CRPO: A New Approach for Safe Reinforcement Learning with Convergence Guarantee

Tengyu Xu [1] **Yingbin Lang** [1] **Guanghui Lan** [2]

Abstract

In safe reinforcement learning (SRL) problems, an agent explores the environment to maximize an expected total reward and meanwhile avoids violation of certain constraints on a number of expected total costs. In general, such SRL problems have nonconvex objective functions subject to multiple nonconvex constraints, and hence are very challenging to solve, particularly to provide a globally optimal policy. Many popular SRL algorithms adopt a primal-dual structure which utilizes the updating of dual variables for satisfying the constraints. In contrast, we propose a primal approach, called constraint-rectified policy optimization (CRPO), which updates the policy alternatingly between objective improvement and constraint satisfaction. CRPO provides a primal-type algorithmic framework to solve SRL problems, where each policy update can take any variant of policy optimization step. To demonstrate the theoretical performance of CRPO, we adopt natural policy gradient (NPG) for each policy update step and show that CRPO achieves an $\mathcal{O}(1/\sqrt{T})$ convergence rate to the global optimal policy in the constrained policy set and an $\mathcal{O}(1/\sqrt{T})$ error bound on constraint satisfaction. This is the first finite-time analysis of primal SRL algorithms with global optimality guarantee. Our empirical results demonstrate that CRPO can outperform the existing primal-dual baseline algorithms significantly.

1. Introduction

Reinforcement learning (RL) has achieved great success in solving complex sequential decision-making and control problems such as Go (Silver et al., 2017), StarCraft (DeepMind, 2019) and recommendation system (Zheng et al., 2018), etc. In these settings, the agent is allowed to explore the entire state and action space to maximize the expected total reward. However, in safe RL (SRL), in addition to maximizing the reward, an agent needs to satisfy certain constraints. Examples include self-driving cars (Fisac et al., 2018), cellular network (Julian et al., 2002), and robot control (Levine et al., 2016). The global optimal policy in SRL is the one that maximizes the reward and at the same time satisfies the cost constraints.

The current safe RL algorithms can be generally categorized into the **primal** and **primal-dual** approaches. The **primal-dual** approaches (Tessler et al., 2018; Ding et al., 2020a; Stooke et al., 2020; Yu et al., 2019; Achiam et al., 2017; Yang et al., 2019a; Altman, 1999; Borkar, 2005; Bhatnagar & Lakshmanan, 2012; Liang et al., 2018; Paternain et al., 2019a) are most commonly used, which convert the constrained problem into an unconstrained one by augmenting the objective with a sum of constraints weighted by their corresponding Lagrange multipliers (i.e., dual variables). Generally, primal-dual algorithms apply a certain policy optimization update such as policy gradient alternatively with a gradient descent type update for the dual variables. Theoretically, (Tessler et al., 2018) has provided an asymptotic convergence analysis for primal-dual method and established a local convergence guarantee. (Paternain et al., 2019b) showed that the primal-dual method achieves zero duality gap. Recently, (Ding et al., 2020a) proposed a primal-dual type proximal policy optimization (PPO) and established the regret bound for linear constrained MDP. The convergence rate of primal-dual method based on a natural policy gradient algorithm was characterized in (Ding et al., 2020b).

The **primal** type of approaches (Liu et al., 2019b; Chow et al., 2018; 2019; Dalal et al., 2018a) enforce constraints via various designs of the objective function or the update process without an introduction of dual variables. The **primal** algorithms are much less studied than the primal-dual approach. Notably, (Liu et al., 2019b) developed an interior point method, which applies logarithmic barrier functions for SRL. (Chow et al., 2018; 2019) leveraged Lyapunov functions to handle constraints. (Dalal et al., 2018a) introduced a safety layer to the policy network to enforce

[1]Department of Electrical and Computer Engineering, The Ohio State University, OH, United States [2]Industrial and Systems Engineering, Georgia Institute of Technology, GA, United States. Correspondence to: Tengyu Xu <xu.3260@osu.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

constraints. None of the existing primal algorithms are shown to have provable convergence guarantee to a globally optimal feasible policy.

Comparing between the primal-dual and primal approaches, the primal-dual approach can be sensitive to the initialization of Lagrange multipliers and the learning rate, and can thus incur extensive cost in hyperparameter tuning (Achiam et al., 2017; Chow et al., 2019). In contrast, the primal approach does not introduce additional dual variables to optimize and involves less hyperparamter tuning, and hence holds the potential to be much easier to implement than the primal-dual approach. However, the existing **primal** algorithms are not yet popular in practice so far, because of no guaranteed global convergence and no strong demonstrations to have competing performance as the primal-dual algorithms. Thus, in order to take the advantage of the primal approach which is by nature easier to implement, we need to answer the following fundamental questions.

- ▷ Can we design a primal algorithm for SRL, and demonstrate that it achieves competing performance or outperforms the baseline primal-dual approach?
- ▷ If so, can we establish global optimality guarantee and the finite-time convergence rate for the proposed primal algorithm?

In this paper, we will provide the affirmative answers to the above questions, thus establishing appealing advantages of the primal approach for SRL.

1.1. Main Contributions

A New Algorithm: We propose a novel primal approach called **Constraint-Rectified Policy Optimization (CRPO)** for SRL, where all updates are taken in the primal domain. CRPO applies *unconstrained* policy maximization update w.r.t. the reward on the one hand, and if any constraint is violated, momentarily rectifies the policy back to the constraint set along the descent direction of the violated constraint also by applying *unconstrained* policy minimization update w.r.t. the constraint function. From the implementation perspective, CRPO can be implemented as easy as unconstrained policy optimization algorithms. Without introduction of dual variables, it does not suffer from hyperparameter tuning of the learning rates to which the dual variables are sensitive, nor does it require initialization to be feasible. Further, CRPO involves only policy gradient descent for both objective and constraints, whereas the primal-dual approach typically requires *projected* gradient descent, where the projection causes higher complexity to implementation as well as hyperparameter tuning due to the projection thresholds.

To further explain the advantage of CRPO over the primal-dual approach, CRPO features **immediate switches** between optimizing the objective and reducing the constraints whenever constraints are violated. However, the primal-dual approach can respond much slower because the control is based on dual variables. If a dual variable is nonzero, then the policy update will descend along the corresponding constraint function. As a result, even if a constraint is already satisfied, there can often be a significant delay for the dual variable to iteratively reduce to zero to release the constraint, which slows down the algorithm. Our experiments in Section 5 validates such a performance advantage of CRPO over the primal-dual approach.

Theoretical Guarantee: To provide the theoretical guarantee for CRPO, we adopt NPG as a representative policy optimizer and investigate the convergence of CRPO in two settings: tabular and function approximation, where in the function approximation setting the state space can be infinite. For both settings, we show that CRPO converges to a global optimum at a convergence rate of $\mathcal{O}(1/\sqrt{T})$. Furthermore, the constraint violation also converges to zero at a rate of $\mathcal{O}(1/\sqrt{T})$. To the best of our knowledge, we establish the first provably global optimality guarantee for a primal SRL algorithm of CRPO.

To compare with the primal-dual approach in the function approximation setting, the value function gap of CRPO achieves the same convergence rate as the primal-dual approach, but the constraint violation of CRPO decays at a rate of $\mathcal{O}(1/\sqrt{T})$, which is much faster than the rate $\mathcal{O}(1/T^{\frac{1}{4}})$ of the primal-dual approach (Ding et al., 2020b).

Technically, our analysis has the following novel developments. (a) We develop a new technique to analyze a stochastic approximation (SA) that randomly and dynamically switches between the target objectives of the reward and the constraint. Such an SA by nature is different from the analysis of a typical policy optimization algorithm, which has a fixed target objective to optimize. Our analysis constructs novel concentration events for capturing the impact of such a dynamic process on the update of the reward and cost functions in order to establish the high probability convergence guarantee. (b) We also develop new tools to handle multiple constraints, which is particularly non-trivial for our algorithm that involves stochastic selection of a constraint if multiple constraints are violated.

