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 \( O(1/\sqrt{T}) \) convergence rate to the global optimal policy in the constrained policy set and an \( O(1/\sqrt{T}) \) 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 alternately 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...
A New Algorithm: We propose a novel primal approach for Safe Reinforcement Learning with Convergence Guarantee.

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 hyperparameter 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 non-zero, 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 $O(1/\sqrt{T})$. Furthermore, the constraint violation also converges to zero at a rate of $O(1/\sqrt{T})$. 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 $O(1/\sqrt{T})$, which is much faster than the rate $O(1/T^{3/4})$ 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 justi-
A discounted Markov decision process (MDP) is a tuple $(\mathcal{S}, \mathcal{A}, c_0, \mathbb{P}, \xi, \gamma)$, 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; $\mathbb{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0,1]$ is the transition kernel, with $\mathbb{P}(s'|s,a)$ 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 $\mathbb{P}(s'|s,a) = \sum_a \mathbb{P}(s'|s,a)\pi(a|s)$ is ergodic, we denote $\mu_\pi$ as the stationary distribution of this MDP, i.e. $\int_{\mathcal{S}} \mathbb{P}(s'|s,a)\mu_\pi(ds) = \mu_\pi(s')$. Moreover, we define the visitation measure induced by the policy $\pi$ as $\nu_\pi(s,a) = (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \mathbb{P}(s_t = s, a_t = a)$.

For a given policy $\pi$, 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 $J_0(\pi) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t c_i(s_t, a_t, s_{t+1})] = \mathbb{E}_\pi[V_\pi^0(s)] = \mathbb{E}_\pi[Q_\pi^0(s,a) - V_\pi^0(s)]$.

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

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

where $d_i$ is a fixed limit for the $i$-th constraint. We denote the set of feasible policies as $\Omega_C = \{\pi : \forall i, J_i(\pi) \leq d_i\}$, 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 $V_i^\pi$, state-action value function $Q_i^\pi$, and advantage function $A_i^\pi$ analogously to $V_\pi^0$, $Q_\pi^0$, and $A_\pi^0$, 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

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

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

CRPO: A New Approach for Safe Reinforcement Learning with Convergence Guarantee
Algorithm 1 Constraint-Rectified Policy Optimization (CRPO)

1: Initialize: initial parameter $w_0$, empty set $N_0$
2: for $t = 0, \ldots, T - 1$ do
3: Policy evaluation under $\pi_{w_t}$: $Q^i_t(s, a) \approx Q^i_{\pi_{w_t}}(s, a)$
4: Sample $(s_j, a_j) \in B_t \sim \xi \cdot \pi_{w_t}$, compute constraint estimation $\hat{J}_{t, w_t} = \sum_{j \in B_t} \rho_j Q^i_t(s_j, a_j)$ for $i = 0, \ldots, p$.
5: if $\hat{J}_{t, w_t} \leq d_i + \eta$ for all $i = 1, \ldots, p$, then
6: Add $w_t$ into set $N_0$
7: Take one-step policy update towards maximize $J_0(w_t)$: $w_t \rightarrow w_{t+1}$
8: else
9: Choose any $i_t \in \{1, \ldots, p\}$ such that $\hat{J}_{t, w_t} > d_i + \eta$
10: Take one-step policy update towards minimize $J_{i_t}(w_t)$: $w_t \rightarrow w_{t+1}$
11: end if
12: end for
13: Output: $w_{\text{opt}}$ uniformly chosen from $N_0$

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 $J_i(\pi_{w_t})$ (for $i = 1, \ldots, p$) (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 $Q^i_{\pi_{w_t}}(s, a) \approx Q^i_{\pi_{w_t}}(s, a)$ for both reward and costs under current policy $\pi_{w_t}$.

