Journal of Machine Learning Research 16 (2015) 3877-3903 Submitted 12/13; Revised 2/15; Published 12/15

PAC Optimal MDP Planning with Application to Invasive Species Management[*]

Majid Alkaee Taleghan ALKAEE@EECS.OREGONSTATE.EDU
Thomas G. Dietterich TGD@EECS.OREGONSTATE.EDU
Mark Crowley CROWLEY@EECS.OREGONSTATE.EDU
School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR 97331

Kim Hall KIM.HALL@OREGONSTATE.EDU
Department of Forest Ecosystems and Society, Oregon State University, Corvallis, OR 97331

H. Jo Albers JO.ALBERS@UWYO.EDU
Haub School of Environment and Natural Resources and Department of Economics and Finance, University of Wyoming, Laramie, WY 82072

Editor: Peter Auer, Marcus Hutter, and Laurent Orseau

Abstract

In a simulator-defined MDP, the Markovian dynamics and rewards are provided in the form of a simulator from which samples can be drawn. This paper studies MDP planning algorithms that attempt to minimize the number of simulator calls before terminating and outputting a policy that is approximately optimal with high probability. The paper introduces two heuristics for efficient exploration and an improved confidence interval that enables earlier termination with probabilistic guarantees. We prove that the heuristics and the confidence interval are sound and produce with high probability an approximately optimal policy in polynomial time. Experiments on two benchmark problems and two instances of an invasive species management problem show that the improved confidence intervals and the new search heuristics yield reductions of between 8% and 47% in the number of simulator calls required to reach near-optimal policies.

Keywords: invasive species management, Markov decision processes, MDP planning, Good-Turing estimate, reinforcement learning

1. Introduction

The motivation for this paper is the area of ecosystem management in which a manager seeks to maintain the healthy functioning of an ecosystem by taking actions that promote the persistence and spread of endangered species or actions that fight the spread of invasive species, fires, and disease. Most ecosystem management problems can be formulated as MDP (Markov Decision Process) planning problems with separate planning and execution phases. During the planning phase, the algorithm can invoke a simulator to obtain samples of the transitions and rewards. Simulators in these problems typically model the system to high fidelity and, hence, are very expensive to execute. Consequently, the time required to solve such MDPs is dominated by the number of calls to

[*]. Portions of this work appeared in Proceedings of Twenty-Seventh AAAI Conference on Artificial Intelligence (AAAI-2013)

the simulator. A good MDP planning algorithm minimizes the number of calls to the simulator and yet terminates with a policy that is approximately optimal with high probability. This is referred to as being PAC-RL (Fiechter, 1994).

Because of the separation between the exploration phase (where the simulator is invoked and a policy is computed) and the exploitation phase (where the policy is executed in the actual ecosystem), we refer to these ecosystem management problems as problems of *MDP Planning* rather than of *Reinforcement Learning*. In MDP planning, we do not need to resolve the exploration-exploitation tradeoff.

Another aspect of these MDP planning problems that distinguishes them from reinforcement learning is that the planning algorithm must decide when to terminate and output a PAC-optimal policy. Many reinforcement learning algorithms, such as Sparse Sampling (Kearns et al., 1999), FSSS (Walsh et al., 2010), MBIE (Strehl and Littman, 2008), and UCRL2 (Jaksch et al., 2010) never terminate. Instead, their performance is measured in terms of the number of "significantly non-optimal actions" (known as PAC-MDP, Kakade (2003)) or cumulative regret (Jaksch et al., 2010).

A final aspect of algorithms for ecosystem management problems is that they must produce an explicit policy in order to support discussions with stakeholders and managers to convince them to adopt and execute the policy. Hence, receding horizon search methods, such as Sparse Sampling and FSSS, are not appropriate because they do not compute an explicit policy.

A naive approach to solving simulator-defined MDP planning problems is to invoke the simulator a sufficiently large number of times in every state-action pair and then apply standard MDP planning algorithms to compute a PAC-optimal policy. While this is required in the worst case (c.f., Azar et al. (2012)), there are two sources of constraint that algorithms can exploit to reduce simulator calls. First, the transition probabilities in the MDP may be sparse so that only a small fraction of states are directly reachable from any given state. Second, in MDP planning problems, there is a designated starting state s_0, and the goal is to find an optimal policy for acting in that state and in all states *reachable* from that state. In the case where the optimality criterion is cumulative *discounted* reward, an additional constraint is that the algorithm only need to consider states that are reachable within a fixed horizon, because rewards far in the future have no significant impact on the value of the starting state.

It is interesting to note that the earliest PAC-optimal algorithm published in the reinforcement learning community was in fact an MDP planning algorithm: the method of Fiechter (1994) addresses exactly the problem of making a polynomial number of calls to the simulator and then outputting a policy that is approximately correct with high probability. Fiechter's method works by exploring a series of trajectories, each of which begins at the start state and continues to a fixed-depth horizon. By exploring along trajectories, this algorithm ensures that only reachable states are explored. And by terminating the exploration at a fixed horizon, it exploits discounting.

Our understanding of reinforcement learning has advanced considerably since Fiechter's work. This paper can be viewed as applying these advances to develop "modern" MDP planning algorithms. Specifically, we introduce the following five improvements:

1. Instead of exploring along trajectories, we take advantage of the fact that our simulators can be invoked for any state-action pair in any order. Hence, our algorithms perform fine-grained exploration where they iteratively select the state-action pair that they believe will be most informative.

2. By not exploring along trajectories (rooted at the start state), we could potentially lose the guarantee that the algorithm only explores states that are reachable from the start state. We address this by maintaining an estimate of the discounted state occupancy measure. This measure is non-zero only for states reachable from the start state. We also use the occupancy measure in our exploration heuristics.

3. We adopt an extension to the termination condition introduced by Even-Dar et al. (2002, 2006), which is the width of a confidence interval over the optimal value of the start state. We halt when the width of the confidence interval is less than ε, the desired accuracy bound.

4. We replace the Hoeffding-bound confidence intervals employed by Fiechter (and others) with the multinomial confidence intervals of Weissman, Ordentlich, Seroussi, Verdu, and Weinberger (2003) employed in the MBIE algorithm of Strehl and Littman (2008).

5. To take advantage of sparse transition functions, we incorporate an additional confidence interval for the Good-Turing estimate of the “missing mass” (the total probability of all unobserved outcomes for a given state-action pair). This confidence interval can be easily combined with the Weissman et al. interval.

This paper is organized as follows. Section 2 introduces our notation. Section 3 describes a particular ecosystem management problem—control of the invasive plant tamarisk—and its formulation as an MDP. Section 4 reviews previous work on sample-efficient MDP planning and describes in detail the algorithms against which we will evaluate our new methods. Section 5 presents the technical contributions of the paper. It introduces our improved confidence intervals, proves their soundness, and presents experimental evidence that they enable earlier termination than existing methods. It then describes two new exploration heuristics, proves that they achieve polynomial sample size, and presents experimental evidence that they are more effective than previous heuristics. Section 6 concludes the paper.

2. Definitions

We employ the standard formulation of an infinite horizon discounted Markov Decision Process (MDP; Bellman 1957; Puterman 1994) with a designated start state distribution. Let the MDP be defined by $\mathscr{M} = \langle S, A, P, R, \gamma, P_0 \rangle$, where S is a finite set of (discrete) states of the world; A is a finite set of possible actions that can be taken in each state; $P : S \times A \times S \mapsto [0,1]$ is the conditional probability of entering state s' when action a is executed in state s; $R(s,a)$ is the (deterministic) reward received after performing action a in state s; $\gamma \in (0,1)$ is the discount factor, and P_0 is the distribution over starting states. It is convenient to define a special starting state s_0 and action a_0 and define $P(s|s_0,a_0) = P_0(s)$ and $R(s_0,a_0) = 0$. We assume that $0 \leq R(s,a) \leq R_{max}$ for all s,a. Generalization of our methods to (bounded) stochastic rewards is straightforward.

A *strong simulator* (also called a *generative model*) is a function $F : S \times A \mapsto S \times \Re$ that given (s,a) returns (s',r) where s' is sampled according to $P(s'|s,a)$ and $r = R(s,a)$.

A (deterministic) policy is a function from states to actions, $\pi : S \mapsto A$. The value of a policy π at the starting state is defined as $V^{\pi}(s_0) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t))]$, where the expectation is taken with respect to the stochastic transitions. The maximum possible $V^{\pi}(s_0)$ is denoted $V_{max} = R_{max}/(1-\gamma)$. An optimal policy π^* maximizes $V^{\pi}(s_0)$, and the corresponding value is denoted by

$V^*(s_0)$. The action-value of state s and action a under policy π is defined as $Q^\pi(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a) V^\pi(s')$. The optimal action-value is denoted $Q^*(s,a)$.

Define $pred(s)$ to be the set of states s^- such that $P(s|s^-,a) > 0$ for at least one action a and $succ(s,a)$ to be the set of states s' such that $P(s'|s,a) > 0$.

Definition 1 *Fiechter (1994). A learning algorithm is PAC-RL*[1] *if for any discounted MDP defined by* $\langle S,A,P,R,\gamma,P_0\rangle$, $\varepsilon > 0$, $1 > \delta > 0$, *and* $0 \le \gamma < 1$, *the algorithm halts and outputs a policy* π *such that*

$$\mathbb{P}[|V^*(s_0) - V^\pi(s_0)| \le \varepsilon] \ge 1-\delta,$$

in time polynomial in $|S|$, $|A|$, $1/\varepsilon$, $1/\delta$, $1/(1-\gamma)$, *and* R_{max}.

As a learning algorithm explores the MDP, it collects the following statistics. Let $N(s,a)$ be the number of times the simulator has been called with state-action pair (s,a). Let $N(s,a,s')$ be the number of times that s' has been observed as the result. Let $R(s,a)$ be the observed reward.

3. Managing Tamarisk Invasions in River Networks

The tamarisk plant (*Tamarix* spp.) is a native of the Middle East. It has become an invasive plant in the dryland rivers and streams of the western US (DiTomaso and Bell, 1996; Stenquist, 1996). It out-competes native vegetation primarily by producing large numbers of seeds. Given an ongoing tamarisk invasion, a manager must repeatedly decide how and where to fight the invasion (e.g., eradicate tamarisk plants? plant native plants? upstream? downstream?).

A stylized version of the tamarisk management problem can be formulated as an MDP as follows. The state of the MDP consists of a tree-structured river network in which water flows from the leaf nodes toward the root (see Figure 1). The network contains E edges. Each edge in turn has H slots at which a plant can grow. Each slot can be in one of three states: empty, occupied by a tamarisk plant, or occupied by a native plant. In this stylized model, because the exact physical layout of the H slots within each edge is unimportant, the state of the edge can be represented using only the number of slots that are occupied by tamarisk plants and the number of slots occupied by native plants. The number of empty slots can be inferred by subtracting these counts from H. Hence, each edge can be in one of $(H+1)(H+2)/2$ states. Consequently, the total number of states in the MDP is $E^{(H+1)(H+2)/2}$.

