Tighter Value Function Bounds for Bayesian Reinforcement Learning

Kanghoon Lee and **Kee-Eung Kim**
Department of Computer Science
Korea Advanced Institute of Science and Technology
Daejeon 305-701, Korea
khlee@ai.kaist.ac.kr and kekim@cs.kaist.ac.kr

Abstract

Bayesian reinforcement learning (BRL) provides a principled framework for optimal exploration-exploitation tradeoff in reinforcement learning. We focus on model-based BRL, which involves a compact formulation of the optimal tradeoff from the Bayesian perspective. However, it still remains a computational challenge to compute the Bayes-optimal policy. In this paper, we propose a novel approach to compute tighter value function bounds of the Bayes-optimal value function, which is crucial for improving the performance of many model-based BRL algorithms. We then present how our bounds can be integrated into real-time AO* heuristic search, and provide a theoretical analysis on the impact of improved bounds on the search efficiency. We also provide empirical results on standard BRL domains that demonstrate the effectiveness of our approach.

Introduction

Reinforcement learning (RL) is the problem where an agent tries to maximize long-term rewards while acting in an unknown environment. A fundamental problem in RL is the *exploration-exploitation tradeoff* since the agent has to explore the environment to gather information on how to improve long-term rewards, but always doing so leads to very poor rewards. Model-based Bayesian RL (BRL) seeks an optimal solution to the tradeoff from the Bayesian perspective by maintaining the posterior distribution over the models of the environment. However, obtaining the solution, which is the Bayes-optimal policy, is known to be computationally intractable (Poupart et al. 2006).

One of the main approaches in model-based BRL is to construct an *optimistic model* of the environment that is tractable to solve while yielding an approximate Bayes-optimal policy with a probably approximately correct (PAC) guarantee. Such an optimistic model can be obtained by either adding bonus to rewards (Kolter and Ng 2009) or augmenting the action space (Asmuth et al. 2009; Castro and Precup 2010; Araya-López, Thomas, and Buffet 2012). Another class of approaches is to employ a *search algorithm*, such as sample-based Monte-Carlo tree search (MCTS) (Guez, Silver, and Dayan 2012; Asmuth and Littman 2011) or real-time heuristic search (Fonteneau, Busoniu, and Munos 2013).

In this paper, we present a model-based BRL algorithm that combines these two approaches. Our main contribution is in improving the approximate model construction so that the value function from the model is a tighter upper bound of the Bayes-optimal value function. The same idea is used to compute a tighter lower bound. We then integrate these bounds into the AO* heuristic search (Nilsson 1982), where the challenge is in computing these bounds quickly enough inside the heuristic function so that we can benefit from our proposed bounds. The resulting algorithm essentially performs multi-step lookahead search using the approximate model. We provide a theoretical guarantee on the improvement in search performance achieved by improved bounds, as well as empirical results on the improvement in standard BRL domains.

Background and Related Work

A Markov decision process (MDP) is defined as a tuple $\langle S, A, T, R, \gamma \rangle$, where S is a set of states, A is a set of actions, $T : S \times A \times S \rightarrow [0, 1]$ is the transition function $T(s, a, s') = \Pr(s'|s, a)$, $R : S \times A \times S \rightarrow [R_{min}, R_{max}]$ is the non-negative reward function (i.e., $R_{min} \geq 0$), and γ is a discount factor. A policy $\pi : S \rightarrow A$ specifies which action to execute in each state. For a policy π, its value function is defined as $V^{\pi}(s) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t), s_{t+1}) | s_0 = s]$, where s_t is the state at time t. By solving an MDP, we obtain an optimal policy π^*, with its value function V^* satisfying the Bellman optimality equation $V^*(s) = \max_a \sum_{s'} T(s, a, s')[R(s, a, s') + \gamma V^*(s')]$.

In this paper, we consider the reinforcement learning problem where the agent acts in the environment modeled as an MDP with unknown transition function. Representing each unknown transition probability $T(s, a, s')$ as the parameter $\theta^{s,a}(s')$, a model-based BRL method would maintain the posterior distribution over $\theta = \{\theta^{s,a}(s') | s, s' \in S, a \in A\}$ to represent the uncertainty.

We can then formulate model-based BRL as a partially observable MDP (POMDP) planning problem (Duff 2002; Poupart et al. 2006), where the POMDP model is defined as a tuple $\langle S_{\mathcal{P}}, A_{\mathcal{P}}, Z_{\mathcal{P}}, T_{\mathcal{P}}, O_{\mathcal{P}}, R_{\mathcal{P}}, \gamma \rangle$ with the following components: the state space is defined by $S_{\mathcal{P}} = S \times \Theta$ with the "hybrid" state $\langle s, \theta \rangle \in S \times \Theta$ being made up of the environment state s and the parameter value θ. The action space remains the same with the environment action space, i.e. $A_{\mathcal{P}} = A$. The observation space is made up of the environment state space, i.e. $Z_{\mathcal{P}} = S$. The transition function is then $T_{\mathcal{P}}(\langle s, \theta \rangle, a, \langle s', \theta' \rangle) = \Pr(s' | \langle s, \theta \rangle, a) \Pr(\theta' | \langle s, \theta \rangle, a) = \theta^{s,a}(s') \delta_{\theta}(\theta')$ where $\delta_{\theta}(\theta')$ is the Kronecker delta with value 1 when $\theta = \theta'$. The observation function is $O_{\mathcal{P}}(\langle s', \theta' \rangle, a, z) = \delta_{s'}(z)$ since the environment state is observed at each time step. Finally, the reward function is $R_{\mathcal{P}}(\langle s, \theta \rangle, a, \langle s', \theta' \rangle) = R(s, a, s')$.

The belief for the hybrid-state POMDP model is a posterior distribution on $\langle s, \theta \rangle$, denoted as $b(\langle s, \theta \rangle)$. Since the state component is observed at each time step, the uncertainty is confined to θ, and hence we use the notation $\langle s, b \rangle$ for the belief. A commonly used probability distribution for the belief on θ is the flat Dirichlet multinomial (FDM), whose density function is $b(\theta) = \Pi_{s,a} \mathcal{D}(\theta^{s,a}(\cdot); n^{s,a}(\cdot))$ with independent Dirichlet distributions $\mathcal{D}(\cdot; \cdot)$ and the observed transition (s, a, s') counts $n^{s,a}(s')$. Since the Dirichlet distribution is a conjugate prior for the multinomial distribution, the updated belief from observing transition (s, a, s') is given by

$$\begin{aligned} b_a^{s,s'} &= \theta^{s,a}(s') \prod_{\hat{s},\hat{a}} Dir(\theta^{\hat{s},\hat{a}}(\cdot); n^{\hat{s},\hat{a}}(\cdot)) \\ &= \prod_{\hat{s},\hat{a}} Dir(\theta^{\hat{s},\hat{a}}(\cdot); n^{\hat{s},\hat{a}}(\cdot) + \delta_{\hat{s},\hat{a},\hat{s'}}(s, a, s')) \end{aligned}$$

which is equivalent to incrementing the corresponding single parameter $n^{s,a}(s') \leftarrow n^{s,a}(s') + 1$.