1.2. Related Work

Safe RL: Algorithms based on primal-dual methods have been widely adopted for solving constrained RL problems, such as PDO (Chow et al., 2017), RCPO (Tessler et al., 2018), OPDOP (Ding et al., 2020a) and CPPO (Stooke et al., 2020). Constrained policy optimization (CPO) (Achiam et al., 2017) extends TRPO to handle constraints, and is later modified with a two-step projection method (Yang et al., 2019a). The effectiveness of primal-dual methods is justified in (Paternain et al., 2019b), in which zero duality gap is guaranteed under certain assumptions. A recent work (Ding et al., 2020b) established the convergence rate of the primal-dual method under Slater's condition assumption. Other methods have also been proposed. For example, (Chow et al., 2018; 2019) leveraged Lyapunov functions to handle constraints. (Yu et al., 2019) proposed a constrained policy gradient algorithm with convergence guarantee by solving a sequence of sub-problems. (Dalal et al., 2018a) proposed to add a safety layer to the policy network so that constraints can be satisfied at each state. (Liu et al., 2019b) developed an interior point method for safe RL, which augments the objective with logarithmic barrier functions. Our work proposes a CRPO algorithm, which can be implemented as easy as unconstrained policy optimization methods and has global optimality guarantee under general constrained MDP. Our result is the first convergence rate characterization of primal-type algorithms for SRL.

Finite-Time Analysis of Policy Optimization: The finite-time analysis of various policy optimization algorithms under unconstrained MDPs have been well studied. The convergence rate of policy gradient (PG) and actor-critic (AC) algorithms have been established in (Shen et al., 2019; Papini et al., 2017; 2018; Xu et al., 2020a; 2019a; Xiong et al., 2020; Zhang et al., 2019) and (Xu et al., 2020b; Wang et al., 2019; Yang et al., 2019b; Kumar et al., 2019; Qiu et al., 2019), respectively, in which PG or AC algorithm is shown to converge to a local optimal. In some special settings such as tabular and LQR, PG and AC can be shown to convergence to the global optimal (Agarwal et al., 2019; Yang et al., 2019b; Fazel et al., 2018; Malik et al., 2018; Tu & Recht, 2018; Bhandari & Russo, 2019; 2020). Algorithms such as NPG, NAC, TRPO and PPO explore the second order information, and achieve great success in practice. These algorithms have been shown to converge to a global optimum in various settings, where the convergence rate has been established in (Agarwal et al., 2019; Shani et al., 2019; Liu et al., 2019a; Wang et al., 2019; Cen et al., 2020; Xu et al., 2020c).

2. Problem Formulation and Preliminaries

2.1. Markov Decision Process

A discounted Markov decision process (MDP) is a tuple $(\mathcal{S}, \mathcal{A}, c_0, \mathsf{P}, \xi, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ are state and action spaces; $c_0 : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function; $\mathsf{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition kernel, with $\mathsf{P}(s'|s, a)$ denoting the probability of transitioning to state s' from previous state s given action a; $\xi : \mathcal{S} \to [0, 1]$ is the initial state distribution; and $\gamma \in (0, 1)$ is the discount factor. A policy $\pi : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ is a mapping from the state space to the space of probability distributions over the actions, with $\pi(\cdot|s)$ denoting the probability of selecting action a in state s. When the associated Markov chain $\mathsf{P}(s'|s) = \sum_{\mathcal{A}} P(s'|s, a)\pi(a|s)$ is ergodic, we denote μ_π as the stationary distribution of this MDP, i.e. $\int_{\mathcal{S}} \mathsf{P}(s'|s)\mu_\pi(ds) = \mu_\pi(s')$. Moreover, we define the visitation measure induced by the police π as $\nu_\pi(s, a) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \mathsf{P}(s_t = s, a_t = a)$.

For a given policy π, we define the state value function as $V_\pi^0(s) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t c_0(s_t, a_t, s_{t+1})|s_0 = s, \pi]$, the state-action value function as $Q_\pi^0(s, a) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t c_0(s_t, a_t, s_{t+1})|s_0 = s, a_0 = a, \pi]$, and the advantage function as $A_\pi^0(s, a) = Q_\pi^0(s, a) - V_\pi^0(s)$. In reinforcement learning, we aim to find an optimal policy that maximizes the expected total reward function defined as $J_0(\pi) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t c_0(s_t, a_t, s_{t+1})] = \mathbb{E}_\xi[V_\pi^0(s)] = \mathbb{E}_{\xi\cdot\pi}[Q_\pi^0(s, a)]$.

2.2. Safe Reinforcement Learning (SRL) Problem

The SRL problem is formulated as an MDP with additional constraints that restrict the set of allowable policies. Specifically, when taking action at some state, the agent can incur a number of costs denoted by $c_1, \cdots, c_p$, where each cost function $c_i : \mathcal{S}\times\mathcal{A}\times\mathcal{S} \to \mathbb{R}$ maps a tuple (s, a, s') to a cost value. Let $J_i(\pi)$ denotes the expected total cost function with respect to c_i as $J_i(\pi) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t c_i(s_t, a_t, s_{t+1})]$. The goal of the agent in SRL is to solve the following constrained problem

$$\max_\pi J_0(\pi), \text{ s.t. } J_i(\pi) \le d_i, \ \forall i = 1, \cdots, p, \qquad (1)$$

where d_i is a fixed limit for the i-th constraint. We denote the set of feasible policies as $\Omega_C \equiv \{\pi : \forall i, J_i(\pi) \le d_i\}$, and define the optimal policy for SRL as $\pi^* = \arg\min_{\pi\in\Omega_C} J_0(\pi)$. For each cost c_i, we define its corresponding state value function V_π^i, state-action value function Q_π^i, and advantage function A_π^i analogously to V_π^0, Q_π^0, and A_π^0, with c_i replacing c_0, respectively.

2.3. Policy Parameterization and Policy Gradient

In practice, a convenient way to solve the problem eq. (1) is to parameterize the policy and then optimize the policy over the parameter space. Let $\{\pi_w : \mathcal{S} \to \mathcal{P}(\mathcal{A}) | w \in \mathcal{W}\}$ be a parameterized policy class, where $\mathcal{W}$ is the parameter space. Then, the problem in eq. (1) becomes

$$\max_{w\in\mathcal{W}} J_0(\pi_w), \text{ s.t. } J_i(\pi_w) \le d_i, \ \forall i = 1, \cdots, p. \qquad (2)$$

The policy gradient of the function $J_i(\pi_w)$ has been derived by (Sutton et al., 2000) as $\nabla J_i(\pi_w) = \mathbb{E}[Q_{\pi_w}^i(s, a)\phi_w(s, a)]$, where $\phi_w(s, a) := \nabla_w \log \pi_w(a|s)$ is the score function. Furthermore, the natural policy gradient was defined by (Kakade, 2002) as $\Delta_i(w) = F(w)^\dagger \nabla J_i(\pi_w)$, where $F(w)$ is the Fisher information matrix defined as $F(w) = \mathbb{E}_{\nu_{\pi_w}}[\phi_w(s, a)\phi_w(s, a)^\top]$.

