

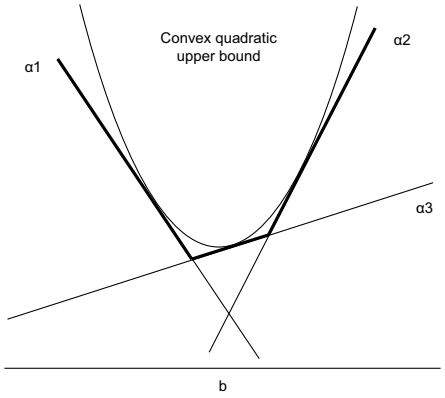

Figure 1: Illustration of a convex quadratic upper bound approximation to a maximum of linear functions $b \cdot \alpha_\pi$.

not compelled to grow the size of the representation at each iteration, we obtain an approach that runs in linear time in the number of value iteration steps.

There are a few drawbacks in dropping the piecewise linear representation however. One drawback is that we lose the 1 to 1 correspondence between α-vectors and behavior strategies π', which means that greedy action selection requires a one step look ahead calculation based on (5). The second drawback is that the convex optimization problem we have to solve at each value iteration is more complex than a simple linear program.

Convex upper bound iteration

The main technical challenge we face is to solve for a tight quadratic upper bound on the value function at each stage of value iteration. Interestingly, this can be done effectively with a convex optimization as follows. We represent the value function approximation over belief states by a quadratic form

$$\hat{V}_n(b) \quad = \quad b^\top W_n b + w_n^\top b + \omega_n \qquad (8)$$

where W_n is a square matrix of weights, w_n is a vector of weights, and ω_n is a scalar offset weight. Equation (8) defines a *convex* function of belief state b if and only if the matrix W_n is positive semidefinite (Boyd & Vandenberghe 2004). We denote the semidefinite constraint on W_n by $W_n \succeq 0$. As shown above, one step of value iteration involves expanding (and back-projecting) a value approximation from stage n; defining the value function at stage $n+1$ by the maximum over the expanded, back-projected set. However, back-projection entails some additional complication in our case because we do not maintain a set of α-vectors, but rather maintain a quadratic function approximation at stage n. That is, our approximate value iteration step has to pull the quadratic form through the backup operator. Unfortunately, the result of a backup is no longer a quadratic, but a rational (quadratic over linear) function. Fortunately, however, the result of this backup is still convex, as we now show.

Let the action-value backup of $\hat{V}$ be denoted by

$$q_a(b) \quad = \quad r(b,a) + \gamma \sum_{o'} p(o'|b,a) \hat{V}(b'_{(b,a,o')}) \qquad (9)$$

To express this as a function of b, we need to expand the definitions of $b'_{(b,a,o')}$ and $\hat{V}_n$ respectively. First, note that $b'_{(b,a,o')}$ is a ratio of a vector linear function of b over a scalar linear function of b by (4), therefore we can represent it by

$$b'_{(b,a,o')} \quad = \quad \frac{M_{a,o'} b}{p(o'|b,a)} \quad = \quad \frac{M_{a,o'} b}{e^\top M_{a,o'} b} \qquad (10)$$

where $M_{a,o'}$ is a matrix such that $M_{a,o'}(s',s) = p(o'|a,s')p(s'|s,a)$, and e denotes the vector of all 1s. Substituting (8) and (10) into (9) yields

$$q_a(b) \; = \; r(b,a) + \gamma \sum_{o'} \frac{b^\top M_{a,o'}^\top W M_{a,o'} b}{e^\top M_{a,o'} b} + (w + \omega e)^\top M_{a,o'} b$$

