Efficient Exploration for Constrained MDPs

Majid Alkaee Taleghan, Thomas G. Dietterich

School of Electrical Engineering and Computer Science
Oregon State University
Corvallis, OR 97331
alkaee,tgd@oregonstate.edu

Abstract

Given a Markov Decision Process (MDP) defined by a simulator, a designated starting state s_0, and a downside risk constraint defined as the probability of reaching catastrophic states, our goal is to find a stationary deterministic policy π that with probability $1 - \delta$ achieves a value $V^{\pi}(s_0)$ that is within ϵ of the value of the optimal stationary deterministic ν-feasible policy, $V^{*}(s_0)$, while economizing on the number of calls to the simulator. This paper presents the first **PAC-Safe-RL** algorithm for this purpose. The algorithm extends PAC-RL algorithms for efficient exploration while providing guarantees that the downside constraint is satisfied. Experiments comparing our ConstrainedDDV algorithm to baselines show substantial reductions in the number of simulator calls required to find a feasible policy.

Introduction

This work is inspired by problems in natural resource management centered on the challenge of invasive species (Dietterich, Alkaee Taleghan, and Crowley, 2013; Taleghan et al., 2015). Computing optimal management policies for ecosystems is challenging because they exhibit complex spatio-temporal interactions at multiple scales. Many ecosystem management problems can be formulated as MDP (Markov Decision Process) planning problems (Sheldon et al., 2010). In a simulator-defined MDP, the Markovian dynamics and rewards are provided by a simulator from which samples can be drawn. Simulators in natural resource management can be very expensive to execute, so that the time required to solve such MDPs is dominated by the number of calls to the simulator.

Efficient MDP planning algorithms attempt to minimize the number of simulator calls before terminating and outputting a policy that is approximately optimal with high probability (Dietterich, Alkaee Taleghan, and Crowley, 2013). For unconstrained MDPs, the standard formulation of this is the notion of PAC-RL, first introduced by Fiechter (1994). This is in contrast to the PAC-MDP formalization, which minimizes various measures of infinite-horizon regret (Strehl and Littman, 2008). A common component of PAC-RL algorithms is to compute confidence intervals and explore using the optimism principle.

In many practical scenarios, such as natural resource management, a desirable policy needs to satisfy certain constraints imposed by decision makers. In these scenarios, maximizing the expected reward does not necessarily avoid rare catastrophic or dangerous situations. For example, in conservation problems, catastrophic outcomes include species extinction, long-term establishment of an invasive species, and severe wildfires. A standard approach to finding policies that avoid catastrophic states is to assign a large negative reward to those states (García and Fernández, 2015; Geibel and Wysotzki, 2005). This is equivalent to a so-called Big M method for establishing a lexicographic preference for policies that do not enter catastrophic states. However, this approach does not quantify the risk (probability) of entering a catastrophic state, nor does it determine whether there are policies that control this risk. A better approach is to adopt the Constrained MDP (C-MDP) formalism (Altman, 1999), which seeks to maximize one objective (e.g., economic value) while satisfying one or more constraints probabilistically. For example, in invasive species management, we can define a C-MDP to minimize the economic cost of invasive species management while ensuring that the probability of native species extinction is less than a specified threshold.

Recently, Geibel and Wysotzki (2005) developed a model-free Q-learning algorithm for C-MDPs. Their formulation is applicable to episodic tasks with a combination of absorbing catastrophic and goal states. As Geramifard (2012) pointed out, the Geibel, et al., work does not provide a performance guarantee on the result.

An alternative to constrained MDPs is to consider risk-sensitive objectives such as variance penalties, value at risk (VaR), and conditional value at risk (CVaR) (García and Fernández, 2015; Altman, 1999). Var and CVar optimize the α-quantile of the expected return, and CVaR has favorable mathematical properties. While these are all very interesting approaches, we find the constrained MDP formulation easier to understand and explain to stakeholders, and for this reason, we focus our efforts on C-MDPs.

A drawback of C-MDPs is that the optimal policy can be stochastic in some cases. Specifically, if there are c constraints, then the optimal policy may be stochastic in up to c states. From the perspective of our stakeholders, this stochastic behavior is confusing and undesirable. Hence, in this paper, we aim to find a stationary deterministic policy that satisfies a downside risk constraint as well as maximizing the discounted reward. We seek to do this while economizing on the number of calls to the simulator and while providing PAC guarantees both that the constraints are satisfied and that the resulting policy is within a fixed bound of optimality. This provides the first PAC-RL algorithm for deterministic policies in C-MDPs.

The paper is organized as follows. Section 2 introduces our notation for MDPs, C-MDPs, and confidence intervals. Section 3 introduces our new planning algorithm CONSTRAINEDDDV. Section 4 provides theoretical results. Section 5 presents an experimental evaluation of CONSTRAINEDDDV and a comparison with other methods. Section 6 concludes the paper. We evaluate our algorithms on an invasive species problem as well as on standard reinforcement learning benchmarks.