Algorithm 1 Constraint-Rectified Policy Optimization (CRPO)

1: **Initialize:** initial parameter w_0, empty set $\mathcal{N}_0$
2: **for** $t = 0, \cdots, T-1$ **do**
3: Policy evaluation under π_{w_t}: $\bar{Q}_t^i(s, a) \approx Q_{\pi_{w_t}}^i(s, a)$
4: Sample $(s_j, a_j) \in \mathcal{B}_t \sim \xi \cdot \pi_{w_t}$, compute constrain estimation $\bar{J}_{i,\mathcal{B}_t} = \sum_{j \in \mathcal{B}_t} \rho_{j,t} \bar{Q}_t^i(s_j, a_j)$ for $i = 0, \cdots, p$, ($\rho_{j,t}$ is the weight)
5: **if** $\bar{J}_{i,\mathcal{B}_t} \leq d_i + \eta$ for all $i = 1, \cdots, p$, **then**
6: Add w_t into set $\mathcal{N}_0$
7: Take one-step policy update towards maximize $J_0(w_t)$: $w_t \to w_{t+1}$
8: **else**
9: Choose any $i_t \in \{1, \cdots, p\}$ such that $\bar{J}_{i_t,\mathcal{B}_t} > d_{i_t} + \eta$
10: Take one-step policy update towards minimize $J_{i_t}(w_t)$: $w_t \to w_{t+1}$
11: **end if**
12: **end for**
13: **Output:** w_{out} uniformly chosen from $\mathcal{N}_0$

3. Constraint-Rectified Policy Optimization (CRPO) Algorithm

In this section, we propose the CRPO approach (see Algorithm 1) for solving the SRL problem in eq. (2). The idea of CRPO lies in updating the policy to maximize the unconstrained objective function $J_0(\pi_{w_t})$ of the reward, alternatingly with rectifying the policy to reduce a constraint function $J_i(\pi_{w_t})$ $(i \geq 1)$ (along the descent direction of this constraint) if it is violated. Each iteration of CRPO consists of the following three steps.

Policy Evaluation: At the beginning of each iteration, we estimate the state-action value function $\bar{Q}_{\pi_t}^i(s, a) \approx Q_{\pi_{w_t}}^i(s, a)$ $(i = \{0, \cdots, p\})$ for both reward and costs under current policy π_{w_t}.

Constraint Estimation: After obtaining $\bar{Q}_{\pi_t}^i$, the constraint function $J_i(w_t) = \mathbb{E}_{\xi \cdot \pi_{w_t}}[Q_{w_t}^i(s, a)]$ can then be approximated via a weighted sum of approximated state-action value function: $\bar{J}_{i,\mathcal{B}_t} = \sum_{j \in \mathcal{B}_t} \rho_{j,t} \bar{Q}_t^i(s_j, a_j)$. Note this step does not take additional sampling cost, as the generation of samples $(s_j, a_j) \in \mathcal{B}_t$ from distribution $\xi \cdot \pi_{w_t}$ does not require the agent to interact with the environment.

Policy Optimization: We then check whether there exists an $i_t \in \{1, \cdots, p\}$ such that the approximated constraint $\bar{J}_{i_t,\mathcal{B}_t}$ violates the condition $\bar{J}_{i_t,\mathcal{B}_t} \leq d_i + \eta$, where η is the tolerance. If so, we take **one-step** update of the policy towards minimizing the corresponding constraint function $J_{i_t}(\pi_{w_t})$ to enforce the constraint. If multiple constraints are violated, we can choose to minimize any one of them. If all constraints are satisfied, we take **one-step** update of the policy towards maximizing the objective function $J_0(\pi_{w_t})$. To apply CRPO in practice, we can use any policy optimization update such as natural policy gradient (NPG) (Kakade, 2002), trust region policy optimization (TRPO) (Schulman et al., 2015), proximal policy optimization (PPO) (Schulman et al., 2017), ACKTR (Wu et al., 2017), DDPG (Lillicrap et al., 2015) and SAC (Haarnoja et al., 2018), etc, in the policy optimization step (line 7 and line 10).

The advantage of CRPO over the primal-dual approach can be readily seen from its design. CRPO features **immediate** switches between optimizing the objective and reducing the constraints whenever they are violated. However, the primal-dual approach can respond much slower because the control is based on dual variables. If a dual variable is nonzero, then the policy update will descend along the corresponding constraint function. As a result, even if a constraint is already satisfied, there can still be a delay (sometimes a significant delay) for the dual variable to iteratively reduce to zero to release the constraint, which yields unnecessary sampling cost and slows down the algorithm. Our experiments in Section 5 validates such a performance advantage of CRPO over the primal-dual approach.

From the implementation perspective, CRPO can be implemented as easy as unconstrained policy optimization such as *unconstrained* policy gradient algorithms, whereas the primal-dual approach typically requires the *projected* gradient descent to update the dual variables, which is more complex to implement. Further, without introduction of the dual variables, CRPO does not suffer from hyperparameter tuning of the learning rates and projection threshold of the dual variables, whereas the primal-dual approach can be very sensitive to these hyperparamters. Nor does CRPO require initialization to be feasible, whereas the primal-dual approach can suffer significantly from bad initialization. We also empirically verify that the performance of CRPO is robust to the value of η over a wide range, which does not cause additional tuning effort compared to unconstrained algorithms. More discussions can be referred to Section 5.

CRPO algorithm is inspired by, yet very different from the cooperative stochastic approximation (CSA) method (Lan & Zhou, 2016) in optimization literature. First, CSA is designed for convex optimization subject to convex constraint, and is not readily capable of handling the more challenging SRL problems eq. (2), which are nonconvex optimization subject to nonconvex constraints. Second, CSA is designed to handle only a single constraint, whereas CRPO can handle multiple constraints with guaranteed constraint satisfaction and global optimality. Thus, the finite-time analysis for CSA and CRPO feature different approaches due to the aforementioned differences in their designs.

4. Convergence Analysis of CRPO

In this section, we take NPG as a representative optimizer in CRPO, and establish the global convergence rate of CRPO in both the tabular and function approximation settings. Note that TRPO and ACKTR update can be viewed as the NPG approach with adaptive stepsize. Thus, the convergence we establish for NPG implies similar results for CRPO that takes TRPO or ACKTR as the optimizer.

4.1. Tabular Setting

In the tabular setting, we consider the softmax parameterization. For any $w \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$, the corresponding softmax policy π_w is defined as

$$\pi_w(a|s) := \frac{\exp(w(s,a))}{\sum_{a'\in\mathcal{A}}\exp(w(s,a'))}, \quad \forall (s,a)\in\mathcal{S}\times\mathcal{A}. \tag{3}$$

Clearly, the policy class defined in eq. (3) is complete, as any stochastic policy in the tabular setting can be represented in this class.

Policy Evaluation: To perform the policy evaluation in Algorithm 1 (line 3), we adopt the temporal difference (TD) learning, in which a vector $\theta^i \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$ is used to estimate the state-action value function $Q^i_{\pi_w}$ for all $i = 0, \cdots, p$. Specifically, each iteration of TD learning takes the form of

$$\begin{aligned}\theta^i_{k+1}(s,a) &= \theta^i_k(s,a)\\ &\quad + \beta_k[c_i(s,a,s') + \gamma\theta^i_k(s',a') - \theta^i_k(s,a)],\end{aligned} \tag{4}$$

where $s \sim \mu_{\pi_w}$, $a \sim \pi_w(\cdot|s)$, $s' \sim \mathsf{P}(\cdot|s,a)$, $a' \sim \pi_w(\cdot|s')$, and β_k is the learning rate. In line 3 of Algorithm 1, we perform the TD update in eq. (4) for K_{in} iterations. It has been shown in (Sutton, 1988; Bhandari et al., 2018; Dalal et al., 2018b) that the iteration in eq. (4) of TD learning converges to a fixed point $\theta^i_*(\pi_w) \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$, where each component of the fixed point is the corresponding state-action value: $\theta^i_*(\pi_w)(s,a) = Q^i_{\pi_w}(s,a)$. After performing K_{in} iterations of TD learning as eq. (4), we let $\bar{Q}^i_t(s,a) = \theta^i_{K_{\text{in}}}(s,a)$ for all $(s,a)\in\mathcal{S}\times\mathcal{A}$ and all $i = \{0,\cdots,p\}$.

