Generalized Constraint for Probabilistic Safe Reinforcement Learning

Weiqin Chen CHENW18@RPI.EDU
Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180, USA

Santiago Paternain PATERS@RPI.EDU
Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180, USA

Editors: A. Abate, K. Margellos, A. Papachristodoulou

Abstract

In this paper, we consider the problem of learning policies for probabilistic safe reinforcement learning (PSRL). Specifically, a safe policy or controller is one that, with high probability, maintains the trajectory of the agent in a given safe set. While the explicit gradient of the probabilistic constraint for solving PSRL directly exists, the high variance in the estimate of the gradient hinders its performance in problems with long-horizons. An alternative that is frequently explored in the literature is to consider a cumulative safe reinforcement learning (CSRL) setting. In this setting, the estimates of the constraint's gradient have less variance but are biased (worse solutions than the PSRL) and they provide an approximate solution since they solve a relaxation of the PSRL formulation. In this work, we propose a safe reinforcement learning framework with a generalized constraint for solving the PSRL problems, which we term Generalized Safe Reinforcement Learning (GSRL). Our theoretical contributions substantiate that the proposed GSRL can recover both the PSRL and CSRL settings. In addition, it can be naturally combined with any state-of-the-art safe RL algorithms like PPO-Lagrangian, TD3-Lagrangian, CPO, PCPO, etc. We evaluate the GSRL by a series of empirical experiments in the well-known safe RL benchmark *Bullet-Safety-Gym*, which exhibit a better return-safety trade-off than both the PSRL and CSRL formulations.

Keywords: Constrained Policy Optimization, Safe Reinforcement Learning, Probabilistic Constraints

1. Introduction

Reinforcement learning (RL) has succeeded in solving sequential decision-making problems, e.g., control problems (Farias et al., 2020), robotic manipulation (Nguyen and La, 2019) and robot locomotion (Li et al., 2021). Markov Decision Processes (MDPs) (Sutton and Barto, 2018) are commonly considered to formulate the RL problems. In cases where the underlying system dynamics are unknown, the acquisition of optimal policies necessitates the process of *learning* from system samples or data. The objective of RL is to maximize the expected return (Watkins and Dayan, 1992; Sutton et al., 1999), which, in general, may lead to unsafe/risky behaviors (Garcıa and Fernández, 2015).

Safety represents a fundamental cornerstone in the conceptualization and design of control systems governing physical entities. For instance, the imperative of guaranteeing collision avoidance (Kahn et al., 2018) stands as a critical requirement in the robot navigation. Moreover, it serves as a paramount measure to uphold human safety within their proximity. In addition, controllers utilized in power systems are intricately designed with precision to forestall voltage instabilities (Van Cutsem, 2000), which, if left unaddressed, could potentially precipitate perilous operational conditions.

Taking into account the safety requirements or constraints motivates the development of policy optimization under safety guarantees (Geibel, 2006; Kadota et al., 2006; Chow et al., 2017). A common approach is to consider the framework of Constrained MDPs (CMDPs) (Altman, 1999), where the auxiliary cumulative reward or cost needs to be maintained within a desired threshold. We term this framework Cumulative Safe RL (CSRL). The CSRL has garnered extensive adoption for instigating safe behaviors (Borkar, 2005; Bhatnagar and Lakshmanan, 2012; Achiam et al., 2017; Liang et al., 2018; Tessler et al., 2018; Yang et al., 2020; Zhang et al., 2020b; Shen et al., 2022; Chen et al., 2024a). However, the CSRL is generally not suitable for safety-critical applications (expect zero safety violation) (Cheng et al., 2019; Zhang et al., 2020a; Corsi et al., 2021), since even safety violations in all trajectories are acceptable in the CSRL as long as the amount of violations does not exceed the desired threshold.

An alternative notion in safety-critical contexts, also known as state-wise safe RL (Zhao et al., 2023), is to guarantee that every state of the system remains within a set recognized as safe. This notion, however, requires system-dependent assumptions for state-wise safety guarantees. In this work, we are interested in a general notion of safety that guarantees the entire trajectory being in the safe set with high probability. Problems with such probabilistic safety have been considered in (Geibel, 2006; Delage and Mannor, 2010). We term this setting Probabilistic Safe RL (PSRL). (Chen et al., 2023, 2024b) tackle the PSRL problems using the Safe Primal-Dual algorithm, where the main contribution is to provide an explicit expression for the gradient of probabilistic safety constraint. However, the estimate of the gradient shows high variance which hinders its performance in long-horizon problems and systems with complex dynamics.

Consequently, we propose in this paper a framework of Generalized Safe RL (GSRL) that trades off the PSRL and CSRL properties. We describe the settings of PSRL, CSRL in detail in Section 2. The theoretical developments of Section 3 indicate that by selecting an appropriate safety threshold the solutions to the GSRL problem guarantee the feasibility in the sense of PSRL (the problem of interest). In addition, we show that the GSRL can recover both the PSRL and CSRL settings by selecting the hyper-parameters properly. Immediately afterwards, we propose a Generalized Safe Primal-Dual (GSPD) algorithm which trades-off the return and safety better than the PSRL and CSRL formulations, upon the appropriate selection of hyper-parameters. Other than concluding remarks (Section 5), the paper finishes with a series of numerical experiments in Section 4, which are implemented on safe RL benchmarks *Bullet-Safety-Gym* (Gronauer, 2022). These experiments illustrate (i) the ability to learn safe policies through implementing the GSPD algorithm and (ii) the improved return-safety trade-offs that the GSRL provides over the PSRL and CSRL settings.