Problem Definition and Notation

Let a simulator-defined MDP consist of a start state s_0, a set of possible states S, a set of possible actions A, a discount factor $\gamma \in (0, 1]$ and a stochastic function F that maps from an input state-action pair (s, a) to a resulting state s' and reward r, where $s' \sim P(s'|s, a)$ is sampled according to the (unknown) transition function, $r \sim R(r|s, a)$ is sampled according to the unknown reward function, and $0 \leq r \leq R_{max}$. In this paper, we will assume that the reward is deterministic; our methods can be easily extended to handle stochastic rewards. A (deterministic) policy π is a function mapping from states s to actions $a = \pi(s)$. The value of the policy in the start state, $V^\pi(s_0)$, is the expected discounted cumulative reward:

$$V^\pi(s_0) = E\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s = s_0\right].$$

Let $V_{max} = \frac{R_{max}}{1-\gamma}$ be the maximum possible value of any state under any policy. The corresponding minimum possible value is zero.

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 $Q^*(s, a)$. Later, we indicate these functions with subscript R to distinguish them from the catastrophe value function.

Definition 1 *The occupancy measure μ of an MDP under policy π is defined as*

$$\mu^\pi(s) = \mathbb{E}_P\left[\sum_{t=0}^{\infty} \gamma^t I[s_t = s] | s_0, \pi\right],$$

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

This is the cumulative discounted probability that the MDP will occupy state s under policy π for discount factor γ. It can be computed via dynamic programming on the Bellman flow equation (Syed, Bowling, and Schapire, 2008):

$$\mu^\pi(s) = I[s = s_0] + \gamma \sum_{s^-} \mu(s^-)P(s|s^-, \pi(s^-)). \quad (1)$$

This says that the discounted probability of visiting state s is equal to the sum of the probability that s is the starting state and the probability of reaching s by first visiting state s^- and then executing an action that leads to state s.

It is easy to show that

$$V^\pi(s_0) = \sum_s \mu^\pi(s)R(s, \pi(s)). \quad (2)$$

We adopt $\mu^{\pi^{UCB}}$ (also written as μ^{UCB}) as the occupancy measure computed based on the principle of optimism under uncertainty and maximum likelihood estimates of transition probabilities.

Let a subset of states $S_C \subset S$ be "catastrophic" states in the sense that we want to limit the probability of entering those states. Let us assume that all states in S_C are absorbing.

Definition 2 *For a policy π, the risk in state s is defined as*

$$\xi^\pi(s) = \sum_t \gamma_C^t P(s_t \in S_C|s, \pi), \quad (3)$$

which is the (discounted) probability of entering a catastrophic state when following π. γ_C denotes the catastrophe discount factor.

As a learning algorithm explores the MDP, it collects the following statistics. Let $N(s, a)$ be the number of times state-action pair (s, a) is simulated during learning and $N(s) = \sum_a N(s, a)$. Let $N(s, a, s')$ be the corresponding number of times that s' has been observed as the resulting state. Let $R(s, a)$ be the observed reward. Let $\hat{P}(s'|s, a) = N(s, a, s')/N(s, a)$ be the maximum likelihood estimate for $P(s'|s, a)$.

A $1 - \delta$ confidence interval is a pair of random variables $\underline{V}(s_0), \overline{V}(s_0)$ such that with probability $1 - \delta$, $\underline{V}(s_0) \leq V^\pi(s_0) \leq \overline{V}(s_0)$. Similarly, $\underline{Q}(s, a)$ and $\overline{Q}(s, a)$ denote the confidence bounds over the action-value functions. We follow the "Optimism Under Uncertainty" principle, and denote by π^{UCB} the policy based on an upper confidence bound on the action-value function, $\pi^{UCB}(s) = \arg\max_a \overline{Q}(s, a)$.

Definition 3 *(Fiechter, 1994). A learning algorithm is PAC-RL if for any discounted MDP $(S, A, P, R, \gamma, P_0)$, $\epsilon > 0$, $1 > \delta > 0$, and $0 \leq \gamma < 1$, the algorithm halts and outputs a policy π such that*

$$\mathbb{P}[|V^*(s_0) - V^\pi(s_0)| \leq \epsilon] \geq 1 - \delta,$$

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

Optimal Policies for C-MDPs

Before delving into additional definitions for C-MDPs, let's clarify the class of optimal policies for C-MDPs. It has been shown that, unlike unconstrained MDPs, the optimal policies in C-MDPs are not necessarily stationary and deterministic and may depend on the starting state (Feinberg and Shwartz, 1996; Zadorojniy, Even, and Shwartz, 2009). In standard discounted unconstrained MDPs, one can find optimal policies that are stationary and deterministic from any state in $O\left(|S|^2|A|\right)$. In a C-MDP with two objectives (the standard value function and the risk of catastrophe), if the two objectives have unequal discount factors, then finding deterministic and stationary policies is NP-complete (Dolgov and Durfee, 2005; Feinberg, 2000; Chang, 2016). Optimal policies in C-MDPs with equal discount factors are randomized and stationary for a fixed starting state. The solution can be found by solving a linear program, where the dual variables represent the state occupancy measure, if the model is known. In our case where we only have one constraint, the optimal randomized policy is called a "1-randomized" policy (Zadorojniy, Even, and Shwartz, 2009). This means the difference between deterministic and the 1-randomized policy will arise in at most one state, where the randomized policy may choose probabilistically between two actions (Feinberg and Rothblum, 2012).

