Learning Safe Policies via Primal-Dual Methods

Santiago Paternain, Miguel Calvo-Fullana, Luiz F. O. Chamon and Alejandro Ribeiro

***Abstract*— In this paper, we study the learning of safe policies in the setting of reinforcement learning problems. This is, we aim to control a Markov Decision Process (MDP) of which we do not know the transition probabilities, but we have access to sample trajectories through experiments. We define safety as the agent remaining in a desired safe set with high probability for every time instance. We therefore consider a constrained MDP where the constraints are probabilistic. Due to the difficulty of addressing these constraints in a reinforcement learning framework, we propose an ergodic relaxation of the problem. Nonetheless, this relaxation is such that we are able to provide safety guarantees on the resulting policies. To compute these policies, we resource to a stochastic primal-dual method. We test the proposed approach in a navigation task in a grid world. The numerical results show that our algorithm is capable of dynamically adapting the policy to the environment and the required safety levels.**

I. Introduction

Markov decision processes (MDPs) [1] are stochastic control processes used ubiquitously to study robotic systems [2], control problems [3], and financal models [4]. When these models are available, optimal control laws—or *policies*—can be obtained for these processes using dynamic programming [5]. In contrast, when the underlying MDP is unknown, the policy needs to be learned from samples of the system. Typically, this is done by assigning an instantaneous *reward* to the system actions that describes the task to be learned. We then measure the total accumulated reward (known as the *value functions*) obtained by the policy to assess its quality and update it so as to maximize the expected value of this quantity [6].

A notable drawback of this method is that is not always suitable for learning dangerous, risky tasks [7]–[9]. Indeed, many applications require robust control strategies which also take into account, for instance, the variance of the accumulated reward to avoid situations in which its value on a specific realization of the process is considerably worse than its mean. Consider the case of a self-driving car deployed in an urban environment. To reach a destination as fast as possible, the optimal policy may be such that it makes risky maneuvers, such as driving close to other cars or crossing pedestrians. Due to the random components in the vehicle actions and the behavior of other cars and pedestrians, collision avoidance cannot be guaranteed.

Strategies used to overcome this limitation can be mapped in four approaches. The first formulates a robust problem in which the policy is optimized over its worst case return [7], [10]. However, these techniques generally yield policies too conservative for the average scenario and make it hard to control the trade-off between safety and performance. The second family of solutions propose to modify the instantaneous reward function so as to reflect a subjective measure balancing risk and task learning [9], [11]. Although this approach makes the risk-performance trade-off more transparent, it requires this balance to be hand-tuned, an often time consuming and challenging task that requires application- and domain-specific expert knowledge. What is more, the function of the reward is to inform the goal of the agent, not prior knowledge on how to complete it. Indeed, "the reward signal is your way of communicating to the robot *what* you want it to achieve, not *how* you want it achieved" [6, Section 3.2]. The third approach addresses this issue by modifying the learning procedure instead of the the reward. By performing safe exploration [12], [13], the agent learns from safe trajectories and is therefore biased to learn safe policies.

The last class of solutions addresses the issue of safety by including explicit constraints in the optimization problem used to learn the policy [14]–[19]. This is the approach taken in this paper. These constraints are typically probabilistic in nature, in the sense that they require certain requirements to hold with some given minimum probability. These requirements can involve, for instance, lower bounds on the value function or additional value functions [14]–[16], thus relaxing the worst case approach from [7], [10], or arbitrary functions of the state-action space [19], [20]. These constrained learning problems are solved by using regularization and relaxations so they can be written as linear programs [14], [18], by leveraging approximate trust region methods [20], or by applying primal-dual algorithms [19]. A comprehensive review of this topic can be found in [21].

In this work, we formulate safety constraints by imposing a lower bound on the probability of remaining in the safe set for all times. We then propose relaxations for the finite and infinite time operations (Section II) and provide guarantees on the ergodic safety of policies learned using our relaxed formulations (Section III-A). Namely, we show that these relaxations do not affect the safety level of the finite horizon problem and establish a safe operation horizon in the infinite case. Finally, we propose to solve the constrained optimization problems using a saddle point algorithm (Section IV) and conclude with numerical experiments in which we show that primal-dual methods are able to automatically adjust the trade-off between goal and safety (Section V).

Work supported by ARL DCIST CRA W911NF-17-2-0181 and the Intel Science and Technology Center for Wireless Autonomous Systems. The authors are with the Department of Electrical and Systems Engineering, University of Pennsylvania. Email: {spater, cfullana, luizf, aribeiro}@seas.upenn.edu.

II. Problem Formulation

Our goal is to find safe policies in reinforcement learning problems. Formally, let $\mathcal{S}$ and $\mathcal{A}$ be compact sets describing the states and actions of the agent respectively. A *policy* is a distribution $\pi_\theta(a|s)$ from which the agent draws its action $a \in \mathcal{A}$ when in state $s \in \mathcal{S}$. We assume that this distribution is parametrized by $\theta \in \mathcal{H}$, where $\mathcal{H}$ is an arbitrary Hilbert space. Every action of the agent has two consequences. First, it drives the agent to another state through the transition dynamics defined by the conditional probability $P^{a_t}_{s_t \to s_{t+1}}(s) := p(s_{t+1} = s \mid s_t, a_t)$, for time $t \in \mathbb{N}$, $s_t, s_{t+1} \in \mathcal{S}$, and $a_t \in \mathcal{A}$. This process is assumed to satisfy the Markov property $P(s_{t+1} = s \mid (s_u, a_u), \forall u \leq t) = p(s_{t+1} = s \mid s_t, a_t)$. This system is known as a Markov decision process. Second, it yields a reward taken from the function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ that informs how good was the chosen action.