The optimal policy $\pi^*(\langle s, b \rangle)$ of the hybrid-state POMDP is the Bayes-optimal policy since it chooses the best action under model uncertainty. Its value function satisfies the Bellman optimality equation

$$V^*(\langle s, b \rangle) = \max_a \sum_{s'} T_b(s, a, s')[R(s, a, s') + \gamma V^*(\langle s', b_a^{s,s'} \rangle)]$$

where $T_b(s, a, s') = \frac{n_b^{s,a}(s')}{n_b^{s,a}}$ and $n_b^{s,a} = \sum_{s'} n_b^{s,a}(s')$ with the Dirichlet parameter $n_b^{s,a}(\cdot)$ of b.

Although the hybrid-state POMDP model offers an elegant formulation, obtaining the optimal policy of the model is a computationally challenging task. One way is to construct an approximate and tractable MDP model of the hybrid-state POMDP. Many model-based BRL algorithms with PAC guarantees take this approach, following the principle of *optimism in the face of uncertainty* to construct an optimistic MDP model: Bayesian exploration bonus (BEB) (Kolter and Ng 2009) introduces additional reward bonuses proportional to $1/(1 + n^{s,a})$ encouraging execution of relatively inexperienced state-action pairs. Best of sampled set (BOSS) (Asmuth et al. 2009) and its variant, SBOSS (Castro and Precup 2010) take a number of samples from the belief and assume the most favorable one as the transition function. Bayesian optimistic local transitions (BOLT) (Araya-López, Thomas, and Buffet 2012) uses the concept of virtual transition experiences to assume optimistic transition to the state with the highest value.

An alternative approach is to focus on solving the hybrid-state POMDP without relying on an approximate model. This approach often employs efficient search algorithms: Bayes-adaptive Monte-Carlo Planning (BAMCP) (Guez, Silver, and Dayan 2012) is an MCTS algorithm, improving UCT (Kocsis and Szepesvári 2006) in a number of aspects. Bayesian forward search sparse sampling (BFS3) (Asmuth and Littman 2011) uses also an MCTS algorithm, forward search sparse sampling (FSSS) (Walsh, Goschin, and Littman 2010), to act like near Bayes-optimal behavior. Bayesian optimistic planning (BOP) (Fonteneau, Busoniu, and Munos 2013) is an instance of AO* search that uses the simple bounds of the value function for the heuristic.

Tighter Upper and Lower Bounds

Bounds on the Bayes-optimal value function are very important for BRL. Ideally, we want tight bounds that can be computed very fast. In fact, BRL algorithms with PAC guarantees generally rely on efficient upper bounds on the value function. In this section, we present a novel upper (and lower) bound that improves on BOLT (Araya-López, Thomas, and Buffet 2012).

Let us start the discussion with the simplest bounds

$$U^0(\langle s, b \rangle) = \tfrac{R_{\max}}{1-\gamma} \quad \text{and} \quad L^0(\langle s, b \rangle) = \tfrac{R_{\min}}{1-\gamma} \qquad (1)$$

which are trivial and very loose. We can slightly improve the bounds by leveraging the fact that the hybrid-state POMDP is basically an MDP with uncertain transition function, and use the optimistic and pessimistic value iterations for bounded parameter MDPs (Givan, Leach, and Dean 2000):

$$U^0(\langle s, b \rangle) = \max_a \textstyle\sum_{s'} \left(R(s, a, s') + \gamma \max_{s'} U^0(\langle s', b' \rangle)\right) \qquad (2)$$

$$L^0(\langle s, b \rangle) = \max_a \textstyle\sum_{s'} \left(R(s, a, s') + \gamma \min_{s'} L^0(\langle s', b' \rangle)\right) \qquad (3)$$

This is equivalent to maximizing/minimizing over all possible transition functions, but it does not take into account that the belief b may be highly concentrated at a specific transition function.

The optimistic model construction in BOLT (Araya-López, Thomas, and Buffet 2012) uses an augmented action space $A \times S$ so that executing an action (a, σ) with target state σ results in η identical transition experiences $\lambda_{s,a,\sigma}^{\eta} = [(s, a, \sigma), ..., (s, a, \sigma)]$ in one step. By ignoring the belief update, the model becomes an MDP that is tractable to solve. Although BOLT gives an idea for an upper bound of the value function, in reality, η should be set to infinite for the infinite-horizon value function. This essentially reduces the BOLT upper bound to Eq. (2).

We address this problem by performing a finite number (no more than η) of value iterations starting from a trivial upper bound (Eq. (1) or (2)). In addition, we decrease the number of virtual transition experiences in each iteration for further tightening the bound. These changes provide a valid yet tighter upper bound for any intermediate horizon $1 \leq T \leq \eta$. We also take advantage of the fact that augmenting the action space is not necessary, as the optimal target state σ is straightforward to be determined for each action. This reduces the time complexity from $O(|S|^3|A|)$ to $O(|S|^2|A|)$ for each iteration.

More formally, we start from the trivial upper bound $U^0_{b,\eta}(s,a) = R_{\max}/(1-\gamma)$. We then refine it via η iterations of value update, using the artificial transition function at the i-th iteration given as

$$\tilde{T}^i_{b,\sigma}(s,a,s') = \tfrac{n_b^{s,a}(s')+(\eta-i+1)\cdot\delta_\sigma(s')}{n_b^{s,a}+(\eta-i+1)}$$

and update equation given as

$$U^i_{b,\eta}(s,a) = \max_{\sigma\in S} \textstyle\sum_{s'} \tilde{T}^i_{b,\sigma}(s,a,s')[R(s,a,s') + \gamma U^{i-1}_{b,\eta}(s')]$$
$$U^i_{b,\eta}(s) = \max_a U^i_{b,\eta}(s,a)$$

for $i = 1,\ldots,\eta$. We then obtain the following result:

Theorem 1. *Given any state s, belief b, and integer $\eta > 0$,*

$$U^i_{b,\eta}(s) \geq V^*(\langle s, b^{\eta-i}\rangle)$$

for any belief $b^{\eta-i}$ that is reachable from b by any sequence of $(\eta-i)$ actions.