2. Probabilistic Safe Reinforcement Learning

2.1. Problem Formulation

In this work, we consider the problem of learning probabilistic safe policies for the PSRL problem (Geibel, 2006; Delage and Mannor, 2010; Chen et al., 2024b). The latter is built on the framework of finite-horizon Markov Decision Processes (MDPs) with additional probabilistic safety constraints. A finite-horizon MDP, see e.g., (Sutton and Barto, 2018), is defined by the tuple $(\mathcal{S}, \mathcal{A}, r, \mathbb{P}, \mu, T)$, where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ denotes the action space, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ denotes the reward function which evaluates the quality of the decision (action). For any $\hat{\mathcal{S}} \subset \mathcal{S}, s_t \in \mathcal{S}, a_t \in \mathcal{A}, t \in \{0\} \cup \mathbb{N}$, $\mathbb{P}^{a_t}_{s_t \to s_{t+1}}(\hat{\mathcal{S}}) := \mathbb{P}(s_{t+1} \in \hat{\mathcal{S}} \mid s_t, a_t)$ denotes the transition probability representing the dynamics of the system, $\mu(\hat{\mathcal{S}}) := \mathbb{P}(s \in \hat{\mathcal{S}})$ denotes the initial state distribution, and T denotes the time horizon. Denote by S_t and A_t the state and the selected action at time t. The agent selects the action A_t based on the state S_t following a parameterized policy (a conditional distribution) $\pi_\theta(A_t|S_t)$. In this work, we focus on the model-free RL (Çalışır and Pehlivanoğlu, 2019) where the transition probability is unknown, and thus the policies need to be learned from system samples by maximizing the value function

$$V(\theta) = \mathbb{E}_{\mathbf{a}\sim\pi_\theta(\mathbf{a}|\mathbf{s}),S_0\sim\mu}\left[\sum_{t=0}^{T} r(S_t, A_t)\right], \tag{1}$$

where $\mathbf{s}$ and $\mathbf{a}$ denote the sequences $\{S_0, S_1, \cdots, S_T\}$ and $\{A_0, A_1, \cdots, A_T\}$, respectively. Note that for the sake of notation simplicity, the subscripts of the expectation are omitted throughout the rest of the paper.

In striving to solely maximize the objective (1), the optimal policies might lead to unsafe/risky behaviors (Garcıa and Fernández, 2015). Consequently, we impose probabilistic safety as a requirement to overcome this limitation. We formally define the notion of probabilistic safety next.

Definition 1 *A policy π_θ is $(1-\delta)$-safe for the set $\mathcal{S}_{safe} \subset \mathcal{S}$ if and only if $\mathbb{P}\left(\cap_{t=0}^{T}\{S_t \in \mathcal{S}_{safe}\}|\pi_\theta\right) \geq 1-\delta$.*

In the previous definition, $\cap_{t=0}^{T}\{S_t \in \mathcal{S}_{safe}\}$ refers to the intersection of events $\{S_t \in \mathcal{S}_{safe}\}$ at all times. This is, we require the state S_t to belong to the safe set $\mathcal{S}_{\text{safe}}$ for all times $t \in \{0\} \cup \mathbb{N}$ with high probability. Under Definition 1, we formulate the PSRL problem as the following constrained optimization problem

$$P_p^\star = \max_\theta V(\theta) \quad \text{s.t.} \quad V_p(\theta) := \mathbb{P}\left(\bigcap_{t=0}^{T}\{S_t \in \mathcal{S}_{\text{safe}}\}|\pi_\theta\right) \geq 1-\delta. \tag{2}$$

(Chen et al., 2024b) tackles this problem using the Safe Primal-Dual algorithm, where the main contribution is to provide an explicit expression for the gradient of probabilistic safety. Nevertheless, the unbiased estimate of the gradient suffers from high variance (resulting in slow convergence which sometimes prevents solving the problem altogether) attributed to the fact that it takes into account the whole episode (from $t = 0$ to $t = T$). This issue is common to all Monte Carlo (MC) methods for sequential decision-making (Sutton and Barto, 2018).

2.2. Guaranteeing Safety through CMDPs

An alternative is to reformulate problem (2) as a CMDP (Altman, 1999) so that the probabilistic safety constraint can be relaxed to a cumulative version of constraints. In this setting, an auxiliary reward function $r_c : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is defined, and the following CMDP is considered

$$P_c^\star(\xi_c) = \max_\theta V(\theta) \quad \text{s.t.} \quad V_c(\theta) := \mathbb{E}\left[\sum_{t=0}^{T} r_c(S_t, A_t)\right] \geq \xi_c, \tag{3}$$

where ξ_c is a hyper-parameter that induces different levels of safety. While problems (2) and (3) may appear distinct, they share a significant connection. Specifically, (Paternain et al., 2022) indicates that the feasible solution of (3) is guaranteed to be feasible for problem (2) by selecting $r_c(S_t, A_t) = \mathbb{1}(S_t \in \mathcal{S}_{\text{safe}})$ and $\xi_c = T + 1 - \delta$. In this case (3) results in

$$P_c^\star = \max_\theta V(\theta) \quad \text{s.t.} \quad V_c(\theta) = \mathbb{E}\left[\sum_{t=0}^{T} \mathbb{1}(S_t \in \mathcal{S}_{\text{safe}})|\pi_\theta\right] \geq T + 1 - \delta. \tag{4}$$

Unlike problem (2), by considering Lagrangian relaxations of (3) and (4) (See Section 3 for more details), these problems can be tackled using state-of-the-art policy-based methods, e.g., PPO-Lagrangian (Ray et al., 2019), CPO (Achiam et al., 2017), RCPO (Tessler et al., 2018), FOCOPS (Zhang et al., 2020b). These approaches, based on temporal-difference (TD) learning, exhibit lower variance in the gradient estimations but are biased. In addition, Theorem 1 in (Chen et al., 2024b) establishes that the optimal value of (4) is lower than that of (2). We can interpret this fact as an additional bias in these algorithms.