The goal of the agent is to find a parametrization θ of the policy that maximizes the value function of the MDP, i.e., the expected value of the cumulative rewards obtained along a trajectory. For finite horizons, i.e., when we are concerned about the evolution of the system until a given time $T \geq 0$, the value function is defined as

$$V_T(\theta) = \mathbb{E}_{\mathbf{a}\sim\pi_\theta(\mathbf{a}|\mathbf{s})}\left[\sum_{t=0}^{T} r(s_t, a_t)\right], \tag{1}$$

where $\mathbf{a} = \{a_0, \ldots, a_T\}$ and $\mathbf{s} = \{s_0, \ldots, s_T\}$. Alternatively, we may consider the infinite horizon problem in which we want to maximize the expectation of the discounted cumulative cost

$$V_\infty(\theta) = \mathbb{E}_{\mathbf{a}\sim\pi_\theta(\mathbf{a}|\mathbf{s})}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right], \tag{2}$$

where $\gamma \in (0, 1)$ is the discount factor. The parameter γ defines how myopic the agent is. For small γ, the geometric sequence vanishes fast and the initial rewards are weighted more than those in the future. On the other hand, γ close to one corresponds to an agent that weights rewards at all times similarly. Although the formulations (1) and (2) capture different operation principles it is possible to show their equivalence when the horizon is selected randomly (see Remark 1).

As we argued in Section I, simply maximizing V_T or V_∞ in (1) and (2) may lead to unsafe or risky policies. To formalize this concept, let $\mathcal{S}_0 \subset \mathcal{S}$ denote a set of safe states. Then, we consider the following definition of safety:

Definition 1. *We say a policy π_θ is $(1-\delta)$-safe for the set $\mathcal{S}_0 \subset \mathcal{S}$ if for every $t \geq 0$ we have that $P(s_t \in \mathcal{S}_0) \geq 1 - \delta$.*

Hence, we can write the problem of finding safe policies in reinforcement learning as the following constrained optimization problem

$$\begin{aligned} \underset{\theta\in\mathcal{H}}{\text{maximize}} \quad & V_{T/\infty}(\theta) \\ \text{subject to} \quad & P(s_t \in \mathcal{S}_0 \mid \pi_\theta) \geq 1 - \delta \text{ for all } t \geq 0 \end{aligned} \tag{3}$$

Note that because the MDP is not available in reinforcement learning problems, $P(s_t \in \mathcal{S}_0)$ can only be evaluated through experiments. Thus, there is no straightforward relation between θ (i.e., the policy) and the constraint in (3). A common approach to deal with this issue is to modify the reward function in (1) and (2) so it is risk aware. Explicitly, we define

$$r(s_t, a_t) = \bar{r}(s_t, a_t) + \lambda \mathbb{1}(s_t \in \mathcal{S}_0), \tag{4}$$

where $\bar{r}$ is the original reward function describing the agent task, $\lambda > 0$ is a safety-related reward, and the indicator function is such that $\mathbb{1}(s_t \in \mathcal{S}_0) = 1$ if $s_t \in \mathcal{S}_0$ and zero otherwise. In other words, the agent receives an extra reward of λ for respecting the safety specifications. Since only the reward function was modified, common learning techniques used to maximize V_T and V_∞ still apply [6]. Nevertheless, selecting the value of λ is not straightforward. Besides depending on the values of $\bar{r}$, it must strike a balance between safety and task completion. Indeed, large values of λ can lead to policies that are safe *because* they do not achieve the goal (see Section V).

In the next section, we provide an alternative relaxation of (3) that leads to guaranteed $(1-\delta)$-safe policies. To do so, we relax the probability constraint so it has a form similar to V_T/V_∞. Thus, we can leverage existing procedures used to maximize (1) and (2) to solve these relaxed constrained learning problems (Section IV). Additionally, we derive safety guarantees for the relaxed problem by showing how much the probabilistic constraint must be tightened to obtain $(1-\delta)$-safe policies.

Remark 1. In this remark we discuss the equivalence between the formulations in (1) and (2). This discussion is inspired in [5, Section 2.3] and in the proofs of [22, Proposition 2 and 3]. Let us start by considering the finite horizon value function in (1) with a horizon chosen from a geometric distribution with parameter $\gamma \in (0, 1)$. Then, it is possible to write (1) as

$$\mathbb{E}\left[\sum_{t=0}^{T} r(s_t, a_t)\right] = \mathbb{E}\left[\sum_{t=0}^{\infty} \mathbb{1}(t \leq T) r(s_t, a_t)\right]. \tag{5}$$

Under mild assumptions on the reward function it is possible to exchange the sum and the expectation (see e.g., [22, Proposition 2]). Also assuming that the horizon is drawn independently from the trajectory, we can write $\mathbb{E}[\mathbb{1}(t \leq T) r(s_t, a_t)] = \mathbb{E}[\mathbb{1}(t \leq T)]\,\mathbb{E}[r(s_t, a_t)]$. This yields

$$\mathbb{E}\left[\sum_{t=0}^{T} r(s_t, a_t)\right] = \sum_{t=0}^{\infty} \mathbb{E}[\mathbb{1}(t \leq T)]\,\mathbb{E}[r(s_t, a_t)]. \tag{6}$$

Further notice that the expectation of the indicator function is the probability of t begin less than T. Since T is drawn from a geometric distribution it follows that $\mathbb{E}[\mathbb{1}(t \leq T)] = \gamma^t(1-\gamma)$. Thus, (6) reduces to

$$\mathbb{E}\left[\sum_{t=0}^{T} r(s_t, a_t)\right] = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \mathbb{E}[r(s_t, a_t)]. \tag{7}$$

Exchanging back the expectation and the sum establishes the equivalence between the two formulations.