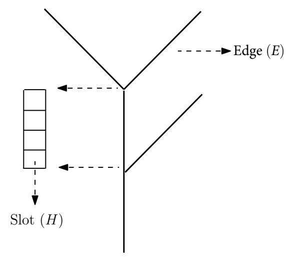


Figure 1: Tamarisk structure

The dynamics are defined as follows. In each time step, each plant (tamarisk or native) dies with probability 0.2. The remaining plants each produce 100 seeds. The seeds then disperse according to a spatial process such that downstream spread is much more likely than upstream spread. We employ the dispersal model of Muneepeerakul et al. (2007, Appendix B) with an upstream parameter of 0.1 and a downstream parameter of 0.5. An important aspect of the dispersal model is that there is a

1. In retrospect, it would have been better if Fiechter had called this PAC-MDP, because he is doing MDP planning. In turn, PAC-MDP has come to refer to *reinforcement learning* algorithms with polynomial time or regret bounds, which would be more appropriately called PAC-RL algorithms. At some point, the field should swap the meaning of these two terms.

non-zero probability for a propagule to travel from any edge to any other edge. Each propagule that arrives at an edge lands in a slot chosen uniformly at random. Hence, after dispersal, each propagule has landed in one of the slots in the river network. The seeds that arrive at an occupied slot die and have no effect. The seeds that arrive at an empty slot compete stochastically to determine which one will occupy the site and grow. In the MDPs studied in this paper, this competition is very simple: one of the arriving seeds is chosen uniformly at random to occupy the slot.

Many variations of the model are possible. For example, we can allow the tamarisk plants to be more fecund (i.e., produce more seeds) than the native plants. The seeds can have differential competitive advantage. The plants can have differential mortality, and so on. One variation that we will employ in one of our experiments is to include "exogenous arrivals" of tamarisk seeds. This models the process by which new seeds are introduced to the river network from some external source (e.g., fishermen transporting seeds on their clothes or equipment). Specifically, in the exogenous arrivals condition, in addition to the seeds that arrive at an edge via dispersal, up to 10 additional seeds of each species arrive in each edge. These are sampled by taking 10 draws from a Bernoulli distribution for each species. For tamarisk, the Bernoulli parameter is 0.1; for the native seeds, the Bernoulli parameter is 0.4.

The dynamics can be represented as a very complex dynamic Bayesian network (DBN). However, inference in this DBN is intractable, because the induced tree width is immense. One might hope that methods from the factored MDP literature could be applied, but the competition between the seeds that arrive at a given slot means that every slot is a parent of every other slot, so there is no sparseness to be exploited. An additional advantage of taking a simulation approach is that our methods can be applied to any simulator-defined MDP. We have therefore constructed a simulator that draws samples from the DBN. Code for the simulator can be obtained from `http://2013.rl-competition.org/domains/invasive-species`.

The actions for the management MDP are defined as follows. At each time step, one action can be taken in each edge. The available actions are "do nothing", "eradicate" (attempt to kill all tamarisk plants in all slots in the edge), and "restore" (attempt to kill all tamarisk plants in all slots in the edge and then plant native plants in every empty slot). The effects are controlled by two parameters: the probability that killing a tamarisk plant succeeds ($\chi = 0.85$) and the probability that planting a native plant in an empty slot succeeds ($\beta = 0.65$). Taken together, the probability that the "restore" action will change a slot from being occupied by a tamarisk plant to being occupied by a native plant is the product $\chi \times \beta = 0.5525$. Because these actions can be taken in each edge, the total number of actions for the MDP is 3^E. However, we will often include a budget constraint that makes it impossible to treat more than one edge per time step.

The reward function assigns costs as follows. There is a cost of 1.0 for each edge that is invaded (i.e., that has at least one slot occupied by a tamarisk plant) plus a cost of 0.1 for each slot occupied by a tamarisk plant. The cost of applying an action to an edge is 0.0 for "do nothing", 0.5 for "eradicate", and 0.9 for "restore".

The optimization objective is to minimize the infinite horizon discounted sum of costs. However, for notational consistency we will describe our algorithms in terms of maximizing the discounted sum of rewards throughout the paper.

It is important to note that in real applications, all of the parameters of the cost function and transition dynamics may be only approximately known, so another motivation for developing sample-efficient algorithms is to permit experimental analysis of the sensitivity of the optimal policy to the values of these parameters. The techniques employed in this paper are closely-related to those used to compute policies that are robust to these uncertainties (Mannor et al., 2012; Tamar et al., 2014).

Now that we have described our motivating application problem, we turn our attention to developing efficient MDP planning algorithms. We start by summarizing previous research.

4. Previous Work on Sample-Efficient MDP Planning

Fiechter (1994) first introduced the notion of PAC reinforcement learning in Definition 1 and presented the PAC-RL algorithm shown in Figure 1. Fiechter's algorithm defines a measure of uncertainty $\tilde{d}_h^{\pi}(s)$, which with high probability is an upper bound on the difference $|V_h^*(s) - V_h^{\pi}(s)|$ between the value of optimal policy and the value of the "maximum likelihood" policy that would be computed by value iteration using the current transition probability estimates. The subscript h indicates the depth of state s from the starting state. Fiechter avoids dealing with loops in the MDP by computing a separate transition probability estimate for each combination of state, action and depth (s,a,h) up to $h \leq H$, where H is the maximum depth ("horizon") at which estimates are needed. Hence, the algorithm maintains separate counts $N_h(s,a,s')$ and $N_h(s,a)$ to record the results of exploration for each depth h. To apply this algorithm in practice, Fiechter (1997) modifies the algorithm to drop the dependency of the related statistics on h.

Fiechter's algorithm explores along a sequence of trajectories. Each trajectory starts at state s_0 and depth 0 and follows an exploration policy π^e until reaching depth H. The exploration policy is the optimal policy for an "exploration MDP" whose transition function is $P_h(s'|s,a)$ but whose reward function for visiting state s at depth h is equal to

$$R_h(s,a) = \frac{6}{\varepsilon}\frac{V_{max}}{1-\delta}\sqrt{\frac{2\ln 4H|S||A| - 2\ln\delta}{N_h(s,a)}}.$$

This reward is derived via an argument based on the Hoeffding bound. The transition probabilities $P_h(s'|s,a)$ are computed from the observed counts.

The quantity $d^{\pi^e}(s)$ is the value function corresponding to π^e. Because the MDP is stratified by depth, π^e and d^{π_e} can be computed in a single sweep starting at depth H and working backward to depth 0. The algorithm alternates between exploring along a single trajectory and recomputing π^e and d^{π^e}. It halts when $d_0^{\pi^e}(s_0) \leq 2/(1-\gamma)$. By exploring along π^e, the algorithm seeks to visit a sequence of states whose total uncertainty is maximized in expectation.

A second important inspiration for our work is the Model-Based Action Elimination (MBAE) algorithm of Even-Dar et al. (2002, 2006). Their algorithm maintains confidence intervals $Q(s,a) \in [Q_{lower}(s,a), Q_{upper}(s,a)]$ on the action-values for all state-action pairs in the MDP. These confidence intervals are computed via "extended value iteration" that includes an additional term derived from the Hoeffding bounds:

$$Q_{upper}(s,a) = R(s,a) + \gamma\sum_{s'}\hat{P}(s'|s,a)V_{upper}(s') + V_{max}\sqrt{\frac{\ln ct^2|S||A| - \ln\delta}{|N(s,a)|}} \quad (1)$$

$$V_{upper}(s) = \max_a Q_{upper}(s,a) \quad (2)$$

$$Q_{lower}(s,a) = R(s,a) + \gamma\sum_{s'}\hat{P}(s'|s,a)V_{lower}(s') - V_{max}\sqrt{\frac{\ln ct^2|S||A| - \ln\delta}{|N(s,a)|}} \quad (3)$$

$$V_{lower}(s) = \max_a Q_{lower}(s,a). \quad (4)$$

```
Algorithm 1: Fiechter($s_0, \gamma, F, \varepsilon, \delta$)
  Input: $s_0$: start state; $\gamma$: discount rate; $F$: a simulator
  Initialization:
      $H = \left\lceil \frac{1}{1-\gamma}\left(\ln V_{max} + \ln \frac{6}{\varepsilon}\right)\right\rceil$ // horizon depth
      for $s, s' \in S, a \in A(s), h = 0, \ldots, H-1$ do
          $N_h(s,a) = 0$
          $N_h(s,a,s') = 0$
          $R_h(s,a,s') = 0$
          $\pi^e_h(s) = a_1$
  Exploration:
      while $d_0^{\pi^e}(s_0) > 2/(1-\gamma)$ do
          reset $h = 0$ and $s = s_0$
          while $h < H$ do
              $a = \pi^e_h(s)$
              $(r, s') \sim F(s,a)$ // draw sample
              update $N_h(s,a)$, $N_h(s,a,s')$, and $R_h(s,a,s')$
              $h = h + 1$
              $s = s'$
          Compute new policy $\pi^e$ (and values $d^{\pi^e}$) using the following dynamic program
          $d_{max} = (12V_{max})/(\varepsilon(1-\gamma))$
          $P_h(s'|s,a) = N_h(s,a,s')/N_h(s,a)$
          $d_H^{\pi^e}(s) = 0, \forall s \in S$
          for $h = H-1, \ldots, 0$ do
              $e_h^{\pi^e}(s,a) = \min\left\{d_{max}, \frac{6}{\varepsilon}\frac{V_{max}}{1-\delta}\sqrt{\frac{2\ln 4H|S||A| - 2\ln\delta}{N_h(s,a)}} + \gamma\sum_{s' \in succ(s,a)} P_h(s'|s,a) d_{h+1}^{\pi^e}(s')\right\}$
              $\pi^e_h(s) = \text{argmax}_{a \in A(s)}\, e_h^{\pi^e}(s,a)$
              $d_h^{\pi^e}(s) = e_h^{\pi^e}(s, \pi^e_h(s))$
  Compute policy $\pi$, and return it.
```

In these equations, t is a counter of the number of times that the confidence intervals have been computed and c is an (unspecified) constant. Even-Dar et al. prove that the confidence intervals are sound. Specifically, they show that with probability at least $1-\delta$, $Q_{lower}(s,a) \leq Q^*(s,a) \leq Q_{upper}(s,a)$ for all s, a, and iterations t.