In this paper, we focus on finding a best policy in the class of stationary and deterministic policies with performance guarantees, even when a randomized policy is the optimal policy. It is a challenge to present a randomized policy to stakeholders. Feinberg (2008) points out that implementation of randomized policies is not natural in many applications, and the use of randomization procedures could increase the variance of the expected return. Boutilier and Lu (2016) also give an example of how randomized policy could be undesirable.

Additional Definitions for C-MDPs

Let Π be the space of deterministic polices over the constrained MDP $\mathcal{M}(\tau) = \langle S, A, P, R_R, R_C, \tau, \gamma, s_0\rangle$. Every policy π induces two value functions V_R^π and V_C^π. We will say two policies π_1 and π_2 are equivalent if $V_R^{\pi_1} = V_R^{\pi_2}$ and $V_C^{\pi_1} = V_C^{\pi_2}$ over all states $s \in S$. Let $\overline{\pi}$ denote the set of policies equivalent to π. Let $\overline{\pi}_1$ and $\overline{\pi}_2$ be two distinct equivalence classes of policies. We will say that $\overline{\pi}_1$ dominates $\overline{\pi}_2$ if $V_R^{\overline{\pi}_1}(s_0) \geq V_R^{\overline{\pi}_2}(s_0)$ and $V_C^{\overline{\pi}_1} \leq V_C^{\overline{\pi}_2}$. That is, $\overline{\pi}_1$ is superior in either R_R or R_C or both. An equivalence class is non-dominated if there does not exist an equivalence class that dominates it.

Let $\Pi(\tau)$ be the space of deterministic policies such that $\forall \pi \in \Pi(\tau), V_C^\pi(s_0) \leq \tau$. These are the feasible deterministic policies. An optimal feasible deterministic policy $\pi_\tau^* \in \Pi(\tau)$ satisfies

$$V_R^{\pi_\tau^*}(s_0) \geq V_R^\pi(s_0)\ \forall \pi \in \Pi(\tau).$$

Values are defined in the usual way as the expected cumulative discounted return:

$$V_C(s_0) = \mathbb{E}[\sum_t \gamma^t R_C(s_t, \pi(s_t))],$$

and

$$V_R(s_0) = \mathbb{E}[\sum \gamma^t R_R(s_t, \pi(s_t))].$$

An optimal feasible policy π_τ^* is not necessarily non-dominated. There might be another policy π' that achieves the same $V_R(s_0)$ but has larger $V_C^{\pi'}(s_0) > V_C^{\pi_\tau^*}(s_0)$ that is still feasible.

Define the Lagrangian MDP $\mathcal{L}(\lambda) = \langle S, A, P, \lambda R_R - (1-\lambda)R_C, \gamma, s_0\rangle$ whose reward function is a linear combination of R_R and R_C.

PAC-RL for Constrained MDPs

We now consider the problem of finding an approximately optimal policy by sampling from a simulator-defined Constrained MDP. We introduce the following parameters:

- τ defines the feasibility constraint. A policy π is feasible if $V_C^\pi(s_0) \leq \tau$.
- ϵ defines a tolerance on the optimality of $V_R^\pi(s_0)$.
- ν defines a tolerance on feasibility. We will accept any policy for which $|V_C^\pi(s_0) - V_C^*(s_0)| \leq \nu$, which means that in the worst case, $V_C^\pi(s_0) = \tau + \nu$.
- η controls the numerical precision of the λ values.
- δ is the confidence parameter.

Definition 4 *(Chang (2016)). A deterministic policy π is called ν-feasible if $V_C^\pi(s_0) \leq \tau + \nu$ for $\nu \geq 0$.*

Definition 5 *Let Π_L be the set of all stationary deterministic policies that are solutions to the Lagrangian MDP for some value of λ.*

Definition 6 *An algorithm is Lagrangian-PAC-SAFE-RL if, for any C-MDP $M(\tau) = \langle S, A, P, R_R, R_C, \tau, \gamma, s_0\rangle$ and any parameters $\epsilon > 0, \delta \in (0,1), \tau \in (0,1], \eta > 0$, and $\nu > 0$ the algorithm halts in time polynomial in $|S|, |A|, 1/(1-\gamma), 1/\epsilon, 1/\nu, 1/\delta$, and $1/\eta$ and does one of the following two things:*

1. *Outputs a policy $\pi \in \Pi_{\mathcal{L},\eta}$ such that with probability $1-\delta$ the following are simultaneously true:*
   (a) *$V_C^\pi(s_0) < \tau + \nu$ (π is $\tau + \nu$ feasible)*
   (b) *$V_R^{*(-\nu)}(s_0) - V_R^\pi(s_0) \leq \epsilon$ (the value of π is never less than ϵ below the value of the optimal $\tau - \nu$ feasible policy, and it may be significantly higher)*
2. *Outputs the message Fail, in which case with probability $1-\delta$ there does not exist any policy $\pi \in \Pi_{\mathcal{L},\eta}$ such that $V_C^\pi(s_0) \leq \tau + \nu$.*

This definition gives us control over how close to feasible the policy is (via ν) and how close to the optimal feasible policy its V_R return is (via ϵ).