Proof. (By mathematical induction) First, $U^0_{b,\eta}(s) \geq V^*(\langle s, b^\eta\rangle)$ is trivially true since $U^0_{b,\eta}(s) = R_{\max}/(1-\gamma)$. Next, using the induction hypothesis $U^i_{b,\eta}(s) \geq V^*(\langle s, b^{\eta-i}\rangle)$, we have

$$U^{i+1}_{b,\eta}(s) - V^*(\langle s, b^{\eta-(i+1)}\rangle)$$
$$\geq U^{i+1}_{b,\eta}(s,a^*) - V^*(\langle s, b^{\eta-(i+1)}\rangle)$$

(where $a^* = \operatorname{argmax}_a Q^*(\langle s, b^{\eta-(i+1)}\rangle, a)$)

$$\begin{aligned}
&= \max_{\sigma\in S} \textstyle\sum_{s'} \tilde{T}^{i+1}_{b,\sigma}(s,a,s')[R(s,a^*,s') + \gamma U^i_{b,\eta}(s')] \\
&\quad - \textstyle\sum_{s'} T_{b^{\eta-(i+1)}}(s,a,s')[R(s,a^*,s') + \gamma V^*(\langle s', b^{\eta-i}\rangle)] \\
&\geq \max_{\sigma\in S} \textstyle\sum_{s'} \tilde{T}^{i+1}_{b,\sigma}(s,a,s')[R(s,a^*,s') + \gamma V^*(\langle s', b^{\eta-i}\rangle)] \\
&\quad - \textstyle\sum_{s'} T_{b^{\eta-(i+1)}}(s,a,s')[R(s,a^*,s') + \gamma V^*(\langle s', b^{\eta-i}\rangle)] \\
&= \max_{\sigma\in S} \sum_{s'} \tfrac{n_b^{s,a}(s')+(\eta-i)\delta_\sigma(s')}{n_b^{s,a}+(\eta-i)}[R(s,a^*,s')+\gamma V^*(\langle s', b^{\eta-i}\rangle)] \\
&\quad - \textstyle\sum_{s'} \tfrac{n_b^{s,a}(s')+\phi^{s,a}_{b,b^{\eta-i}}(s')}{n_b^{s,a}+\phi^{s,a}_{b,b^{\eta-i}}}[R(s,a^*,s') + \gamma V^*(\langle s', b^{\eta-i}\rangle)]
\end{aligned}$$

where $\phi^{s,a}_{b,b^{\eta-i}}(s')$ is the observation count of (s,a,s') in the path from b to $b^{\eta-i}$, i.e., $\phi^{s,a}_{b,b^{\eta-i}}(s') = n^{s,a}_{b^{\eta-i}}(s') - n_b^{s,a}(s')$, and $\phi^{s,a}_{b,b^{\eta-i}} = \sum_{s'} \phi^{s,a}_{b,b^{\eta-i}}(s')$. Note that $\eta - i \geq \phi^{s,a}_{b,b^{\eta-i}}$. Thus,

$$\begin{aligned}
&= \max_{\sigma\in S} \textstyle\sum_{s'} \tfrac{(\eta-i)\cdot\delta_\sigma(s')}{n_b^{s,a}+(\eta-i)}[R(s,a^*,s') + \gamma V^*(\langle s', b^{\eta-i}\rangle)] \\
&\quad - \textstyle\sum_{s'} \left(\tfrac{n_b^{s,a}(s')+\phi^{s,a}_{b,b^{\eta-i}}(s')}{n_b^{s,a}+\phi^{s,a}_{b,b^{\eta-i}}} - \tfrac{n_b^{s,a}(s')}{n_b^{s,a}+(\eta-i)}\right)\cdot \\
&\qquad\qquad [R(s,a^*,s') + \gamma V^*(\langle s', b^{\eta-i}\rangle)] \\
&= \max_{\sigma\in S} \tfrac{(\eta-i)}{n_b^{s,a}+(\eta-i)}[R(s,a^*,\sigma) + \gamma V^*(\langle \sigma, b^{\eta-i}\rangle)] \\
&\quad - \textstyle\sum_{s'} \left(\tfrac{n_b^{s,a}(s')+\phi^{s,a}_{b,b^{\eta-i}}(s')}{n_b^{s,a}+\phi^{s,a}_{b,b^{\eta-i}}} - \tfrac{n_b^{s,a}(s')}{n_b^{s,a}+(\eta-i)}\right)\cdot \\
&\qquad\qquad [R(s,a^*,s') + \gamma V^*(\langle s', b^{\eta-i}\rangle)] \\
&\geq \max_{\sigma\in S} \left[\tfrac{(\eta-i)}{n_b^{s,a}+(\eta-i)} - \tfrac{\sum_{s'} n_b^{s,a}(s')+\phi^{s,a}_{b,b^{\eta-i}}(s')}{n_b^{s,a}+\phi^{s,a}_{b,b^{\eta-i}}} - \tfrac{\sum_{s'} n_b^{s,a}(s')}{n_b^{s,a}+(\eta-i)}\right]\cdot \\
&\qquad\qquad [R(s,a^*,\sigma) + \gamma V^*(\langle \sigma, b^{\eta-i}\rangle)] \\
&= \; 0
\end{aligned}$$

where the last step is due to

$$\tfrac{(\eta-i)}{n_b^{s,a}+(\eta-i)} - \tfrac{\sum_{s'} n_b^{s,a}(s')+\phi^{s,a}_{b,b^{\eta-i}}(s')}{n_b^{s,a}+\phi^{s,a}_{b,b^{\eta-i}}} + \tfrac{\sum_{s'} n_b^{s,a}(s')}{n_b^{s,a}+(\eta-i)} = 0$$

by the definitions of $n_b^{s,a}(s')$ and $\phi^{s,a}_{b,b^{\eta-i}}(s')$. □

This theorem yields the following result as a special case:

Corollary 1. *Given any state s, belief b, and integer $\eta > 0$,*

$$U^\eta_{b,\eta}(s) \geq V^*(\langle s, b\rangle)$$

Corollary 1 states that our optimistic value iteration for horizon η yields a valid upper bound of the *infinite* horizon value function. Note that, in contrast, BOLT yields the upper bound of the η-horizon value function.

Another advantage of our bound is that, once we compute the bound for horizon η, we can reuse intermediate bounds $U^i_{b,\eta}$, $i = \eta,\ldots,0$ as the upper bound for any belief b' that we encounter in subsequent time steps (Theorem 1). This property will be leveraged in the real-time search algorithm we present in the next section.