Their MBAE algorithm does not provide a specific exploration policy. Instead, the primary contribution of their work is to demonstrate that these confidence intervals can be applied as a termination rule. Specifically, if for all (s,a), $|Q_{upper}(s,a) - Q_{lower}(s,a)| < \frac{\varepsilon(1-\gamma)}{2}$, then the policy that chooses actions to minimize $Q_{lower}(s,a)$ is ε-optimal with probability at least $1-\delta$. Note that the iteration over s' in these equations only needs to consider the observed transitions, as $\hat{P}(s'|s,a) = 0$ for all transitions where $N(s,a,s') = 0$.

An additional benefit of the confidence intervals is that any action a' can be eliminated from consideration in state s if $Q_{upper}(s,a') < Q_{lower}(s,a)$. Even-Dar et al. demonstrate experimentally that this can lead to faster learning than standard Q learning (with either uniform random action selection or ε-greedy exploration).

The third important source of ideas for our work is the Model-Based Interval Estimation (MBIE) algorithm of Strehl and Littman (2008). MBIE maintains an upper confidence bound on the action-value function, but unlike Fiechter and Even-Dar et al., this bound is based on a confidence region for the multinomial distribution developed by Weissman et al. (2003).

Let $\hat{P}(s'|s,a) = N(s,a,s')/N(s,a)$ be the maximum likelihood estimate for $P(s'|s,a)$, and let $\hat{P}$ and $\tilde{P}$ denote $\hat{P}(\cdot|s,a)$ and $\tilde{P}(\cdot|s,a)$. Define the confidence set CI as

$$CI(\hat{P}|N(s,a),\delta) = \left\{\tilde{P} \;\middle|\; \|\tilde{P}-\hat{P}\|_1 \leq \omega(N(s,a),\delta)\right\}, \tag{5}$$

where $\|\cdot\|_1$ is the L_1 norm and $\omega(N(s,a),\delta) = \sqrt{\frac{2[\ln(2^{|S|}-2)-\ln\delta]}{N(s,a)}}$. The confidence interval is an L_1 "ball" of radius $\omega(N(s,a),\delta)$ around the maximum likelihood estimate for P. Weissman et al. (2003) prove that with probability $1-\delta$, $P(\cdot|s,a) \in CI(\hat{P}(\cdot|s,a)|N(s,a),\delta)$.

Given confidence intervals for all visited (s,a), MBIE computes an upper confidence bound on Q and V as follows. For any state where $N(s,a)=0$, define $Q_{upper}(s,a) = V_{max}$. Then iterate the following dynamic programming equations to convergence:

$$Q_{upper}(s,a) = R(s,a) + \max_{\tilde{P}(s,a)\in CI(P(s,a),\delta_1)} \gamma \sum_{s'} \tilde{P}(s'|s,a) \max_{a'} Q_{upper}(s',a') \quad \forall s,a \tag{6}$$

At convergence, define $V_{upper}(s) = \max_a Q_{upper}(s,a)$. Strehl and Littman (2008) prove that this converges.

Strehl and Littman provide Algorithm UPPERP (Algorithm 2) for solving the optimization over $CI(P(s,a),\delta_1)$ in (6) efficiently. If the radius of the confidence interval is ω, then we can solve for $\tilde{P}$ by shifting $\Delta\omega = \omega/2$ of the probability mass from outcomes s' for which $V_{upper}(s') = \max_{a'} Q_{upper}(s',a')$ is low ("donor states") to outcomes for which it is maximum ("recipient states"). This will result in creating a $\tilde{P}$ distribution that is at L_1 distance ω from $\hat{P}$. The algorithm repeatedly finds a pair of successor states $\underline{s}$ and $\overline{s}$ and shifts probability from one to the other until it has shifted $\Delta\omega$. Note that in most cases, $\overline{s}$ will be a state for which $N(s,a,\overline{s}) = 0$—that is, a state we have never visited. In such cases, $V_{upper}(\overline{s}) = V_{max}$.

As with MBAE, UPPERP only requires time proportional to the number of transitions that have been observed to have non-zero probability.

The MBIE algorithm works as follows. Given the upper bound Q_{upper}, MBIE defines an exploration policy based on the optimism principle (Buşoniu and Munos, 2012). Specifically, at each state s, it selects the action a that maximizes $Q_{upper}(s,a)$. It then performs that action in the MDP simulator to obtain the immediate reward r and the resulting state s'. It then updates its statistics $N(s,a,s')$, $R(s,a)$, and $N(s,a)$ and recomputes Q_{upper}.

MBIE never terminates. However, it does compute a constant m such that if $N(s,a) > m$, then it does not draw a new sample from the MDP simulator for (s,a). Instead, it samples a next state according to its transition probability estimate $\hat{P}(s'|s,a)$. Hence, in an ergodic[2] or unichain[3] MDP, it will eventually stop drawing new samples, because it will have invoked the simulator on all actions a in all non-transient states s at least m times.

2. An ergodic MDP is an MDP where every state can be accessed in a finite number of steps from any other state
3. In unichain MDP, every policy in an MDP result in a single ergodic class

```
Algorithm 2: UPPERP($s,a,\delta,M_0$)
    Input: $s,a$
    $\delta$: Confidence parameter
    $M_0$: missing mass limit
    Lines marked by GT: are for the Good-Turing extension
    $N(s,a) := \sum_{s'} N(s,a,s')$
    $\hat{P}(s'|s,a) := N(s,a,s')/N(s,a)$ for all $s'$
    $\tilde{P}(s'|s,a) := \hat{P}(s'|s,a)$ for all $s'$
    $\Delta\omega := \omega(N(s,a),\delta)/2$
GT: $N_0(s,a) := \{s'|N(s,a,s') = 0\}$
GT: $\Delta\omega := \min\left(\omega(N(s,a),\delta/2)/2, (1+\sqrt{2})\sqrt{\frac{ln(2/\delta)}{N(s,a)}}\right)$
    while $\Delta\omega > 0$ do
        $S' := \{s' : \hat{P}(s'|s,a) < 1\}$     recipient states
GT:     if $M_0 = 0$ then $S' := S' \setminus N_0(s,a)$
        $\underline{s} := \text{argmin}_{s':\tilde{P}(s'|s,a)>0} V_{upper}(s')$     donor state
        $\bar{s} := \text{argmax}_{s' \in S', \tilde{P}(s'|s,a)<1} V_{upper}(s')$ recipient state
        $\xi := \min\{1 - \tilde{P}(\bar{s}|s,a), \tilde{P}(\underline{s}|s,a), \Delta\omega\}$
        $\tilde{P}(\underline{s}|s,a) := \tilde{P}(\underline{s}|s,a) - \xi$
        $\tilde{P}(\bar{s}|s,a) := \tilde{P}(\bar{s}|s,a) + \xi$
        $\Delta\omega := \Delta\omega - \xi$
GT:     if $\bar{s} \in N_0(s,a)$ then $M_0 := M_0 - \xi$
    return $\tilde{P}$
```

Because MBIE does not terminate, it cannot be applied directly to MDP planning. However, we can develop an MDP planning version by using the horizon time H computed by Fiechter's method and forcing MBIE to jump back to s_0 each time it has traveled H steps away from the start state. Algorithm 3 provides the pseudo-code for this variant of MBIE, which we call MBIE-reset.

Now that we have described the application goal and previous research, we present the novel contributions of this paper.

5. Improved Model-Based MDP Planning

We propose a new algorithm, which we call DDV. Algorithm 4 presents the general schema for the algorithm. For each state-action (s,a) pair that has been explored, DDV maintains upper and lower confidence limits on $Q(s,a)$ such that $Q_{lower}(s,a) \leq Q^*(s,a) \leq Q_{upper}(s,a)$ with high probability. From these, we compute a confidence interval on the value of the start state s_0 according to $V_{lower}(s_0) = \max_a Q_{lower}(s_0,a)$ and $V_{upper}(s_0) = \max_a Q_{upper}(s_0,a)$. Consequently, $V_{lower}(s_0) \leq V^*(s_0) \leq V_{upper}(s_0)$ with high probability. The algorithm terminates when the width of this confidence interval, which we denote by $\Delta V(s_0) = V_{upper}(s_0) - V_{lower}(s_0)$, is less than ε.

The confidence intervals for Q_{lower} and Q_{upper} are based on an extension of the Weissman, et al. confidence interval of Equation (5), which we will refer to as $CI^{GT}(P(s,a),\delta_1)$ (which will be described below). The confidence intervals are computed by iterating the following equations to

```
Algorithm 3: MBIE-reset($s_0, \gamma, F, H, \varepsilon, \delta$)
  Input: $s_0$:start state, $\gamma$: discount rate, $F$: a simulator, $H$ : horizon,$\varepsilon, \delta$: accuracy and
         confidence parameters
  $N(s,a,s') = 0$ for all $(s,a,s')$
  $m = c\left[\frac{|S|}{\varepsilon^2(1-\gamma)^4} + \frac{1}{\varepsilon^2(1-\gamma)^4}\ln\frac{|S||A|}{\varepsilon(1-\gamma)\delta}\right]$
  repeat forever
      $s = s_0$
      $h = 1$
      while $h \le H$ do
          update $Q_{upper}$ and $V_{upper}$ by iterating equation 6 to convergence
          $a = \text{argmax}_a Q_{upper}(s)$
          if $N(s,a) < m$ then
              $(r,s') \sim F(s,a)$ // draw sample
              update $N(s,a,s'), N(s,a)$, and $R(s,a)$
          else
              $s' \sim \hat{P}(s'|s,a)$
              $r = R(s,a)$
          $h = h + 1$
```

convergence:

$$Q_{lower}(s,a) = R(s,a) + \min_{\tilde{P}(s,a)\in CI^{GT}(P(s,a),\delta_1)} \gamma \sum_{s'} \tilde{P}(s'|s,a) \max_{a'} Q_{lower}(s',a') \quad \forall s,a. \tag{7}$$

$$Q_{upper}(s,a) = R(s,a) + \max_{\tilde{P}(s,a)\in CI^{GT}(P(s,a),\delta_1)} \gamma \sum_{s'} \tilde{P}(s'|s,a) \max_{a'} Q_{upper}(s',a') \quad \forall s,a. \tag{8}$$

The Q values are initialized as follows: $Q_{lower}(s,a) = 0$ and $Q_{upper}(s,a) = V_{max}$. At convergence, define $V_{lower}(s) = \max_a Q_{lower}(s,a)$ and $V_{upper}(s) = \max_a Q_{upper}(s,a)$.

