Semi-Infinitely Constrained Markov Decision Processes and Provably Efficient Reinforcement Learning

Liangyu Zhang, Yang Peng, Wenhao Yang, and Zhihua Zhang

***Abstract*—We propose a novel generalization of constrained Markov decision processes (CMDPs) that we call the *semi-infinitely constrained Markov decision process* (SICMDP). Particularly, we consider a continuum of constraints instead of a finite number of constraints as in the case of ordinary CMDPs. We also devise two reinforcement learning algorithms for SICMDPs that we refer to as SI-CMBRL and SI-CPO. SI-CMBRL is a model-based reinforcement learning algorithm. Given an estimate of the transition model, we first transform the reinforcement learning problem into a linear semi-infinitely programming (LSIP) problem and then use the dual exchange method in the LSIP literature to solve it. SI-CPO is a policy optimization algorithm. Borrowing ideas from the cooperative stochastic approximation approach, we make alternative updates to the policy parameters to maximize the reward or minimize the cost. To the best of our knowledge, we are the first to apply tools from semi-infinitely programming (SIP) to solve constrained reinforcement learning problems. We present theoretical analysis for SI-CMBRL and SI-CPO, identifying their iteration complexity and sample complexity. We also conduct extensive numerical experiments to illustrate the SICMDP model and demonstrate that our proposed algorithms are able to solve complex control tasks leveraging modern deep reinforcement learning techniques.**

***Index Terms*—Constrained Markov decision process, policy gradients, reinforcement learning, semi-infinitely programming.**

I. Introduction

Reinforcement learning has achieved great success in areas such as game-playing [51], [60], robotics [30], [37], large language models [42], [43], etc. However, due to safety concerns or physical limitations, in some real-world reinforcement learning problems, we must consider additional constraints that may influence the optimal policy and the learning process [21], [68]. A standard framework to handle such cases is the constrained Markov Decision Process (CMDP) [6]. Within the CMDP framework, the agent has to maximize the expected cumulative reward while obeying a finite number of constraints, which are usually in the form of expected cumulative cost criteria.

However, we are sometimes concerned with problems with a continuum of constraints. For example, the constraints we meet might be time-evolving or subject to uncertain parameters, which cannot be formulated as an ordinary CMDP (see Examples 1 and 2 in Section III). In this paper we study a generalized CMDP to address the above problem. Because the constraints are not only infinitely many but also lie in a continuous set, the generalization is not trivial. Fortunately, we find that we can borrow the idea behind semi-infinite programming (SIP) [26], [47] to deal with the semi-infinite constraints. Accordingly, we propose *semi-infinitely constrained Markov decision processes* (SICMDPs) as a novel complement to the ordinary CMDP framework.

We also present two reinforcement learning algorithms to solve SICMDPs called *semi-infinitely constrained model-based reinforcement learning* (SI-CMBRL) and *semi-infinitely constrained policy optimization* (SI-CPO), respectively. SI-CMBRL is a model-based reinforcement learning algorithm designed for tabular cases, and SI-CPO is a policy optimization algorithm for non-tabular cases. The main challenge is that we need to deal with a continuum of constraints, thus reinforcement learning algorithms for ordinary CMDPs do not work anymore. In SI-CMBRL, we tackle this difficulty by first transforming the reinforcement learning problem to an equivalent linear semi-infinite programming (LSIP, or also SILP in some literature) problem, which can then be solved using methods in the LSIP literature like the dual exchange methods [27], [46]. In SI-CPO, we resort to the idea of cooperative stochastic approximation developed in [33], [65]. As far as we know, we are the first to introduce tools from SIP into the reinforcement learning community for solving constrained reinforcement learning problems.

Furthermore, we give theoretical analysis for both SI-CMBRL and SI-CPO. For SI-CMBRL, we decompose the error into two parts: the statistical error from approximating the true SICMDP with an offline dataset and the optimization error due to the fact that the solution of the LSIP problem obtained by the dual exchange method is inexact. On the

Manuscript received 10 June 2023; revised 26 October 2023; accepted 19 December 2023. Date of publication 1 January 2024; date of current version 3 April 2024. This work was supported in part by the National Key Research and Development Project of China under Grant 2022YFA1004002 and in part by the National Natural Science Foundation of China under Grants 12350001 and 12271011. Recommended for acceptance by C. Wolf. *(Corresponding author: Zhihua Zhang.)*

Liangyu Zhang and Wenhao Yang are with the Academy for Advanced Interdisciplinary Studies, Peking University, Beijing 100871, China (e-mail: zhangliangyu@pku.edu.cn; yangwenhaosms@pku.edu.cn).