In a similar manner, we can compute the lower bound. Starting from the trivial lower bound $L^0_{b,\eta}(s,a) = R_{\min}/(1-\gamma)$, we refine it via η iterations $i = 1,\ldots,\eta$

$$L^i_{b,\eta}(s,a) = \min_{\sigma\in S} \textstyle\sum_{s'} \tilde{T}^i_{b,\sigma}(s,a,s')[R(s,a,s') + \gamma L^{i-1}_{b,\eta}(s')]$$
$$L^i_{b,\eta}(s) = \max_a L^i_{b,\eta}(s,a).$$

We have the following statements on the lower bound, the proofs of which are almost identical.

Theorem 2. *Given any state s, belief b, and integer $\eta > 0$,*

$$L^i_{b,\eta}(s) \leq V^*(\langle s, b^{\eta-i}\rangle)$$

for any $b^{\eta-i}$ that is reachable from b by any sequence of $(\eta-i)$ actions.

Corollary 2. *Given any state s, belief b, and integer $\eta > 0$,*

$$L^\eta_{b,\eta}(s) \leq V^*(\langle s, b\rangle)$$

Real-Time Heuristic Search for BRL

AEMS (Ross et al. 2008) is a well-known real-time AO* heuristic search algorithm for POMDPs. The execution is divided into the offline and online phases. In the offline phase, which is before the first execution of an action, AEMS computes the estimates of the upper and lower bounds of the optimal value function. In the online phase, which is between the consecutive executions of actions, it performs AO* to construct an AND-OR search tree, and selects the best action for execution upon timeout. The AND-OR search tree represents reachable beliefs from the current belief, where OR-nodes represent beliefs and AND-nodes represent action choices. The search tree is iteratively grown by selecting and

Algorithm 1 AEMS-BRL Algorithm

```
Input: $\langle s_{init}, b_{init}\rangle$ : initial state $s_{init}$ and prior on transition function $b_{init}$
Static: $\langle s, b\rangle$ : current belief state of the agent
        $\mathcal{T}$ : current AND-OR search tree
$\langle s, b\rangle \leftarrow \langle s_{init}, b_{init}\rangle$
Initialize $\mathcal{T}$ to single root node $\langle s, b\rangle$
while not ExecutionTerminated() do
    while not SearchTerminated() do
        $\langle s^*, b^*\rangle \leftarrow$ ChooseNextNodeToExpand()
        Expand($\langle s^*, b^*\rangle$)
        UpdateAncestor($\langle s^*, b^*\rangle$)
    end while
    Execute best action $a^*$ for $\langle s, b\rangle$
    Observe new state $s'$
    $\langle s, b\rangle \leftarrow \langle s', (b)^{s,s'}_{a^*}\rangle$
    Update tree $\mathcal{T}$ so that $\langle s, b\rangle$ is the new root
end while
```

expanding the most promising fringe node, using a heuristic based on the upper and lower bounds of the optimal value.

AEMS can be straightforwardly adapted as a model-based BRL algorithm since BRL can be formulated as a hybrid-state POMDP planning problem. Algorithm 1 shows the main procedure of the approach, which we refer to as AEMS-BRL. A fringe node in the search tree $\mathcal{T}$ is chosen for expansion based on its error contribution to the root node:

$$e(\langle s, b\rangle) = \gamma^{d_{\mathcal{T}}(\langle s,b\rangle)} \Pr(h(\langle s, b\rangle))[U^0(\langle s, b\rangle) - L^0(\langle s, b\rangle)]$$

where $h(\langle s, b\rangle)$ is the path from the root node $\langle s_0, b_0\rangle$ to the fringe node $\langle s, b\rangle$, $d_{\mathcal{T}}(\cdot)$ is the depth of the fringe node, $U^0(\cdot)$ is the upper bound value, and $L^0(\cdot)$ is the lower bound value at the fringe node. The path probability in the above equation is defined as

$$\Pr(h(\langle s, b\rangle)) = \prod_{i=0}^{d_{\mathcal{T}}(\langle s,b\rangle)-1} T_{b_i}(s_i, a_i, s_{i+1}) \Pr(a_i | \langle s_i, b_i\rangle)$$

for the path

$$h(\langle s, b\rangle) = [\langle s_0, b_0\rangle, a_0, \langle s_1, b_1\rangle, a_1, \ldots, a_{d-1}, \langle s, b\rangle]$$

with the action probability defined as

$$\Pr(a_i | \langle s_i, b_i\rangle) = \begin{cases} 1 & \text{if } a_i = \operatorname{argmax}_{a \in A} U_{\mathcal{T}}(\langle s_i, b_i\rangle, a) \\ 0 & \text{otherwise} \end{cases}$$

where $U_{\mathcal{T}}(\langle s_i, b_i\rangle, a)$ is the upper bound action value at an internal node $\langle s_i, b_i\rangle$ in the search tree $\mathcal{T}$. The fringe node with the maximum error contribution is expanded (Algorithm 2) and the bounds $U_{\mathcal{T}}$ and $L_{\mathcal{T}}$ are updated (Algorithm 3). Since AEMS-BRL is essentially AEMS, it inherits desirable properties of AEMS such as completeness and ϵ-optimality of the search (Ross, Pineau, and Chaib-Draa 2007).

One of the main factors that affect the performance is the initial upper and lower bounds used for newly expanded nodes (Algorithm 2, line 4-5). We can use the following three bounds as the initial bounds: [A] trivial bounds in Eq. (1), [B] optimistic/pessimistic value iteration bounds in Eq. (2) and (3), and [C] our proposed bound in the previous section. Note that [A] and [B] are offline bounds that need to be computed only once before the search, whereas [C] is an online bound that, when used naively, requires value iteration for every evaluation.

Algorithm 2 Expand($\langle s, b\rangle$)

```
Input: $\langle s, b\rangle$ : OR-Node to be expanded
Static: $U$ : upper bound on $V^*$,   $L$ : lower bound on $V^*$
        $\mathcal{T}$: current AND-OR search tree
 1: for $a \in A$ do
 2:     for $s' \in S$ do
 3:         Create child node $\langle s', b^{s,s'}_a\rangle$
 4:         $U_{\mathcal{T}}(s', b^{s,s'}_a) \leftarrow U^0(s', b^{s,s'}_a)$
 5:         $L_{\mathcal{T}}(s', b^{s,s'}_a) \leftarrow L^0(s', b^{s,s'}_a)$
 6:     end for
 7:     $U_{\mathcal{T}}(\langle s, b\rangle, a)$
            $\leftarrow \sum_{s' \in S} T_b(s, a, s') \, [R(s, a, s') + \gamma U_{\mathcal{T}}(\langle s', b^{s,s'}_a\rangle)]$
 8:     $L_{\mathcal{T}}(\langle s, b\rangle, a)$
            $\leftarrow \sum_{s' \in S} T_b(s, a, s') \, [R(s, a, s') + \gamma L_{\mathcal{T}}(\langle s', b^{s,s'}_a\rangle)]$
 9: end for
10: $U_{\mathcal{T}}(\langle s, b\rangle) \leftarrow \min(U_{\mathcal{T}}(\langle s, b\rangle), \max_a U_{\mathcal{T}}(\langle s, b\rangle, a))$
11: $L_{\mathcal{T}}(\langle s, b\rangle) \leftarrow \max(L_{\mathcal{T}}(\langle s, b\rangle), \max_a L_{\mathcal{T}}(\langle s, b\rangle, a))$
```