III. Safe Policy Learning

If the transition probabilities of the system were known, (3) could be solved by directly imposing constraints on the probabilities, using for instance Model Predictive Control [23]. However, this is not the scenario in reinforcement learning problems, where the transition probabilities can only be accessed through experiments. To overcome this difficulty, we consider the following relaxations of the chance constraint in (3). In the case of finite time horizon problems, we replace the conjunction in (3) by the average safety probability defined as

$$U_T(\theta) = \frac{1}{T+1}\sum_{t=0}^{T} P(s_t \in \mathcal{S}_0|\pi_\theta). \tag{8}$$

For the infinite horizon problem, we use the geometrically discounted average

$$U_\infty(\theta) = \sum_{t=0}^{\infty} \delta^t P(s_t \in \mathcal{S}_0|\pi_\theta). \tag{9}$$

The relaxation (8) is related to the idea of online learning [24], where instead of satisfying the constraint $P(s_t \in \mathcal{S}_0|\pi_\theta) \geq 1-\delta$ for all $t \geq 0$, we aim to satisfy the constraint in average. In view of the equivalence between formulations (1) and (2) discussed in Remark 1, note that the proposed relaxations are also equivalent when the horizon T is drawn from a geometric distribution.

Using Definition 1, note from (8) and (9) that $U_T(\theta) > 1-\delta$ and that $U_\infty(\theta) > 1$ for any $(1-\delta)$-safe policy. However, since these are necessary but not sufficient conditions for safety, we introduce a slack variable $\varepsilon \geq 0$ to tighten the constraints. Hence, the safe learning problem is given by

$$\begin{aligned} \theta_T^* \triangleq \underset{\theta\in\mathcal{H}}{\operatorname{argmax}} \quad & V_T(\theta) \\ \text{subject to} \quad & U_T(\theta) \geq 1-\delta+\varepsilon, \end{aligned} \tag{10}$$

for finite horizon and for infinite horizon yields

$$\begin{aligned} \theta_\infty^* \triangleq \underset{\theta\in\mathcal{H}}{\operatorname{argmax}} \quad & V_\infty(\theta) \\ \text{subject to} \quad & U_\infty(\theta) \geq 1+\frac{\varepsilon}{1-\delta}, \end{aligned} \tag{11}$$

In Section III-A, we establish values of ε that guarantee the policies obtained from (10) and (11) are $(1-\delta)$-safe policies.

Before proceeding, however, note that (8) and (9) still involve $P(s_t \in \mathcal{S}_0|\pi_\theta)$, which we can only evaluate through experiments. However, we can maximize U_T and U_∞ without explicitly computing this probability. To see why this is the case, define the reward function $r_{\text{safe}}(s,a) = \mathbb{1}(s \in \mathcal{S}_0)$ for the indicator function defined as in (4). By noticing that $P(s_t \in \mathcal{S}_0) = \mathbb{E}\left[\mathbb{1}(s \in \mathcal{S}_0)\right]$, we immediately obtain that $U_T(\theta) = V_T(\theta)$ and $U_\infty(\theta) = V_\infty(\theta)$ for the reward function r_{safe} and the discount factor $\gamma = \delta$. Hence, we can maximize U_T and U_∞ using the same methods used to maximize value functions [6]. We leverage this observation in Section IV to learn $(1-\delta)$-safe policies.

A. Safety Guarantees

In this section we establish the safety guarantees of the policy π_θ^* that arises from solving the problems formulated in (10) and (11). Before doing so, we define $\mathcal{T}_{\text{safe}}$ to be the set of time indices in which the policy is indeed safe $\mathcal{T}_{safe} := \{t = 0, \dots, T \mid P(s_t \in S_0|\pi_\theta)\}$, with cardinality $T_{\text{safe}} = |\mathcal{T}_{\text{safe}}|$. We define as well $\mathcal{T}_{\text{unsafe}} = \mathcal{T}_{\text{safe}}^c$, with cardinality $T_{\text{unsafe}} = |\mathcal{T}_{\text{unsafe}}| = T+1-T_{\text{safe}}$. In the next proposition we bound the fraction of times for which the policy achieved is not $(1-\delta)$-safe.

Proposition 1. *Let π_θ^* be a solution of* (10) *with $\varepsilon \in (0, \delta)$. Then, the proportion of unsafe times is bounded by*

$$\frac{T_{unsafe}}{T+1} \leq 1 - \frac{\varepsilon}{\delta} \tag{12}$$

Proof. Split the summation in (8) in one sum with the safe times and another one with the unsafe ones

$$\begin{aligned} U_T(\theta^*) &= \frac{1}{T+1}\sum_{t=0}^{T} P(s_t \in \mathcal{S}_0|\pi_\theta^*) \qquad (13)\\ &= \frac{1}{T+1}\sum_{t\in\mathcal{T}_{\text{safe}}} P(s_t \in \mathcal{S}_0|\pi_\theta^*) + \frac{1}{T+1}\sum_{t\in\mathcal{T}_{\text{unsafe}}} P(s_t \in \mathcal{S}_0|\pi_\theta^*), \end{aligned}$$