Yang Peng and Zhihua Zhang are with the School of Mathematical Sciences, Peking University, Beijing 100871, China (e-mail: pengyang@pku.edu.cn; zhangpku@pku.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2023.3348460, provided by the authors.

Digital Object Identifier 10.1109/TPAMI.2023.3348460

optimization side, we show that the iteration complexity of SI-CMBRL is $O(\{\text{diam}(Y)L\sqrt{|\mathcal{S}|^2|\mathcal{A}|m}/[(1-\gamma)\epsilon]\}^m)$, where $Y \subset \mathbb{R}^m$ denotes the set a continuum of constraints and $\text{diam}(Y) := \sup_{y_1,y_2\in Y} \|y_1 - y_2\|_\infty$. On the statistical side, we show that the sample complexity of SI-CMBRL is $\widetilde{O}(\frac{|\mathcal{S}|^2|\mathcal{A}|^2}{\epsilon^2(1-\gamma)^3})$ if the offline dataset is generated by a generative model, and $\widetilde{O}(\frac{|\mathcal{S}||\mathcal{A}|}{\nu_{\min}\epsilon^2(1-\gamma)^3})$ if the dataset is generated by a probability measure ν as considered in [14]. Here $\widetilde{O}$ means that all logarithm terms are discarded. For SI-CPO, things become a little more complicated because, other than the statistical error and the optimization error, we also need to consider the function approximation error, which comes from imperfect policy parametrizations. It is shown that if the function approximation error can be controlled to $O(\epsilon)$ order, the iteration complexity of SI-CPO is $\widetilde{O}(\frac{1}{\epsilon^2(1-\gamma)^6})$ and the sample complexity of SI-CPO is $\widetilde{O}(\frac{1}{\epsilon^4(1-\gamma)^{10}})$. Here our iteration complexity bound is equivalent to a typical $\widetilde{O}(1/\sqrt{T})$ global convergence rate.

At the technical level, we make the following new developments to the literature: 1) our analysis of the statistical errors of SI-CMBRL is new and never appears in previous works; 2) our analysis of SI-CPO is the first try on theoretical analysis of cooperative stochastic approximation in the context of non-convex SIP.

We perform a set of numerical experiments to illustrate the SICMDP model and validate our proposed algorithms. Specifically, we examine two numerical examples, namely the discharge of sewage and ship route planning. Through the discharge of sewage example, we show the advantage of the SICMDP framework over the CMDP baseline obtained by naive discretization in modeling realistic sequential decision-making problems. Moreover, we demonstrate the effectiveness of the SI-CMBRL and SI-CPO algorithms in such tabular environments. In the ship route planning example, we illustrate the benefits of the SICMDP framework and the ability of the SI-CPO algorithm to address complex continuous control tasks involving continuous state spaces with modern deep reinforcement learning techniques.

II. Related Work

Constrained Markov decision processes (CMDPs) have been extensively applied in areas like robotics [41], communication and networks [36], [52] and finance [1]. For a detailed treatment of CMDPs readers may refer to [6]. A number of reinforcement learning algorithms for CMDPs are proposed, which can be divided into model-based methods and model-free methods. For model-based methods, [61], [71] considered the case where the reward and cost are random but the transition dynamics are known. [7], [20], [25] considered the case where the transition dynamics are unknown and need to be estimated, which is a more common setting in the literature of reinforcement learning. And [59] gave a near-minimax-optimal sample complexity bound of learning CMDPs. Most model-free methods can indeed be categorized as policy optimization methods. [19], [58] utilized a primal-dual approach that transforms the constrained problem into an unconstrained one by considering the Lagrange functions. [2], [35], [67] addressed the constrained problem by adding constraints to the sub-problems used to compute the updating direction in each iteration step. [66] proposed to solve the CMDP problem by performing alternating updates to maximize the reward or minimize the cost.