Confidence intervals for V_R and V_C for policy evaluation

Suppose we have drawn a set of samples for various states and actions. For any fixed policy π, we can perform extended policy evaluation (i.e., extended value iteration with a fixed policy) to obtain lower and upper confidence bounds on $V_C(s_0)$ and $V_R(s_0)$. We will denote these as $\underline{V}_C^\pi(s_0)$, $\overline{V}_C^\pi(s_0)$, $\underline{V}_R^\pi(s_0)$, and $\overline{V}_R^\pi(s_0)$. Suppose our goal is to determine whether π is feasible and if it is, then to determine confidence intervals on $V_R^\pi(s_0)$. The policy π will be feasible with probability $1-\delta$ if $\overline{V}_C^\pi(s_0) \leq \tau$. Conversely, π is not feasible with probability $1-\delta$ if $\underline{V}_C^\pi(s_0) > \tau$.

Confidence intervals for V_R and V_C for policy optimization

Instead of using a fixed policy, we can set a value of λ and perform extended value iteration based on the upper confidence bound of the Lagrangian objective. This will define the $\pi^{UCB(\lambda)}$ policy. More generally, we can perform binary search on λ to find three values:

- $\underline{\lambda}$ is the largest value of $\lambda \in \Lambda$ such that $\overline{V}_C^{UCB(\lambda)}(s_0) \leq \tau$. This means that given our current sample, $\pi^{UCB(\lambda)}$ is the "best" policy (in the sense of having the largest λ) for which we can guarantee with probability $1 - \delta$ that it is feasible.
- $\overline{\lambda}$ is the largest value of $\lambda \in \Lambda$ such that $\underline{V}_C^{UCB(\lambda)}(s_0) \leq \tau$. This means that given our current sample, this is the largest value of λ that we cannot prove is not feasible.

The solid lines denote the true values of V_C and V_R. The dashed lines denote the corresponding upper and lower confidence bounds. For purposes of this section, let π^* be the policy in $\Pi(\tau, \eta)$ that maximizes $V_R^{\pi}(s_0)$. That is, π^* is τ-feasible and among all such policies it maximizes the V_R return.

Extended Value Iteration

Classical value iteration computes an optimal policy for a fixed MDP. Extended value iteration can compute optimal policy for finite-sampled optimistic/pessimistic MDPs by defining confidence intervals on the value function at each state of the MDP based on samples from that MDP. Different confidence interval methods (e.g., Hoeffding bound (Hoeffding, 1963), empirical Bernstein bound (Audibert, Munos, and Szepesvári, 2009), multinomial confidence interval (Weissman et al., 2003), etc.) at each state lead to different confidence intervals throughout the MDP. One can obtain robust policies from pessimistic MDPs (Tamar, Mannor, and Xu, 2014). Based on our experiments, the empirical Bernstein bound is the tightest bound compared to the other bounds.

The Empirical Bernstein Method: This approach uses the empirical Bernstein bound. Let $M(s, a)$ denote the sample mean of the discounted backed-up values from the successor states that result from taking action a in state s, and $Var(s, a)$ denote the sample variance of these values. We denote the upper and lower bounds on these values as $\overline{M}(s, a)$, $\underline{M}(s, a)$,$\overline{Var}(s)$, and $\underline{Var}(s)$.

$$\begin{aligned}
\overline{M}(s,a) &= \sum_{s'} \hat{P}(s'|s,a)\gamma\overline{V}(s') \\
\overline{Var}(s,a) &= \sum_{s'} \hat{P}(s'|s,a)[\gamma\overline{V}(s') - \overline{M}(s,a)]^2 \\
\overline{V}(s) &= \max_a R(s,a) + \overline{M}(s,a) + \\
&\quad \sqrt{\frac{2\overline{Var}(s)\ln(3/\delta_0)}{N(s,a)}} + \frac{3\gamma V_{max}\ln(3/\delta_0)}{N(s,a)}
\end{aligned} \tag{4}$$

The lower bounds could be defined in a similar way as above. We need to define δ_0 so that the confidence intervals hold simultaneously with probability $1 - \delta$. These equations can be iterated to convergence. At convergence, with probability $1 - \delta$, $\underline{V}(s_0) \leq V^*(s_0) \leq \overline{V}(s_0)$.

Algorithm

The extended value iteration for the Lagrangian objective computes upper and lower bounds on V_R and V_C in all states and on $Q_R(s, a)$ and $Q_C(s, a)$ in all state-action pairs. A binary search algorithm (see supplementary materials) on λ finds $\underline{\lambda}$ and $\overline{\lambda}$ to within tolerance η for a given set of samples. We will apply BINARYSEARCH to find $\underline{\lambda}$ and $\overline{\lambda}$. For $\underline{\lambda}$, we are looking for the point λ where $\overline{V}_C^{\lambda}(s_0)$ crosses τ, which is exactly what BINARYSEARCH does. For $\overline{\lambda}$, we need to find the point where $\underline{V}_C^{\lambda}(s_0)$ crosses τ, determine the value on the larger side, and then find the largest value of $\overline{\lambda}$ that achieves that value. The function NEXTLARGERLAMBDA finds the next larger value of λ that will cause the UCB policy to change by calling LAGRANGIANEVI.