Notice that by definition for those times in which the policy is unsafe we have that $P(s_t \in \mathcal{S}_0|\pi_\theta^*) \leq 1-\delta$. This allows us to upper bound all the terms in the second sum by $1-\delta$. The sum of the safe terms can be always bounded by T_{safe} since each probability is upper bounded by 1. Thus, it follows that

$$U_T(\theta^*) \leq \frac{T_{\text{safe}}}{T+1} + \frac{T_{\text{unsafe}}}{T+1}(1-\delta). \tag{14}$$

Moreover, since $T_{\text{safe}} + T_{\text{unsafe}} = T+1$ it follows that

$$U_T(\theta^*) \leq 1 - \delta\frac{T_{\text{unsafe}}}{T+1}. \tag{15}$$

For the constraint considered in (10) we have that

$$1-\delta+\varepsilon \leq U_T(\theta^*) \leq 1 - \delta\frac{T_{\text{unsafe}}}{T+1}. \tag{16}$$

This completes the proof of the result. ■

The previous results establishes a bound on the proportion of times where the policy achieved by solving (10) is not safe. Notice that by setting $\varepsilon = \delta$ the number of unsafe times is zero. However, that means that the solution set of problem (10) has no interior, which violates the constraint qualification conditions necessary for solving the problem in practice. The next corollary uses the bound in (12) to establish conditions under which the solution of (10) is a $(1-\delta)$-safe policy.

Corollary 1. *Assume that for set $\mathcal{S}_0$ there exists a $1-\delta/(T+1)$-safe policy $\tilde{\pi}_\theta$. Then, the solution of* (10) *with $\varepsilon > \delta T/(T+1)$ yields a $(1-\delta)$-safe policy.*

Proof. Using the bound for the unsafe times derived in Proposition 1 to write

$$T_{\text{unsafe}} \leq (T+1)\left(1-\frac{\varepsilon}{\delta}\right) < (T+1)\left(1-\frac{T}{T+1}\right) = 1. \tag{17}$$

The latter implies that there are no unsafe times as long as the problem (10) with the choice of epsilon is feasible. Notice, that for $\varepsilon > \delta T/(T+1)$, the constraint in (10) reduces to

$$U_T(\theta) \geq 1 - \frac{\delta}{T+1}. \tag{18}$$

The latter is a feasible problem since there exists a policy $\tilde{\pi}_\theta$ that is $(1-\delta/(T+1))$-safe. This completes the proof of the corollary. ■

The above corollary establishes that it is possible to achieve a $(1-\delta)$-safe policy by solving problem (10) with slack variable $\delta T/(T+1)$. In what follows we develop the analogous results of Proposition 1 and Corollary 1 for the discounted formulation (11).

Proposition 2. *Let π_θ^* be a solution of* (11) *with $\varepsilon \in (0, \delta)$. Then, the following bound for the unsafe times holds*

$$\sum_{t \in \mathcal{T}_{unsafe}} \delta^t \leq \frac{1-\varepsilon/\delta}{1-\delta}. \tag{19}$$

Proof. Let us split the summation in (9) in one sum with the safe times and another one with the unsafe ones

$$\begin{aligned} U_\infty(\theta^*) &= \sum_{t=0}^{\infty} \delta^t P(s_t \in \mathcal{S}_0 | \pi_\theta^*) \\ &= \sum_{t \in \mathcal{T}_{\text{safe}}} \delta^t P(s_t \in \mathcal{S}_0 | \pi_\theta^*) + \sum_{t \in \mathcal{T}_{\text{unsafe}}} \delta^t P(s_t \in \mathcal{S}_0 | \pi_\theta^*). \end{aligned} \tag{20}$$

As done in the proof of Proposition 1, we can upper bound the probabilities in the summation of the unsafe times by $1-\delta$ and those in the safe times by 1. Hence it follows that

$$U_\infty(\theta^*) \leq \sum_{t \in \mathcal{T}_{\text{safe}}} \delta^t + \sum_{t \in \mathcal{T}_{\text{unsafe}}} \delta^t (1-\delta). \tag{21}$$

Notice that the terms δ^t of both sums add to $1/(1-\delta)$ since it is the sum of a geometric sequence. Hence, the previous upper bound reduces to

$$U_\infty(\theta^*) \leq \frac{1}{1-\delta} - \delta \sum_{t \in \mathcal{T}_{\text{unsafe}}} \delta^t. \tag{22}$$

If a policy π_θ^* satisfies the constraint in (11) it holds

$$\sum_{t \in \mathcal{T}_{\text{unsafe}}} \delta^t \leq \frac{1}{\delta(1-\delta)} - \frac{1}{\delta} - \frac{\varepsilon}{\delta(1-\delta)} = \frac{\delta - \varepsilon}{\delta(1-\delta)}. \tag{23}$$

This completes the proof of the result. ■

The previous proposition does not allow us to establish a bound on the unsafe times but it suggests that either the unsafe times are concentrated towards the initial times but there are few or that they happen far away in the future. In the next corollary we establish a condition for the latter being the case, which ensures safety until a desired time horizon.