3. Generalized Safe Reinforcement Learning

Notice that there is a tension between the algorithms to solve problems (2) and (4). Specifically, the estimate of the gradient for probabilistic constraints in problem (2) provides an unbiased solution to (2) at the cost of high variance (in the gradient estimates). On the other hand, the current state-of-the-art algorithms (e.g., PPO-Lagrangian) for solving (4) have low variance but yield a biased solution. This tension is similar to that observed between MC and TD methods. Akin to the n-step TD learning (Sutton and Barto, 2018) (which aims to find an intermediate solution between the 1-step TD and MC methods), we also consider a problem formulation in between the CSRL (4) and PSRL (2) settings

$$P_g^\star(\xi_g) = \max_\theta V(\theta) \quad \text{s.t.} \quad V_g(\theta) := \mathbb{E}\left[\sum_{t=N}^{T} \prod_{u=t-N}^{t} \mathbb{1}\left(S_u \in \mathcal{S}_{\text{safe}}\right)|\pi_\theta\right] \geq \xi_g, \tag{5}$$

where $N \in \{0, 1, \cdots, T\}$ is a hyper-parameter that trades off the PSRL and CSRL properties. Indeed, the constraint in (5) reduces to the cumulative safe constraint of (4) when $N = 0$, $\xi_g = T + 1 - \delta$, while transforming into the probabilistic safe constraint of (2) as $N = T$, $\xi_g = 1 - \delta$. Thus, we term the problem (5) GSRL. This generalization requires an appropriate choice of $\xi_g(N)$ to make the problem safe under Definition 1. We will formalize this claim in the following theorem.

Theorem 2 *For all $N \in \{0, 1, \cdots, T\}$, denote by θ_g a feasible solution to problem (5) with $\xi_g = T + 1 - N - \delta$. Then, θ_g is a feasible solution to problem (2) as well, i.e., the policy induced by θ_g guarantees the probabilistic safety in the sense of Definition 1.*

Proof We start by defining a new event $\mathcal{E}_t = \cap_{u=t-N}^{t} \{S_u \in \mathcal{S}_{safe}\}$. In particular, notice that $\mathcal{E}_N = \cap_{u=0}^{N} \{S_u \in \mathcal{S}_{safe}\}$ can be written as $\{S_0, S_1, \ldots, S_N \in \mathcal{S}_{safe}\}$. Likewise, the event $\mathcal{E}_{N+1}$ indicates $\{S_1, S_2, \ldots, S_{N+1} \in \mathcal{S}_{safe}\}$. Hence $\mathcal{E}_N \cap \mathcal{E}_{N+1} = \{S_0, S_1, \ldots, S_{N+1} \in \mathcal{S}_{safe}\}$. Applying this argument recursively it follows that $\cap_{t=N}^{T} \mathcal{E}_t = \cap_{t=0}^{T}\{S_t \in \mathcal{S}_{\text{safe}}\}$.

Using De Morgan's law (see e.g., (Durrett, 2019)), we proceed to rewrite $\mathbb{P}\left(\cap_{t=N}^{T} \mathcal{E}_t\right)$ as

$$\mathbb{P}\left(\cap_{t=N}^{T} \mathcal{E}_t\right) = 1 - \mathbb{P}\left(\cup_{t=N}^{T} \bar{\mathcal{E}}_t\right), \tag{6}$$

where $\bar{\mathcal{E}}_t$ denotes the complement of $\mathcal{E}_t$. Since $\mathbb{P}\left(\cup_{t=N}^{T}\bar{\mathcal{E}}_t\right) \leq \sum_{t=N}^{T}\mathbb{P}\left(\bar{\mathcal{E}}_t\right)$, the previous expression can be lower bounded as follows

$$\mathbb{P}\left(\cap_{t=N}^{T}\mathcal{E}_t\right) \geq 1 - \sum_{t=N}^{T}\mathbb{P}\left(\bar{\mathcal{E}}_t\right) = 1 - \sum_{t=N}^{T}\left(1 - \mathbb{P}\left(\mathcal{E}_t\right)\right), \tag{7}$$

where the equality follows from the definition of the complement. The previous inequality is equivalent to

$$\mathbb{P}\left(\cap_{t=N}^{T}\mathcal{E}_t\right) \geq 1 - (T + 1 - N) + \sum_{t=N}^{T}\mathbb{P}\left(\mathcal{E}_t\right). \tag{8}$$

From the previous inequality it follows that to establish $\mathbb{P}\left(\cap_{t=N}^{T}\mathcal{E}_t\right) \geq 1 - \delta$, it is sufficient to show

$$\sum_{t=N}^{T}\mathbb{P}\left(\mathcal{E}_t\right) \geq T + 1 - N - \delta. \tag{9}$$

Note that $\sum_{t=N}^{T}\mathbb{P}\left(\mathcal{E}_t\right) = V_g(\theta)$ due to the fact that $\mathbb{P}\left(\mathcal{E}_t\right) = \mathbb{E}\left[\prod_{u=t-N}^{t}\mathbb{1}\left(S_u \in \mathcal{S}_{\text{safe}}\right)|\pi_\theta\right]$. Therefore, selecting $\xi_g = T + 1 - N - \delta$ completes the proof of Theorem 2. ■

Theorem 2 indicates that $\forall N \in \{0, 1, \cdots, T\}$ selecting $\xi_g = T + 1 - N - \delta$ guarantees that solutions to (5) are safe in the sense of Definition 1. Furthermore, the optimal value of (2) is not less than that of (5), i.e., $P_p^\star \geq P_g^\star$. This can be explained by that problem (5) has smaller feasible set than problem (2), as indicated by Theorem 2. On the other hand, we claim that $V_g(\theta)$ is monotonically decreasing with the increasing N, and selecting $\xi_g = T + 1 - N - \delta$ in (5) results in both formulations of PSRL (2) when $N = T$ and CSRL (4) when $N = 0$. We formalize these claims in the following proposition.