Algorithm 3 UpdateAncestor($\langle s', b'\rangle$)

```
Input: $\langle s', b'\rangle$ : OR-Node chosen to update its bounds
Static: $U$ : upper bound on $V^*$,   $L$ : lower bound on $V^*$
        $\mathcal{T}$: current AND-OR search tree
while $\langle s', b'\rangle$ is not root of $\mathcal{T}$ do
    Set $\langle s, b\rangle$ to be the parent of $\langle s', b'\rangle$ and $a$ to be the
    corresponding action
    $U_{\mathcal{T}}(\langle s, b\rangle, a)$
        $\leftarrow \sum_{s' \in S} T_b(s, a, s') \, [R(s, a, s') + \gamma U_{\mathcal{T}}(\langle s', b^{s,s'}_a\rangle)]$
    $L_{\mathcal{T}}(\langle s, b\rangle, a)$
        $\leftarrow \sum_{s' \in S} T_b(s, a, s') \, [R(s, a, s') + \gamma L_{\mathcal{T}}(\langle s', b^{s,s'}_a\rangle)]$
    $U_{\mathcal{T}}(\langle s, b\rangle) \leftarrow \min(U_{\mathcal{T}}(\langle s, b\rangle), \max_a U_{\mathcal{T}}(\langle s, b\rangle, a))$
    $L_{\mathcal{T}}(\langle s, b\rangle) \leftarrow \max(L_{\mathcal{T}}(\langle s, b\rangle), \max_a L_{\mathcal{T}}(\langle s, b\rangle, a))$
    $\langle s', b'\rangle \leftarrow \langle s, b\rangle$
end while
```

In Algorithm 4, we leverage the property mentioned in the previous section, that the intermediate results from the bound calculation can be reused. This is achieved by introducing parameter η_{min} that controls the degree of reuse. Specifically, if the intermediate bound we are attempting to reuse is computed for less than η_{min} horizon ($\eta - i < \eta_{min}$), we recompute the upper bound for η horizon since the bound is determined to be too loose. By reducing η_{min}, we increase the degree of reuse (i.e. reduce overall running time) by sacrificing the accuracy of the bound, and vice versa.

Analysis on Tighter Initial Bounds for Search

In this section, we provide a theoretical analysis on the search efficiency achieved by our tighter bounds.

In (Bonet and Geffner 2012; Hansen and Zilberstein

Algorithm 4 Online Initial Bound Computation

```
Input: $\langle s_0, b_0 \rangle$ : belief state
Static: $\eta$ : value update horizon
        $\eta_{min}$ : minimum horizon for reusing bounds
if the bound of the $d$-th ancestor $\langle \hat{s}, \hat{b} \rangle$ was computed
  & $d \leq \eta - \eta_{min}$ then
    return $U_{\hat{b},\eta}^{\eta-d}(s_0)$ and $L_{\hat{b},\eta}^{\eta-d}(s_0)$
end if
Initialize $U_{b,\eta}^0$ and $L_{b,\eta}^0$ with offline bounds
for $i = 1, ..., \eta$ do
    for $s \in S$ do
        for $a \in A$ do
            for $s' \in S$ do
                $V_U(s, a, s') = R(s, a, s') + \gamma U_{b,\eta}^{i-1}(s')$
                $V_L(s, a, s') = R(s, a, s') + \gamma L_{b,\eta}^{i-1}(s')$
            end for
            $\sigma_U = \text{argmax}_{s'} V_U(s, a, s')$
            $\sigma_L = \text{argmin}_{s'} V_L(s, a, s')$
            $U_{b,\eta}^i(s, a) = \sum_{s'} \tilde{T}_{b,\sigma_U}^i(s, a, s') V_U(s, a, s')$
            $L_{b,\eta}^i(s, a) = \sum_{s'} \tilde{T}_{b,\sigma_L}^i(s, a, s') V_L(s, a, s')$
        end for
        $U_{b,\eta}^i(s) = \max_a U_{b,\eta}^i(s, a)$
        $L_{b,\eta}^i(s) = \max_a L_{b,\eta}^i(s, a)$
    end for
end for
return $U_{b,\eta}^{\eta}(s_0)$ and $L_{b,\eta}^{\eta}(s_0)$ as the upper and lower
  bound values
```

2001), it was shown that a d-horizon (PO)MDP planning problem can be treated as a search for the optimal solution graph (tree) from the complete (i.e. implicit) AND-OR tree of depth d. A solution graph is a sub-graph of the complete AND-OR tree with the following three properties: the root OR-node belongs to the solution graph, every internal OR-node in the solution graph has exactly one child, and every fringe node in the solution graph is a terminal node in the complete AND-OR tree. The solution graph corresponds to a d-step policy, and the optimal solution graph corresponds to the optimal policy with the maximum value.

The AO* algorithm searches for the optimal solution graph by maintaining the best partial solution graph. The search is guided by an admissible heuristic function h that overestimates the optimal evaluation function f^*, choosing the best successor of OR-node using h. We can note that AEMS-BRL is an instance of AO* algorithm since it maintains the best partial solution graph using the upper bound on the optimal value. Hence, the upper bound is the admissible heuristic function being used for the search.

Although AO* is guaranteed to find the optimal solution graph, the search performance depends on the quality of the heuristic, which can be measured in terms of *worst-case set of nodes W* (Chakrabarti, Ghose, and DeSarkar 1987). In essence, it is the set of nodes that are expanded in the worst case until the optimal solution graph is found, and thus it directly relates to the worst-case running time.

Theorem 3. *(Chakrabarti, Ghose, and DeSarkar 1987) Given two admissible heuristic functions h_1 and h_2, such that $h_1(n) \geq h_2(n) \geq f^*(n)$ for all nodes n, the worst-case set of nodes expanded by AO* algorithm using h_2 is a subset of those using h_1.*

We then immediately have the following result since the algorithm in the previous section uses the upper bound of the optimal value as the heuristic for choosing actions:

Theorem 4. *Given two upper bounds U_1^0 and U_2^0, such that $U_1^0(\langle s, b \rangle) \geq U_2^0(\langle s, b \rangle) \geq V^*(\langle s, b \rangle)$ for all beliefs $\langle s, b \rangle$, the worst-case set of nodes expanded using U_2^0 is a subset of those using U_1^0.*

We can further refine this classical analysis by noting that when a lower bound is also used in search, we can detect the discovery of the optimal solution graph earlier.