Constraint Estimation: In the tabular setting, we let the sample set $\mathcal{B}_t$ include all state-action pairs, i.e., $\mathcal{B}_t = \mathcal{S}\times\mathcal{A}$, and the weight factor be $\rho_{j,t} = \xi(s_j)\pi_{w_t}(a_j|s_j)$ for all $t = 0,\cdots,T-1$. Then, the estimation error of the constraints can be upper bounded as $|\bar{J}_i(\theta^i_t) - J_i(w_t)| = |\mathbb{E}[\bar{Q}^i_t(s,a)] - \mathbb{E}[Q^i_{\pi_{w_t}}(s,a)]| \leq ||\bar{Q}^i(\theta^i_t) - Q^i_{\pi_{w_t}}||^2$. Thus, our approximation of constraints is accurate when the approximated value function $\bar{Q}^i_t(s,a)$ is accurate.

Policy Optimization: In the tabular setting, it can be checked that the natural policy gradient of $J_i(\pi_w)$ is $\Delta_i(w)_{s,a} = (1-\gamma)^{-1}Q^i_{\pi_w}(s,a)$ (see Appendix B). Once we obtain an approximation $\bar{Q}^i_t(s,a) \approx Q^i_{\pi_w}(s,a)$, we can use it to update the policy in the upcoming policy optimization step:

$$\begin{aligned} &w_{t+1} = w_t + \alpha\bar{\Delta}_t, \text{ (line 7)}\\ \text{or}\quad &w_{t+1} = w_t - \alpha\bar{\Delta}_t \text{ (line 10)},\end{aligned} \tag{5}$$

where $\alpha > 0$ is the stepsize and $\bar{\Delta}_t(s,a) = (1-\gamma)^{-1}\bar{Q}^0_t(s,a)$ (line 7) or $(1-\gamma)^{-1}\bar{Q}^{i_t}_t(s,a)$ (line 10).

Our **main technical challenge** lies in the analysis of policy optimization, which runs as a stochastic approximation (SA) process with **random and dynamical switches** between optimization objectives of the reward and cost targets. Moreover, since critics estimate the constraints and help actor to estimate the policy update, the interaction error between actor and critics affects how the algorithm switches between objective and constraints. The typical analysis technique for NPG (Agarwal et al., 2019) is not applicable here, because NPG has a fixed objective to optimize, and its analysis technique does not capture the overall convergence performance of an SA with dynamically switching optimization objective. Furthermore, the updates with respect to the constraint functions involve the stochastic selection of a constraint if multiple constraints are violated, which further complicates the random events to analyze. To handle these issues, we develop a **novel analysis approach**, in which we focus on the event in which critic returns almost accurate value function estimation. Such an event greatly facilitates us to capture how CRPO switches between objective and multiple constraints and establish the convergence rate.

The following theorem characterizes the convergence rate of CRPO in terms of the objective function and constraint error bound.

Theorem 1. *Consider Algorithm 1 in the tabular setting with softmax policy parameterization defined in eq. (3) and any initialization $w_0 \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$. Suppose the policy evaluation update in eq. (4) takes $K_{in} = \Theta(T^{1/\sigma}(1-\gamma)^{-2/\sigma}\log^{2/\sigma}(T^{1+2/\sigma}/\delta))$ iterations. Let the tolerance $\eta = \Theta(\sqrt{|\mathcal{S}|\,|\mathcal{A}|}/((1-\gamma)^{1.5}\sqrt{T}))$ and perform the NPG update defined in eq. (5) with $\alpha = (1-\gamma)^{1.5}/\sqrt{|\mathcal{S}|\,|\mathcal{A}|\,T}$. Then, with probability at least $1-\delta$, we have*

$$J_0(\pi^*) - \mathbb{E}[J_0(w_{out})] \leq \Theta\left(\frac{\sqrt{|\mathcal{S}|\,|\mathcal{A}|}}{(1-\gamma)^{1.5}\sqrt{T}}\right),$$

$$\mathbb{E}[J_i(w_{out})] - d_i \leq \Theta\left(\frac{\sqrt{|\mathcal{S}|\,|\mathcal{A}|}}{(1-\gamma)^{1.5}\sqrt{T}}\right)$$

for all $i = \{1,\cdots,p\}$, where the expectation is taken with respect to selecting w_{out} from $\mathcal{N}_0$.

As shown in Theorem 1, starting from an arbitrary initialization, CRPO algorithm is guaranteed to converge to the globally optimal policy π^* in the feasible set Ω_C at a sublinear rate $\mathcal{O}(1/\sqrt{T})$, and the constraint violation of the output policy also converges to zero also at a sublinear rate $\mathcal{O}(1/\sqrt{T})$. Thus, to attain a w_{out} that satisfies $J_0(\pi^*) - \mathbb{E}[J_0(w_{\text{out}})] \leq \epsilon$ and $\mathbb{E}[J_i(w_{\text{out}})] - d_i \leq \epsilon$ for all $1 \leq i \leq p$, CRPO needs at most $T = \mathcal{O}(\epsilon^{-2})$ iterations, with each policy evaluation step consists of approximately

$K_{\text{in}} = \mathcal{O}(T)$ iterations when σ is close to 1. Theorem 1 is the first global convergence for a primal-type algorithm even under the nonconcave objective with nonconcave constraints.

Outline of Proof Idea. We briefly explain the idea of the proof of Theorem 1, and the detailed proof can be referred to Appendix B. The key challenge here is to analyze an SA process that randomly and dynamically switches between the target objectives of the reward and the constraint. To this end, we construct novel concentration events for capturing the impact of such a dynamic process on the update of the reward and cost functions in order to establish the high probability convergence guarantee.

More specifically, we focus on the event in which all policy evaluation step returns an estimation with high accuracy. Then we show that under the parameter setting specified in Theorem 1, either the size of the approximated feasible policy set $\mathcal{N}_0$ is large, or the average policies in the set $\mathcal{N}_0$ is at least as good as π^*. In the first case we have enough candidate policies in the set $\mathcal{N}_0$, which guarantees the convergence of CRPO within the set $\mathcal{N}_0$. In the second case we can directly conclude that $J(w_{\text{out}}) \geq J(\pi^*)$. To establish the convergence rate of the constraint violation, note that w_{out} is selected from the set $\mathcal{N}_0$, and thus the violation cost is not worse than the summation of constraint estimation error and the tolerance. $\square$

4.2. Function Approximation Setting

In the function approximation setting, we parameterize the policy by a two-layer neural network together with the softmax policy. We assign a feature vector $\psi(s,a) \in \mathbb{R}^d$ with $d \geq 2$ for each state-action pair (s,a). Without loss of generality, we assume that $\|\psi(s,a)\|_2 \leq 1$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$. A two-layer neural network $f((s,a);W,b)$ with input $\psi(s,a)$ and width m takes the form of

$$f((s,a);W,b) = \frac{1}{\sqrt{m}} \sum_{r=1}^{m} b_r \cdot \text{ReLU}(W_r^\top \psi(s,a)), \quad (6)$$

for any $(s,a) \in \mathcal{S} \times \mathcal{A}$, where $\text{ReLU}(x) = \mathbb{1}(x > 0) \cdot x$, $b = [b_1, \cdots, b_m]^\top \in \mathbb{R}^m$, and $W = [W_1^\top, \cdots, W_m^\top]^\top \in \mathbb{R}^{md}$ are the parameters. When training the two-layer neural network, we initialize the parameter via $[W_0]_r \sim D_w$ and $b_r \sim \text{Unif}[-1,1]$ independently, where D_w is a distribution that satisfies $d_1 \leq \|[W_0]_r\|_2 \leq d_2$ (where d_1 and d_2 are positive constants), for all $[W_0]_r$ in the support of D_w. During training, we only update W and keep b fixed, which is widely adopted in the convergence analysis of neural networks (Cai et al., 2019; Du et al., 2018). For notational simplicity, we write $f((s,a);W,b)$ as $f((s,a);W)$ in the sequel. Using the neural network in eq. (6), we define the softmax policy

$$\pi_W^\tau(a|s) := \frac{\exp(\tau \cdot f((s,a);W))}{\sum_{a'\mathcal{A}} \exp(\tau \cdot f((s,a');W))}, \quad (7)$$

for all $(s,a) \in \mathcal{S} \times \mathcal{A}$, where τ is the temperature parameter, and it can be verified that $\pi_W^\tau(a|s) = \pi_{\tau W}(a|s)$. We define the feature mapping $\phi_W(s,a) = [\phi_W^1(s,a)^\top, \cdots, \phi_W^m(s,a)^\top]^\top \colon \mathbb{R}^d \to \mathbb{R}^{md}$ as

$$\phi_W^r(s,a)^\top = \frac{b_r}{\sqrt{m}} \mathbb{1}(W_r^\top \psi(s,a) > 0) \cdot \psi(s,a),$$

for all $(s,a) \in \mathcal{S} \times \mathcal{A}$ and for all $r \in \{1, \cdots, m\}$.