Proposition 3 *Consider the GSRL formulation* (5) *with* $\xi_g = T + 1 - N - \delta$.

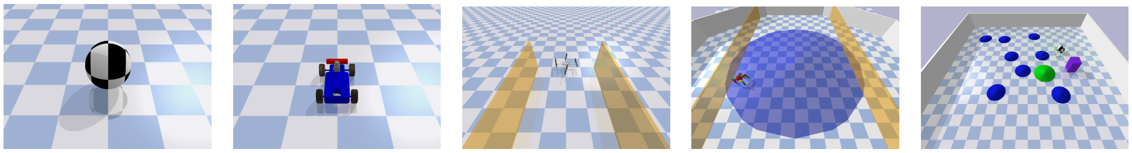
(I) $V_g(\theta)$ *is monotonically non-increasing with* N.

(II) It recovers problems (2) *when* $N = T$ *and* (4) *when* $N = 0$.

Proof Let us start by proving (I). Observe that $\mathbb{E}\left[\prod_{u=t-N}^{t}\mathbb{1}(S_u \in \mathcal{S}_{\text{safe}})\right] = \mathbb{P}(\cap_{u=t-N}^{t}\{S_u \in \mathcal{S}_{\text{safe}}\})$ is non-increasing as N increases. Indeed, larger N implies that more random variables need to belong to the safe set. This results in a smaller joint probability. In addition, the definition of $V_g(\theta)$ in (5) consists of $T - N$ terms of the form discussed. The larger N the fewer the terms in the summation. Since these are all positive, it follows that $V_g(\theta)$ is monotonically non-increasing as N increases.

We now turn to prove (II). When $N = 0$, one obtains $V_g(\theta) = \mathbb{E}\left[\sum_{t=0}^{T}\mathbb{1}(S_t \in \mathcal{S}_{\text{safe}})|\pi_\theta\right] = V_c(\theta)$ and $\xi_g = T + 1 - \delta$, by which the problem (5) reduces to (4). On the other extreme of $N = T$, it shows that $V_g(\theta) = \mathbb{E}\left[\prod_{t=0}^{T}\mathbb{1}(S_t \in \mathcal{S}_{\text{safe}})|\pi_\theta\right] = V_p(\theta)$ and $\xi_g = 1 - \delta$, thus transforming the problem (5) to (2). These complete the proof of (II). ■

As indicated by Proposition 3, $V_g(\theta)$ approaches $V_p(\theta)$ as N increases, resulting in the reduced bias of the algorithm since it is closer to the problem of interest, i.e., problem (2). On the other



(*a*) Ball (*b*) Car (*c*) Run (*d*) Circle (*e*) Reach

Figure 1: Agents and Tasks in Bullet-Safety-Gym (Gronauer, 2022): (a) The Ball agent; (b) The Car agent; (c) The Run task; (d) The Circle task; (e) The Reach task.

hand, it increases the variance of the algorithm since the constraint includes an indicator function that depends on multiple steps.

Having established the properties of the GSRL formulation, we now turn to explore the approaches to solve problem (5). It is conceivable to employ gradient-based methods e.g., regularization (Censor, 1977) and primal-dual (Arrow et al., 1958) to achieve local optimal solutions. For instance, consider the regularization method with a fixed penalty. This is, for $\lambda > 0$ we formulate the following *unconstrained* problem as an approximation to the *constrained* problem (5)

$$\max_{\theta} V(\theta) + \lambda \left(V_g(\theta) - \xi_g \right). \tag{10}$$

It is worth noting that, in general, there is no guarantee that a fixed coefficient λ achieves the same solution as (5) (an exception is, for example, in cases where (5) is convex (Boyd et al., 2004)). However, λ trades off the optimality of the objective and the safety constraint. Indeed, for large values of λ solutions to (10) will prioritize safe behaviors, whereas for small values of λ the solutions will focus on maximizing the value function (1). Alternatively, one approach to automatically search appropriate values of λ is to use iterative methods such as the primal-dual type algorithm. Therefore, we propose a Generalized Safe Primal Dual (GSPD) algorithm that is summarized under Algorithm 1 for solving problem (10). The intuition of the algorithm is to update the policy variable as in (10) using any RL algorithm of choice. Indeed, at each step, the GSPD constructs the Lagrangian where the reward is augmented by the product of the last N indicator function with the weight λ^k, i.e., $\mathcal{L}(s_t, a_t) = r(s_t, a_t) + \lambda^k \prod_{u=t-N}^{t} \mathbb{1}(s_u, a_u)$. Then we append the trajectory $(s_t, a_t, \mathcal{L}(s_t, a_t))$ into the replay buffer $\mathcal{B}$. Then, the policy is updated using any unconstrained RL algorithm (this is step 6 in Algorithm 1, where *RL-Algo* denotes the algorithm of choice, e.g., PPO (Schulman et al., 2017), TD3 (Fujimoto et al., 2018), SAC (Haarnoja et al., 2018),).

To update λ we follow the intuition described earlier regarding how λ trades off constraint satisfaction and optimality. We estimate the probabilistic safety by $\mathbb{S} = \prod_{t=0}^{T} \mathbb{1}(s_t, a_t)$ at each episode k, and thus update the dual variable λ^{k+1} using $\lambda^{k+1} = \left[\lambda^k - \eta_\lambda \left(\mathbb{S} - (1 - \delta)\right)\right]_+$, where η_λ denotes the dual step-size. Thus, if the trajectories are safer than $1 - \delta$ on average, it will result in a decrease in λ whereas if they are safe less than $1 - \delta$ fraction of the time λ will increase. This update can also be understood as a stochastic gradient descent on the dual function. It is worth noting that Algorithm 1 reduces to the algorithms presented in (Chen et al., 2024b) and (Paternain et al., 2022) when N is set to be T and 0, respectively.