Lemma 2 *If $\delta_1 = \delta/(|S||A|)$, then with probability $1-\delta$, $Q_{lower}(s,a) \le Q^*(s,a) \le Q_{upper}(s,a)$ for all (s,a) and $V_{lower}(s) \le V^*(s) \le V_{upper}$ for all s.*

Proof Strehl and Littman (2008) prove this for Q_{upper} and V_{upper} by showing that it is true at the point of initialization and that Equation (8) is a contraction. Hence, it remains true by induction on the number of iterations of value iteration. The proof for Q_{lower} and V_{lower} is analogous. ■

The exploration heuristic for DDV is based on exploring the state-action pair (s,a) that maximizes the expected decrease in $\Delta V(s_0)$. We write this quantity as $\Delta\Delta V(s_0|s,a)$, because it is a change (Δ) in the confidence interval width $\Delta V(s_0|s,a)$. Below, we will describe two different heuristics that are based on two different approximations to $\Delta\Delta V(s_0|s,a)$.

We now present the improved confidence interval, CI^{GT}, and evaluate its effectiveness experimentally. Then we introduce our two search heuristics, analyze them, and present experimental evidence that they improve over previous heuristics.

Algorithm 4: DDV $(s_0, \gamma, F, \varepsilon, \delta)$

```
Input: $s_0$:start state
$\gamma$: discount rate
$F$: a simulator
$\varepsilon, \delta$: accuracy and confidence parameters
$m = c\left[\frac{|S|}{\varepsilon^2(1-\gamma)^4} + \frac{1}{\varepsilon^2(1-\gamma)^4}\ln\frac{|S||A|}{\varepsilon(1-\gamma)^\delta}\right]$
$\delta' = \delta/(|S||A|m)$
$\tilde{S} = \{s_0\}$ // observed and/or explored states
$N(s,a,s') = 0$ for all $(s,a,s')$
repeat forever
    update $Q_{upper}, Q_{lower}, V_{upper}, V_{lower}$ by iterating equations 7 and 8 using $\delta'$ to compute
    the confidence intervals
    if $V_{upper}(s_0) - V_{lower}(s_0) \leq \varepsilon$ then
        // compute a good policy and terminate
        $\pi_{lower}(s) = \arg\max_a Q_{lower}(s,a)$
        return $\pi_{lower}$
    forall the explored or observed states $s$ do
        forall the actions $a$ do
            compute $\Delta\Delta V(s_0|s,a)$
    compute $(s,a) := \text{argmax}_{(s,a)} \Delta\Delta V(s_0|s,a)$
    $(r,s') \sim F(s,a)$ // draw sample
    $\tilde{S} := \tilde{S} \cup \{s'\}$ // update the set of discovered states
    update $N(s,a,s'), N(s,a)$, and $R(s,a)$
```

5.1 Tighter Statistical Analysis for Earlier Stopping

The first contribution of this paper is to improve the confidence intervals employed in equation (6). In many real-world MDPs, the transition probability distributions are sparse in the sense that there are only a few states s' such that $P(s'|s,a) > 0$. A drawback of the Weissman et al. confidence interval is that $\omega(N,\delta)$ scales as $O(\sqrt{|S|/N})$, so the intervals are very wide for large state spaces. We would like a tighter confidence interval for sparse distributions.

Our approach is to intersect the Weissman et al. confidence interval with a confidence interval based on the Good-Turing estimate of the missing mass (Good, 1953).

Definition 3 *For a given state-action pair (s,a), let $N_k(s,a) = \{s'|N(s,a,s') = k\}$ be the set of all result states s' that have been observed exactly k times. We seek to bound the total probability of those states that have never been observed: $M_0(s,a) = \sum_{s' \in N_0(s,a)} P(s'|s,a)$. The Good-Turing estimate of $M_0(s,a)$ is*

$$\widehat{M}_0(s,a) = \frac{|N_1(s,a)|}{N(s,a)}.$$

In words, Good and Turing count the number of successor states that have been observed exactly once and divide by the number of samples. The following lemma follows directly from Kearns and Saul (1998), McAllester and Schapire (2000), and McAllester and Ortiz (2003).

Lemma 4 *With probability* $1-\delta$,

$$M_0(s,a) \le \widehat{M}_0(s,a) + (1+\sqrt{2})\sqrt{\frac{\ln(1/\delta)}{N(s,a)}}. \tag{9}$$

Proof Let $S(M_0(s,a),x)$ be the Chernoff "entropy", defined as

$$S(M_0(s,a),x) = \sup_{\beta} x\beta - \ln Z(M_0(s,a),\beta),$$

where $Z(M_0(s,a),\beta) = \mathbb{E}[e^{\beta M_0(s,a)}]$. McAllester and Ortiz (2003, Theorem 16) prove that

$$S(M_0(s,a), \mathbb{E}[M_0(s,a)] + \varepsilon) \ge N(s,a)\varepsilon^2.$$

From Lemmas 12 and 13 of McAllester and Schapire (2000),

$$\mathbb{E}[M_0(s,a)] \le \widehat{M}_0(s,a) + \sqrt{\frac{2\log 1/\delta}{N(s,a)}}.$$

Combining these results yields

$$S\left(M_0(s,a), \widehat{M}_0(s,a) + \sqrt{\frac{2\log 1/\delta}{N(s,a)}} + \varepsilon\right) \ge N(s,a)\varepsilon^2. \tag{10}$$

Chernoff (1952) proves that

$$P(M_0(s,a) \ge x) \le e^{-S(M_0(s,a),x)}.$$

Plugging in (10) gives

$$P\left(M_0(s,a) \ge \widehat{M}_0(s,a) + \sqrt{\frac{2\log 1/\delta}{N(s,a)}} + \varepsilon\right) \le e^{-N(s,a)\varepsilon^2}. \tag{11}$$

Setting $\delta = e^{-N(s,a)\varepsilon^2}$ and solving for ε gives $\varepsilon = \sqrt{(\log 1/\delta)/N(s,a)}$. Plugging this into (11) and simplifying gives the result. ■

Define $CI^{GT}(\hat{P}|N(s,a),\delta)$ to be the set of all distributions $\tilde{P} \in CI(\hat{P}|N(s,a),\delta/2)$ such that $\sum_{s' \in N_0(s,a)} \tilde{P}(s'|s,a) < \widehat{M}_0(s,a) + (1+\sqrt{2})\sqrt{\frac{\ln(2/\delta)}{N(s,a)}}$. This intersects the Weissman and Good-Turing intervals. Note that since we are intersecting two confidence intervals, we must compute both (5) and (9) using $\delta/2$ so that they will simultaneously hold with probability $1-\delta$.

We can incorporate the bound from (9) into UPPERP by adding the lines prefixed by "GT:" in Algorithm 2. These limit the amount of probability that can be shifted to unobserved states according to (9). The modified algorithm still only requires time proportional to the number of states s' where $N(s,a,s') > 0$.

5.1.1 EXPERIMENTAL EVALUATION OF THE IMPROVED CONFIDENCE BOUND

To test the effectiveness of this Good-Turing improvement, we ran MBIE-reset and compared its performance with and without the improved confidence interval.

We experimented with four MDPs. The first is a Combination Lock MDP with 500 states. In each state i, there are two possible actions. The first action makes a deterministic transition to state $i+1$ with reward 0 except for state 500, which is a terminal state with a reward of 1. The second action makes a transition (uniformly) to one of the states $1,\ldots,i-1$ with reward 0. The optimal policy is to choose the first action in every state, even though it doesn't provide a reward until the final state.

The remaining three MDPs are different versions of the tamarisk management MDP. The specific network configurations that we employed in this experiment were the following:

- $E=3,H=2$ with the budget constraint that in each time step we can only choose one edge in which to perform a non-"do nothing" action. This gives a total of 7 actions.
- $E=3,H=3$ with the same constraints as for $E=3,H=2$.
- $E=7,H=1$ with the budget constraint that in each time step we can only choose one edge in which to perform a non-"do nothing" action. The only such action is "restore". This gives a total of 8 actions.

5.1.2 RESULTS

Figure 2 shows the upper and lower confidence bounds, $V_{upper}(s_0)$ and $V_{lower}(s_0)$, on the value of the starting state s_0 as a function of the number of simulator calls. The confidence bounds for the Weissman et al. interval are labeled "V(CI)", whereas the bounds for this interval combined with the Good-Turing interval are labeled "V(CI-GT)".

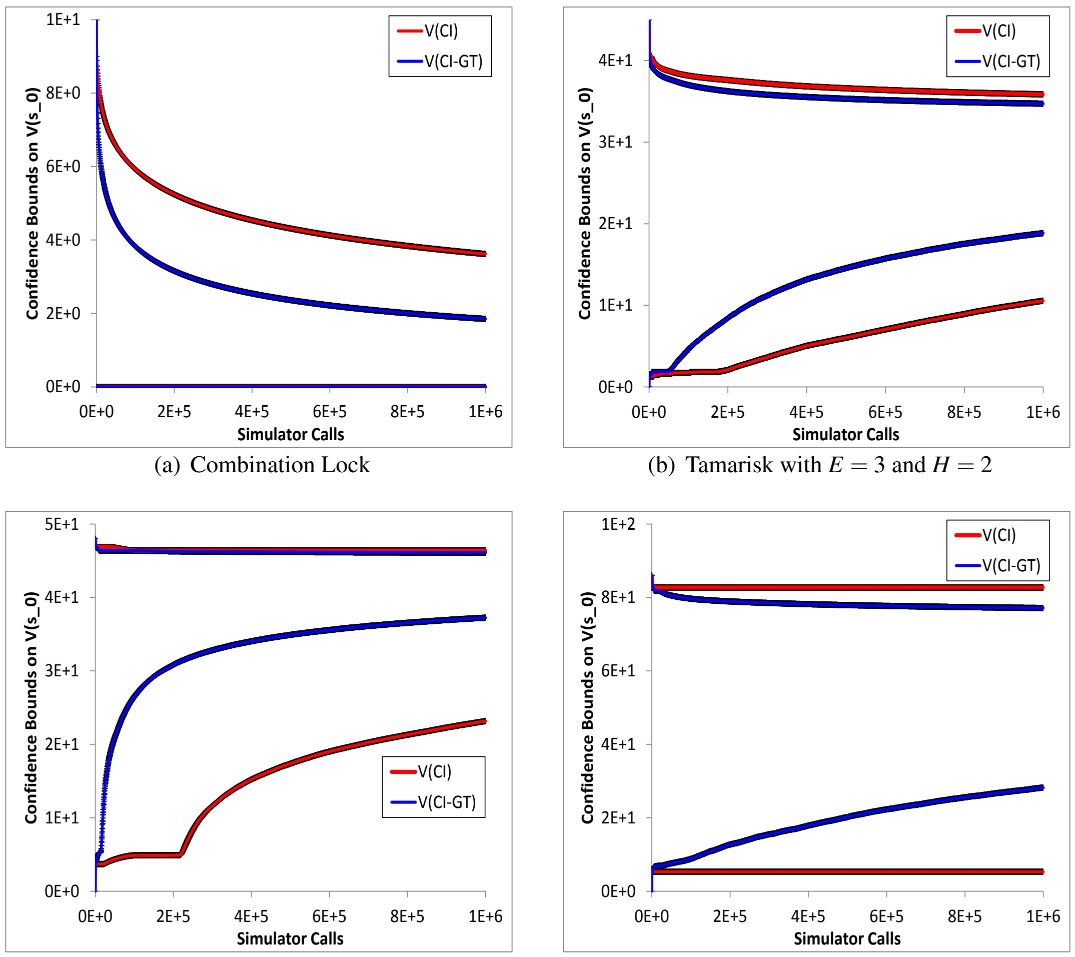
5.1.3 DISCUSSION

The results show that the Good-Turing interval provides a substantial reduction in the number of required simulator calls. On the combination lock problem, the CI-GT interval after 2×10^5 calls is already better than the CI interval after 10^6 calls, for a more than five-fold speedup. On the $E=3,H=2$ tamarisk problem, the speedup is more than a factor of three. On the $E=3,H=3$ version, the speedup is more than five-fold. And on the $E=7,H=1$ problem, the CI interval does not show any progress toward convergence, whereas the CI-GT interval has begun to make progress.