Theorem 1 *$q_a(b)$ is convex in b.*

Proof First note that $M_{a,o'}^\top W M_{a,o'} \succeq 0$ if $W \succeq 0$, and therefore it suffices to show that the function $f(b) = (b^\top N b)/(v^\top b)$ is convex under the assumption $N \succeq 0$ and $v^\top b \geq 0$. Note that $N \succeq 0$ implies $N = QQ^\top$ for some Q, and therefore $f(b) = (b^\top QQ^\top b)/(v^\top b) = (Q^\top b)^\top (v^\top bI)^{-1}(Q^\top b)$. Next, we use a few elementary facts about convexity (Boyd & Vandenberghe 2004). First, a function is convex iff its epigraph is convex, so it suffices to show that the set $\{(b, v^\top bI, \delta) | v^\top bI \geq 0, (Q^\top b)^\top (v^\top bI)^{-1}(Q^\top b) \leq \delta\}$ is convex. By the Schur complement lemma, we have that $\delta - (Q^\top b)^\top (v^\top bI)^{-1}(Q^\top b) \geq 0$ iff $\begin{bmatrix} v^\top bI & Q^\top b \\ (Q^\top b)^\top & \delta \end{bmatrix} \succeq 0$ and therefore $f(b)$ is convex iff the set $\left\{ (b, v^\top bI, \delta) | v^\top bI \geq 0, \begin{bmatrix} v^\top bI & Q^\top b \\ (Q^\top b)^\top & \delta \end{bmatrix} \succeq 0 \right\}$ is convex. The result then follows because this set can be written as a linear matrix inequality. ∎

Corollary 1 *Given a convex quadratic representation for $\hat{V}_n$, $\max_a q_a(b)$, and hence $H\hat{V}_n$, is convex in b.*

So back-projecting the convex quadratic representation still yields a convex result. Our goal is to optimize a tight quadratic upper bound on the maximum of these convex functions (which of course is still convex). In some approaches below we will use the back-projected action-value functions directly. However, in other cases, it will prove advantageous if we can work with linear upper bounds on the back-projections.

Proposition 1 *The tightest linear upper bound on $q_a(b)$ is given by $q_a(b) \leq u_a^\top b$ for a vector u_a such that $u_a^\top 1_s = q_a(1_s)$ for each corner belief state 1_s.*

Algorithmic approach

We would like to solve for a quadratic $\hat{V}_{n+1}$ at stage $n+1$ that obtains as tight an upper bound on $H\hat{V}_n$ as possible. To do this, we appeal to the linear program characterization

of the optimal value function (3) which also is expressed as minimizing an upper bound on the back-projected value function. Unfortunately, here, since we are no longer working with a finite space, we cannot formulate a linear program but rather have to pose a generalized semi-infinite program

$$\min_{W,w,\omega} \int_b \left(b^\top W b + w^\top b + \omega\right)\ \mu(b)\ db \quad \text{subject to} \qquad (11)$$
$$b^\top W b + w^\top b + \omega \geq q_a(b),\ \ \forall\ a, b;\ \ W \succeq 0$$

where $\mu(b)$ is a measure over the space of possible belief states. The semi-infinite program (11) specifies a linear objective subject to linear constraints (albeit infinitely many linear constraints); and hence is a *convex* optimization problem in W, w, ω.

There are two main difficulties in solving this convex optimization problem. First, the objective involves an integral with respect to a measure $\mu(b)$ on belief states. This measure is arbitrary (except that it must have full support on the belief space B) and allows one to control the emphasis the minimization places on different regions of the belief space. For simplicity, we assume the measure is a Dirichlet distribution, specified by a vector of prior parameters $\theta(s)$, $\forall s \in S$. The Dirichlet distribution is particularly convenient in this context since one can specify a uniform distribution over the belief simplex merely by setting $\theta(s) = 1$ for all s. Moreover, the required integrals for the Dirichlet have a closed form solution, which allows us to simply *precompute* the linear coefficients for the weight parameters, by

$$\int_b \left(b^\top W b + w^\top b + \omega\right) \mu(b) db \ \ = \ \ \langle W, \mathrm{E}[bb^\top]\rangle + w^\top \mathrm{E}[b] + \omega$$

where $\mathrm{E}[b] = \theta/\|\theta\|_1$; $\mathrm{E}[bb^\top] = (\mathrm{diag}(\mathrm{E}[b]) + \|\theta\|_1 \mathrm{E}[b]\mathrm{E}[b]^\top)/(1 + \|\theta\|_1)$ (Gelman *et al.* 1995); and $\langle A, B\rangle = \sum_{ij} A_{ij} B_{ij}$. That is, one can specify θ and compute the linear coefficients ahead of time.