Policy Evaluation: To estimate the state-action value function in Algorithm 1 (line 3), we adopt another neural network $f((s,a);\theta^i)$ as an approximator, where $f((s,a);\theta^i)$ has the same structure as $f((s,a);W)$, with W replaced by $\theta \in \mathbb{R}^{md}$ in eq. (7). To perform the policy evaluation step, we adopt the TD learning with neural network parametrization, which has also been used for the policy evaluation step in (Cai et al., 2019; Wang et al., 2019; Zhang et al., 2020). Specifically, we choose the same initialization as the policy neural work, i.e., $\theta_0^i = W_0$, and perform the TD iteration as

$$\begin{aligned}\theta_{k+1/2}^i =& \theta_k^i + \beta(c_i(s,a,s') + \gamma f((s',a');\theta_k^i) \\ & - f((s,a);\theta_k^i)) \nabla_\theta f((s,a);\theta_k^i), \quad (8)\end{aligned}$$

$$\theta_{k+1}^i = \arg\min_{\theta \in \boldsymbol{B}} \left\| \theta - \theta_{k+1/2}^i \right\|_2, \quad (9)$$

where $s \sim \mu_{\pi_W}$, $a \sim \pi_W(\cdot|s)$, $s' \sim \mathsf{P}(\cdot|s,a)$, $a' \sim \pi_W(\cdot|s')$, β is the learning rate, and $\boldsymbol{B}$ is a compact space defined as $\boldsymbol{B} = \{\theta \in \mathbb{R}^{md} : \left\|\theta - \theta_0^i\right\|_2 \leq R\}$. For simplicity, we denote the state-action pair as $x = (s,a)$ and $x' = (s',a')$ in the sequel. We define the temporal difference error as $\delta_k(x,x',\theta_k^i) = f(x_k',\theta_k^i) - \gamma f(x_k,\theta_k^i) - c_i(x_k,x_k')$, stochastic semi-gradient as $g_k(\theta_k^i) = \delta_k(x_k,x_k'.\theta_k^i)\nabla_\theta f(x_k,\theta_k^i)$, and full semi-gradient as $\bar{g}_k(\theta_k^i) = \mathbb{E}_{\mu_{\pi_W}}[\delta_k(x,x',\theta_k^i)\nabla_\theta f(x,\theta_k^i)]$. We then describe the following regularity conditions on the stationary distribution μ_{π_W}, state-action value function $Q_{\pi_W}^i$, and variance, which have been adopted widely in the analysis of TD learning with function approximation and stochastic approximation (SA) (Cai et al., 2019; Wang et al., 2019; Zhang et al., 2020; Fu et al., 2020).

Assumption 1. *There exists a constant $C_0 > 0$ such that for any $\tau \geq 0$, $x \in \mathbb{R}^d$ with $\|x\|_2 = 1$ and π_W, it holds that $\boldsymbol{P}\left(\left|x^\top \psi(s,a)\right| \leq \tau\right) \leq C_0 \cdot \tau$, where $(s,a) \sim \mu_{\pi_W}$.*

Assumption 2. *We define the following function class:*

$$\begin{aligned}\mathcal{F}_{R,\infty} = \Big\{& f((s,a);\theta) = f((s,a);\theta_0) \\ & + \int \mathbb{1}(\theta^\top \psi(s,a) > 0) \cdot \lambda(\theta)^\top \psi(s,a) dp(\theta) \Big\}\end{aligned}$$

where $f((s,a);\theta_0)$ is the two-layer neural network corresponding to the initial parameter $\theta_0 = W_0$, $\lambda(\theta):\mathbb{R}^d \rightarrow \mathbb{R}^d$ is a weighted function satisfying $\|\lambda(w)\|_\infty \leq R/\sqrt{d}$, and $p(\cdot):\mathbb{R}^d \rightarrow \mathbb{R}$ is the density D_w. We assume that $Q^i_{\pi_W} \in \mathcal{F}_{R,\infty}$ for all π_W and $i=\{0,\cdots,p\}$.

Assumption 3. *For any parameterized policy π_W, there exists a constant $C_\zeta > 0$ such that for all $k \geq 0$, $\mathbb{E}_{\mu_{\pi_W}}\left[\exp\left(\left\|\bar{g}_k(\theta^i_k) - g_k(\theta^i_k)\right\|_2^2/C_\zeta^2\right)\right] \leq 1$.*

Assumption 1 implies that the distribution of $\psi(s,a)$ has a uniformly upper bounded probability density over the unit sphere, which can be satisfied for most of the ergodic Markov chain. Assumption 2 is a mild regularity condition on $Q^i_{\pi_W}$, as $\mathcal{F}_{R,\infty}$ is a function class of neural networks with infinite width, which captures a sufficiently general family of functions. Assumption 3 on the variance bound is standard, which has been widely adopted in stochastic optimization literature (Ghadimi & Lan, 2013; Nemirovski et al., 2009; Lan, 2012; Ghadimi & Lan, 2016).

In the following lemma, we characterize the convergence rate of neural TD *in high probability*, which is needed for our the analysis. Such a result is stronger than the convergence *in expectation* provided in (Bhandari et al., 2018; Cai et al., 2019; Wang et al., 2019; Zhang et al., 2020; Srikant & Ying, 2019), which is not sufficient for our need later on.

Lemma 1 (Convergence rate of TD in high probability)**.** *Consider the TD iteration with neural network approximation defined in eq. (8). Let $\bar{\theta}_K = \frac{1}{K}\sum_{k=0}^{K-1}\theta_k$ be the average of the output from $k=0$ to $K-1$. Let $\bar{Q}^i_t(s,a) = f((s,a),\theta^i_{K_{in}})$ be an estimator of $Q^i_{\pi_{\tau_t W_t}}(s,a)$. Suppose Assumptions 1-3 hold, assume that the stationary distribution μ_{π_W} is not degenerate for all $W \in \boldsymbol{B}$, and let the stepsize $\beta = \min\{1/\sqrt{K}, (1-\gamma)/12\}$. Then, with probability at least $1-\delta$, we have*

$$\left\|\bar{Q}^i_t(s,a) - Q^i_{\pi_{\tau_t W_t}}(s,a)\right\|^2_{\mu_\pi} \leq \Theta\Big(\frac{1}{(1-\gamma)^2\sqrt{K}}\sqrt{\log\left(\frac{1}{\delta}\right)}\Big) + \Theta\Big(\frac{1}{(1-\gamma)^3 m^{1/4}}\sqrt{\log\left(\frac{K}{\delta}\right)}\Big).$$

Lemma 1 implies that after performing the neural TD learning in eq. (8)-eq. (9) for $\Theta(\sqrt{m})$ iterations, we can obtain an approximation $\bar{Q}^i_t$ such that $||\bar{Q}^i_t - Q^i_{\pi_{\tau_t W_t}}||_{\mu_\pi} = \mathcal{O}(1/m^{1/8})$ with high probability.