Algorithm 1 Generalized Safe Primal Dual (GSPD)

```
Input: GSRL hyper-parameter $N$, initial primal variable $\theta^0$, initial dual variable $\lambda^0$, primal stepsize
    $\eta_\theta$, dual stepsize $\eta_\lambda$, safety threshold $\delta$, horizon $T$, backbone algorithm RL-Algo, replay buffer
    $\mathcal{B}$, batch size $Z$
for $k = 0, 1, \ldots$ do
    for $t = 0, 1, \ldots, T$ do
        Construct the Lagrangian by
            $$\mathcal{L}(s_t, a_t) = r(s_t, a_t) + \lambda^k \prod_{u=t-N}^{t} \mathbb{1}(s_u, a_u)$$
        Append the trajectory into the buffer $\mathcal{B}$
            $$\mathcal{B} = \mathcal{B} \cup (s_t, a_t, \mathcal{L}(s_t, a_t))$$
    end for
    when RL-Algo update do
        Sample a batch of $Z$ trajectories from the buffer $\mathcal{B}$
        Update the primal variable $\theta$ using RL-Algo and $\eta_\theta$
    Obtain the safety by $\mathbb{S} = \prod_{t=0}^{T} \mathbb{1}(s_t, a_t)$
    Update the dual variable using
        $$\lambda^{k+1} = \left[\lambda^k - \eta_\lambda \left(\mathbb{S} - (1 - \delta)\right)\right]_+$$
end for
```

4. Numerical Results

In this section, we demonstrate the numerical performance of the GSPD algorithm presented in Section 3. To do so we consider the tasks Run, Circle, Reach using the agents Ball and Car from the Bullet Safety Gym (Gronauer, 2022) as shown in Figure 1. In the three tasks, both the Ball and Car agents utilize states that encompass their position, linear, and angular velocities. The Ball agent is controlled by a two-dimensional force vector, while the Car agent is based on a control scheme consisting of the target wheel velocity for all four wheels and the target steering angle.

As illustrated in Figure 1(c), the agent in the Run task (Chow et al., 2019) receives rewards for navigating through a designated avenue delimited by two safety boundaries. These boundaries, though intangible, incur costs upon penetration without collision. Furthermore, the agent incurs additional costs if it surpasses the (agent-specific) velocity threshold. The reward and the cost are defined by

$$\begin{aligned} r(s_t) &= \left\|\boldsymbol{p^{t-1}} - \boldsymbol{g}\right\|_2 - \left\|\boldsymbol{p^t} - \boldsymbol{g}\right\|_2 + r_{\text{robot}}\left(s_t\right), \\ c(s_t) &= \mathbb{1}\left(|p_y^t| > y_{\text{lim}}\right) + \mathbb{1}\left(\left\|\boldsymbol{v^t}\right\|_2 > v_{\text{lim}}\right), \end{aligned} \tag{11}$$

where $r_{\text{robot}}\left(s_t\right)$ specifies the unique reward for various robots, $\boldsymbol{p^t} = \left[p_x^t, p_y^t\right]$ defines the position of the agent at time step t, $\boldsymbol{g} = [g_x, g_y]$ represents the position of a fictitious target, y_{lim} defines the safety region, $\boldsymbol{v^t} = \left[v_x^t, v_y^t\right]$ denotes the agent's velocity at time t, and v_{lim} denotes the speed limit.

In the Circle task (Achiam et al., 2017) (see Figure 1(d)), the agent is tasked with navigating along a circular trajectory in a clockwise direction. The reward structure is characterized by its density, escalating in tandem with the agent's velocity and its proximity to the boundary of the circle. Incurred costs arise when the agent deviates from the designated safety zone, delineated by two yellow boundaries. The reward and cost functions for the Circle task are delineated as follows

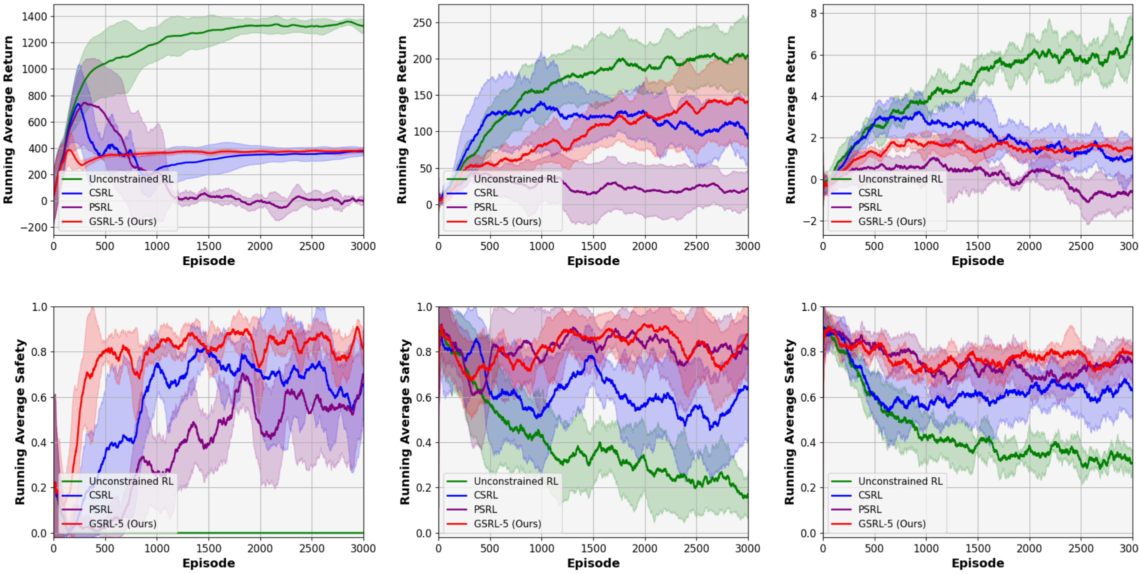
$$\begin{aligned} r\left(s_t\right) &= \frac{-p_y^t v_x^t + p_x^t v_y^t}{1 + \left| \, \|\boldsymbol{p^t}\|_2 - o \right|} + r_{\text{robot}} \left(s_t\right), \\ c\left(s_t\right) &= \mathbb{1}\left(\left|p_x^t\right| > x_{\text{lim}}\right), \end{aligned} \tag{12}$$

where o denotes the radius of the circle and x_{lim} represents the boundaries of the safety region.