5.2 Improved Exploration Heuristics for MDP Planning

The second contribution of this paper is to define two new exploration heuristics for MDP planning and compare them to existing algorithms. As with previous work, we wish to exploit reachability and discounting to avoid exploring unnecessarily. However, we want to take advantage of the fact that our simulators are "strong" in the sense that we can explore any desired state-action pair in any order.

As discussed above, our termination condition is to stop when the width of the confidence interval $\Delta V(s_0)=V_{upper}(s_0)-V_{lower}(s_0)$ is less than ε. Our heuristics are based on computing the state-action pair (s,a) that will lead to the largest (one step) reduction in $\Delta V(s_0)$. Formally, let



(a) Combination Lock
(b) Tamarisk with $E = 3$ and $H = 2$
(c) Tamarisk with $E = 3$ and $H = 3$
(d) Tamarisk with $E = 7$ and $H = 1$

Figure 2: Plots of $V_{upper}(s_0)$ and $V_{lower}(s_0)$ for MBIE-reset on $V(s_0)$ with and without incorporating Good-Turing confidence intervals. Values are the mean of 15 independent trials. Error bars (which are barely visible) show 95% confidence intervals computed from the 15 trials.

$\Delta\Delta V(s_0|s,a) = \mathbb{E}[\Delta V(s_0) - \Delta V'(s_0)|(s,a)]$ be the expected change in $\Delta V(s_0)$ if we draw one more sample from (s,a). Here the prime in $\Delta V'(s_0)$ denotes the value of $\Delta V(s_0)$ after exploring (s,a). The expectation is taken with respect to two sources of uncertainty: uncertainty about the reward $R(s,a)$ and uncertainty about the resulting state $s' \sim P(s'|s,a)$.

Suppose we are considering exploring (s,a). We approximate $\Delta\Delta V(s_0|s,a)$ in two steps. First, we consider the reduction in our uncertainty about $Q(s,a)$ if we explore (s,a). Let $\Delta Q(s,a) = Q_{upper}(s,a) - Q_{lower}(s,a)$ and $\Delta\Delta Q(s,a) = \mathbb{E}[\Delta Q(s,a) - \Delta Q'(s,a)|(s,a)]$. Second, we consider the impact that reducing $\Delta Q(s,a)$ will have on $\Delta V(s_0)$.

We compute $\Delta\Delta Q(s,a)$ as follows.

Case 1: $N(s,a) = 0$. In this case, our current bounds are $Q_{lower}(s,a) = 0$ and $Q_{upper}(s,a) = V_{max}$. After we sample $(r,s') \sim F(s,a)$, we will observe the actual reward $R(s,a) = r$ and we will observe one of the possible successor states s'. For purposes of deriving our heuristic, we will assume a uniform[4] prior on $R(s,a)$ so that the expected value of R is $\overline{R} = R_{max}/2$. We will assume that s' will be a "new" state that we have never observed before, and hence $V_{upper}(s') = V_{max}$ and $V_{lower}(s') = 0$. This gives us

$$Q'_{upper}(s,a) = \overline{R}(s,a) + \gamma R_{max}/(1-\gamma) \tag{12a}$$

$$Q'_{lower}(s,a) = \overline{R}(s,a), \tag{12b}$$

If a more informed prior is known for $R(s,a)$, then it could be employed to derive a more informed exploration heuristic.

Case 2: $N(s,a) > 0$. In this case, we have already observed $R(s,a)$, so it is no longer a random variable. Hence, the expectation is only over s'. For purposes of deriving our exploration heuristic, we will assume that s' will be drawn according to our current maximum likelihood estimate $\hat{P}(s'|s,a)$ but that $N_1(s,a)$ will not change. Consequently, the Good-Turing estimate will not change. Under this assumption, the expected value of Q will not change, $M_0(s,a)$ will not change, so the only change to Q_{upper} and Q_{lower} will result from replacing $\omega(N(s,a),\delta)$ by $\omega(N(s,a)+1,\delta)$ in the Weissman et al. confidence interval.

Note that DDV may explore a state-action pair (s,a) even if a is not currently the optimal action in s. That is, even if $Q_{upper}(s,a) < Q_{upper}(s,a')$ for some $a' \neq a$. An alternative rule would be to only explore (s,a) if it would reduce the expected value of $\Delta V(s) = V_{upper}(s) - V_{lower}(s)$. However, if there are two actions a and a' such that $Q_{upper}(s,a) = Q_{upper}(s,a')$, then exploring only one of them will not change $\Delta V(s)$. Our heuristic avoids this problem. We have studied another variant in which we defined $V_{upper}(s) = \text{softmax}_a(\tau)\, Q_{upper}(s,a)$ (the softmax with temperature τ). This gave slightly better results, but it requires that we tune τ, which is a nuisance.

The second component of our heuristic is to estimate the impact of $\Delta\Delta Q(s_0|s,a)$ on $\Delta\Delta V(s_0|s,a)$. To do this, we appeal to the concept of an occupancy measure.

Definition 5 *The* occupancy measure $\mu^\pi(s)$ *is the expected discounted number of times that policy* π *visits state* s*,*

$$\mu^\pi(s) = \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^t \mathbb{I}[s_t = s] \,\middle|\, s_0, \pi\right], \tag{13}$$

where $\mathbb{I}[\cdot]$ *is the indicator function and the expectation taken is with respect to the transition distribution.*

This can be computed via dynamic programming on the Bellman flow equation (Syed et al., 2008):

$$\mu^\pi(s) := P_0(s) + \gamma \sum_{s^-} \mu^\pi(s^-) P(s|s^-, \pi(s^-)).$$

This says that the (discounted) probability of visiting state s is equal to the sum of the probability of starting in state s (as specified by the starting state distribution $P_0(s)$) and the probability of reaching s by first visiting state s^- and then executing an action that leads to state s.

4. Any symmetric prior centered on $R_{max}/2$ would suffice.

The intuition behind using an occupancy measure is that if we knew that the optimal policy would visit s with measure $\mu^*(s)$ and if exploring (s,a) would reduce our uncertainty at state s by approximately $\Delta\Delta Q(s_0|s,a)$, then a reasonable estimate of the impact on $\Delta V(s_0)$ would be $\mu^*(s)\Delta\Delta Q(s_0|s,a)$. Unfortunately, we don't know μ^* because we don't know the optimal policy. We consider two other occupancy measures instead: μ^{OUU} and $\overline{\mu}$.

The first measure, μ^{OUU} is computed based on the principle of optimism under uncertainty. Specifically, define $\pi^{OUU}(s) := \max_a Q_{upper}(s,a)$ to be the policy that chooses the action that maximizes the upper confidence bound on the Q function. This is the policy followed by MBIE and most other upper-confidence bound methods. This gives us the DDV-OUU heuristic.

Definition 6 *The DDV-OUU heuristic explores the state action pair (s,a) that maximizes*

$$\mu^{OUU}(s)\Delta\Delta Q(s_0|s,a).$$

The second measure $\overline{\mu}$ is computed based on an upper bound of the occupancy measure over all possible policies. It gives us the DDV-UPPER heuristic.

Definition 7 *The DDV-UPPER heuristic explores the state action pair (s,a) that maximizes*

$$\overline{\mu}(s)\Delta\Delta Q(s_0|s,a).$$

The next section defines $\overline{\mu}$ and proves a property that may be of independent interest.

5.2.1 AN UPPER BOUND ON THE OCCUPANCY MEASURE

The purpose of this section is to introduce $\overline{\mu}$, which is an upper bound on the occupancy measure of any optimal policy for a restricted set of MDPs $\widetilde{\mathscr{M}}$. This section defines this measure and presents a dynamic programming algorithm to compute it. The attractive aspect of $\overline{\mu}$ is that it can be computed without knowing the optimal policy. In this sense, it is analogous to the value function, which value iteration computes in a policy-independent way.

To define $\overline{\mu}$, we must first define the set $\widetilde{\mathscr{M}}$ of MDPs. At each point during the execution of DDV, the states S of the unknown MDP can be partitioned into three sets: (a) the *unobserved states s* (i.e., $N(s^-,a^-,s) = 0$ for all s^-,a^-); (b) the *observed but unexplored states s* (i.e., $\exists(s^-,a^-)N(s^-,a^-,s) > 0$ but $N(s,a) = 0$ for all a), and (c) *the (partially) explored states s* (i.e., $N(s,a,s') > 0$ for some a). Consider the set $\widetilde{\mathscr{M}} = \langle \tilde{S}, \tilde{A}, \tilde{T}, \tilde{R}, s_0 \rangle$ of MDPs satisfying the following properties:

- $\tilde{S}$ consists of all states s that have been either observed or explored,
- $\tilde{A} = A$, the set of actions in the unknown MDP,
- $\tilde{T}$ consists of any transition function T such that for explored states s and all actions a, $T(s,a,\cdot) \in CI^{GT}(\hat{P}(s,a),\delta)$. For all observed but not explored states s, $T(s,a,s) = 1$ for all a, so they enter self-loops.
- $\tilde{R}$: For explored (s,a) pairs, $\tilde{R}(s,a) = R(s,a)$. For unexplored (s,a) pairs, $\tilde{R}(s,a) \in [0, R_{max}]$.
- s_0 is the artificial start state.

The set $\widetilde{\mathscr{M}}$ contains all MDPs consistent with the observations with the following restrictions. First, the MDPs do not contain any of the unobserved states. Second, the unexplored states contain self-loops and hence do not transition to any other states.