The origination of semi-infinite programming (SIP) can date back to [47]. From then on, SIP has been widely used in quantum physics [13], signal processing [38], [40], finance [18], stochastic control [9], and engineering [26]. One may refer to [23], [26] for a detailed overview of SIP as well as its recent advances. One important class of SIP problems is called linear semi-infinite programming (LSIP). [22] provides a thorough survey about the LSIP theory. Various numerical methods are proposed to solve SIP problems, including discretization methods [11], [45], [55], exchange methods [27], [70], interior point methods [54], and Newton-type methods [39], [44]. In SI-CMBRL, we choose to use the dual exchange method in [27] to solve the LSIP problem therein. Compared to other numerical methods especially Lagrangian-based methods, the chosen dual exchange method is conceptually simpler because there is no need to introduce concepts like Lagrangians or KKT conditions. Another advantage is that it is a one-phase algorithm and we do not need to make extra efforts to find a feasible starting solution. Recently, [65] proposed to solve SIP problems via the cooperative stochastic approximation method, which was first developed in [33] to solve convex stochastic optimization problems with functional or expectation constraints.

Our SI-CMBRL algorithm uses a similar strategy as in [20] in the sense that they all use the optimistic method to transform the reinforcement learning problem into a linear (semi-infinite) programming problem, which resolves the feasibility issue and makes the theoretical analysis easier as well. However, our work and [20] are very different at the technical level: 1) our theoretical guarantees are in the form of sample complexity bounds, while the results in [20] are in the form of online regret bounds; the proof techniques are quite different; and 2) [20] considered episodic MDPs, while we consider the infinite-horizon case.

The algorithmic framework of the proposed SI-CPO method may seem similar to methods proposed in [65]. Our SI-CPO algorithm differs from the comirror algorithm in [65] because we focus on tackling non-convex loss functions with semi-infinite constraints, whereas their focus remains on solving convex SIP problems." Because the non-convex loss function is induced by large-scale reinforcement learning problems, we choose policy gradient algorithms with function approximations to update the optimization variables. In contrast, the comirror method uses plain gradient descent to update the optimization variables. Also, the theoretical guarantees of SI-CPO are obtained with different analysis techniques. The CRPO algorithm in [66] can be viewed as a special example of our SI-CPO algorithm. Readers could refer to Remark 6 for more details.

This work is an extended version of the proceedings paper [69]. We extend [69] in the following ways: 1) We propose a new reinforcement learning algorithm named SI-CPO for solving large-scale SICMDPs and validate its efficacy in numerical experiments; 2) we give a theoretical analysis of SI-CPO including the iteration complexity bounds and sample complexity bounds; 3) we also refine the theoretical analysis of SI-CRL, where we discard the assumption that a subproblem must be exactly solved and give new iteration complexity bounds.

III. The SICMDP Model

A semi-infinitely constrained MDP (SICMDP) is defined by a tuple $M = \langle \mathcal{S}, \mathcal{A}, Y, P, r, c, u, \mu, \gamma \rangle$. Here $\mathcal{S}, \mathcal{A}, P, r, \mu, \gamma$ are defined in a similar manner as in common infinite-horizon discounted MDPs. Specifically, $\mathcal{S}$ and $\mathcal{A}$ are the finite sets of states and actions, respectively. P is the transition dynamics and $P(s'|s,a)$ represents the probability of transitioning to state s' when playing action a at state s. And $r : \mathcal{S} \times \mathcal{A} \to [0,1]$ is the reward function, μ is the fixed initial distribution over $\mathcal{S}$, and γ is the discount factor. Y is the set of constraints, which we define as a compact set in $\mathbb{R}^m$, and $\mathrm{diam}(Y) < \infty$ denotes its diameter, that is, $\mathrm{diam}(Y) := \sup_{y,y' \in Y} \|y - y'\|_\infty$. In addition, $c : Y \times \mathcal{S} \times \mathcal{A} \to [0,1]$ is used to denote a continuum of cost functions and the value for constraints (bounds that must be satisfied) is determined by the function $u : Y \to \mathbb{R}$. Note that when Y is finite, we get an ordinary constrained MDP, which is indeed a special case of SICMDP.