Figure 1(e) depicts the Reach task (Ray et al., 2019), where the reward system comprises a dense component, rewarding the agent for advancing closer to the goal, and a sparse component, granted upon successfully entering the goal zone. Upon the agent's entry into the goal zone, the goal is promptly regenerated, necessitating the agent to reach the next position. Obstacles are strategically positioned to impede the agent from effortlessly discovering solutions. These obstacles are designed with physical bodies, serving as collision points that incur costs upon impact, and also include elements without collision shapes, imposing costs for mere contact. The reward and cost functions are defined as

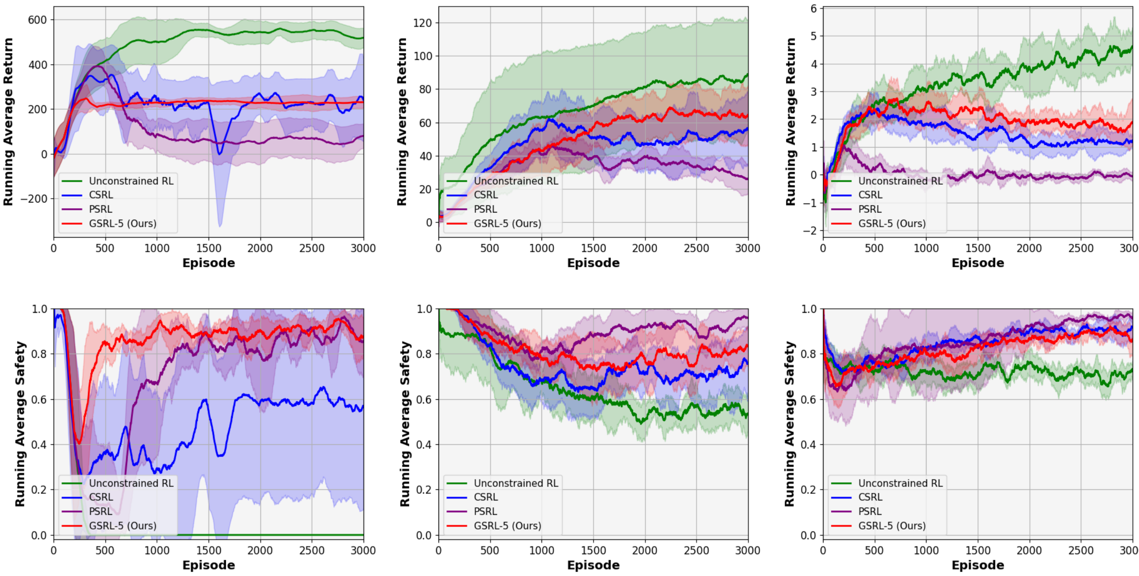
$$\begin{aligned} r(s_t) &= \text{Distance}(\text{target}, s_{t-1}) - \text{Distance}(\text{target}, s_t), \\ c(s_t) &= \mathbb{1}(\mathcal{C}). \end{aligned} \tag{13}$$

where $\mathcal{C}$ represents the collision between the agent and the hazards populated in the environment.



(*a*) *SafetyBallRun-v0* (*b*) *SafetyBallCircle-v0* (*c*) *SafetyBallReach-v0*

Figure 2: The training curves of the running average return and safety (window=100) with the Ball agent in the Run, Circle, and Reach Task. The *RL-Algo* employed in Algorithm 1 is PPO.



(a) *SafetyCarRun-v0* (b) *SafetyCarCircle-v0* (c) *SafetyCarReach-v0*

Figure 3: The training curves of the running average return and safety (window=100) with the Car agent in the Run, Circle, and Reach Task. The *RL-Algo* employed in Algorithm 1 for *SafetyCarRun-v0* and *SafetyCarCircle-v0* is PPO, while TD3 is selected for *SafetyCarReach-v0*.

Having established the general RL settings as above, we are in the stage of validating the GSPD algorithm (Algorithm 1). We consider six environments that combines the two agents and three tasks aforementioned. We implement Algorithm 1 in each environment, compared with three baselines: the unconstrained RL (maximizing solely (1)), the PSRL (2) and the CSRL (4). The unconstrained RL baseline is to estimate the upper bound of the achievable returns when safety aspects are not regarded. As discussed in Proposition 3, the PSRL and CSRL settings are the special cases of the GSRL when $N = T$ and $N = 0$, respectively. We choose $N = 5$ in all experiments for our implementation of Algorithm 1, which we term "GSRL-5". In addition, we use the same settings in all six implementations for a fair comparison. In all six implementations, the initial value of λ is set to be 0. The desired level of safety, expressed as $1 - \delta$, is set to be 0.95. The total number of iterations for the algorithm is established as 3000. The time horizon T is selected to be 200. To substantiate our assertion allowing flexibility in the choice of any RL algorithm as the *RL-Algo* input, we opt for PPO (Schulman et al., 2017) as the designated *RL-Algo* in all experiments, with the exception of *SafetyCarReach-v0*, where TD3 (Fujimoto et al., 2018) is selected.