Define $P_{upper}(s'|s,a)$ as follows:

$$P_{upper}(s'|s,a) = \max_{\tilde{P}(s,a) \in CI^{GT}(P,\delta)} \tilde{P}(s'|s,a).$$

Define $\overline{\mu}$ as the solution to the following dynamic program. For all states s,

$$\overline{\mu}(s) = \sum_{s^- \in pred(s)} \max_{a^-} \gamma P_{upper}(s|s^-,a^-)\overline{\mu}(s^-). \tag{14}$$

The intuition is that we allow each predecessor s^- of s to choose the action a^- that would send the most probability mass to s and hence give the biggest value of $\overline{\mu}(s)$. These action choices a^- are not required to be consistent for multiple successors of s^-. We fix $\overline{\mu}(s_0) = \mu(s_0) = 1$. (Recall, that s_0 is an artificial start state. It is not reachable from any other state—including itself—so $\mu(s_0) = 1$ for all policies.)

Lemma 8 *For all MDPs $\widetilde{M} \in \widetilde{\mathscr{M}}$, $\overline{\mu}(s) \geq \mu^{\pi^*(\widetilde{M})}(s)$, where $\pi^*(\widetilde{M})$ is any optimal policy of $\widetilde{M}$.*

Proof By construction, $P_{upper}(s'|s,a)$ is the maximum over all transition distributions in $\widetilde{\mathscr{M}}$ of the probability of $(s,a) \to s'$. According to (14), the probability flowing to s is the maximum possible over all policies executed in the predecessor states $\{s^-\}$. Finally, all probability reaching a state s must come from its known predecessors $pred(s)$, because all observed but unexplored states only have self-transitions and hence cannot reach s or any of its predecessors. ■

In earlier work, Smith and Simmons (2006) employed a less general path-specific bound on μ as a heuristic for focusing Real-Time Dynamic Programming (a method that assumes a full model of the MDP is available).

5.2.2 SOUNDNESS OF DDV-OUU AND DDV-UPPER

We now show that DDV, using either of these heuristics, produces an ε-optimal policy with probability at least $1-\delta$ after making only polynomially-many simulator calls. The steps in this proof closely follow previous proofs by Strehl and Littman (2008) and Even-Dar et al. (2006).

Theorem 9 (DDV is PAC-RL) *There exists a sample size m polynomial in $|S|$, $|A|$, $1/\varepsilon$, $1/\delta$, $1/(1-\gamma)$, R_{max}, such that* DDV$(s_0, F, \varepsilon, \delta/(m|S||A|))$ *with either the DDV-OUU or the DDV-UPPER heuristic terminates after no more than $m|S||A|$ calls on the simulator and returns a policy π such that $|V^\pi(s_0) - V^*(s_0)| < \varepsilon$ with probability $1-\delta$.*

Proof First, note that every sample drawn by DDV will shrink the confidence interval for some $Q(s,a)$. Hence, these intervals will eventually become tight enough to make the termination condition true. To establish a rough bound on sample complexity, let us suppose that each state must be sampled enough so that $\Delta Q(s,a) = Q_{upper}(s,a) - Q_{lower}(s,a) \leq \varepsilon$.

This will cause termination. Consider state s_0 and let $a_{upper} = \text{argmax}_a Q_{upper}(s_0,a)$ be the action chosen by the OUU policy. Then the upper bound on s is $V_{upper}(s) = Q_{upper}(s,a_{upper})$, and the lower bound on S is $V_{lower}(s_0) = \max_a Q_{lower}(s_0, a) \geq Q_{lower}(s_0, a_{upper})$. Hence, the difference $V_{upper}(s_0) - V_{lower}(s_0) \leq \varepsilon$.

How many samples are required to ensure that $\Delta Q(s,a) \leq \varepsilon$ for all (s,a)? We can bound $Q_{upper}(s,a) - Q_{lower}(s,a)$ as follows.

$$\begin{aligned} Q_{upper}(s,a) - Q_{lower}(s,a) &= R(s,a) + \gamma \max_{\tilde{P} \in CI(\hat{P}(s,a),\delta')} \sum_{s'} \tilde{P}(s'|s,a) V_{upper}(s') \\ &\quad -R(s,a) - \gamma \min_{\tilde{P} \in CI(\hat{P}(s,a),\delta')} \sum_{s'} \tilde{P}(s'|s,a) V_{lower}(s') \end{aligned}$$

Let P_{upper} be the $\tilde{P}$ chosen in the max and P_{lower} be the $\tilde{P}$ chosen in the min. At termination, we know that in every state $V_{upper} \leq V_{lower} + \varepsilon$. Substituting these and simplifying gives

$$Q_{upper}(s,a) - Q_{lower}(s,a) \leq \gamma \sum_{s'} [P_{upper}(s'|s,a) - P_{lower}(s'|s,a)] V_{lower}(s') + \gamma\varepsilon.$$

We make two approximations: $P_{upper}(s'|s,a) - P_{lower}(s'|s,a) \leq |P_{upper}(s'|s,a) - P_{lower}(s'|s,a)|$ and $V_{lower}(s') \leq \frac{R_{max}}{1-\gamma}$. This yields

$$Q_{upper}(s,a) - Q_{lower}(s,a) \leq \gamma \frac{R_{max}}{1-\gamma} \sum_{s'} |P_{upper}(s'|s,a) - P_{lower}(s'|s,a)| + \gamma\varepsilon.$$

We know that $\|P_{upper}(\cdot|s,a) - P_{lower}(\cdot|s,a)\|_1 \leq 2\omega$, because both distributions belong to the L_1 ball of radius ω around the maximum likelihood estimate $\hat{P}$.

$$Q_{upper}(s,a) - Q_{lower}(s,a) \leq \gamma \frac{R_{max}}{1-\gamma} 2\omega + \gamma\varepsilon.$$

Setting this less than or equal to ε and solving for ω gives

$$\omega \leq \frac{\varepsilon(1-\gamma)^2}{2\gamma R_{max}}.$$

We know that

$$\omega = \sqrt{\frac{2[\ln(2^{|S|}-2) - \ln\delta']}{N}}.$$

To set δ', we must divide δ by the maximum number of confidence intervals computed by the algorithm. This will be $2|S||A|N$, because we compute two intervals (upper and lower) for ever (s,a). Plugging the value for δ' in and simplifying gives the following equation:

$$N \geq \frac{\gamma^2 8 R_{max}^2 [\ln(2^{|S|}-2) - \ln\delta + \ln 2|S||A| + \ln N]}{\varepsilon^2(1-\gamma)^4}.$$

This has no closed form solution. However, as Strehl and Littman note, there exists a constant C such that if $N \geq 2C \ln C$ then $N \geq C \ln N$. Hence, the $\ln N$ term on the right-hand side will only require a small increase in N. Hence

$$N = O\left(\frac{\gamma^2 R_{max}^2 |S| + \ln|S||A|/\delta}{\varepsilon^2(1-\gamma)^4}\right).$$

In the worst case, we must draw N samples for every state-action pair, so

$$m = O\left(|S|^2|A|\frac{\gamma^2 R_{max}^2 + \ln|S||A|/\delta}{\varepsilon^2(1-\gamma)^4}\right),$$

which is polynomial in all of the relevant parameters.

To prove that the policy output by DDV is within ε of optimal with probability $1-\delta$, note that the following relationships hold:

$$V_{upper}(s_0) \geq V^*(s_0) \geq V^{\pi_{lower}}(s_0) \geq V_{lower}(s_0).$$

The inequalities $V_{upper}(s_0) \geq V^*(s_0) \geq V_{lower}(s_0)$ hold (with probability $1-\delta$) by the admissibility of the confidence intervals. The inequality $V^*(s_0) \geq V^{\pi_{lower}}(s_0)$ holds, because the true value of any policy is no larger than the value of the optimal policy. The last inequality, $V^{\pi_{lower}}(s_0) \geq V_{lower}(s_0)$, holds because extended value iteration estimates the value of π_{lower} by backing up the values V_{lower} of the successor states. At termination, $V_{upper}(s_0) - V_{lower}(s_0) \leq \varepsilon$. Hence, $V^*(s_0) - V^{\pi_{lower}}(s_0) \leq \varepsilon$. ■

5.3 Experimental Evaluation on Exploration Heuristics

We conducted an experimental study to assess the effectiveness of DDV-OUU and DDV-UPPER and compare them to the exploration heuristics of MBIE (with reset) and Fiechter's algorithm.

5.3.1 METHODS

We conducted two experiments. The goal of both experiments was to compare the number of simulator calls required by each algorithm to achieve a target value ε for the width of the confidence interval, $\Delta V(s_0)$, on the value of the optimal policy in the starting state s_0. For problems where the value $V^*(s_0)$ of the optimal policy is known, we define $\varepsilon = \alpha V^*(s_0)$ and plot the required sample size as a function of α. For the tamarisk problems, where $V^*(s_0)$ is not known, we define $\varepsilon = \alpha R_{max}$ and again plot the required sample size as a function of α. This is a natural way for the user to define the required accuracy ε.

In the first experiment, we employed four MDPs: the RiverSwim and SixArms benchmarks, which have been studied by Strehl and Littman (2004, 2008), and two instances of our tamarisk management MDPs ($E = 3$, $H = 1$) and ($E = 3$, $H = 2$). Each of the tamarisk MDPs implemented a budget constraint that permits a non-"do nothing" action in only one edge in each time step. In the $E = 3, H = 2$ MDP, we included exogenous arrivals using the parameters described in Section 3 (up to 10 seeds per species per edge; Bernoulli parameters are 0.1 for tamarisk and 0.4 for native plants). The $E = 3, H = 1$ tamarisk MDP has 7 actions and 27 states, and the $E = 3, H = 2$ MDP has 7 actions and 216 states. The discount factor was set to 0.9 in all four MDPs.