For a given policy π, we define the value function

$$V_r^\pi(s) = \mathbb{E}_{a_t|s_t \sim \pi(\cdot|s_t)} \left(\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s \right),$$

the state-action value function

$$Q_r^\pi(s,a) = \mathbb{E}_{a_t|s_t \sim \pi(\cdot|s_t)} \left(\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) | s_0 = s, a_0 = a \right),$$

and the advantage function $A_r^\pi(s,a) = Q_r^\pi(s,a) - V_r^\pi(s)$. Here $V_{c_y}^\pi(s)$, $Q_{c_y}^\pi(s,a)$ and $A_{c_y}^\pi(s,a)$ are defined in a similar manner. Let the occupancy measure on $\mathcal{S} \times \mathcal{A}$ introduced by policy π be $\nu_\pi \in \Delta(\mathcal{S} \times \mathcal{A})$ and $\nu_\pi(s,a) = (1-\gamma) \sum_{t=0}^\infty \gamma^t \mathbb{P}_\pi(s_t = s, a_t = a)$.

The general SICMDP problem is to find a stationary policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, where $\Delta(\mathcal{A})$ is the set of probability measure supported on $\mathcal{A}$, to maximize the value function while complying with a continuum of constraints. In other words, we consider the following optimization problem:

$$\max_\pi \ V_r^\pi(\mu) \quad \text{s.t. } V_{c_y}^\pi(\mu) \leq u_y, \ \forall y \in Y. \tag{M}$$

Let us see two concrete examples of SICMDPs.

Example 1 (Spatio-temporal Constraints): Consider an ordinary CMDP problem with a single constraint:

$$\max_\pi \ V_r^\pi(\mu) \quad \text{s.t. } V_c^\pi(\mu) \leq u. \tag{1}$$

In some cases the constraint would be spatio-temporal, i.e., the cost function $c(s,a)$ and the value for constraints u are no longer constant functions and would change with time $\tau \in [0,T]$ or location $x \in \mathcal{X} \subset \mathbb{R}^3$. Then we should use the SICMDP model with $Y = [0,T]$ or $Y = \mathcal{X}$ rather than the ordinary CMDP framework to model such problems:

$$\max_\pi \ V_r^\pi(\mu) \quad \text{s.t. } V_{c_\tau}^\pi(\mu) \leq u_\tau, \ \forall \tau \in [0,T], \tag{2}$$

or

$$\max_\pi \ V_r^\pi(\mu) \quad \text{s.t. } V_{c_x}^\pi(\mu) \leq u_x, \ \forall x \in \mathcal{X}. \tag{3}$$

Load Balancing: We consider a problem in mobile communications. Suppose a mobile network operator wants to deploy an RL agent to balance the load between multiple cell sites using some policy π. The objective is to minimize the total maintenance cost of all cell sites $-V_r^\pi(\mu)$ and the constraint is that, at every place x in the region $\mathcal{X}$, the cumulative communication capacity $V_{c_x}^\pi(\mu)$ is above some adaptive threshold u_x.

Ship Route Planning: Suppose we need to navigate a ship using some policy π. Our objective is to minimize the voyage duration. The constraint is that at every place x in the region $\mathcal{X}$ the cumulative environmental pollution $V_{c_x}^\pi(\mu)$ is below some adaptive threshold u_x.

Example 2 (Constraints with Uncertainty): Again we consider a problem like Problem (1). In many application scenarios the cost function $c(s,a)$ is handcrafted and the construction of $c(s,a)$ is not guaranteed to be correct. Hence it may be helpful to include an additional parameter $\epsilon \in E$ representing our uncertainty in the construction of the cost function $c(s,a)$ as well as the value of constraints u. Even if the constraint is not handcrafted and has clear physical meaning, it may still be subject to uncertain parameters $\epsilon \in E$ that cannot be observed in advance. Therefore, we should use the SICMDP model with $Y = E$ rather than the ordinary CMDP framework to model such problems:

$$\max_\pi \ V_r^\pi(\mu) \quad \text{s.t. } V_{c_\epsilon}^\pi(\mu) \leq u_\epsilon, \forall \epsilon \in E. \tag{4}$$

Underwater Drone: Suppose an underwater drone needs to maximize $V_r^\pi(\mu)$ to accomplish some tasks. When the unknown environment feature (salinity, temperature, ocean current, etc,) is $\epsilon \in E$, for a given state s the drone needs to expend an amount of energy denoted by $c_\epsilon(s,a)$ to execute action a. The constraint is that total energy consumption $V_{c_\epsilon}^\pi(\mu)$ cannot be larger than an adaptive threshold u_ϵ, ensuring that the remaining battery capacity is adequate for the return journey.