The main algorithm works by maintaining an upper bound $\overline{V}_R^{UCB(\overline{\lambda}^{(-\nu)})}(s_0)$ on the value of the best $(\tau - \nu)$-feasible policy and a lower bound $\underline{V}_R^{UCB(\underline{\lambda}^{(+\nu)})}(s_0)$ on the value of the best $(\tau + \nu)$-feasible policy. Here the notation $\lambda^{(-\nu)}$ refers the $(\tau - \nu)$ feasibility and $\lambda^{(+\nu)}$ refers to $(\tau + \nu)$ feasibility. Sampling proceeds in a series of minibatches that cause these bounds to shrink toward one another. Execution terminates when $\overline{V}_R^{UCB(\overline{\lambda}^{(-\nu)})}(s_0) - \underline{V}_R^{UCB(\underline{\lambda}^{(+\nu)})}(s_0) \leq \epsilon$. This is summarized in Algorithm1).

The rationale is the following. The largest value that $V_R^{*(-\nu)}(s_0)$ could have is $\overline{V}_R^{UCB(\overline{\lambda}^{(-\nu)})}(s_0)$. The smallest value that $\pi^{UCB(\underline{\lambda}^{(+\nu)})}$ could have is $\underline{V}_R^{UCB(\underline{\lambda}^{(+\nu)})}(s_0)$. We want the value of $\pi^{UCB\underline{\lambda}^{(+\nu)})}$ to be no less than ϵ below the value of $V_R^{*(-\nu)}(s_0)$. We attain this by ensuring that $\overline{V}_R^{UCB(\underline{\lambda}^{(+\nu)})}(s_0) - \underline{V}_R^{UCB(\overline{\lambda}^{(-\nu)})}(s_0) < \epsilon$.

Correctness and Polynomial Running Time

The proofs for the following claims and theorem are provided in supplementary materials.

Claim 1 *For any fixed λ, the optimal policy π^*_λ for $\mathcal{L}(\lambda)$ is a non-dominated policy.*

Claim 2 *Let λ_1 and λ_2 be a pair of values such that $\lambda_2 = \lambda_1 - \delta$ for some positive δ. Let π_1 be a policy that optimizes the Lagrangian for $\lambda = \lambda_1$ and π_2 be the policy that optimizes the Lagrangian for $\lambda = \lambda_2$. Then one of two cases holds:*

Case 1: $V_C^{\pi_2}(s_0) = V_C^{\pi_1}(s_0)$, *and* $V_R^{\pi_2}(s_0) = V_R^{\pi_1}(s_0)$ *or*

Case 2: $\pi_1 \neq \pi_2$, $V_C^{\pi_2}(s_0) < V_C^{\pi_1}(s_0)$, *and* $V_R^{\pi_2}(s_0) < V_R^{\pi_1}(s_0)$.

Claim 3 *There exists a value λ^* such that $\forall \lambda \leq \lambda^*$, the optimal policy, π^*_λ, of the Lagrangian MDP $\mathcal{L}(\lambda)$ is feasible for $\mathcal{M}(\tau)$; that is $V_C^{\pi^*_\lambda}(s_0) \leq \tau$.*

For computational efficiency, we will not consider all possible values of λ. Instead, we discretize the space by introducing a precision parameter η. Define $\Pi_{\mathcal{L},\eta}$ to be the class of all policies in $\Pi_{\mathcal{L}}$ where $\lambda = k\eta$, for $k \in \{0, 1, \ldots, 1/\eta\}$. We will restrict our attention to only these policies.

To obtain a polynomial time sampling algorithm, we need to relax our goal (based on ideas from Chang (2016)). Let $\Pi_{\mathcal{L},\eta}(\tau)$ be the set of all policies $\pi \in \Pi_{\mathcal{L},\eta}$ such that $V_C^{\pi}(s_0) \leq \tau$. These are the τ-feasible policies. We will

Algorithm 1: CONSTRAINEDDDV$(s_0, \tau, \nu, F, \epsilon, \delta, \gamma, R_{max})$

```
1: $\underline{\lambda}^{(+\nu)} := 0$; $\overline{\lambda}^{(-\nu)} := 1$
2: CheckFeasibility:=true
3: **loop**
4:   $\overline{\lambda}^{(-\nu)} = \text{FINDUPPER}(0, 1, \max(0, \tau - \nu), \eta)$
5:   $\underline{\lambda}^{(+\nu)} = \text{FINDLOWER}(0, 1, \min(1, \tau + \nu), \eta)$
6:   **if** CheckFeasibility **then**
7:     LAGRANGIANEVI$(0, \eta, \delta)$
8:     **if** $\underline{V}_C^{UCB(0)}(s_0) \geq \tau - \nu$ **then**
9:       {there is no $(\tau - \nu)$-feasible policy}
10:      **return** No feasible policy
11:    **else if** $\overline{V}_C^{UCB(0)}(s_0) < \tau - \nu$ **then**
12:      {there is a $(\tau - \nu)$-feasible policy}
13:      CheckFeasibility:=false
14:    **end if**
15:  **end if**
16:  **if** $\left(\overline{\lambda}^{(-\nu)} = \underline{\lambda}^{(+\nu)}\right)$ **and**
     $\left(\overline{V}_R^{UCB(\overline{\lambda}^{(-\nu)})}(s_0) - \underline{V}_R^{UCB(\underline{\lambda}^{(+\nu)})}(s_0) \leq \epsilon\right)$ **then**
17:    **return** $\left(Success, \pi^{UCB(\underline{\lambda}^{(+\nu)})}\right)$
18:  **end if**
19:  Explore for a minibatch of $B$ samples using DDV
     on $\pi^{UCB(\overline{\lambda}^{(-\nu)})}$
20: **end loop**
```