Each algorithm was executed for one million simulator calls. Instead of performing dynamic programming updates (for extended value iteration and occupancy measure computation) after every simulator call, we computed them on the following schedule. For MBIE-reset, we performed dynamic programming after each complete trajectory. For DDV-OUU and DDV-UPPER, we performed dynamic programming after every 10 simulator calls. Extended value iteration gives us the confidence limits $V_{lower}(s_0)$ and $V_{upper}(s_0)$ for the starting state from which we also computed

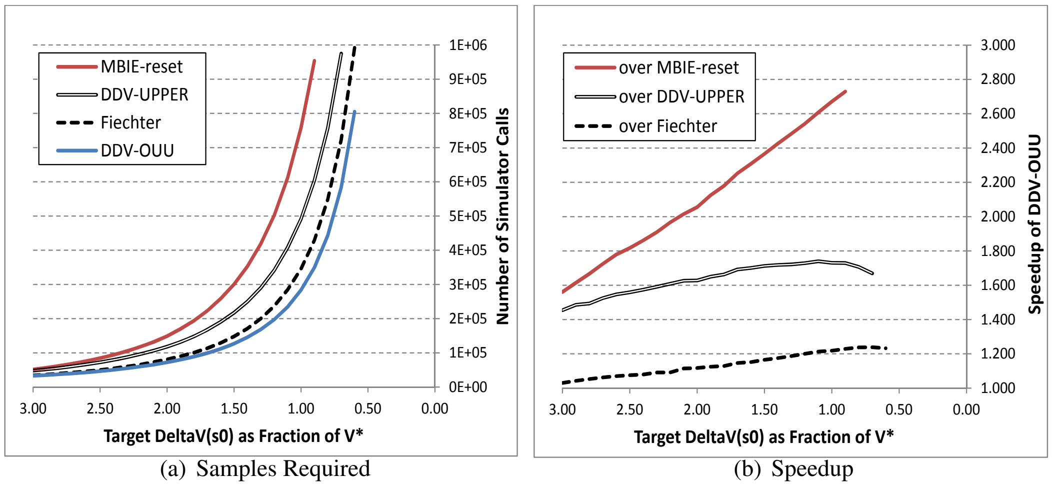


Figure 3: RiverSwim results: (a) Number of samples required by MBIE-reset, Fiechter, DDV-UPPER, and DDV-OUU to achieve various target confidence interval widths $\Delta V(s_0)$. (b) Speedup of DDV-OUU over the algorithms.

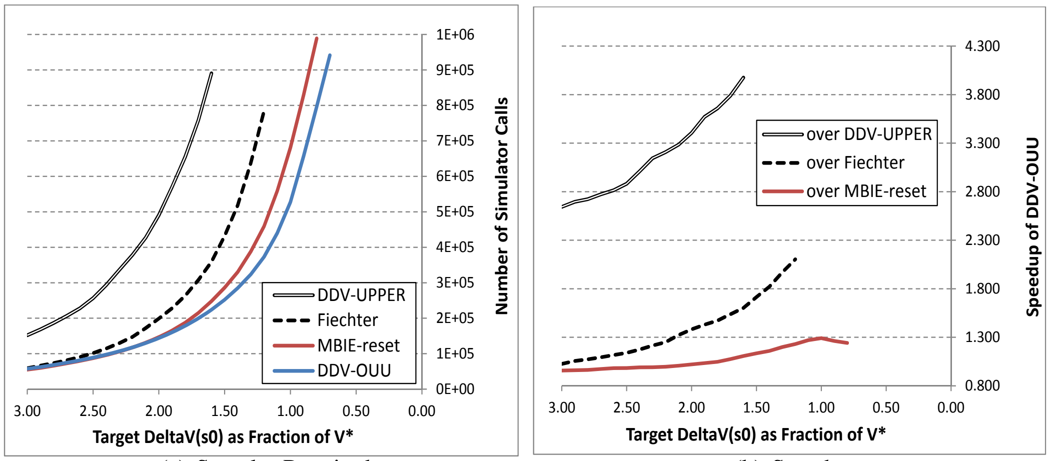
$\Delta V(s_0) = V_{upper}(s_0) - V_{lower}(s_0)$. The experiment was repeated 15 times, and the average value of $\Delta V(s_0)$ was computed. For each MDP, we defined a range of target values for $\Delta V(s_0)$ and computed the average number of samples m required by each algorithm to achieve each target value. By plotting these values, we can see how the sample size increases as we seek smaller target values for $\Delta V(s_0)$. We also computed the speedup of DDV-OUU over each of the other algorithms, according to the formula $m_{alg}/m_{\text{DDV-OUU}}$, and plotted the result for each MDP.

We also measured the total amount of CPU time required by each algorithm to complete the one million simulator calls. Because the simulators in these four MDPs are very efficient, the CPU time primarily measures the cost of the various dynamic programming computations. For Fiechter, these involve setting up and solving the exploration MDP. For MBIE-reset, the primary cost is performing extended value iteration to update V_{upper} and π^{OUU}. For the DDV methods, the cost involves extended value iteration for both V_{upper} and V_{lower} as well as the dynamic program for μ.

In the second experiment, we ran all four algorithms on the RiverSwim and SixArms problems until either 40 million calls had been made to the simulator or until $\Delta V(s_0) \leq \alpha R_{max}$, where $\alpha = 0.1$ and $R_{max} = 10000$ (for RiverSwim) and $R_{max} = 6000$ (for SixArms).

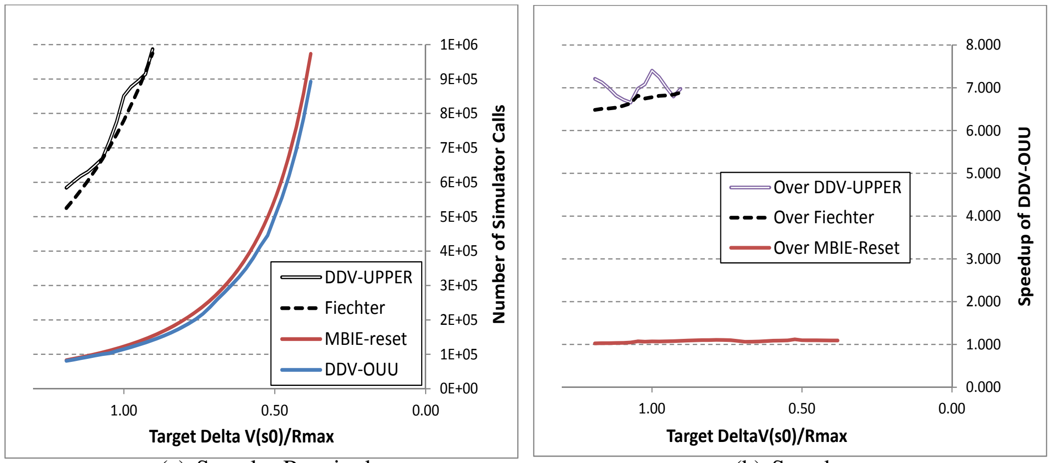
5.3.2 RESULTS

Figures 3, 4, 5, and 6 show the results for the first experiment. In each figure, the left plot shows how the required sample size increases as the target width for $\Delta V(s_0)$ is made smaller. In each figure, the right plot shows the corresponding speedup of DDV-OUU over each of the other algorithms. In all cases, DDV-OUU generally requires the fewest number of samples to reach the target width, and DDV-UPPER generally requires the most. The poor behavior of DDV-UPPER suggests that the policy-free occupancy measure $\overline{\mu}$ is too loose to provide a competitive heuristic.



(a) Samples Required (b) Speedup

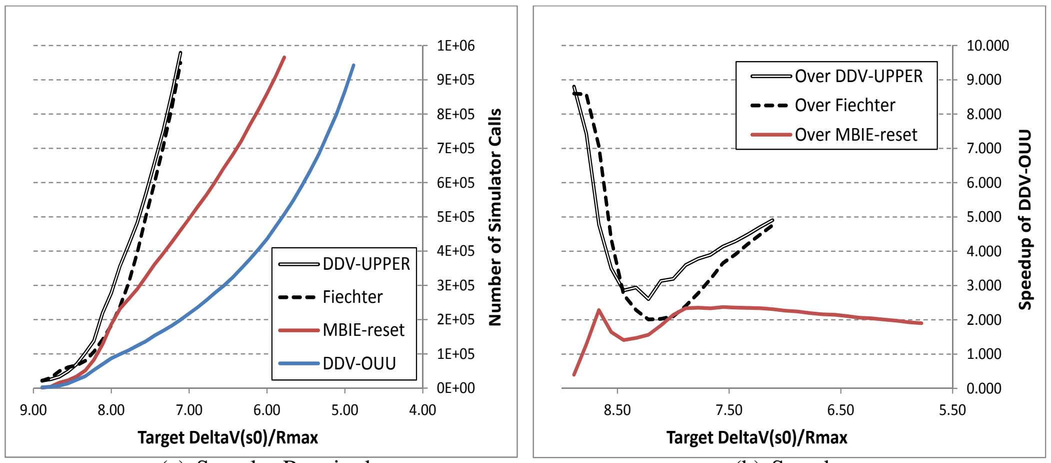
Figure 4: SixArms results: (a) Number of samples required by MBIE-reset, Fiechter, DDV-UPPER, and DDV-OUU to achieve various target confidence interval widths $\Delta V(s_0)$. (b) Speedup of DDV-OUU over the other algorithms.



(a) Samples Required (b) Speedup

Figure 5: Tamarisk with $E = 3$ and $H = 1$ results: (a) Number of samples required by MBIE-reset, Fiechter, DDV-UPPER, and DDV-OUU to achieve various target confidence interval widths $\Delta V(s_0)$. (b) Speedup of DDV-OUU over the other algorithms.

The relative performance of MBIE-reset and Fiechter's algorithm varies dramatically across the four MDPs. On RiverSwim, Fiechter's method is almost as good as DDV-OUU: DDV-OUU shows a speedup of at most 1.23 (23%) over Fiechter. In contrast, MBIE-reset performs much worse. But on SixArms, it is MBIE-reset that is the closest competitor to DDV-OUU. In fact, MBIE-reset is



(a) Samples Required (b) Speedup

Figure 6: Tamarisk with $E = 3$ and $H = 2$ results: (a) Number of samples required by MBIE-reset, Fiechter, DDV-UPPER, and DDV-OUU to achieve various target confidence interval widths $\Delta V(s_0)$. (b) Speedup of DDV-OUU over the other algorithms.

MDP	Algorithm			
	DDV-UPPER (ms/call)	DDV-OUU (ms/call)	MBIE-reset (ms/call)	Fiechter (ms/call)
RiverSwim	9.59	9.92	3.73	3.29
SixArms	15.54	48.97	10.53	4.87
Tamarisk (E=3 and H=1)	11.93	8.13	4.81	4.68
Tamarisk (E=3 and H=2)	187.30	166.79	12.63	18.79

Table 1: RiverSwim clock time per simulator call.

Quantity	Algorithm				
	DDV-UPPER	DDV-OUU	MBIE-reset	Fiechter	Optimal
$V_{upper}(s_0)$	2967.2	2936.6	3001.5	2952.6	2203
$V_{lower}(s_0)$	1967.2	1936.6	2001.5	1952.6	2203
$\Delta V(s_0)$	1000	1000	1000	1000	
Simulator Calls ($\times 10^6$)	2.31	**1.44**	4.05	1.76	

Table 2: RiverSwim confidence intervals and required sample size to achieve target $\Delta V(s_0) = 1000$.