Remark 1: An alternative approach to solving problems such as Examples 1 and 2 is to naively discretize the constraint set Y, and then the discretized problem can be fit into the conventional CMDP framework. We call this strategy naive discretization. The problem with this naive method is that the prior knowledge, i.e., the constraint function is continuous w.r.t. y, would be lost, which makes the method extremely inefficient. In Section VI we demonstrate this issue via numerical examples.

When a SICMDP M is known to us, we may do the planning by solving a linear semi-infinite programming (LSIP) problem. Problem (M) can be reformulated as the following LSIP problem:

$$\begin{aligned} &\max_\nu \ \nu^\top r \\ &\text{s.t. } \frac{1}{1-\gamma} \nu^\top c_y \leq u_y, \ \forall y \in Y. \\ &\qquad \sum_{s',a} \nu(s',a)(\mathbf{1}_{\{s'=s\}} - \gamma P(s|s',a)) = (1-\gamma)\mu(s), \ \forall s \in \mathcal{S}. \end{aligned}$$

$$\nu \succeq 0. \tag{5}$$

Here ν represents the occupancy measure on $\mathcal{S}\times\mathcal{A}$ induced by some policy π. The reformulation is valid because we have $V_r^\pi(\mu)=\frac{1}{1-\gamma}\nu^\top r$ and $V_{c_y}^\pi(\mu)=\frac{1}{1-\gamma}\nu^\top c_y$. And the constraints on ν guarantee it is a valid occupancy measure. For more details one may refer to Chapter 8 of [5]. Therefore, when M is already known the optimal policy π^* can be found by solving Problem (5) and use the $\pi(a|s)=\frac{\nu_\pi(s,a)}{\sum_{a'\in\mathcal{A}}\nu_\pi(s,a')}$. And we always assume such a policy π^* exists.

Assumption 1: Problem (M) is feasible with an optimal solution π^*, or equivalently, Problem (5) is feasible with an optimal solution ν^*.

IV. Algorithms

In this section, we present two reinforcement learning algorithms called *semi-infinitely constrained model-based reinforcement learning* (SI-CMBRL) and *semi-infinitely constrained policy optimization* (SI-CPO), respectively. SI-CMBRL is a model-based reinforcement learning algorithm that can solve tabular SICMDPs in a sample-efficient way. The SI-CPO algorithm is a policy optimization algorithm and it works for large-scale SICMDPs where we can use complex function approximators such as deep neural networks to approximate the policy and the value function.

A. The SI-CMBRL Algorithm

From a high-level point of view, the SI-CMBRL algorithm is a semi-infinite version of the algorithms proposed in [20], [25]. In the first stage, SI-CMBRL takes an offline dataset $\{(s_i,a_i,s_i')|i=1,2,\ldots,n\}$ as input and generates an empirical estimate $\widehat{P}$ of the true transition dynamic P. Then the algorithm constructs a confidence set (the optimistic set) according to $\widehat{P}$ that would cover the true SICMDP with high probability. For each policy π we would view its return as the largest possible return in SICMDPs in the confidence set. This method is also called the optimistic approach. In the second stage, we reformulate the problem as an LSIP problem and find the optimistic policy $\hat{\pi}$ using an LSIP solver. It can be shown that the resulting policy $\hat{\pi}$ is guaranteed to be nearly optimal, and the theoretical analysis can be found in Section V-A.

Now we give a more detailed description of SI-CMBRL. First, the empirical estimate $\widehat{P}$ is calculated as: $\widehat{P}(s'|s,a):=\frac{n(s,a,s')}{\max(1,n(s,a))}$, where $n(s,a,s'):=\sum_{i=1}^m \mathbf{1}\{s_i=s,a_i=a,s_i'=s'\}$ and $n(s,a)=\sum_{s'}n(s,a,s')$. The reason why we do not directly plug $\widehat{P}$ into Problem (5) and solve the resulting LSIP problem is due to the lack of guarantee that the LSIP problem w.r.t. $\widehat{P}$ is feasible. To address this issue, we construct an optimistic set M_δ such that, with high probability, the true SICMDP M lies in M_δ. In particular, M_δ is defined via the empirical Bernstein's bound and the Hoeffding's bound [34]:

$$M_\delta := \Big\{\langle\mathcal{S},\mathcal{A},Y,P',r,c,u,\mu,\gamma\rangle :$$
$$|P'(s'|s,a)-\widehat{P}(s'|s,a)|\le d_\delta(s,a,s'),\forall s,s'\in\mathcal{S},a\in\mathcal{A}\Big\},$$

where

$$d_\delta(s,a,s') := \min\Bigg\{\sqrt{\frac{2\widehat{P}(s'|s,a)(1-\widehat{P}(s'|s,a))\log(4/\delta)}{n(s,a,s')}} + \frac{4\log(4/\delta)}{n(s,a,s')},\ \sqrt{\frac{\log(2/\delta)}{2n(s,a,s')}}\Bigg\}.$$

The next step is to solve the optimistic planning problem:

$$\max_{M'\in M_\delta,\pi} V_r^{\pi,M'}(\mu),\quad \text{s.t. } V_{c_y}^{\pi,M'}(\mu)\le u_y,\ \forall y\in Y, \tag{6}$$

where the superscript M' denotes that the expectation is taken w.r.t. SICMDP M'.

Theorem 1: Suppose $n/|\mathcal{S}||\mathcal{A}|\ge 3$. With probability at least $1-2|\mathcal{S}|^2|\mathcal{A}|\delta$, we have that $M\in M_\delta$, and Problem (6) is feasible.

The proof is in Appendix A, available online. Note that the optimization variables include both M' and π, and LSIP reformulations like Problem (5) would no longer be possible. Instead, we shall introduce the state-action-state occupancy measure $z(s,a,s')$. In particular, assuming $z_{P,\pi}(s,a,s'):=P(s'|s,a)\nu_\pi(s,a)$, we have $P(s'|s,a)=\frac{z_{P,\pi}(s,a,s')}{\sum_{x\in\mathcal{S}}z_{P,\pi}(s,a,x)}$, and $\pi(a|s)=\frac{\sum_{s'\in\mathcal{S}}z_{P,\pi}(s,a,s')}{\sum_{s'\in\mathcal{S},a'\in\mathcal{A}}z_{P,\pi}(s,a',s')}$. Problem (6) can be reformulated as the following extended LSIP problem:

$$\begin{aligned}
&\max_z \sum_{s,a,s'} z(s,a,s')r(s,a)\\
&\text{s.t. } \frac{1}{1-\gamma}\sum_{s,a,s'} z(s,a,s')c_y(s,a)\le u_y,\ \forall y\in Y.\\
&\quad z(s,a,s')\le(\widehat{P}(s'|s,a)+d_\delta(s,a,s'))\sum_{x\in\mathcal{S}}z(s,a,x),\\
&\quad \forall s,s'\in\mathcal{S},\ a\in\mathcal{A}.\\
&\quad z(s,a,s')\ge(\widehat{P}(s'|s,a)-d_\delta(s,a,s'))\sum_{x\in\mathcal{S}}z(s,a,x),\\
&\quad \forall s,s'\in\mathcal{S},\ a\in\mathcal{A}.\\
&\quad \sum_{x\in\mathcal{S},b\in\mathcal{A}} z(s,b,x)=(1-\gamma)\mu(s)+\gamma\sum_{x\in\mathcal{S},b\in\mathcal{A}}z(x,b,s),\\
&\quad \forall s\in\mathcal{S}.\\
&\quad z\succeq 0.
\end{aligned} \tag{7}$$

However, compared to LP problems, LSIP problems are typically harder to solve and there are no all-purpose LSIP solvers. Here, we choose the simple yet effective dual exchange methods [27], [46] to solve Problem (7). The SI-CMBRL algorithm can be summarized in Algorithm 1. At the t th iteration of Algorithm 1, we solve a finite LP problem with a set of constraints indexed by $Y_0\in Y$ and get a solution $z^{(t)}$. The finite LP problem serves as an approximation of the original LSIP problem. Then we determine if $z^{(t)}$ is feasible. If the answer is