The second and more difficult problem with solving (11) is to find a way to cope with the infinite number of linear constraints. Here, we address the problem with a straightforward constraint generation approach. The idea is to solve (11), iteratively, by keeping a finite set of constraints, each corresponding to a belief state, and solving the finite *semi-definite* program

$$\min_{W,w,\omega} \ \langle W, E[bb^\top]\rangle + w^\top E[b] + \omega \quad \text{subject to} \qquad (12)$$
$$b_i^\top W b_i + w^\top b_i + \omega \geq q_a(b_i),\ \ \forall\ a, b_i \in \mathcal{C};\ \ W \succeq 0$$

Given a putative solution, W, w, ω, a new constraint can be obtained by finding a belief state b that solves

$$\min_b b^\top W b + w^\top b + \omega - q_a(b) \quad \text{subject to}$$
$$b \geq 0,\ \ \textstyle\sum_s b(s) = 1 \qquad (13)$$

for each a. If the minimum value is nonnegative for all a then there are no violated constraints and we have a solution to (11).

Unfortunately, (13) cannot directly be used for constraint generation, since $q_a(b)$ is a convex function of b (Theorem 1) and hence $-q_a(b)$ is concave; yielding a non-convex objective. Thus, to use (13) for constraint generation we need to follow an alternative approach. We have pursued three different approaches to this problem thus far.

Our first strategy maintains a provable upper bound on the optimal value function by strengthening the constraint threshold with the linear upper bound $u_a^\top b \geq q_a(b)$ from Proposition 1. Replacing $q_a(b)$ with $u_a^\top b$ in (11) and (13) ensures that an upper bound will be maintained, but also reduces (13) to a quadratic program that can be efficiently minimized to produce a belief state with maximum constraint violation.

Our second strategy relaxes the upper bound guarantee by only substituting $u_a^\top b$ for $q_a(b)$ in the constraint generation procedure, maintaining an efficient quadratic programming formulation there, but keeping $q_a(b)$ in the main optimization (12). This no longer guarantees an upper bound, but can still produce better approximations in practice because the bounds do not have to be artificially strengthened.

Our final strategy side-steps optimal constraint generation entirely, and instead chooses a fixed set of belief states for the constraint set $\mathcal{C}$ in (12). In this way, the semidefinite program (12) needs to be solved only once per value iteration step. This strategy doesn't produce an upper bound either but the resulting approximation is fast and effective in practice.

Finally, to improve approximation quality, one could augment the approximate value function representation with a maximum over a set of quadratics, much as with α-vectors. One natural way to do this would be to maintain a separate quadratic for each action, a, in (11).

Experimental results

We implemented the proposed approach using SDPT3 (Toh, Todd, & Tutuncu 1999) as the semidefinite program solver for (12). Specifically, in our initial experiments, we have investigated the third (simplest) strategy mentioned above, CQUB, which only used a random sample of belief states to specify the constraints in $\mathcal{C}$. We compared this method to two current value function approximation strategies in the literature: Perseus (Spaan & Vlassis 2005), and PBVI (Pineau, Gordon, & Thrun 2003). Here, both Perseus and PBVI were run with the number of belief states fixed at 1000, whereas the convex quadratic method, CQUB, was run with 100 random belief states.

In our initial experiments, we considered the benchmark problems: Maze (Hauskrecht 1997), Tiger-grid, Hallway, Hallway2, Aircraft available from http://www.cassandra.org/pomdp/examples. Table 1 gives the problem characteristics. In each case, a number of value iteration steps was fixed as shown in Table 1, and each method was run 10 times to generate an estimate of value function approximation quality.