Constraint Estimation: After obtaining $Q^i_{\pi_{w_t}}$, the constraint function $J_i(w_t) = E_{\xi \cdot \pi_{w_t}}[Q^i_{\pi_{w_t}}(s, a)]$ can then be approximated via a weighted sum of approximated state-action value function: $\hat{J}_{i, w_t} = \sum_{j \in B_t} \rho_j Q^i_{\pi_{w_t}}(s_j, a_j)$. Note this step does not take additional sampling cost, as the generation of samples $(s_j, a_j) \in B_t$ 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 $i_t \in \{1, \ldots, p\}$ such that the approximated constraint $\hat{J}_{i_t, w_t}$ violates the condition $\hat{J}_{i_t, w_t} \leq d_i + \eta$, where $\eta$ is the tolerance. If so, we take a 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 a 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 hyperparameters. 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 $\eta$ 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.
we obtain an approximation \( \bar{s} \) (Algorithm 1 (line 3), we adopt the temporal difference (TD) iteration step:
\[
J = \text{checked that the natural policy gradient of approximated value function} \ \bar{s}.
\]
and the weight factor be \( \rho \). In the tabular setting, we let the action value:
\[
K = \text{the component of the fixed point is the corresponding state-}
\]
\( \theta \) has been shown in (Sutton, 1988; Bhandari et al., 2018; Dalal perform the TD update in eq. (4) for \( \beta \) is the learning rate. In line 3 of Algorithm 1, we
\[
\pi \in \text{in} \ \{0, \cdots, p\}.
\]
\( J = \text{Policy Evaluation: To perform the policy evaluation in} \ \text{Algorithm 1 (line 3), we adopt the temporal difference (TD)} \text{learning, in which a vector} \ \theta' \in \mathbb{R}^{[S \times |A|]} \text{is used to estimate the state-action value function} \ Q'_{\pi_{w}} \text{for all} \ i = 0, \cdots, p. \text{Specifically, each iteration of TD learning takes the form of}
\]
\[
\theta'_{i+1}(s, a) = \theta'_{i}(s, a) + \beta_{i}c_{i}(s, a, s') + \gamma \theta'_{i}(s', a') - \theta'_{i}(s, a), \quad (4)
\]
where \( s \sim \mu_{\pi_{w}}, a \sim \pi_{w}(\cdot | s), s' \sim P(\cdot | s, a), a' \sim \pi_{w}(\cdot | s'), \) and \( \beta_{i} \) is the learning rate. In line 3 of Algorithm 1, we perform the TD update in eq. (4) for \( K_{\gamma} \) 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 \( \theta'_{i}(\pi_{w}) \in \mathbb{R}^{[S \times |A|]} \) where each component of the fixed point is the corresponding state-action value:
\[
\theta'_{i}(\pi_{w})(s, a) = Q'_{\pi_{w}}(s, a). \text{After performing} \ K_{\gamma} \text{iterations of TD learning as eq. (4), we let}
\]
\( Q_{i}(s, a) = \theta_{K_{\gamma}}(s, a) \) for all \( (s, a) \in \mathcal{S} \times \mathcal{A} \) and all \( i = \{0, \cdots, p\} \).
\( \text{Constraint Estimation: In the tabular setting, we let the sample set} \ \mathcal{B}_{1} \text{include all state-action pairs, i.e.,} \ \mathcal{B}_{1} = \mathcal{S} \times \mathcal{A}, \text{and the weight factor be} \ \rho_{j,t} = \xi(s_{j})\pi_{w}(a_{j}|s_{j}) \text{for all} \ t = 0, \cdots, T - 1. \text{Then, the estimation error of the}
\]
\( J_{0}(\pi) - E[J_{0}(w_{out})] \leq \Theta\left(\frac{\sqrt{|S||A|}}{(1-\gamma)1.5\sqrt{T}}\right), \]
\( E[J_{i}(w_{out})] - d_{i} \leq \Theta\left(\frac{\sqrt{|S||A|}}{(1-\gamma)1.5\sqrt{T}}\right) \]
\( \text{for all} \ i = \{1, \cdots, p\}, \text{where the expectation is taken with respect to selecting} \ w_{out} \text{from} \ \mathcal{N}_{0}. \)
\( \text{As shown in Theorem 1, starting from an arbitrary initialization, CRPO algorithm is guaranteed to converge} \ \text{to the globally optimal policy} \ \pi^{*} \text{in the feasible set} \ \Omega_{C}; \text{at a sublinear rate} \ \mathcal{O}(1/\sqrt{T}), \text{and the constraint violation of the output policy also converges to zero also at a sublinear rate} \ \mathcal{O}(1/\sqrt{T}). \text{Thus, to attain a} \ w_{out} \text{that satisfies}
\]
\( J_{0}(\pi^{*}) - E[J_{0}(w_{out})] \leq \epsilon \text{and} \ E[J_{i}(w_{out})] - d_{i} \leq \epsilon \text{for all} \ 1 \leq i \leq p, \text{CRPO needs at most} \ T = \mathcal{O}(\epsilon^{-2}) \text{iterations, with each policy evaluation step consists of approximately} \)
\[
\text{where} \ \alpha > 0 \text{is the stepsize and} \ \hat{\Delta}_{t}(s, a) = (1 - \gamma)^{-1}Q'_{\pi_{w}}(s, a) \text{line 7) or} (1 - \gamma)^{-1}Q_{\pi_{w}}(s, a) \text{line 10).}
\]
\( \text{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.}
\text{Theorem 1. Consider Algorithm 1 in the tabular setting with softmax policy parameterization defined in eq. (3) and any initialization} \ \pi_{w} \in \mathbb{R}^{[S \times |A|]}. \text{Suppose the policy evaluation update in eq. (4) takes} \ K_{\gamma} = \Theta(T^{1/\sigma}(1 - \gamma)^{-2/\sigma} \log^{2/\sigma}(T^{1+2/\sigma}/\delta)) \text{iterations. Let the tolerance} \ \eta = \Theta(\sqrt{|S||A|}/((1 - \gamma)(1.5\sqrt{T})) \text{and perform the NPG update defined in eq. (5) with} \ \alpha = (1 - \gamma)^{1.5}/\sqrt{|S||A|} \cdot T. \text{Then, with probability at least} \ 1 - \delta, \text{we have}
\]
\( J_{0}(\pi^{*}) - E[J_{0}(w_{out})] \leq \Theta\left(\frac{\sqrt{|S||A|}}{(1-\gamma)1.5\sqrt{T}}\right), \]
\( E[J_{i}(w_{out})] - d_{i} \leq \Theta\left(\frac{\sqrt{|S||A|}}{(1-\gamma)1.5\sqrt{T}}\right) \]
\( \text{for all} \ i = \{1, \cdots, p\}, \text{where the expectation is taken with respect to selecting} \ w_{out} \text{from} \ \mathcal{N}_{0}. \)
CRPO: A New Approach for Safe Reinforcement Learning with Convergence Guarantee