Constraint Estimation: Since the state space is usually very large or even infinite in the function approximation setting, we cannot include all state-action pairs to estimate the constraints as for the tabular setting. Instead, we sample a batch of state-action pairs $(s_j,a_j) \in \mathcal{B}_t$ from the distribution $\xi(\cdot)\pi_{W_t}(\cdot|\cdot)$, and let the weight factor $\rho_j = 1/|\mathcal{B}_t|$ for all j. In this case, the estimation error of the constrains $\left|\bar{J}_i(\theta^i_t) - J_i(w_t)\right|$ is small when the policy evaluation $\bar{Q}^i_t$ is accurate and the batch size $|\mathcal{B}_t|$ is large. We assume the following concentration property for the sampling process in the constraint estimation step. Similar assumptions have also been taken in (Ghadimi & Lan, 2013; Nemirovski et al., 2009; Lan, 2012; Ghadimi & Lan, 2016).

Assumption 4. *For any parameterized policy π_W, there exists a constant $C_f > 0$ such that for all $k \geq 0$, $\mathbb{E}_{\xi\cdot\pi_W}\left[\exp([\bar{Q}^i_t(s,a) - \mathbb{E}_{\xi\cdot\pi_{\tau_t W_t}}[\bar{Q}^i_t(s,a)]^2/C_f^2)\right] \leq 1$.*

Policy Optimization: In the neural softmax approximation setting, at each iteration t, an approximation of the natural policy gradient can be obtained by solving the following linear regression problem (Agarwal et al., 2019; Wang et al., 2019; Xu et al., 2019b):

$$\begin{aligned}\Delta_i(W_t) &\approx \bar{\Delta}_t \\ &= \arg\min_{\theta\in\boldsymbol{B}} \mathbb{E}_{\nu_{\pi_{\tau_t W_t}}}[(\bar{Q}^i_t(s,a) - \phi_{W_t}(s,a)^\top\theta)^2]. \end{aligned} \quad (10)$$

Given the approximated natural policy gradient $\bar{\Delta}_t$, the policy update takes the form of

$$\begin{aligned} &\tau_{t+1} = \tau_t + \alpha,\ \ \tau_{t+1}\cdot w_{t+1} = \tau_t\cdot w_t + \alpha\bar{\Delta}_t \text{ (line 7)} \\ \text{or}\quad &\tau_{t+1}\cdot w_{t+1} = \tau_t\cdot w_t - \alpha\bar{\Delta}_t \text{ (line 10)}. \end{aligned} \quad (11)$$

Note that in eq. (11) we also update the temperature parameter by $\tau_{t+1} = \tau_t + \alpha$ simultaneously, which ensures $w_t \in \boldsymbol{B}$ for all t. The following theorem characterizes the convergence rate of Algorithm 1 in terms of both the objective function and the constraint violation.

Theorem 2. *Consider Algorithm 1 in the function approximation setting with neural softmax policy parameterization defined in eq. (7). Suppose Assumptions 1-4 hold. Suppose the same setting of policy evaluation step stated in Lemma 1 holds, and consider performing the neural TD in eq. (8) and eq. (9) with $K_{in} = \Theta((1-\gamma)^2\sqrt{m})$ at each iteration. Let the tolerance $\eta = \Theta(m(1-\gamma)^{-1}/\sqrt{T} + (1-\gamma)^{-2.5}m^{-1/8})$ and perform the NPG update defined in eq. (11) with $\alpha = \Theta(1/\sqrt{T})$. Then with probability at least $1-\delta$, we have*

$$J_0(\pi^*) - \mathbb{E}[J_0(\pi_{\tau_{out}W_{out}})] \leq \Theta\left(\frac{1}{(1-\gamma)\sqrt{T}}\right) + \Theta\left(\frac{1}{(1-\gamma)^{2.5}m^{1/8}}\log^{\frac{1}{4}}\left(\frac{(1-\gamma)^2T\sqrt{m}}{\delta}\right)\right),$$

and for all $i=1,\cdots,p$, we have

$$\mathbb{E}[J_i(\pi_{\tau_{out}W_{out}})] - d_i \leq \Theta\left(\frac{1}{(1-\gamma)\sqrt{T}}\right) + \Theta\left(\frac{1}{(1-\gamma)^{2.5}m^{1/8}}\log^{\frac{1}{4}}\left(\frac{(1-\gamma)^2T\sqrt{m}}{\delta}\right)\right).$$

where the expectation is taken only with respect to the randomness of selecting W_{out} from $\mathcal{N}_0$.

Theorem 2 guarantees that CRPO converges to the global optimal policy π^* in the feasible set at a sublinear rate $\mathcal{O}(1/\sqrt{T})$ with a approximation error $\mathcal{O}(m^{-1/8})$ vanishes as the network width m increases. The constraint violation bound also converges to zero at a sublinear rate $\mathcal{O}(1/\sqrt{T})$ with a vanishing error $\mathcal{O}(m^{-1/8})$ decreases as m increase. The approximation error arises from both the policy evaluation and policy optimization due to the limited expressive power of neural networks.

To compare with the primal-dual approach in the function approximation setting, Theorem 2 shows that while the value function gap of CRPO achieves the same convergence rate as the primal-dual approach, the constraint violation of CRPO decays at a convergence rate of $\mathcal{O}(1/\sqrt{T})$, which substantially outperforms the rate $\mathcal{O}(1/T^{\frac{1}{4}})$ of the primal-dual approach (Ding et al., 2020b). Such an advantage of CRPO is further validated by our experiments in Section 5, which show that the constraint violation of CRPO vanishes much faster than that of the primal-dual approach.

Remark 1. *Our convergence analysis for Theorem 2 can still hold without Assumptions 3 and 4. As a result, the convergence rate of CRPO would have polynomial dependence on δ rather than logarithmic dependence.*

Remark 2. *Both Theorems 1 and 2 can be extended to cases with Markovian sampling, where an additional bias error due to Markovian sampling can be bounded using the standard techniques (e.g. (Bhandari et al., 2018; Tagorti & Scherrer, 2015)).*

Remark 3. *The result in Theorem 2 can be extended to scenarios with a continuous action space and with a generally parametrized policy (not necessarily softmax), by leveraging the analysis in (Agarwal et al., 2019) for proving the global convergence of NPG with general function approximation.*

5. Experiments

In this section, we conduct simulation experiments on different SRL tasks to compare our CRPO with the other baseline SRL algorithms: primal-dual optimization (PDO), constrained policy optimization (CPO), and interior point optimization (IPO). We consider two tasks based on OpenAI gym (Brockman et al., 2016) with each having multiple or a single constraints given as follows:

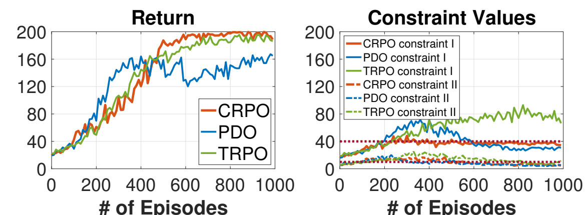
Cartpole: The agent is rewarded for keeping the pole upright, but is penalized with cost if (1) entering into some specific areas, or (2) having the angle of pole being large.

Acrobot: The agent is rewarded for swing the end-effector at a specific height, but is penalized with cost if (1) applying torque on the joint when the first link swings in a prohibited direction, or (2) when the the second link swings in a prohibited direction with respect to the first link. In the single constraint setting, we only consider the fist penalty.

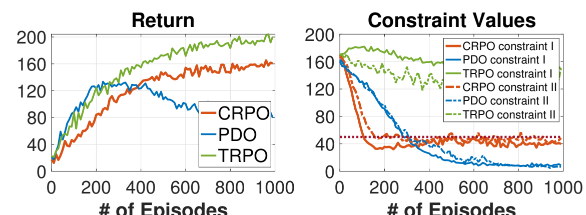
The detailed experimental setting is described in Appendix A. For all experiments, we use neural softmax policy with two hidden layers of size $(128, 128)$. We adopt TRPO as the optimizer for CRPO, PDO and CPO, and PPO as the optimizer for IPO, which is the approach taken in the original IPO algorithm in (Liu et al., 2019b). It remains unclear how to develop an IPO approach based on TRPO. In CRPO, we let the tolerance $\eta = 0.5$. In PDO, we initialize the Lagrange multiplier as zero, and select the best tuned stepsize for dual variable update. In CPO, we select the best tuned size of the line search region for both the reward and cost optimization. In IPO, the regularization factor of the barrier function is set to be 20 as suggested in (Liu et al., 2019b).