Corollary 2. *Assume that there exists a policy $\tilde{\pi}_\theta$ that is $(1-\delta^{T+1}(1-\delta))$-safe for the set $\mathcal{S}_0$. Then, the solution of* (11) *with $\varepsilon > \delta(1-\delta^T(1-\delta))$ is such that is safe for all times $t \leq T$.*

Proof. We will argue at the end of the proof that a $(1-\delta^{T+1}(1-\delta))$-safe policy makes the problem (19) with slack variable $\varepsilon > \delta(1-\delta^T(1-\delta))$ feasible. That being the case, replacing ε in (19) yields

$$\sum_{t \in \mathcal{T}_{\text{unsafe}}} \delta^t < \delta^T. \tag{24}$$

We next argue that the latter implies that there can not be any terms $t \in \mathcal{T}_{\text{unsafe}}$ such that $t \leq T$. Notice that if that were the case, the sum on the left hand side of the previous equation should be lower bounded by δ^T. Hence, all $t \in \mathcal{T}_{\text{unsafe}}$ are such that $t > T$. To complete the proof, we need to verify that the problem (11) is feasible for the slack variable $\varepsilon > \delta(1-\delta^T(1-\delta))$. Since there exists a policy $\tilde{\pi}_\theta$ such that the set $\mathcal{S}_0$ is $(1-\delta^{T+1}(1-\delta))$-safe, we have that

$$U_\infty(\theta) > \frac{1-\delta^{T+1}(1-\delta)}{1-\delta}. \tag{25}$$

Adding and subtracting δ on the numerator allows us to write the previous expression as

$$U_\infty(\theta) > 1 + \frac{\delta(1-\delta^T(1-\delta))}{1-\delta}. \tag{26}$$

Thus, the constraint in (11) is feasible for the slack ε selected. ■

Having established the aforementioned safety guarantees we set the focus into solving them. In the next section we propose a primal-dual algorithm to do so.

IV. Primal-Dual Algorithm

We start the development of the algorithm by writing a relaxation for the problems (10) and (11). Let $\lambda \geq 0$ be a multiplier and define the corresponding Lagrangian as

$$\mathcal{L}(\theta, \lambda) = V(\theta) + \lambda(U(\theta) - s), \tag{27}$$

where s is the slack for each one of the problems defined. This slack takes the value $s = 1 - \delta + \varepsilon$ in the finite time problem (10) and $s = 1 + \varepsilon/(1-\delta)$ in the discounted infinite horizon problem (11). We then define the dual function as the point-wise maximum of the Lagrangian $\mathcal{L}(\theta, \lambda)$ with respect to the primal variable θ. Namely,

$$g(\lambda) = \max_{\theta \in \mathcal{H}} \mathcal{L}(\theta, \lambda). \tag{28}$$

Since the dual function is a point-wise maximum of linear functions with respect to λ, it is a convex function. Moreover, the dual function is an upper bound on the optimal value of the original problem. To see why this is the case, notice that by definition of the maximum, we have that

$$g(\lambda) \geq \mathcal{L}(\theta^*, \lambda) = V(\theta^*) + \lambda(U(\theta^*) - s). \tag{29}$$

Since θ^* is the solution of the primal problem, it is feasible as well, which implies that $U(\theta^*) - s \geq 0$. Thus $g(\lambda) \geq V(\theta^*)$, which in turn means that the dual function is an upper bound of the optimal problem. Of all possible upper bounds, we

are interested in the tightest one, this is to find λ such to minimize the dual function

$$D^* := \min_{\lambda \geq 0} g(\lambda). \tag{30}$$

As previously argued, since $g(\lambda)$ is a convex function, one can run gradient descent on the dual variable in order to solve (30). It follows from Danskin's theorem [25] that the gradient of the dual function can be computed by evaluating the constraints of (10) and (11) at the primal maximizer. For the k–th iteration of the gradient ascent, the $\nabla g(\lambda)$ can be computed as $\nabla g(\lambda^k) = U(\theta^k) - s$, where

$$\theta^k = \operatorname*{argmax}_{\theta \in \mathcal{H}} \mathcal{L}(\theta, \lambda^k). \tag{31}$$

Hence a feasible solution of problems (10) and (11) can be computed through the following algorithm consisting of the primal update (31) followed by the dual gradient descent step

$$\lambda^{k+1} = \lambda^k - \eta_\lambda \nabla g(\lambda^k) = \lambda^k - \eta_\lambda (U(\theta^k) - s). \tag{32}$$

A common approach to solve the primal maximization (31)—at least locally—is to use gradient ascent on the parametrization of the policy. Alternatively, instead of solving (31) before taking the dual step, one can update the primal and dual variable at the same time. That is, update (32) is computed for the current value of the iterate θ^k and the primal variable update is computed as

$$\theta^{k+1} = \theta^k + \eta_\theta \nabla_\theta \mathcal{L}(\theta^k, \lambda^k). \tag{33}$$

The previous expression involves the computation of the gradient of the expected value of the cumulative reward of the system under policy $\pi_\theta(a|s)$. To give the expression for said quantity, we require the following definitions. Let

$$R_T^\lambda(\mathbf{s}, \mathbf{a}) = \sum_{t=0}^{T} r(s_t, a_t), \tag{34}$$

$$R_\infty^\lambda(\mathbf{s}, \mathbf{a}) = \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \tag{35}$$

be the cumulative weighted rewards—as in (4)—for both the finite horizon and infinite horizon formulations. Further, define $d_{\theta,T}(\mathbf{a}|\mathbf{s}) = \prod_{t=0}^{T} \pi_\theta(a_t|s_t)$ and $d_{\theta,\infty}(\mathbf{a}|\mathbf{s}) = \prod_{t=0}^{\infty} \pi_\theta(a_t|s_t)^{\gamma^t}$ for both formulations. Then the gradient of the Lagrangian (27) with respect to the parameters of the policy θ for both formulations yields [6]

$$\nabla_\theta \mathcal{L}(\theta, \lambda^k) = \mathbb{E}_{\mathbf{a} \sim \pi_\theta(\mathbf{a}|\mathbf{s})} \left[R^\lambda(\mathbf{s}, \mathbf{a}) \nabla_\theta \log \left(d_\theta(\mathbf{a}|\mathbf{s}) \right) \right], \tag{36}$$