Quantity	Algorithm				
	DDV-UPPER	DDV-OUU	MBIE-reset	Fiechter	Optimal
$V_{upper}(s_0)$	5576.7	5203.9	5242.4	5672.8	4954
$V_{lower}(s_0)$	4140.4	4603.9	4642.4	3997.7	4954
$\Delta V(s_0)$	1436.3	600	600	1675.1	
Simulator Calls ($\times 10^6$)	40.0	**14.5**	19.3	40.0	

Table 3: SixArms confidence intervals and required sample size to achieve the target $\Delta V(s_0) = 600$.

actually better than DDV-OUU for target values larger than 2.1, but as the target width for $\Delta V(s_0)$ is made smaller, DDV-OUU scales much better. On the tamarisk $R = 3$ $H = 1$ problem, MBIE-reset is again almost as good as DDV-OUU. The maximum speedup produced by DDV-OUU is 1.11. Finally, on the tamarisk $R = 3$ $H = 2$ problem, DDV-OUU is definitely superior to MBIE-reset and achieves speedups in the 1.9 to 2.3 range. Surprisingly, on this problem, Fiechter's method is sometimes worse than DDV-UPPER.

The CPU time consumed per simulator call by each algorithm on each problem is reported in Table 1. Not surprisingly, MBIE-reset and Fiechter have much lower cost than the DDV methods. All of these methods are designed for problems where the simulator is extremely expensive. For example, in the work of Houtman et al. (2013) on wildfire management, one call to the simulator can take several minutes. In such problems, the overhead of complex algorithms such as DDV more than pays for itself by reducing the number of simulator calls.

Tables 2 and 3 report the results of the second experiment. The results are consistent with those of the first experiment. DDV-OUU reaches the target $\Delta V(s_0)$ with the smallest number of simulator calls on both problems. On RiverSwim, Fiechter's method is second best, whereas on SixArms, MBIE-reset is second best. On SixArms, DDV-UPPER and Fiechter did not reach the target accuracy within the limit of 40 million simulator calls.

5.3.3 DISCUSSION

The experiments show that DDV-OUU is the most effective of the four algorithms and that it achieves substantial speedups over the other three algorithms (maximum speedups of 2.73 to 7.42 across the four problems).

These results contrast with our previous work (Dietterich, Alkaee Taleghan, and Crowley, 2013) in which we showed that DDV-UPPER is better than MBIE. The key difference is that in the present paper, we are comparing against MBIE with reset, whereas in the previous work, we compared against MBIE without reset. Without resetting, MBIE can spend most of its time in regions of the MDP that are far from the start state, so it can fail to find a good policy for s_0. This behavior also explains the poor performance of Q-learning reported in Dietterich et al. (2013).

6. Summary and Conclusions

This paper has addressed the problem of MDP planning when the MDP is defined by an expensive simulator. In this setting, the planning phase is separate from the execution phase, so there is no tradeoff between exploration and exploitation. Instead, the goal is to compute a PAC-optimal policy while minimizing the number of calls to the simulator. The policy is designed to optimize the cumulative discounted reward starting in the current real-world state s_0. Unlike in most published RL papers, which typically assume that the MDP is ergodic, the starting state of our ecosystem management problems is typically a transient state.

The paper makes two contributions. First, it shows how to combine the Good-Turing estimate with the L_1-confidence region of Weissman et al. (2003) to obtain tighter confidence intervals (and hence, earlier termination) in sparse MDPs. Second, it shows how to use occupancy measures to create better exploration heuristics. The paper introduced a new policy-independent upper bound $\overline{\mu}$ on the occupancy measure of the optimal policy and applied this to define the DDV-UPPER algorithm. The paper also employed an occupancy measure μ^{OUU} based on the "optimism under uncertainty" principle to define the DDV-OUU algorithm.

The $\overline{\mu}$ measure is potentially of independent interest. Like the value function computed during value iteration, it does not quantify the behavior of any particular policy. This means that it can be computed without needing to have a specific policy to evaluate. However, the DDV-UPPER exploration heuristic did not perform very well. We have two possible explanations for this. First, $\overline{\mu}$ can be a very loose upper bound on the optimal occupancy measure μ^*. Perhaps this leads DDV-UPPER to place too much weight on unfruitful state-action pairs. Second, it is possible that while DDV-UPPER is optimizing the one-step gain in $\Delta\Delta V(s_0)$ (as it is designed to do), DDV-OUU does a better job of optimizing gains over the longer term. Further experimentation is needed to determine which of these explanations is correct.

Our DDV-OUU method gave the best performance in all of our experiments. This is yet another confirmation of the power of the "Optimism Principle" (Buşoniu and Munos, 2012) in exploration. Hence, we recommend it for solving simulator-defined MDP planning problems. We are applying it to solve moderate-sized instances of our tamarisk MDPs. However, additional algorithm innovations will be required to solve much larger tamarisk instances.

Three promising directions for future research are (a) exploiting tighter confidence interval methods, such as the Empirical Bernstein Bound (Audibert et al., 2009; Szita and Szepesvári, 2010) or improvements on the Good-Turing estimate (Orlitsky et al., 2003; Valiant and Valiant, 2013), (b) explicitly formulating the MDP planning problem in terms of sequential inference (Wald, 1945), which would remove the independence assumption in the union bound for partitioning δ, and (c) studying exploration methods based on posterior sampling (Thompson, 1933).

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grants 0832804 and 1331932.

References

Jean Yves Audibert, Remi Munos, and Csaba Szepesvári. Exploration-exploitation trade-off using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902, 2009.

Mohammad Gheshlaghi Azar, Remi Munos, and Hilbert J Kappen. On the sample complexity of reinforcement learning with a generative model. In *Proceedings of the 29th International*

Conference on Machine Learning (ICML 2012), 2012.

Richard Bellman. *Dynamic Programming*. Princeton University Press, New Jersey, 1957.

Lucian Buşoniu and Remi Munos. Optimistic planning for Markov decision processes. In *15th International Conference on Artificial Intelligence and Statistics (AI-STATS-12)*, 2012.

Herman Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, 23(4):493–507, 1952.

Thomas G Dietterich, Majid Alkaee Taleghan, and Mark Crowley. PAC optimal planning for invasive species management: improved exploration for reinforcement learning from simulator-defined MDPs. In *Association for the Advancement of Artificial Intelligence AAAI 2013 Conference (AAAI-2013)*, 2013.

Joseph M DiTomaso and Carl E Bell. *Proceedings of the Saltcedar Management Workshop*. www.invasivespeciesinfo.gov/docs/news/workshopJun96/index.html, Rancho Mirage, CA, 1996.

Eyal Even-Dar, Shie Mannor, and Yishay Mansour. PAC bounds for multi-armed bandit and Markov decision processes. In *Proceedings of the 15th Annual Conference on Computational Learning Theory*, pages 255–270, London, 2002. Springer-Verlag.

Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of Machine Learning Research*, 7:1079–1105, 2006.

Claude-Nicolas Fiechter. Efficient reinforcement learning. In *Proceedings of the Seventh Annual ACM Conference on Computational Learning Theory*, pages 88–97. ACM Press, 1994.

Claude-Nicolas Fiechter. *Design and Analysis of Efficient Reinforcement Learning Algorithms*. PhD thesis, University of Pittsburgh, Pittsburgh, PA, USA, 1997.

Irving John Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(3):237–264, 1953.

Rachel M. Houtman, Claire A Montgomery, Aaron R Gagnon, David E. Calkin, Thomas G. Dietterich, Sean McGregor, and Mark Crowley. Allowing a wildfire to burn: estimating the effect on future fire suppression costs. *International Journal of Wildland Fire*, 22:871–882, 2013.

Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11:1563–1600, 2010.

Sham M. Kakade. *On the Sample Complexity of Reinforcement Learning*. Doctoral dissertation, University College London, 2003.

Michael Kearns and Lawrence Saul. Large deviation methods for approximate probabilistic inference, with rates of convergence. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pages 311–319, 1998.

Michael J Kearns, Yishay Mansour, and Andrew Y Ng. A sparse sampling algorithm for near-optimal planning in large Markov decision processes. In *IJCAI*, pages 1231–1324, 1999.

Shie Mannor, O Mebel, and H Xu. Lightning does not strike twice: robust mdps with coupled uncertainty. In *Proceedings of the 29th International Conference on Machine Learning (ICML 2012)*, 2012.

David McAllester and Luis Ortiz. Concentration inequalities for the missing mass and for histogram rule error. *Journal of Machine Learning Research*, 4:895–911, 2003.

David McAllester and Robert E Schapire. On the convergence rate of Good-Turing estimators. In *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory*, pages 1–6, 2000.

Rachata Muneepeerakul, Simon A Levin, Andrea Rinaldo, and Ignacio Rodriguez-Iturbe. On biodiversity in river networks: a trade-off metapopulation model and comparative analysis. *Water Resources Research*, 43(7):1–11, 2007.

Alon Orlitsky, Narayana P. Santhanam, and Junan Zhang. Always Good Turing: asymptotically optimal probability estimation. *Science*, 302(5644):427–431, 2003.

Martin Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Mathematical Statistics. Wiley, 1994.

Trey Smith and Reid Simmons. Focused real-time dynamic programming for MDPs: squeezing more out of a heuristic. In *AAAI 2006*, pages 1227–1232, 2006.

Scott M Stenquist. *Saltcedar Management and Riparian Restoration Workshop*. www.invasivespeciesinfo.gov/docs/news/workshopSep96/index.html, Las Vegas, NV, 1996.

Alexander Strehl and Michael Littman. An analysis of model-based interval estimation for Markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331, 2008.

Alexander L Strehl and Michael L Littman. An empirical evaluation of interval estimation for Markov decision processes. In *16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2004)*, pages 128–135, 2004.

Umar Syed, Michael Bowling, and Robert Schapire. Apprenticeship learning using linear programming. In *International Conference on Machine Learning*, Helsinki, Finland, 2008.

Istvan Szita and Csaba Szepesvári. Model-based reinforcement learning with nearly tight exploration complexity bounds. In Johannes Fürnkranz and Thorsten Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 1031–1038, 2010.

Aviv Tamar, Shie Mannor, and Huan Xu. Scaling up robust MDPs using function approximation. In *ICML 2014*, volume 32, 2014.

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3):285–294, 1933.

Gregory Valiant and Paul Valiant. Estimating the unseen: improved estimators for entropy and other properties. In *Neural Information Processing Systems 2013*, pages 1–9, 2013.

Abraham Wald. Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, 16(2):117–186, 1945.

Thomas J Walsh, Sergiu Goschin, and Michael L Littman. Integrating sample-based planning and model-based reinforcement learning. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, Atlanta, GA, 2010.

Tsachy Weissman, Erik Ordentlich, Gadiel Seroussi, Sergio Verdu, and Marcelo J Weinberger. Inequalities for the L1 deviation of the empirical distribution. Technical report, HP Labs, 2003.