Table 2 shows the results obtained by the various value function approximation strategies on these domains, reporting the expected discounted reward obtained by the greedy policies defined with respect to the value function estimates, as well as the average time and the size of the value function

Problems	$\|S\|$	$\|A\|$	$\|O\|$	value iters
Maze	20	6	8	40
Tiger-grid	33	5	17	76
Hallway	57	5	21	55
Hallway2	89	5	17	33
Aircraft	100	10	31	10

Table 1: Problem characteristics.

	CQUB	Perseus	PBVI
Maze			
Avg. reward	45.35 $\pm$3.28	30.49 $\pm$0.75	46.70 $\pm$2.0
Run time (s)	197.71	60.00	0.66
Size	231	460	1160
Tiger-grid			
Avg. reward	2.16 $\pm$0.02	2.34 $\pm$0.02	2.25 $\pm$0.06
Run time (s)	7.5×10^3	61.36	28.47
Size	595	4422	15510
Hallway			
Avg. reward	0.58 $\pm$0.14	0.51 $\pm$0.06	0.53 $\pm$0.03
Run time (s)	7.5×10^3	61.26	39.79
Size	1711	3135	4902
Hallway2			
Avg. reward	0.43 $\pm$0.25	0.34 $\pm$0.16	0.35 $\pm$0.03
Run time (s)	1.8×10^4	63.72	27.97
Size	4095	4984	8455
Aircraft			
Avg. reward	16.70 $\pm$0.58	12.73 $\pm$4.63	16.37 $\pm$0.42
Run time (s)	3.8×10^5	60.01	8.03
Size	5151	10665	47000

Table 2: Mean discounted reward obtained over 1000 trajectories using the greedy policy for each value function approximation, averaged over 10 runs of value iteration.

approximation.[1] Interestingly, the convex quadratic strategy CQUB performed surprisingly well in these experiments, competing with state of the art value function approximations while only using 100 random belief states for constraint generation in (12). The result is slightly weaker in the Tiger-grid domain, but significantly stronger in the Hallway domains; supporting the thesis that convex quadratics capture value function structure more efficiently than linear approaches.

Conclusions

We have introduced a new approach to value function approximation for POMDPs that is based on a convex quadratic bound rather than a piecewise linear approximation. We have found that quadratic approximators can achieve highly competitive approximation quality without growing the size of the representation, even while explicitly focusing on only a tiny fraction of the belief states. We expect that this approach can lead to new avenues of research in value approximation for POMDPs.

We are currently considering extensions to this approach based on belief state compression (Poupart & Boutilier 2002; 2004; Roy, Gordon, & Thrun 2005), and factored models (Boutilier & Poole 1996; Feng & Hansen 2001; Poupart 2005) to tackle POMDPs with large state spaces. We also plan to combine our quadratic value function approximation with policy based and sampling based approaches. A further idea we are exploring is the interpretation of convex quadratics as second order Taylor approximations to the optimal value function, which offers further algorithmic approaches with the potential for tight theoretical guarantees on approximation quality.

[1] For Perseus and PBVI, the size is $|S|$ times the number of α-vectors. For CQUB, the size is just $|S|(|S|+1)/2+|S|+1$, which corresponds to the number of variables in the quadratic approximator.

Acknowledgments

Research supported by the Alberta Ingenuity Centre for Machine Learning, NSERC, MITACS, CFI, and the Canada Research Chairs program.

References

Amato, C.; Bernstein, D.; and Zilberstein, S. 2006. Solving POMDPs using quadratically constrained linear programs. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Bertsekas, D. 1995. *Dynamic Programming and Optimal Control*, volume 2. Athena Scientific.

Boger, J.; Poupart, P.; Hoey, J.; Boutilier, C.; Fernie, G.; and Mihailidis, A. 2005. A decision-theoretic approach to task assistance for persons with dementia. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI)*.

Bonet, B. 2002. An ϵ-optimal grid-based algorithm for partially observable Markov decision processes. In *Proceedings of the Nineteenth International Conference on Machine Learning (ICML)*.

Boutilier, C., and Poole, D. 1996. Computing optimal policies for partially observable decision processes using compact representations. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI)*.

Boyd, S., and Vandenberghe, L. 2004. *Convex Optimization*. Cambridge Univ. Press.