where $R^\lambda(\mathbf{s}, \mathbf{a})$ is either (34) or (35), depending on the formulation being used. As discussed in Section II the main advantage of the relaxations introduced in (8) and (9) is that they allow us to compute the gradient of the Lagrangian with respect to the parametrization of the policy. On the other hand, a difficulty, in the computation of the dual gradient in (32) and the primal gradient (36) is the need of computing expectations with respect to the trajectories of the system. To avoid the need of sampling a large number of trajectories, one can compute a stochastic approximation

$$\hat{U}(\theta^k) = \sum_{t=0}^{T} \mathbb{1}(s_t \in \mathcal{S}_0), \tag{37}$$

$$\hat{\nabla}_\theta \mathcal{L}(\theta^k, \lambda^k) = R^{\lambda^k}(\mathbf{s}, \mathbf{a}) \nabla_\theta \log(d_{\theta^k}(\mathbf{a}|\mathbf{s})). \tag{38}$$

In cases where the horizon is finite, the previous expressions can be computed without any additional steps and they yield unbiased estimates of the quantities that they estimate. However, for the infinite horizon case, one would require an infinite trajectory for the later to hold. An alternative, and given the equivalence between the finite and infinite time horizon problem discussed in Remark 1 is to sample a horizon from a geometric distribution. By computing the expressions in (38) and (37) over the randomly drawn horizon the estimates obtained are unbiased [22]. The stochastic primal-dual algorithm is summarized in Algorithm 1 and in the next section we show how it can be used to safely navigate a grid world.

Algorithm 1 Stochastic Primal-Dual for Safe Policies

Input: $\theta^0, \lambda^0, T, \eta_\theta, \eta_\lambda, \delta, \epsilon$

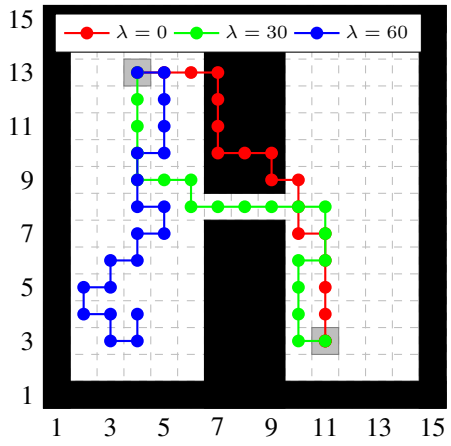
for $k = 0, 1, \ldots$ **do**
  Simulate a trajectory with the policy $\pi_{\theta^k}(\mathbf{a}|\mathbf{s})$
  Estimate primal gradient $\hat{\nabla}_\theta \mathcal{L}(\theta^k, \lambda^k)$ as in (38)
  Estimate dual gradient $\hat{U}(\theta^k) - s$ as in (37)
  Update primal $\theta^{k+1} = \theta^k + \eta_\theta \hat{\nabla}_\theta \mathcal{L}(\theta^k, \lambda^k)$
  Update dual $\lambda^{k+1} = \lambda^k + \eta_\lambda \left(\hat{U}(\theta^k) - s \right)$
end for

V. Numerical Results

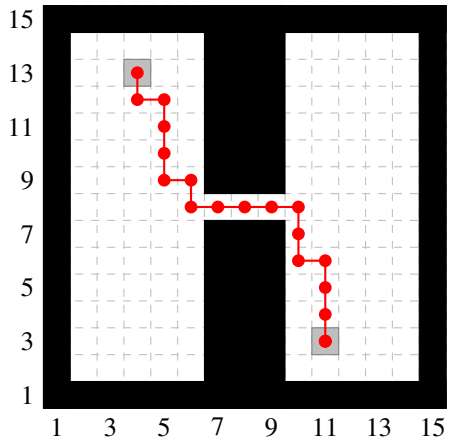
In this section, we study the performance of our proposed safe primal-dual policy gradient algorithm. For comparison, we also provide simulations with a classical policy gradient, in which the reward function has been modified to include the notion of safety as in (4). The scenario that we consider is one in which an agent is performing a navigation task. The high-level description of the navigation task is the following: two safe areas are connected by a bridge and the objective of the agent is to go from one safe area to the other without falling off (i.e., going to the unsafe areas). The specific map representing this task is shown in Figure 1. The actual description of the MDP is given by a discrete state space composed of 15×15 states and where each state has four possible actions (moving `up`, `right`, `down`, and `left`). The policy that the agent is aiming to learn is a softmax policy on the possible actions. More specifically,

$$\pi_\theta(a|s) = \frac{e^{\theta_{s,a}}}{\sum_{a' \in \mathcal{A}} e^{\theta_{s,a'}}} \tag{39}$$

where, $\theta = \{\theta_{s,a}\}_{(s,a) \in \mathcal{S} \times \mathcal{A}}$. In this space, the agent attempts to learn how to *safely* navigate from start position $s_{\text{start}} = [4, 13]$ to goal position $s_{\text{goal}} = [11, 3]$, over a time horizon



(a) Policy gradient with different rewards.



(b) Primal-dual policy gradient.