\[ K_{\text{in}} = O(T) \text{ iterations when } \sigma \text{ is close to 1. Theorem 1} \]

\[ \text{is the first global convergence for a primal-type algorithm even under the nonconcave objective with nonconcave constraints.} \]

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 \( \pi^* \). 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 \( w_{\text{out}} \) 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.

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.

\[ f((s,a); W, b) = \frac{1}{\sqrt{m}} \sum_{r=1}^{m} b_r \cdot \text{ReLU}(W_r^T \psi(s,a)), \]

for any \((s,a) \in S \times A\), where \( \text{ReLU}(x) = 1(x > 0) \cdot x \), \( b = [b_1, \ldots, b_m]^T \in \mathbb{R}^m \), and \( W = [W_1^T, \ldots, W_m^T]^T \in \mathbb{R}^{md} \) are the parameters. When training the two-layer neural network, we initialize the parameter via \( W_0 \sim \text{Unif} -1, 1 \) independently, where \( D_w \) is a distribution that satisfies \( d_1 \leq \|W_0\|_2 \leq d_2 \) (where \( d_1 \) and \( d_2 \) are positive constants), for all \( W_0 \) 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

\[ \pi^*_W(a|s) = \frac{\exp(r \cdot f((s,a); W))}{\sum_{a' \in A} \exp(r \cdot f((s,a'); W))}, \]

for all \((s,a) \in S \times A\), where \( r \) is the temperature parameter, and it can be verified that \( \pi^*_W(a|s) = \pi_{\tau W}(a|s) \). We define the feature mapping \( \phi_W(s,a) = [\phi_W^1(s,a)^T, \ldots, \phi_W^m(s,a)^T]^T : \mathbb{R}^d \rightarrow \mathbb{R}^{md} \)

\[ \phi_W^i(s,a)^T = \frac{b_r}{\sqrt{m}} \mathbb{I}(W_r^T \psi(s,a) > 0) \cdot \psi(s,a), \]