Cassandra, A.; Littman, M.; and Zhang, N. 1997. Incremental pruning: A simple, fast, exact method for partially observable Markov decision processes. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence (UAI)*.

Feng, Z., and Hansen, E. A. 2001. Approximate planning for factored POMDPs. In *Proceedings of the Sixth European Conference on Planning*.

Gelman, A.; Carlin, J.; Stern, H.; and Rubin, D. 1995. *Bayesian Data Analysis*. Chapman & Hall.

Gordon, G. 1995. Stable function approximation in dynamic programming. In *Proceedings of the Twelfth International Conference on Machine Learning (ICML)*.

Hauskrecht, M. 1997. Incremental methods for computing bounds in partially observable markov decision processes. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI)*.

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

Kearns, M.; Mansour, Y.; and Ng, A. 2002. A sparse sampling algorithm for near-optimal planning in large Markov decision processes. *Machine Learning* 49(2-3):193–208.

Littman, M.; Cassandra, A.; and Kaelbling, L. 1995. Learning policies for partially observable environments: scaling up. In *Proceedings of the Twelfth International Conference on Machine Learning (ICML)*.

Madani, O.; Hanks, S.; and Condon, A. 2003. On the undecidability of proabilistic planning and related stochastic optimization problems. *Artificial Intelligence* 147:5–34.

Mundhenk, M.; Goldsmith, J.; Lusena, C.; and Allender, E. 2000. Complexity of finite-horizon Markov decision processes. *Journal of the Association for Computing Machinery* 47(4):681–720.

Ng, A., and Jordan, M. 2000. Pegasus: A policy search method for large MDPs and POMDPs. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI)*.

Parr, R., and Russell, S. 1995. Approximating optimal policies for partially observable stochastic domains. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI)*.

Pineau, J.; Montemerlo, M.; Pollack, M.; Roy, N.; and Thrun, S. 2003. Towards robotic assistants in nursing homes: Challenges and results. *Robotics and Autonomous Systems* 42:271–281.

Pineau, J.; Gordon, G.; and Thrun, S. 2003. Point-based value iteration: An anytime algorithm for POMDPs. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*.

Poupart, P., and Boutilier, C. 2002. Value-directed compression of POMDPs. In *Advances in Neural Information Processing Systems (NIPS 15)*.

Poupart, P., and Boutilier, C. 2003. Bounded finite state controllers. In *Advances in Neural Information Processing Systems (NIPS 16)*.

Poupart, P., and Boutilier, C. 2004. VDCBPI: An approximate scalable algorithm for large POMDPs. In *Advances in Neural Information Processing Systems (NIPS 17)*.

Poupart, P. 2005. *Exploiting Structure to efficienty solve large scale partially observable Markov decision processes*. Ph.D. Dissertation, Department of Computer Science, University of Toronto.

Roy, N.; Gordon, G.; and Thrun, S. 2005. Finding approximate POMDP solutions through belief compression. *Journal of Artificial Intelligence Research* 23:1–40.

Smith, T., and Simmons, R. 2005. Point-based POMDP algorithms: Improved analysis and implementation. In *Proceedings of the Twenty-first Conference on Uncertainty in Artificial Intelligence (UAI)*.

Sondik, E. 1978. The optimal control of partially observable Markov processes over the infinite horizon: Discounted costs. *Operations Research* 26:282–304.

Spaan, M., and Vlassis, N. 2005. Perseus: Randomized point-based value iteration for POMDPs. *Journal of Artificial Intelligence Research* 24:195–220.

Thrun, S.; Burgard, W.; and Fox, D. 2005. *Probabilistic Robotics*. MIT Press.

Thrun, S. 2000. Monte Carlo POMDPs. In *Advances in Neural Information Processing Systems (NIPS 12)*.

Toh, K.; Todd, M.; and Tutuncu, R. 1999. SDPT3–a Matlab software package for semidefinite programming. *Optimization Methods and Software* 11.

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

Zhou, R., and Hansen, E. 2001. An improved grid-based approximation algorithm for POMDPs. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI)*.