Fig. 1. Example trajectories for each of the trained policies. The safe set $\mathcal{S}_0$ is represented by the white pixels in the image, while black pixels represent the unsafe set. The agent starts the navigation task at position $s_{\text{start}} = [4, 13]$ and attempts to reach the goal position $s_{\text{goal}} = [11, 3]$.

of $T = 20$ time slots. The reward received by the agent is

$$r(s, a) = \lambda \mathbb{1}(s \in \mathcal{S}_0) + 100 \mathbb{1}(s = s_{\text{goal}}) - 10 \mathbb{1}(s \neq s_{\text{goal}})$$

for both the policy gradient and the primal-dual policy gradient. This function indicates that the agent receives a reward of $+100$ for reaching the goal and -10 for stepping anywhere else in the map, and staying in the safe set is rewarded by λ. While the reward function for both algorithms is the same, for the classic policy gradient, the value λ is a system parameter that has to be manually selected. This is not the case of the primal-dual policy gradient, in which λ corresponds to the dual variable, which is dynamically adapted according to the the dual update (32).

We run a simulation over $t = 10{,}000$ iterations with a primal step size of $\eta_\theta = 0.001$ (for both algorithms) and a dual step size of $\eta_\lambda = 0.15$ (for the primal-dual policy gradient). For the primal-dual policy gradient, we also choose a safety level of $1 - \delta = 0.95$ and a slack of $\epsilon = 0$.

In Figure 1 we plot a sample trajectory of the trained policies, for both the policy gradient (Figure 1(a)) and the primal-dual policy gradient (Figure 1(b)). In the case of the former, safety needs to be manually specified by the choice of λ. In this sense, the sample trajectories show the effect of this parameter. A parameter $\lambda = 0$ is equivalent to ignoring the safety constraint. Hence, the algorithm attempts to reach the goal completely disregarding the safe "bridge" passing. For $\lambda = 30$, the sample trajectory is safe, as it crosses through the bridge. On the other hand, for $\lambda = 60$, the demanded safety is so large, that the algorithm stops attempting to reach the goal (the algorithm is too cautious to explore the environment), and, as shown by the sample trajectory, it stays in the safe area without attempting to cross the bridge to reach the goal.

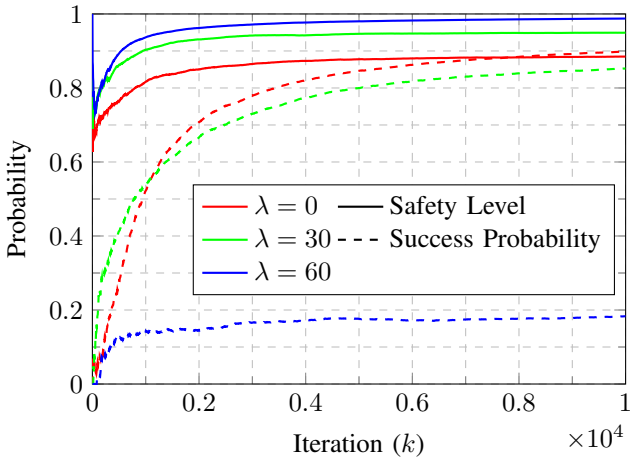
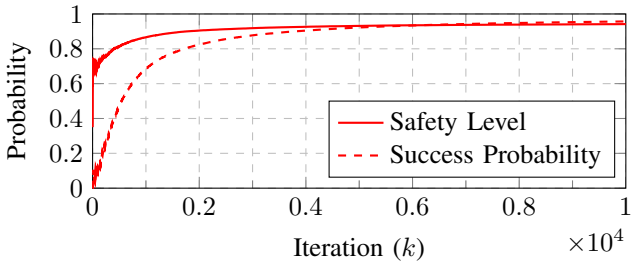


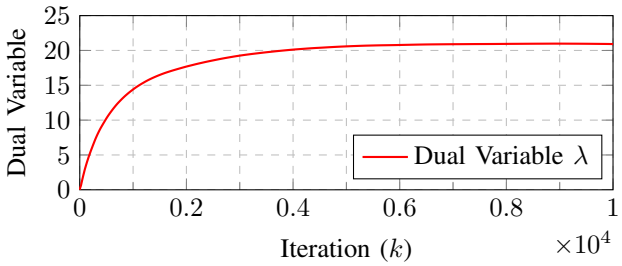
Fig. 2. Convergence of policy gradient. We plot the probability of safety of a trajectory and the probability of reaching the goal.

For the case of the primal-dual policy gradient, we do not need to specify the value of λ, as this is dynamically selected by the primal-dual policy gradient algorithm. We see that a sample trajectory of the converged policy is safe. The λ to which the algorithm converges in this case is $\lambda \approx 21$ and the behavior is similar to the policy gradient with $\lambda = 30$.

Now we will delve deeper into the properties and behavior of these different policies. While previously we looked at a single sample trajectory of each one of these policies after convergence, we now look at how their behavior evolves during training. Specifically, we are interested in the safety probability of a trajectory $U_T(\theta)$ (cf. equation (8)), and the probability of a trajectory reaching the goal state. We plot the resulting values for the policy gradient with different rewards in Figure 2. As previously observed in sample trajectories, different values of the safety parameter λ lead to different levels of safety and task accomplishment. For $\lambda = 0$, the probability of safety is the lowest, but the probability of success is the highest. This behavior is apparent, since the agent is disregarding any notion of safety and simply focusing on reaching the goal. At the other extreme, for $\lambda = 60$, the safety probability is close to 1, but this is due to almost completely disregarding the task, as shown by the small probability of success. Therefore, safety increases with λ, while the probability of reaching the goal decrease. This illustrates that there exist a trade-off between task



(a) Probabilities of safety and task accomplishment.



(b) Value of the dual variable.

Fig. 3. Convergence of the primal-dual policy gradient. We plot the safety level of a trajectory and the probability of reaching the goal.

accomplishment and safety (i.e., too safe and the agent will not accomplish the goal).