be interested in two other policy classes: $\Pi_{\mathcal{L},\eta}(\tau - \nu)$ and $\Pi_{\mathcal{L},\eta}(\tau + \nu)$.

Let $\pi^{*(-\nu)} \in \Pi_{\mathcal{L},\eta}(\tau - \nu)$ be a policy that is feasible with respect to the threshold $\tau - \nu$ and that among all such policies maximizes $V_R(s_0)$. More precisely, $\pi^{*(-\nu)} = \arg\max_{\pi \in \Pi_{\mathcal{L},\eta}(\tau-\nu)} V_R^{\pi}(s_0)$.

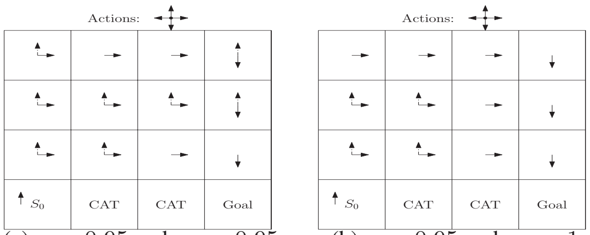
Denote the value of $\pi^{*(-\nu)}$ by $V_R^{*(-\nu)}(s_0)$. Our goal will be to output a policy $\pi \in \Pi_{\mathcal{L},\eta}(\tau + \nu)$ such that $V_R^{*(-\nu)}(s_0) - V_R^{\pi}(s_0) \leq \epsilon$ and to do so in polynomial time.

Claim 4 *The optimal value* $\lambda^* \in [\underline{\lambda}, \overline{\lambda}]$ *with probability* $1 - \delta$.

Claim 5 $\underline{V}_R^{UCB(\underline{\lambda})}(s_0) \leq V_R^*(s_0) \leq \overline{V}_R^{UCB(\overline{\lambda})}(s_0)$ *with probability* $1 - \delta$.

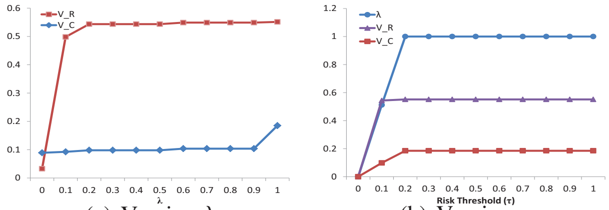
Note that the gap between $\overline{V}_R^{UCB(\overline{\lambda})}(s_0)$ and $\underline{V}_R^{UCB(\underline{\lambda})}(s_0)$ is composed of three parts. First, there is the width of the upper confidence interval $\overline{V}_R^{UCB(\overline{\lambda})}(s_0) - V_R^{UCB(\overline{\lambda})}(s_0)$. Second, there is the difference in the values of the policies $\pi^{UCB(\overline{\lambda})}$ and $\pi^{UCB(\underline{\lambda})}$, which we can write as $V_R^{UCB(\overline{\lambda})}(s_0) - V_R^{UCB(\underline{\lambda})}(s_0)$. Finally, there is the width of the lower confidence interval $V_R^{UCB(\underline{\lambda})}(s_0) - \underline{V}_R^{UCB(\underline{\lambda})}(s_0)$.

To prove correctness, we must show that, under appropriate conditions, the CONSTRAINEDDDV algorithm will terminate at line 17. Specifically, we will prove the following claim:



(a) $\gamma = 0.95$ and $\gamma_C = 0.95$ (b) $\gamma = 0.95$ and $\gamma_C = 1$

Figure 1: Derived policies for the GridWorld domain; solid arrows are when $\lambda = 1$ and dotted arrows are when $\lambda = 0$. When both policies agree on an action in a cell, only one is shown.



(a) Varying λ (b) Varying τ

Figure 2: Value of reward and risk while varying λ and risk threshold (τ) for the GridWorld domain.

Claim 6 *If* $\Pi_{\mathcal{L},\eta}(\tau - \nu)$ *and* $\Pi_{\mathcal{L},\eta}(\tau + \nu)$ *are non-empty and* $0 < \lambda^* < 1$, *then with probability* $1 - \delta$, CONSTRAINEDDDV *will terminate at line 17.*

We can also show the following.