Definition 1. OptimalityCondition($\langle U^0, L^0 \rangle$) *is true when $U_{\mathcal{T}}(\langle s, b \rangle, a) \leq L_{\mathcal{T}}(\langle s, b \rangle, a^*)$ for all $a \in A \backslash a^*$ and nodes $\langle s, b \rangle$ in the partial solution graph, where $\mathcal{T}$ is the search tree constructed using initial bounds $\langle U^0, L^0 \rangle$, $U_{\mathcal{T}}$ and $L_{\mathcal{T}}$ are the upper and lower bounds computed in $\mathcal{T}$, a^* is the best upper bound action for $\langle s, b \rangle$.*

It is easy to see that when OptimalityCondition is true, we have obtained the optimal solution graph and thus we can safely terminate the search. Using the definition, we can refine the analysis on the worst-case set of nodes:

Theorem 5. *Given two initial bounds $\langle U_1^0, L_1^0 \rangle$ and $\langle U_2^0, L_2^0 \rangle$ such that $U_1^0(\langle s, b \rangle) \geq U_2^0(\langle s, b \rangle) \geq V^*(\langle s, b \rangle) \geq L_2^0(\langle s, b \rangle) \geq L_1^0(\langle s, b \rangle)$ for all beliefs $\langle s, b \rangle$, the worst-case set of nodes expanded using U_2^0 and stopping with* OptimalityCondition ($\langle U_2^0, L_2^0 \rangle$) *is a subset of those using U_1^0 and stopping with* OptimalityCondition ($\langle U_1^0, L_1^0 \rangle$)

Proof. Let $\hat{W}_1$ and $\hat{W}_2$ be the worst-case sets of nodes expanded by search using the initial bounds $\langle U_1^0, L_1^0 \rangle$ and $\langle U_2^0, L_2^0 \rangle$ and stopping with the corresponding OptimalityCondition.

Suppose that there exists a node $\langle s, b \rangle$ such that $\langle s, b \rangle \in \hat{W}_2$ but $\langle s, b \rangle \notin \hat{W}_1$. Without loss of generality, let the node $\langle s, b \rangle$ be at the smallest depth so that its parent OR-node $\langle s^p, b^p \rangle$ belongs to both $\hat{W}_2$ and $\hat{W}_1$.

Since the search tree only containing the nodes $\hat{W}_2 \backslash \langle s, b \rangle$ does not satisfy OptimalityCondition($\langle U_2^0, L_2^0 \rangle$), we know that $U_2^0(\langle s^p, b^p \rangle, a) > L_2^0(\langle s^p, b^p \rangle, a^*)$ for some $a \in A \backslash a^*$. From $U_1^0(\langle s^p, b^p \rangle) \geq U_2^0(\langle s^p, b^p \rangle)$ and $L_2^0(\langle s^p, b^p \rangle) \geq L_1^0(\langle s^p, b^p \rangle)$, we can conclude that $U_1^0(\langle s^p, b^p \rangle, a) > L_1^0(\langle s^p, b^p \rangle, a^*)$. This is a contradiction since $\langle s^p, b^p \rangle$ belongs to $\hat{W}_1$. □

Theorems 4 and 5 state that tighter upper and lower bounds lead to a more efficient AO* search. In the next section, we show empirical results on the search efficiency achieved by our improved bounds.

Experiments

We first present our experimental results on four standard BRL domains: **Chain** (Strens 2000) consists of a 5 state linear chain with 2 actions. The transitions are stochastic with

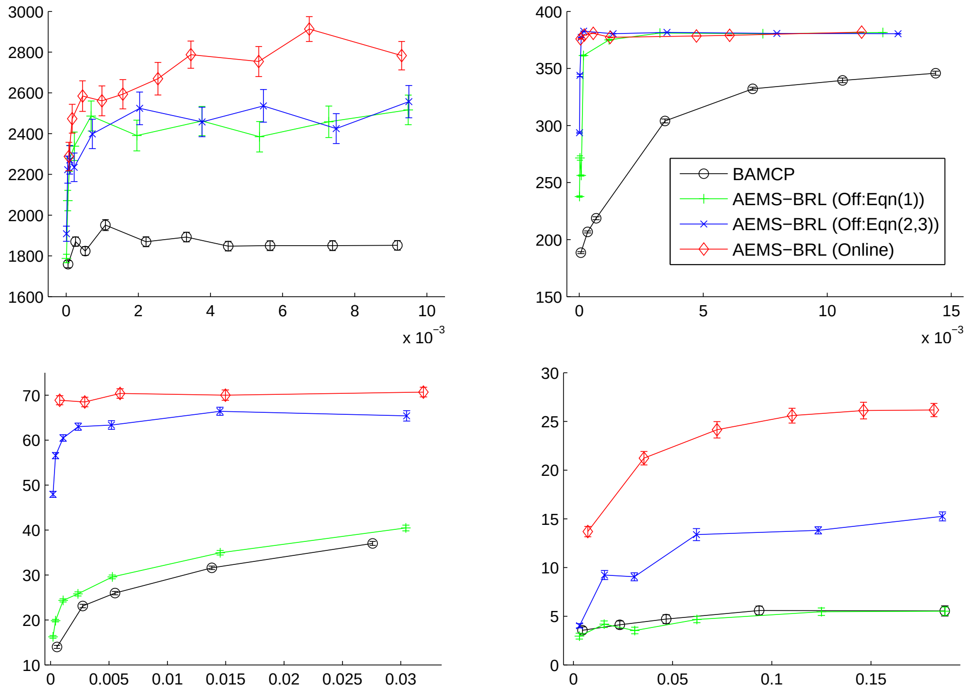

Figure 1: Average undiscounted return vs. CPU time (sec/step) in Chain (top left), Double-loop (top right), Grid5 (bottom left), Grid10 (bottom right). We compare three variants of AEMS-BRL with different bound initializations and BAMCP. The graphs are obtained by changing the number of node expansions during search. The error bar represents the 95% confidence interval.

a slipping probability of 0.2. The agent needs to traverse to the other end of the chain in order to obtain a higher reward. **Double-loop** (Dearden, Friedman, and Russell 1998) consists of two loops of length 5 with a shared starting state (9 states total) and 2 actions. The transitions are deterministic, but the agent has to close the loop with a less obvious path in order to obtain a higher reward. **Grid5** (Guez, Silver, and Dayan 2012) is a 2D grid of 25 states and 4 directional movement actions. The agent starts from the lower left corner and goal state is at the upper right corner. The reward is given upon reaching the goal state. **Grid10** (Guez, Silver, and Dayan 2012) is an enlarged version of Grid5 with 100 states.