We summarize the numerical results in Figures 2 and 3. We plot the running average return and running average safety of a window of 100. Note that the safety in each episode is estimated by $\mathbb{S} = \prod_{t=0}^{T} \mathbb{1}(s_t, a_t)$. In all six experiments, the unconstrained RL baseline achieves the highest return accompanied by however, low safety. Especially in *SafetyBallRun-v0* and *SafetyCarRun-v0* (see Figures 2(a) and 3(a)), the unconstrained RL leads to a safety of zero. As observed in all experiments

Table 1: The running average of return and safety upon reaching convergence.

Environment	Algorithm	Return	Safety	Environment	Algorithm	Return	Safety
BallRun	Unconstrained	1335.50 ± 7.18	0.0 ± 0.0	*CarRun*	Unconstrained	515.72 ± 1.28	0.0 ± 0.0
	CSRL	373.84 ± 0.39	0.60 ± 0.03		CSRL	$\mathbf{240.69 \pm 12.56}$	0.56 ± 0.01
	PSRL	3.14 ± 4.71	0.61 ± 0.05		PSRL	71.16 ± 7.24	0.87 ± 0.01
	GSRL-5	$\mathbf{379.93 \pm 1.41}$	$\mathbf{0.87 \pm 0.03}$		GSRL-5	230.29 ± 0.27	$\mathbf{0.88 \pm 0.01}$
BallCircle	Unconstrained	204.02 ± 1.65	0.17 ± 0.01	*CarCircle*	Unconstrained	86.05 ± 1.57	0.55 ± 0.02
	CSRL	103.30 ± 5.69	0.63 ± 0.01		CSRL	54.94 ± 0.72	0.75 ± 0.02
	PSRL	18.05 ± 1.78	0.81 ± 0.01		PSRL	26.88 ± 0.70	$\mathbf{0.96 \pm 0.01}$
	GSRL-5	$\mathbf{142.89 \pm 2.15}$	$\mathbf{0.84 \pm 0.02}$		GSRL-5	$\mathbf{63.69 \pm 0.64}$	0.82 ± 0.01
BallReach	Unconstrained	6.33 ± 0.32	0.32 ± 0.01	*CarReach*	Unconstrained	4.46 ± 0.06	0.71 ± 0.01
	CSRL	0.93 ± 0.06	0.66 ± 0.01		CSRL	1.18 ± 0.05	0.89 ± 0.01
	PSRL	-0.66 ± 0.06	0.76 ± 0.01		PSRL	0.01 ± 0.04	$\mathbf{0.97 \pm 0.01}$
	GSRL-5	$\mathbf{1.48 \pm 0.04}$	$\mathbf{0.80 \pm 0.01}$		GSRL-5	$\mathbf{1.69 \pm 0.11}$	0.88 ± 0.01

except *SafetyCarCircle-v0* and *SafetyCarReach-v0*, our Algorithm 1 achieves both higher return and higher safety than the PSRL and CSRL baselines. In *SafetyCarCircle-v0* and *SafetyCarReach-v0*, Algorithm 1 still outperforms the CSRL setting by achieving both higher return and higher safety. While the PSRL obtains a higher safety than Algorithm 1, it sacrifices the return a lot. Hence, Algorithm 1 exhibits an overall better return-safety trade-off than both the PSRL and CSRL formulations across all experiments. For a better readability for the return and safety upon reaching convergence, we compute the average and the standard deviation of the last 100 running average return/safety in Figures 2 and 3, and summarize the results in Table 1. We focus on comparing the return-safety trade-off between the CSRL, the PSRL and our implemented GSRL-5. Therefore, the unconstrained RL case is not taken into consideration for the comparison. As observed, in all experiments with the Ball agent, our GSRL-5 yields both higher return and safety upon reaching convergence. In the Car-agent experiments, our GSRL-5 method achieves high return while not sacrificing the safety too much. These numerical evaluations in *Bullet-Safety-Gym* validate that our Algorithm 1 finds an overall better trade-off than both the PSRL and CSRL baselines between finding a better solution and reducing the variance of the estimate in the gradient steps.

5. Conclusion

In this work, we have studied the problem of learning safe policies under the PSRL setting. Concretely, a safe policy is defined as one that guarantees, with high probability, that the agent remains in a desired safe set across the whole trajectory. We have proposed a GSRL framework that can recover both the PSRL setting and the CSRL formulation that is commonly considered in the literature. Indeed, our theoretical results validate that the GSRL finds safe policies for PSRL problems. The intuition of the proposed method is that it finds a better trade-off than both the PSRL and CSRL in terms of the return and safety. Accordingly, a GSPD algorithm is proposed that can be combined with any state-of-the-art RL algorithms. The GSPD algorithm in this work is substantiated by a series of numerical experiments in *Bullet-Safety-Gym*, indicating that the GSPD outperforms the CSRL and PSRL baselines. Future work includes: (i) combining the GSPD with more cutting-edge RL algorithms, e.g., SAC-Lagrangian, CPO, PCPO, FOCOPS, etc, (ii) characterizing the running time and sample complexity of the GSPD algorithm, (iii) applying the GSPD algorithm to the systems with more complicated dynamics.

Acknowledgments

This work is supported by the Rensselaer-IBM AI Research Collaboration. The authors thank Zuxin Liu for his thoughtful suggestions and valuable insights.

References

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International conference on machine learning*, pages 22–31. PMLR, 2017.

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

Kenneth Joseph Arrow, Hirofumi Azawa, Leonid Hurwicz, and Hirofumi Uzawa. *Studies in linear and non-linear programming*, volume 2. Stanford University Press, 1958.