Claim 7 *If there is no* $(\tau - \nu)$*-feasible policy, then the* CONSTRAINEDDDV *algorithm will terminate at line 10.*

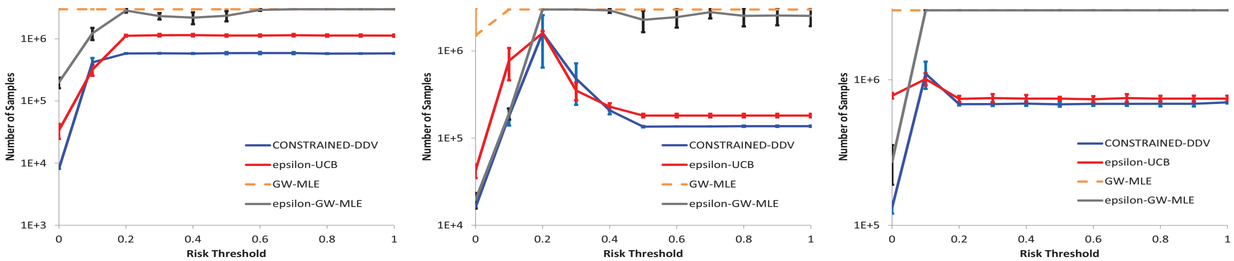
Theorem 1 CONSTRAINEDDDV *requires polynomial sample size and terminates in polynomial computation time.*

Experiments

We report three experiments. First, we study the GridWord domain shown in Figure 1(a) (there is one starting state, one goal state, and two catastrophic states). Our goal is to gain some intuition about the C-MDP formulation. Specifically, we look at the policies for $\lambda = 0$ and $\lambda = 1$.

In Figure 1, we assume the model is known. The solid lines show the optimal policy for $\lambda = 1$ (maximizing the reward), and the dotted actions show the optimal policy for $\lambda = 0$ (minimizing the risk). Notice that even for unequal discount factors, we are able to find a desirable policy, which may not be optimal. The main difference between the policies for discounted and undiscounted risk is that for discounted risk the best stationary deterministic policy that minimizes the risk takes the discount into account and moves toward the goal more slowly than the undiscounted risk policy.

In the second experiment, we solve for the optimal policy when the MDP is known while varying λ and the constraint



(a) GridWorld (b) Tamarisk $R = 3$ and $H = 1$ (c) Tamarisk $R = 3$ and $H = 2$

Figure 3: Comparison of number of samples taken by each algorithm to reach to the termination point.

threshold τ. Our goal is to determine the right answer and see the impact of τ and λ. Figure 2 shows the value of reward (V_R) and value of risk (V_C) in the starting state for the GridWorld domain while varying the value of λ (2(a)) and while varying the value of τ (2(b)). There is no feasible policy when $\tau = 0$.

In Figure 2(a), we see that when λ is close to 1, we can easily reduce V_C without any impact on V_R. As λ shrinks, V_C and V_R both shrink gradually, so that for values of τ in the range (0.185 to 0.1), there continues to be little impact on V_R. However, when λ goes from 0.1 to 0.0, we see a huge drop in V_R for very little gain in V_C. This kind of sudden drop causes difficulty for obtaining PAC results. The problem is that in this region, the confidence intervals on V_R will be very wide, and it can require a huge number of training samples to shrink them enough to achieve a width of ϵ.

In the third experiment, we compare the sample complexity of CONSTRAINEDDDV against three benchmark algorithms: GW-MLE, ϵ_g-greedy GW-MLE, ϵ_g-greedy UCB. GW-MLE is the improved version of the algorithm of Geibel and Wysotzki (2005), which basically maximizes the Lagrangian defined as $\mathcal{L}(\hat{\lambda}) = \langle S, A, \hat{P}, \hat{\lambda} R_R - (1 - \hat{\lambda}) R_C, \gamma, s_0 \rangle$, where $\hat{\lambda}$ is the maximum likelihood estimate of λ calculated over the MDP with transition probability $\hat{P}$ and reward functions R_R and R_C. The GW-MLE algorithm samples along the induced $\pi^{\hat{\lambda}}$ policy at each mini-batch. UCB algorithm calculates $\pi^{UCB} = \arg\max_a \overline{Q}_R(s, a)$ and samples along the π^{UCB} policy. Since the UCB algorithm ignores the risk in its default operation, we have added an adjustable ϵ_g parameter for better exploration. The algorithms are modified to have stopping condition similar to the lines 8 and 16 in Algorithm 1 .

We compared these algorithms on the GridWorld MDP and two instances of the tamarisk domain. In these experiments, we learn the model by sampling from the simulator. Tamarisk problem instances are configured with the number of river segments ($E = 3$) and the number of slots ($H = 1$) and ($H = 2$) (for more detail see Taleghan et al. (2015)). For the ($E = 3, H = 1$) problem, the starting state was NTE (one site contains a native species, one is invaded by tamarisk, and one site at the bottom of river is empty). For the ($E = 3, H = 2$) instance, the starting state is NTEEEE (one site contains a native species and an invasive species and the rest of the sites in the river are empty). A catastrophic state is any state in which there are no natives (species extinction). The goal state is that all sites are fully occupied by native species. We optimized the value of ϵ_g for ϵ_g-greedy GW-MLE and ϵ_g-greedy UCB algorithms among the candidate values $\epsilon_g \in \{0.01, 0.1, 0.25\}$. After sampling a minibatch of size $B = 1000$ we update the model and calculate the corresponding confidence bounds. We calculate $\overline{\lambda}$ and $\underline{\lambda}$ every 8000 samples.