Figure 1 compares AEMS-BRL with BAMCP (Guez, Silver, and Dayan 2012), a state-of-the-art MCTS algorithm. We experimented with 3 different bound initializations: [A] trivial offline bound in Eq. (1), [B] optimistic/pessimistic value iteration offline bound in Eq. (2) & Eq. (3), and [C] our online bound. The average undiscounted return was collected from 500 runs of 1000 time steps, except for Grid10 in which we used 100 runs of 2000 time steps. In all experiments, we set $\gamma = 0.95$ for the search and used simple Dirichlet-Multinomial model with symmetric Dirichlet parameter $\alpha_0 = 1/|S|$ except for Double-loop in which we used parameter $\alpha_0 = 1$. For the online bound initialization, we set $\eta = 40$ and $\eta_{min} = 30$ in all experiments. We can observe that our online bound initialization significantly improves the performance of real-time heuristic search with the computational overhead being already taken into account.

Algorithm	**Return**
BAMCP	347 ± 9.6
AEMS-BRL(Off:Eqn(1))	95 ± 1.9
AEMS-BRL(Off:Eqn(2,3))	110 ± 1.3
AEMS-BRL(Online)	1031 ± 7.9

Table 1: Results on Maze (0.25 sec/step)

In Figure 2, we show the effect of η_{min} on the actual undiscounted return and the computation time. Remind that η_{min} is the parameter that controls the frequency of recomputing the bounds. More frequent recomputation by increasing η_{min} generally led to higher undiscounted returns at the cost of more computation time.

In Figure 3, we compare the trends of the upper and lower bounds of the starting state at each time steps. The gap between the bounds shrinks much faster for the online bound compared to other offline bounds. It is also interesting to note that the online upper bound tends to grow larger for Double-loop. This situation arises particularly when the upper bound is very tight — as the agent gathers more observations from the environment, the belief moves towards a corner of the simplex, hence the Bayes-optimal value will actually increase for the same environment state.

In Table 1, we show the average returns (20 runs of 20000 time steps) of the algorithms on **Maze** (Dearden, Friedman, and Russell 1998), consisting of 264 states and 4 actions where the agent has to navigate in a grid environment to capture 3 flags and reach the goal location. We used simple Dirichlet-Multinomial model with symmetric Dirichlet parameter $\alpha_0 = 1/|S|$. Our online bound initialization achieved an average return of 1031±7.9

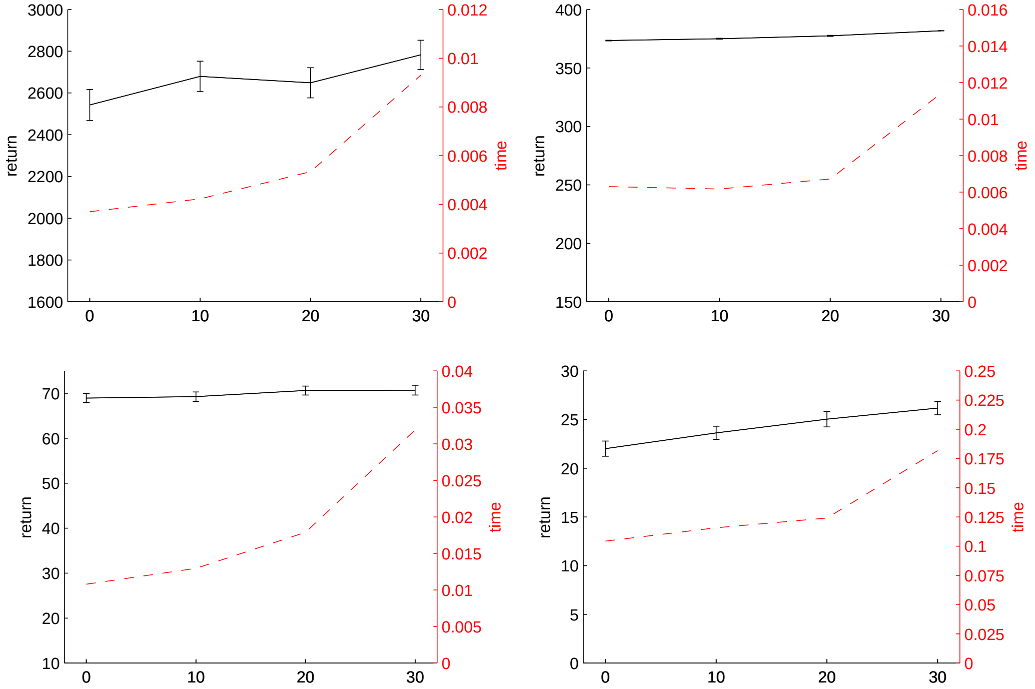

Figure 2: Average undiscounted return (left axis, black solid line) and CPU time (sec/step) (right axis, red dotted line) vs. η_{min} in Chain (top left), Double-loop (top right), Grid5 (bottom left), and Grid10 (bottom right). The results are obtained by varying the parameter $\eta_{min} = \{0, 10, 20, 30\}$. All other settings are the same as in Figure 1 (i.e., the returns for $\eta_{min} = 30$ in the figure are equal to those of AEMS-BRL with online bounds for the longest computation time. The error bars are the 95% confidence intervals.