A trade-off that can methodically be found by the use of the primal-dual policy gradient. In Figure 3 we plot the same results as previously, but for the primal-dual policy gradient. In this case, we are demanding a safety level of $1-\delta = 0.95$ with a slack of $\epsilon = 0$. The plot shows that the algorithm attains the desired probability of safety while maintaining a high probability of task accomplishment. Further insight can be obtained by looking at the value of the dual variable λ as the algorithm iterates. This is shown in Figure 3(b). We see that this dual variable converges to $\lambda \approx 21$, a value close to the $\lambda = 30$, previously seen to work for the policy gradient with manually specified safety rewards. In this sense, the primal-dual policy gradient attempts to find the required weight of safety (given by λ) that allows to maintain the probability of safety demanded $1-\delta$. Since the resulting safety are similar, the resulting λ variables are similar. If, e.g., one where to demand less safety of the primal-dual policy gradient, the price of safety would go down, and hence the dual variables λ would converge to smaller values.

VI. Conclusions

In this paper, we have studied the problem of learning safe policies in reinforcement learning problems. More specifically, we have introduced safety into the problem through probabilistic constraints that we then relax for both finite and infinite horizons, hence formulating a constrained optimization problem. The advantages of the proposed relaxations are twofold. First, they allow us to compute the primal and dual gradients of the Lagrangian associated to the optimization problem. Which can be solved by running a stochastic primal-dual method. Second, these relaxations do not come at the cost of safety. In particular we established that the finite horizon problem remains safe and we established a safe horizon for the discounted optimization problem. Numerical results for an agent navigating a grid world show that the proposed scheme dynamically adapts the cost of safety to the environment. Compared to previous approaches, our proposed scheme provides safe policies with guarantees and a systematic way of achieving them, without being reliant on the manual tuning of parameters.

References

[1] R. A. Howard, *Dynamic programming and Markov processes*. Wiley for The Massachusetts Institute of Technology, 1964.

[2] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[3] S. E. Shreve and D. P. Bertsekas, "Alternative theoretical frameworks for finite horizon discrete-time stochastic optimal control," *SIAM J. on control and optimization*, vol. 16, no. 6, pp. 953–978, 1978.

[4] M. Rásonyi, L. Stettner, *et al.*, "On utility maximization in discrete-time financial market models," *The Annals of Applied Probability*, vol. 15, no. 2, pp. 1367–1395, 2005.

[5] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-dynamic programming*, vol. 5. Athena Scientific Belmont, MA, 1996.

[6] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[7] M. Heger, "Consideration of risk in reinforcement learning," in *Machine Learning Proceedings 1994*, pp. 105–111, Elsevier, 1994.

[8] O. Mihatsch and R. Neuneier, "Risk-sensitive reinforcement learning," *Machine learning*, vol. 49, no. 2-3, pp. 267–290, 2002.

[9] P. Geibel and F. Wysotzki, "Risk-sensitive reinforcement learning applied to control under constraints," *Journal of Artificial Intelligence Research*, vol. 24, pp. 81–108, 2005.

[10] S. P. Coraluppi and S. I. Marcus, "Risk-sensitive and minimax control of discrete-time, finite-state markov decision processes," *Automatica*, vol. 35, no. 2, pp. 301–309, 1999.

[11] R. A. Howard and J. E. Matheson, "Risk-sensitive markov decision processes," *Management science*, vol. 18, no. 7, pp. 356–369, 1972.

[12] T. M. Moldovan and P. Abbeel, "Safe exploration in markov decision processes," *arXiv preprint arXiv:1205.4810*, 2012.

[13] M. Turchetta, F. Berkenkamp, and A. Krause, "Safe exploration in finite markov decision processes with gaussian processes," in *Advances in Neural Information Processing Systems*, pp. 4312–4320, 2016.

[14] P. Geibel, "Reinforcement learning for mdps with constraints," in *European Conference on Machine Learning*, pp. 646–653, Springer, 2006.

[15] Y. Kadota, M. Kurano, and M. Yasuda, "Discounted markov decision processes with utility constraints," *Computers & Mathematics with Applications*, vol. 51, no. 2, pp. 279–284, 2006.

[16] E. Delage and S. Mannor, "Percentile optimization for markov decision processes with parameter uncertainty," *Operations research*, vol. 58, no. 1, pp. 203–213, 2010.

[17] D. Di Castro, A. Tamar, and S. Mannor, "Policy gradients with variance related risk criteria," *arXiv preprint arXiv:1206.6404*, 2012.

[18] M. Hutter, "Self-optimizing and pareto-optimal policies in general environments based on bayes-mixtures," in *International Conference on Computational Learning Theory*, pp. 364–379, Springer, 2002.

[19] Y. Chow, M. Ghavamzadeh, L. Janson, and M. Pavone, "Risk-constrained reinforcement learning with percentile risk criteria.," *Journal of Machine Learning Research*, vol. 18, no. 167, pp. 1–167, 2017.

[20] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 22–31, JMLR. org, 2017.

[21] J. Garcıa and F. Fernández, "A comprehensive survey on safe reinforcement learning," *Journal of Machine Learning Research*, vol. 16, no. 1, pp. 1437–1480, 2015.

[22] S. Paternain, J. A. Bazerque, A. Small, and A. Ribeiro, "Stochastic policy gradient ascent in reproducing kernel hilbert spaces," *arXiv preprint arXiv:1807.11274*, 2018.

[23] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[24] V. Vapnik, *The nature of statistical learning theory*. Springer, 2000.

[25] D. P. Bertsekas, *Nonlinear programming*. Athena Sci., Belmont, 1999.