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

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 network, i.e., \( \theta_0^i = W_0 \), and perform the TD iteration as

\[ \theta_{k+1/2}^i = \theta_k^i + \beta(c_k(s,a,s') + \gamma f((s',a'); \theta_k^i) - f((s,a);\theta_k^i)) \nabla_{\theta_k^i} f((s,a);\theta_k^i), \]

\[ \theta_{k+1} = \arg \min_{\theta \in B} \left\| \theta - \theta_{k+1/2}^i \right\|_2, \]

where \( s \sim \mu_{\pi_W} \cdot a \sim \pi_W(\cdot|s), \ s' \sim P(\cdot|s,a), \ a' \sim \pi_W(\cdot|s'), \beta \) is the learning rate, and \( B \) is a compact space defined as \( B = \{ \theta \in \mathbb{R}^{md} : \|\theta - \theta_0^i\|_2 \leq R \} \). 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 \( \delta_k(x,x',\theta_k^i) = f(x_k^{i'},\theta_k^i) - \gamma f(x_k,\theta_k^i) - c_k(x_k, x_k') \), stochastic semi-gradient as \( \delta_k(\theta_k^i) = \delta_k(x_k, x_k', \theta_k^i) \nabla_{\theta_k^i} f(x_k, \theta_k^i) \), and full semi-gradient as \( \bar{\delta}_k(\theta_k^i) = \mathbb{E}_{\mu_{\pi_W}} [\delta_k(x, x', \theta_k^i) \nabla_{\theta_k^i} f(x, \theta_k^i) \cdot a] \).

We then describe the following regularity conditions on the stationary distribution \( \mu_{\pi_W} \), state-action value function \( Q_{\pi_W} \), 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 \( \pi_W \), it holds that \( P \left( |\tau^T \psi(s,a)| \leq \tau \right) \leq C_0 \cdot \tau, \) where \( (s,a) \sim \mu_{\pi_W} \).

Assumption 2. We define the following function class:

\[ F_{R,\infty} = \left\{ f((s,a); \theta) = f((s,a); \theta_0) + \int \mathbb{I}(\theta^T \psi(s,a) > 0) \cdot \lambda(\theta)^T \psi(s,a) dp(\theta) \right\}. \]
where \( f((s, a); \theta_0) \) is the two-layer neural network corresponding to the initial parameter \( \theta_0 = W_0, \lambda(\theta) : \mathbb{R}^d \to \mathbb{R}^d \) is a weighted function satisfying \( ||\lambda(w)||_\infty \leq R/\sqrt{d} \), and \( p(\cdot) : \mathbb{R}^d \to \mathbb{R} \) is the density \( D_w \). We assume that \( Q^* \in F_{R, \infty} \) for all \( \pi_w \) and \( i \in \{0, \cdots, p\} \).

**Assumption 3.** For any parameterized policy \( \pi_W \), there exists a constant \( C_\zeta > 0 \) such that for all \( k \geq 0 \),

\[
\mathbb{E}_{\mu_{\pi_w}} \left[ \exp \left( \left| g_i(\theta_w^k) - g_k(\theta_k) \right|^2 / C_\phi^2 \right) \right] \leq 1.
\]

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 \( Q^\pi_w \), as \( F_{R, \infty} \) 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 \( \bar{Q}^1(s, a) = f((s, a), \bar{\theta}_K) \) be an estimator of \( Q^\pi_{\tau_t W_t}(s, a) \).