5.1. Comparison with PDO

The learning curves for CRPO and PDO are provided in Figure 1. At each step we evaluate the performance based on two metrics: the return reward and constraint value of the output policy. We also show the learning curve of unconstrained TRPO (the green line), which, although achieves the best reward, does not satisfy the constraints.



(a) Cartpole



(b) Acrobot

Figure 1. Average performance for CRPO, PDO, and unconstrained TRPO over 10 seeds. The red dot lines in (a) and (b) represent the limits of the constraints.

In both tasks, CRPO tracks the constraint returns almost exactly to the limit, indicating that CRPO sufficiently explores the boundary of the feasible set, which results in an optimal return reward. In contrast, although PDO also outputs a constraints-satisfying policy in the end, it tends to over- or under-enforce the constraints, which results in lower return reward and unstable constraint satisfaction performance. In terms of the convergence, the constraints of CRPO drop below the thresholds (and thus satisfy the constraints) much faster than that of PDO, corroborating our theoretical comparison that the constraint violation of CRPO (given in Theorem 2) converges much faster than that of

PDO given in (Ding et al., 2020b).

We also find that the performance of CRPO is robust to the value of η over a wide range, whereas the convergence performance of PDO is very sensitive to the stepsize of the dual variable (see additional experiments of hyperparameters comparison in Appendix A). Thus, in contrast to the difficulty of tuning PDO, CRPO is much less sensitive to hyper-parameters and is hence much easier to tune.

5.2. Comparison with CPO

Since it is very difficult for CPO to solve multi-constraint tasks as discussed in (Liu et al., 2019b), in order to compare the performance between CRPO and CPO, we focus on the 'Acrobot' task with a single constraint. We also add the learning curve of IPO in the plot for comparison. Figure 2 illustrates that CRPO converges faster and achieves higher reward than CPO (and IPO), although all algorithms share similar convergence behavior over the constraint values.

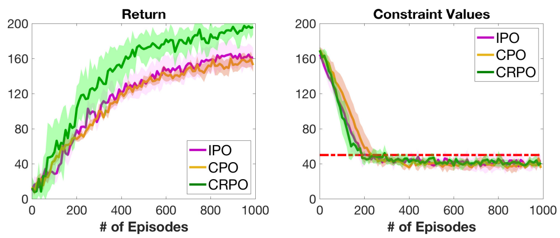


Figure 2. Average performance of CRPO, CPO and IPO in 'Acrobot' with one constraint over 10 seeds.

5.3. Comparison with IPO

We compare the performance between CRPO and IPO over the same setting of 'Acrobot' with two constraints. As discussed in (Liu et al., 2019b), IPO relies on the barrier regularization function to enforce the satisfaction of the constraints, and hence IPO is guaranteed to converge only to a suboptimal point. Such a regularization can also slow down the convergence speed of the constraint value. As shown in Figure 3, our CRPO outperforms IPO in terms of the convergence of both the reward and constraint values in a multi-constraint setting.

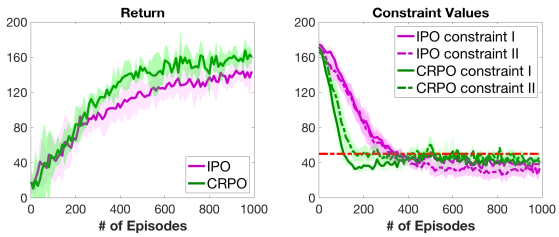


Figure 3. Average performance of CRPO and IPO in 'Acrobot' with two constraints over 10 seeds.

6. Conclusion

In this paper, we propose a novel CRPO approach for policy optimization for SRL, which is easy to implement and has provable global optimality guarantee. We show that CRPO achieves an $\mathcal{O}(1/\sqrt{T})$ convergence rate to the global optimum and an $\mathcal{O}(1/\sqrt{T})$ rate of vanishing constraint error when NPG update is adopted as the optimizer. This is the first primal SRL algorithm that has a provable convergence guarantee to a global optimum. In the future, it is interesting to incorporate various momentum schemes to CRPO to improve its convergence performance.

Acknowledgements

The work of T. Xu and Y. Liang was supported in part by the U.S. National Science Foundation under the grants CCF-1909291 and CCF-1900145. The work of G. Lan was supported by the U.S. National Science Foundation under the grant CCF-1909298.

References

Achiam, J., Held, D., Tamar, A., and Abbeel, P. Constrained policy optimization. In *Proc. International Conference on Machine Learning (ICML)*, pp. 22–31, 2017.

Agarwal, A., Kakade, S. M., Lee, J. D., and Mahajan, G. Optimality and approximation with policy gradient methods in Markov decision processes. *arXiv preprint arXiv:1908.00261*, 2019.

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

Bhandari, J. and Russo, D. Global optimality guarantees for policy gradient methods. *arXiv preprint arXiv:1906.01786*, 2019.

Bhandari, J. and Russo, D. A note on the linear convergence of policy gradient methods. *arXiv preprint arXiv:2007.11120*, 2020.

Bhandari, J., Russo, D., and Singal, R. A finite time analysis of temporal difference learning with linear function approximation. In *Proc. Conference on Learning Theory (COLT)*, pp. 1691–1692, 2018.

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

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

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. OpenAI Gym, 2016.

Cai, Q., Yang, Z., Lee, J. D., and Wang, Z. Neural temporal-difference and q-learning provably converge to global optima. *arXiv preprint arXiv:1905.10027*, 2019.

Cen, S., Cheng, C., Chen, Y., Wei, Y., and Chi, Y. Fast global convergence of natural policy gradient methods with entropy regularization. *arXiv preprint arXiv:2007.06558*, 2020.

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

Chow, Y., Nachum, O., Duenez-Guzman, E., and Ghavamzadeh, M. A Lyapunov-based approach to safe reinforcement learning. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pp. 8092–8101, 2018.

Chow, Y., Nachum, O., Faust, A., Duenez-Guzman, E., and Ghavamzadeh, M. Lyapunov-based safe policy optimization for continuous control. *arXiv preprint arXiv:1901.10031*, 2019.

Dalal, G., Dvijotham, K., Vecerik, M., Hester, T., Paduraru, C., and Tassa, Y. Safe exploration in continuous action spaces. *arXiv preprint arXiv:1801.08757*, 2018a.

Dalal, G., Szörényi, B., Thoppe, G., and Mannor, S. Finite sample analyses for TD (0) with function approximation. In *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, 2018b.

Dalal, G., Szorenyi, B., and Thoppe, G. A tale of two-timescale reinforcement learning with the tightest finite-time bound. *arXiv preprint arXiv:1911.09157*, 2019.

DeepMind, G. A. Mastering the real-time strategy game starcraft ii. 2019.

Ding, D., Wei, X., Yang, Z., Wang, Z., and Jovanović, M. R. Provably efficient safe exploration via primal-dual policy optimization. *arXiv preprint arXiv:2003.00534*, 2020a.

Ding, D., Zhang, K., Basar, T., and Jovanovic, M. Natural policy gradient primal-dual method for constrained Markov decision processes. *In Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 33, 2020b.

Du, S. S., Zhai, X., Poczos, B., and Singh, A. Gradient descent provably optimizes over-parameterized neural networks. In *Proc. International Conference on Learning Representations (ICLR)*, 2018.

Fazel, M., Ge, R., Kakade, S. M., and Mesbahi, M. Global convergence of policy gradient methods for the linear quadratic regulator. *arXiv preprint arXiv:1801.05039*, 2018.