Shalabh Bhatnagar and K Lakshmanan. An online actor–critic algorithm with function approximation for constrained markov decision processes. *Journal of Optimization Theory and Applications*, 153(3):688–708, 2012.

Vivek S Borkar. An actor-critic algorithm for constrained markov decision processes. *Systems & control letters*, 54(3):207–213, 2005.

Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Sinan Çalışır and Meltem Kurt Pehlivanoğlu. Model-free reinforcement learning algorithms: A survey. In *2019 27th signal processing and communications applications conference (SIU)*, pages 1–4. IEEE, 2019.

Yair Censor. Pareto optimality in multiobjective problems. *Applied Mathematics and Optimization*, 4(1):41–59, 1977.

Weiqin Chen, Dharmashankar Subramanian, and Santiago Paternain. Policy gradients for probabilistic constrained reinforcement learning. In *2023 57th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6. IEEE, 2023.

Weiqin Chen, James Onyejizu, Long Vu, Lan Hoang, Dharmashankar Subramanian, Koushik Kar, Sandipan Mishra, and Santiago Paternain. Adaptive primal-dual method for safe reinforcement learning. *arXiv preprint arXiv:2402.00355*, 2024a.

Weiqin Chen, Dharmashankar Subramanian, and Santiago Paternain. Probabilistic constraint for safety-critical reinforcement learning. *IEEE Transactions on Automatic Control*, 2024b.

Richard Cheng, Gábor Orosz, Richard M Murray, and Joel W Burdick. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3387–3395, 2019.

Yinlam Chow, Mohammad Ghavamzadeh, Lucas Janson, and Marco Pavone. Risk-constrained reinforcement learning with percentile risk criteria. *The Journal of Machine Learning Research*, 18(1):6070–6120, 2017.

Yinlam Chow, Ofir Nachum, Aleksandra Faust, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. Lyapunov-based safe policy optimization for continuous control. *arXiv preprint arXiv:1901.10031*, 2019.

Davide Corsi, Enrico Marchesini, and Alessandro Farinelli. Formal verification of neural networks for safety-critical tasks in deep reinforcement learning. In *Uncertainty in Artificial Intelligence*, pages 333–343. PMLR, 2021.

Erick Delage and Shie Mannor. Percentile optimization for markov decision processes with parameter uncertainty. *Operations research*, 58(1):203–213, 2010.

Rick Durrett. *Probability: theory and examples*, volume 49. Cambridge university press, 2019.

Gonzalo Farias, Gonzalo Garcia, Guelis Montenegro, Ernesto Fabregas, Sebastian Dormido-Canto, and Sebastian Dormido. Reinforcement learning for position control problem of a mobile robot. *IEEE Access*, 8:152941–152951, 2020.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.

Javier Garcıa and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

Peter Geibel. Reinforcement learning for mdps with constraints. In *European Conference on Machine Learning*, pages 646–653. Springer, 2006.

Sven Gronauer. Bullet-safety-gym: A framework for constrained reinforcement learning. 2022.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

Yoshinobu Kadota, Masami Kurano, and Masami Yasuda. Discounted markov decision processes with utility constraints. *Computers & Mathematics with Applications*, 51(2):279–284, 2006.

Gregory Kahn, Adam Villaflor, Bosen Ding, Pieter Abbeel, and Sergey Levine. Self-supervised deep reinforcement learning with generalized computation graphs for robot navigation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5129–5136. IEEE, 2018.

Zhongyu Li, Xuxin Cheng, Xue Bin Peng, Pieter Abbeel, Sergey Levine, Glen Berseth, and Koushil Sreenath. Reinforcement learning for robust parameterized locomotion control of bipedal robots. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2811–2817. IEEE, 2021.

Qingkai Liang, Fanyu Que, and Eytan Modiano. Accelerated primal-dual policy optimization for safe reinforcement learning. *arXiv preprint arXiv:1802.06480*, 2018.

Hai Nguyen and Hung La. Review of deep reinforcement learning for robot manipulation. In *2019 Third IEEE International Conference on Robotic Computing (IRC)*, pages 590–595. IEEE, 2019.

Santiago Paternain, Miguel Calvo-Fullana, Luiz FO Chamon, and Alejandro Ribeiro. Safe policies for reinforcement learning via primal-dual methods. *IEEE Transactions on Automatic Control*, 2022.

Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning. *arXiv preprint arXiv:1910.01708*, 7:1, 2019.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Li Shen, Long Yang, Shixiang Chen, Bo Yuan, Xueqian Wang, Dacheng Tao, et al. Penalized proximal policy optimization for safe reinforcement learning. *arXiv preprint arXiv:2205.11814*, 2022.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

Chen Tessler, Daniel J Mankowitz, and Shie Mannor. Reward constrained policy optimization. *arXiv preprint arXiv:1805.11074*, 2018.

Thierry Van Cutsem. Voltage instability: phenomena, countermeasures, and analysis methods. *Proceedings of the IEEE*, 88(2):208–227, 2000.

Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3):279–292, 1992.

Tsung-Yen Yang, Justinian Rosca, Karthik Narasimhan, and Peter J Ramadge. Projection-based constrained policy optimization. *arXiv preprint arXiv:2010.03152*, 2020.

Jesse Zhang, Brian Cheung, Chelsea Finn, Sergey Levine, and Dinesh Jayaraman. Cautious adaptation for reinforcement learning in safety-critical settings. In *International Conference on Machine Learning*, pages 11055–11065. PMLR, 2020a.

Yiming Zhang, Quan Vuong, and Keith Ross. First order constrained optimization in policy space. *Advances in Neural Information Processing Systems*, 33:15338–15349, 2020b.

Weiye Zhao, Tairan He, Rui Chen, Tianhao Wei, and Changliu Liu. State-wise safe reinforcement learning: A survey. *arXiv preprint arXiv:2302.03122*, 2023.