In these experiments, $\gamma = \gamma_C = 0.95$, $\delta = 0.01$, $\eta = 0.01$, and $\nu = 0.025$. For the GridWorld domain, $\epsilon = 0.2$, and for the Tamarisk problems $\epsilon = 1$. The algorithms terminate either if the width of the confidence interval falls below ϵR_{max} or if 3 million samples are drawn.

We report the number of samples drawn at termination in Figure 3. The results are averaged over 10 independent runs, and the vertical axis is plotted on a log scale. Error bars indicate one standard deviation. The GW-MLE and ϵ_g-GW-MLE algorithms perform very poorly; much worse than CONSTRAINEDDDV. In many cases, they hit the 3 million maximum sampling budget without achieving the desired confidence interval width. CONSTRAINEDDDV and ϵ_g-UCB give much more similar performance, if ϵ_g is properly tuned. CONSTRAINEDDDV almost always requires smaller sample sizes, particularly for small values of τ (which would be the values normally encountered in a real application).

Conclusion

Many computational sustainability problems involving MDPs must be concerned with catastrophic outcomes such as species extinction. One approach to this is to limit the probability of catastrophic outcomes by imposing a constraint on the MDP policy, which converts the MDP into a Constrained MDP (C-MDP). Previous work on simulation-based MDP planning for constrained MDPs has not provided formal guarantees. This paper is the first to provide an algorithm with formal guarantees by extending the notion of PAC-RL algorithms to PAC-Safe-RL algorithms. We proved that this new algorithm, CONSTRAINEDDDV, is PAC-Safe-RL. Our experiments demonstrated that CONSTRAINEDDDV is also able to match or beat the sample complexity of very competitive baseline algorithms that lack formal performance guarantees.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant Nos. 1331932 and 1521687.

References

Altman, E. 1999. *Constrained Markov decision processes*, volume 7. CRC Press.

Audibert, J. Y.; Munos, R.; and Szepesvári, C. 2009. Exploration-exploitation trade-off using variance estimates in multi-armed bandits. *Theoretical Computer Science* 410(19):1876–1902.

Boutilier, C., and Lu, T. 2016. Budget allocation using weakly coupled, constrained markov decision processes. https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/45291.pdf.

Chang, H. S. 2016. Sleeping experts and bandits approach to constrained markov decision processes. *Automatica* 63:182 – 186.

Dietterich, T. G.; Alkaee Taleghan, M.; and Crowley, M. 2013. 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)*.

Dolgov, D. A., and Durfee, E. H. 2005. Stationary deterministic policies for constrained mdps with multiple rewards, costs, and discount factors. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-05)*, 1326–1332.

Feinberg, E. A., and Rothblum, U. G. 2012. Splitting randomized stationary policies in total-reward markov decision processes. *Mathematics of Operations Research* 37(1):129–153.

Feinberg, E. A., and Shwartz, A. 1996. Constrained discounted dynamic programming. *Mathematics of Operations Research* 21(4):922–945.

Feinberg, E. A. 2000. Constrained discounted markov decision processes and hamiltonian cycles. *Mathematics of Operations Research* 25(1):130–140.

Feinberg, E. A. 2008. Optimality of deterministic policies for certain stochastic control problems with multiple criteria and constraints. In *Mathematical Control Theory and Finance*. Springer. 137–148.

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

García, J., and Fernández, F. 2015. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research* 16:1437–1480.

Geibel, P., and Wysotzki, F. 2005. Risk-sensitive reinforcement learning applied to control under constraints. *J. Artif. Intell. Res.(JAIR)* 24:81–108.

Geramifard, A. 2012. *Practical Reinforcement Learning Using Representation Learning And Safe Exploration For Large Scale Markov Decision Processes*. Ph.D. Dissertation, Massachusetts Institute of Technology.

Hoeffding, W. 1963. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association* 58(301):13–30.

Sheldon, D.; Dilkina, B.; Elmachtoub, A.; Finseth, R.; Sabharwal, A.; Conrad, J.; Gomes, C.; Shmoys, D.; Allen, W.; and Amundsen, O. 2010. Maximizing the spread of cascades using network design. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, 517–526.

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

Syed, U.; Bowling, M.; and Schapire, R. 2008. Apprenticeship learning using linear programming. In *International Conference on Machine Learning*.

Taleghan, M. A.; Dietterich, T. G.; Crowley, M.; Hall, K.; and Albers, H. J. 2015. PAC optimal MDP planning with application to invasive species management. *Journal of Machine Learning Research* 16:3877–3903.

Tamar, A.; Mannor, S.; and Xu, H. 2014. Scaling up robust MDPs using function approximation. In *ICML 2014*, volume 32.

Weissman, T.; Ordentlich, E.; Seroussi, G.; Verdu, S.; and Weinberger, M. J. 2003. Inequalities for the L1 deviation of the empirical distribution. Technical report, HP Labs.

Zadorojniy, A.; Even, G.; and Shwartz, A. 2009. A strongly polynomial algorithm for controlled queues. *Mathematics of Operations Research* 34(4):992–1007.