Suppose Assumptions 1-3 hold, assume that the stationary distribution \( \mu_{\pi_w} \) is not degenerate for all \( W \in B \), and let the stepsize \( \bar{\beta} = \min \{1 / \sqrt{K}, (1 - \gamma)/12\} \). Then, with probability at least \( 1 - \delta \), we have

\[
\left\| Q^1(s, a) - \bar{Q}^1_{\pi_{\tau_t W_t}}(s, a) \right\|_{\mu_{\pi}}^2 \leq \Theta \left( \frac{1}{(1 - \gamma)^2 \sqrt{K} \sqrt{\log \left( \frac{1}{\delta} \right)}} + \Theta \left( \frac{1}{(1 - \gamma)^4 m^{1/8}} \sqrt{\log \left( \frac{K}{\delta} \right)} \right) \right).
\]

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 \( \bar{Q}^1_i \) such that \( \left\| \bar{Q}^1_i - Q^1_{\pi_{\tau_t W_t}} \right\|_{\mu_{\pi}} = \Theta(1/m^{1/8}) \) 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 \((s_j, a_j) \in B_i \) from the distribution \( \xi(\cdot)_{\pi_{\tau_t W_t}}(\cdot) \), and let the weight factor \( \rho_j = 1 / |B_i| \) for all \( j \). In this case, the estimation error of the constrains \( \left| J_i(\theta^i_w) - J_i(w_t) \right| \) is small when the policy evaluation \( \bar{Q}^1_i \) is accurate and the batch size \( |B_i| \) 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 \( \pi_W \), there exists a constant \( C_\delta > 0 \) such that for all \( k \geq 0 \),

\[
\mathbb{E}_{\xi_{\pi_w}} \left[ \exp \left( \left| \bar{Q}^1_i(s, a) - \bar{Q}^1_i(s, a) \right|^2 / C_\delta^2 \right) \right] \leq 1.
\]

**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):

\[
\Delta_t(W_t) \approx \bar{\Delta}_t = \arg \min_{\theta \in B} \mathbb{E}_{\nu_{\pi_{\tau_t W_t}}} \left[ (\bar{Q}^1_i(s, a) - \phi_{W_t}(s, a) \theta)^2 \right]. \tag{10}
\]

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

\[
\tau_{t+1} = \tau_t + \alpha \cdot \tau_{t+1} \cdot w_{t+1} = \tau_t \cdot w_t + \alpha \bar{\Delta}_t \tag{line 7}
\]

or

\[
\tau_{t+1} \cdot w_{t+1} = \tau_t \cdot w_t - \alpha \bar{\Delta}_t \tag{line 10}.
\]

Note that in eq. (11) we also update the temperature parameter \( \tau_{t+1} = \tau_t + \alpha \) simultaneously, which ensures \( w_t \in B \) 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 \( K_m = \Theta((1 - \gamma)^2 \sqrt{m}) \) 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

\[
\begin{aligned}
J_0(\pi^+) - \mathbb{E}[J_0(\pi_{\tau_m W_m})] &\leq \Theta \left( \frac{1}{(1 - \gamma)\sqrt{T}} + \Theta \left( \frac{1}{(1 - \gamma)^{2.5} m^{1/8}} \log \left( \frac{1}{\delta} \right) \right) \right), \\
\text{and for all } i = 1, \cdots, p, \text{ we have} \quad
\mathbb{E}[J_i(\pi_{\tau_m W_m})] - d_i &\leq \Theta \left( \frac{1}{(1 - \gamma)\sqrt{T}} + \Theta \left( \frac{1}{(1 - \gamma)^{2.5} m^{1/8}} \log \left( \frac{1}{\delta} \right) \right) \right),
\end{aligned}
\]

where the expectation is taken only with respect to the randomness of selecting \( W_{\text{out}} \) from \( B_0 \).
Theorem 2 guarantees that CRPO converges to the global optimal policy \( \pi^* \) in the feasible set at a sublinear rate \( O(1/\sqrt{T}) \) with a approximation error \( O(m^{-1/8}) \) vanishes as the network width \( m \) increases. The constraint violation bound also converges to zero at a sublinear rate \( O(1/\sqrt{T}) \) with a vanishing error \( O(m^{-1/8}) \) 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 \( O(1/\sqrt{T}) \), which substantially outperforms the rate \( O(1/T^{1/4}) \) 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 \( \delta \) 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 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.

![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.](image-url)
CRPO: A New Approach for Safe Reinforcement Learning with Convergence Guarantee

We also find that the performance of CRPO is robust to the value of $\eta$ 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.](image)

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.](image)

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 $O(1/\sqrt{T})$ convergence rate to the global optimum and an $O(1/\sqrt{T})$ 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


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


Tu, S. and Recht, B. The gap between model-based and model-free methods on the linear quadratic regulator: an