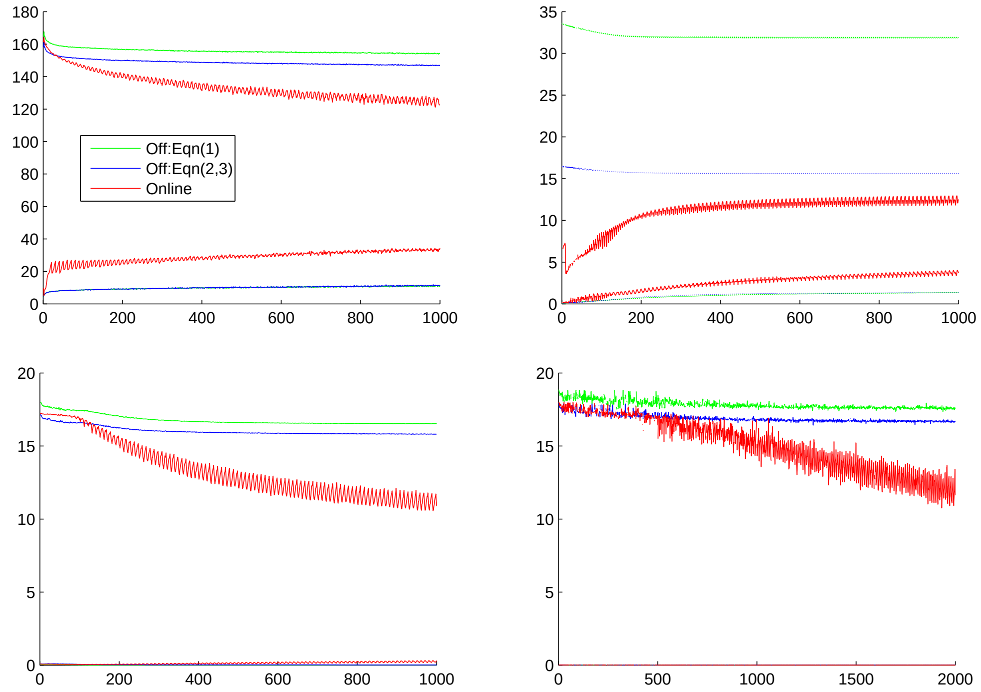

Figure 3: Upper and lower bounds of the start state vs. time steps in Chain (top left), Double-loop (top right), Grid5 (bottom left), and Grid10 (bottom right). The samples for computing the average at each time step were collected by keeping track of whether the start state is actually visited in each run.

even without a sparse prior. By comparison, BAMCP with sparse Dirichlet-Multinomial prior was reported to achieve 965.2±73 in (Guez, Silver, and Dayan 2012).

Conclusion and Future Work

In this paper, we presented a novel approach to computing the bounds of the value function for model-based BRL. Previously proposed bounds were restricted to the finite horizon value function, degenerating to very loose and naive bounds for the infinite horizon. In contrast, our bounds are for the infinite horizon value function, and provably tighter than naive bounds.

We demonstrated how our new bounds can be used in a real-time AO* algorithm for model-based BRL. Our proposed method achieved a significant performance improvement over a state-of-the-art Monte-Carlo tree search algorithm. We also provided a theoretical analysis on an improved bound for AO* search, extending the classical analysis that only involves the upper bound. Our theoretical result can be applied to AO* search for other problems, such as search-based planning with MDPs and POMDPs.

In regards to future work, applying our bound initialization method to other approaches to model-based BRL is a promising direction. Moreover, although we primarily focused on learning with discrete state MDPs, extending the work to larger domains such as factored (Boutilier, Dearden, and Goldszmidt 1995) or continuous state spaces shall be investigated for the challenge in scalability.

Acknowledgments

This work was partly supported by the ICT R&D program of MSIP/IITP [14-824-09-014, Basic Software Research in Human-level Lifelong Machine Learning (Machine Learning Center)], National Research Foundation of Korea (Grant# 2012-007881), and Defense Acquisition Program Administration and Agency for Defense Development under the contract UD140022PD, Korea

References

Araya-López, M.; Thomas, V.; and Buffet, O. 2012. Near-optimal BRL using optimistic local transition. In *Proceedings of the 29th International Conference on Machine Learning*, 97–104.

Asmuth, J., and Littman, M. 2011. Learning is planning: Near Bayes-optimal reinforcement learning via Monte-Carlo tree search. In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence*, 19–26.

Asmuth, J.; Li, L.; Littman, M. L.; Nouri, A.; and Wingate, D. 2009. A Bayesian sampling approach to exploration in reinforcement learning. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, 19–26.

Bonet, B., and Geffner, H. 2012. Action selection for MDPs: Anytime AO* versus UCT. In *Proceedings of Association for the Advancement of Artificial Intelligence*, 1749–1755.

Boutilier, C.; Dearden, R.; and Goldszmidt, M. 1995. Exploiting structure in policy construction. In *Proceedings of International Joint Conferences on Artificial Intelligence*, 1104–1111.

Castro, P. S., and Precup, D. 2010. Smarter sampling in model-based Bayesian reinforcement learning. In *Machine Learning and Knowldege Discovery in Database*. Springer. 200–214.

Chakrabarti, P. P.; Ghose, S.; and DeSarkar, S. C. 1987. Admissibility of AO* when heuristics overestimate. *Artificial Intelligence* 34:97–113.

Dearden, R.; Friedman, N.; and Russell, S. 1998. Bayesian Q-learnging. In *Proceedings of Association for the Advancement of Artificial Intelligence*, 761–768.

Duff, M. O. 2002. *Optimal Learning: Computational Procedures for Bayes-Adaptive Markov Decision Processes*. Ph.D. Dissertation, University of Massachusetts Amherst.

Fonteneau, R.; Busoniu, L.; and Munos, R. 2013. Optimistic planning for belief-augmented Markov decision processes. In *IEEE Symposium on Approximate Dynamic Programming and Reinforcement Learning*, 77–84.

Givan, R.; Leach, S.; and Dean, T. 2000. Bounded-parameter Markov decision processes. *Artificial Intelligence* 122:71–109.

Guez, A.; Silver, D.; and Dayan, P. 2012. Efficient Bayes-adaptive reinforcement learning using sample-based search. In *Advances in Neural Information Processing Systems*, 1034–1042.

Hansen, E. A., and Zilberstein, S. 2001. LAO*: A heuristic search algorithm that finds solutions with loops. *Artificial Intelligence* 129:35–62.

Kocsis, L., and Szepesvári, C. 2006. Bandit based Monte-Carlo planning. In *Machine Learning: ECML 2006*. Springer. 282–293.

Kolter, J. Z., and Ng, A. Y. 2009. Near-Bayesian exploration in polynomial time. In *Proceedings of the 26th International Conference on Machine Learning*, 513–520.

Nilsson, N. J. 1982. *Principles of Artificial Intelligence*. Symbolic Computation / Aritificial Intelligence. Springer.

Poupart, P.; Vlassis, N.; Hoey, J.; and Regan, K. 2006. An analytic solution to discrete Bayesian reinforcement learning. In *Proceedings of the 23rd International Conference on Machine Learning*, 697–704.

Ross, S.; Pineau, J.; Paquet, S.; and Chaib-draa, B. 2008. Online planning algorithms for POMDPs. *Journal of Artificial Intelligence Research* 32:663–704.

Ross, S.; Pineau, J.; and Chaib-Draa, B. 2007. Theoretical analysis of heuristic search methods for online POMDPs. In *Advances in Neural Information Processing Systems*, 1216–1225.

Strens, M. 2000. A Bayesian framework for reinforcement learning. In *Proceedings of the 17th International Conference on Machine Learning*, 943–950.

Walsh, T. J.; Goschin, S.; and Littman, M. L. 2010. Integrating sample-based planning and model-based reinforcement learning. In *Proceedings of Association for the Advancement of Artificial Intelligence*, 612–617.