Fisac, J. F., Akametalu, A. K., Zeilinger, M. N., Kaynama, S., Gillula, J., and Tomlin, C. J. A general safety framework for learning-based control in uncertain robotic systems. *IEEE Transactions on Automatic Control*, 64(7): 2737–2752, 2018.

Fu, Z., Yang, Z., and Wang, Z. Single-timescale actor-critic provably finds globally optimal policy. *arXiv preprint arXiv:2008.00483*, 2020.

Ghadimi, S. and Lan, G. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

Ghadimi, S. and Lan, G. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Mathematical Programming*, 156(1-2):59–99, 2016.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proc. International Conference on Machine Learning (ICML)*, pp. 1861–1870, 2018.

Julian, D., Chiang, M., O'Neill, D., and Boyd, S. Qos and fairness constrained convex optimization of resource allocation for wireless cellular and ad hoc networks. In *Proc. Conference of the IEEE Computer and Communications Societies*, volume 2, pp. 477–486. IEEE, 2002.

Kakade, S. and Langford, J. Approximately optimal approximate reinforcement learning. In *Proc. International Conference on Machine Learning (ICML)*, volume 2, pp. 267–274, 2002.

Kakade, S. M. A natural policy gradient. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1531–1538, 2002.

Kumar, H., Koppel, A., and Ribeiro, A. On the sample complexity of actor-critic method for reinforcement learning with function approximation. *arXiv preprint arXiv:1910.08412*, 2019.

Lan, G. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1):365–397, 2012.

Lan, G. and Zhou, Z. Algorithms for stochastic optimization with functional or expectation constraints. *arXiv preprint arXiv:1604.03887*, 2016.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

Liang, Q., Que, F., and Modiano, E. Accelerated primal-dual policy optimization for safe reinforcement learning. *arXiv preprint arXiv:1802.06480*, 2018.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Liu, B., Cai, Q., Yang, Z., and Wang, Z. Neural proximal/trust region policy optimization attains globally optimal policy. In *Proc. Advances in Neural Information Processing Systems (NeuIPS)*, 2019a.

Liu, Y., Ding, J., and Liu, X. Ipo: interior-point policy optimization under constraints. *arXiv preprint arXiv:1910.09615*, 2019b.

Malik, D., Pananjady, A., Bhatia, K., Khamaru, K., Bartlett, P. L., and Wainwright, M. J. Derivative-free methods for policy optimization: guarantees for linear quadratic systems. *arXiv preprint arXiv:1812.08305*, 2018.

Nemirovski, A., Juditsky, A., Lan, G., and Shapiro, A. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009.

Papini, M., Pirotta, M., and Restelli, M. Adaptive batch size for safe policy gradients. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pp. 3591–3600, 2017.

Papini, M., Binaghi, D., Canonaco, G., Pirotta, M., and Restelli, M. Stochastic variance-reduced policy gradient. In *Proc. International Conference on Machine Learning (ICML)*, pp. 4026–4035, 2018.

Paternain, S., Calvo-Fullana, M., Chamon, L. F., and Ribeiro, A. Safe policies for reinforcement learning via primal-dual methods. *arXiv preprint arXiv:1911.09101*, 2019a.

Paternain, S., Chamon, L., Calvo-Fullana, M., and Ribeiro, A. Constrained reinforcement learning has zero duality gap. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pp. 7555–7565, 2019b.

Qiu, S., Yang, Z., Ye, J., and Wang, Z. On the finite-time convergence of actor-critic algorithm. In *Proc. Optimization Foundations for Reinforcement Learning Workshop at Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Rahimi, A. and Recht, B. Weighted sums of random kitchen sinks: replacing minimization with randomization in learning. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1313–1320, 2009.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *Proc. International Conference on Machine Learning (ICML)*, pp. 1889–1897, 2015.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Shani, L., Efroni, Y., and Mannor, S. Adaptive trust region policy optimization: Global convergence and faster rates for regularized mdps. *arXiv preprint arXiv:1909.02769*, 2019.

Shen, Z., Ribeiro, A., Hassani, H., Qian, H., and Mi, C. Hessian aided policy gradient. In *Proc. International Conference on Machine Learning (ICML)*, pp. 5729–5738, 2019.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

Srikant, R. and Ying, L. Finite-time error bounds for linear stochastic approximation and TD learning. In *Proc. Conference on Learning Theory (COLT)*, 2019.

Stooke, A., Achiam, J., and Abbeel, P. Responsive safety in reinforcement learning by pid lagrangian methods. In *Proc. International Conference on Machine Learning (ICML)*, 2020.

Sutton, R. S. Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44, 1988.

Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1057–1063, 2000.

Tagorti, M. and Scherrer, B. On the rate of convergence and error bounds for lstd (λ). In *Proc. International Conference on Machine Learning (ICML)*, pp. 1521–1529, 2015.

Tessler, C., Mankowitz, D. J., and Mannor, S. Reward constrained policy optimization. In *Proc. International Conference on Learning Representations (ICLR)*, 2018.

Tu, S. and Recht, B. The gap between model-based and model-free methods on the linear quadratic regulator: an asymptotic viewpoint. *arXiv preprint arXiv:1812.03565*, 2018.

Wang, L., Cai, Q., Yang, Z., and Wang, Z. Neural policy gradient methods: Global optimality and rates of convergence. *arXiv preprint arXiv:1909.01150*, 2019.

Wu, Y., Mansimov, E., Grosse, R. B., Liao, S., and Ba, J. Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pp. 5279–5288, 2017.

Xiong, H., Xu, T., Liang, Y., and Zhang, W. Non-asymptotic convergence of adam-type reinforcement learning algorithms under Markovian sampling. *arXiv preprint arXiv:2002.06286*, 2020.

Xu, P., Gao, F., and Gu, Q. An improved convergence analysis of stochastic variance-reduced policy gradient. In *Proc. International Conference on Uncertainty in Artificial Intelligence (UAI)*, 2019a.

Xu, P., Gao, F., and Gu, Q. Sample efficient policy gradient methods with recursive variance reduction. In *Proc. International Conference on Learning Representations (ICLR)*, 2020a.

Xu, T., Zou, S., and Liang, Y. Two time-scale off-policy TD learning: Non-asymptotic analysis over markovian samples. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pp. 10633–10643, 2019b.

Xu, T., Wang, Z., and Liang, Y. Improving sample complexity bounds for actor-critic algorithms. *arXiv preprint arXiv:2004.12956*, 2020b.

Xu, T., Wang, Z., and Liang, Y. Non-asymptotic convergence analysis of two time-scale (natural) actor-critic algorithms. *arXiv preprint arXiv:2005.03557*, 2020c.

Yang, T.-Y., Rosca, J., Narasimhan, K., and Ramadge, P. J. Projection-based constrained policy optimization. In *Proc. International Conference on Learning Representations (ICLR)*, 2019a.

Yang, Z., Chen, Y., Hong, M., and Wang, Z. Provably global convergence of actor-critic: A case for linear quadratic regulator with ergodic cost. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pp. 8351–8363, 2019b.

Yu, M., Yang, Z., Kolar, M., and Wang, Z. Convergent policy optimization for safe reinforcement learning. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pp. 3127–3139, 2019.

Zhang, K., Koppel, A., Zhu, H., and Başar, T. Global convergence of policy gradient methods to (almost) locally optimal policies. *arXiv preprint arXiv:1906.08383*, 2019.

Zhang, Y., Cai, Q., Yang, Z., and Wang, Z. Generative adversarial imitation learning with neural networks: global optimality and convergence rate. *arXiv preprint arXiv:2003.03709*, 2020.

Zheng, G., Zhang, F., Zheng, Z., Xiang, Y., Yuan, N. J., Xie, X., and Li, Z. Drn: A deep reinforcement learning framework for news recommendation. In *Proc. World Wide Web Conference*, pp. 167–176, 2018.