Algorithm 1: SI-CMBRL.

```
Input: state space $\mathcal{S}$, action space $\mathcal{A}$, dataset $\{(s_i, a_i, s_i') | i = 1, 2, \ldots, n\}$, reward function $r$, a continuum of cost functions $c$, index set $Y$, value for constraints $u$, discount factor $\gamma$, tolerance $\eta$, maximum iteration number $T$.
for each $(s, a, s')$ tuple do
  Set $\widehat{P}(s'|s,a) := \frac{\sum_{i=1}^{n} \mathbb{1}\{s_i=s,a_i=a,s_i'=s'\}}{\max(1,\sum_{i=1}^{n} \mathbb{1}\{s_i=s,a_i=a\})}$
end for
Initialize $Y_0 = \{y_0\}$
for $t = 1$ to $T$ do
Use an LP solver to solve a finite version of Problem (7) by only considering constraints in $Y_0$ and store the solution as $z^{(t)}$.
Find $y^{(t)} \approx \text{argmax}_{y \in Y} \sum_{s,a,s'} z^{(t)}(s,a,s')c_y(s,a) - u_y$.
if $\sum_{s,a,s'} z^{(t)}(s,a,s')c_{y^{(t)}}(s,a) - u_{y^{(t)}} \le \eta$ then
    Set $z^{(T)} = z^{(t)}$.
    BREAK
  end if
  Add $y^{(t)}$ to $Y_0$.
end for
for each $(s, a)$ pair do
  Set $\hat{\pi}(a|s) = \frac{\sum_{s'} z^{(T)}(s,a,s')}{\sum_{s',a'} z^{(T)}(s,a',s')}$.
end for
RETURN $\hat{\pi}$.
```

true then $z^{(t)}$ is the desired solution of the LSIP problem. Or we find the most violated constraint and add it to Y_0 to form a new LP, which forms a finer approximation of the LSIP problem.

A key ingredient of Algorithm 1 is solving the inner-loop optimization problem

$$\max_{y \in Y} \sum_{s,a,s'} z(s,a,s')c_y(s,a) - u_y.$$

We can obtain different versions of SI-CMBRL algorithm by choosing different optimization subroutines to solve the inner-loop problem above. If c_y and u_y satisfy conditions like concavity and smoothness, then the inner problem can be solved using methods like projected subgradient ascent [12]. If the inner problem is ill-posed, we may still solve it using methods like random search [8], [53]. An interesting question is what would happen if Assumption 1 is not true. Unfortunately, it may just keep running and never halt providing that the inner optimization problem can be accurately solved.

B. The SI-CPO Algorithm

In SI-CPO, we borrow ideas from the cooperative stochastic approximation [33], [65] to deal with the infinitely many constraints. At a certain iteration, the SI-CPO algorithm first determines whether the constraint violation is below some tolerance or not. It then performs a single step of policy optimization along the direction of maximizing the value of reward if the constraint violation is below some tolerance; or performs a single step of policy optimization along the direction of minimizing the value of some cost corresponding to a violated constraint.

We now describe the SI-CPO algorithm in more detail. We follow the convention to define the parameterized policy class as $\{\pi_\theta, \theta \in \Theta \subset \mathbb{R}^d\}$ and use $\pi^{(t)}$ as short for $\pi_{\theta^{(t)}}$, $V_\diamond^{(t)}$ as short for $V_\diamond^{\pi^{(t)}}$ for ease of notation. Here $\diamond$ represents either the reward r or some cost c_y. Suppose at the t-th iteration our policy parameter is $\theta^{(t)}$, then we first construct an estimate $\widehat{V}_{c_y}^{(t)}(\mu)$ using some policy evaluation subroutine. Next, we are to solve a subproblem using some optimization subroutine

$$y^{(t)} = \underset{y}{\text{argmax}} \ \widehat{V}_{c_y}^{\pi^{(t)}}(\mu) - u_y.$$

If $\widehat{V}_{c^{(t)}}^{\pi^{(t)}}(\mu) - u_{y^{(t)}} \le \eta$, where $c^{(t)} := c_{y^{(t)}}$ and $\eta > 0$ is a tolerance threshold, we say the constraint violation is small and add the time index t to the "good set" $\mathcal{B}$. Then we perform a step of update with a policy optimization subroutine to maximize the value of reward $V_r^{(t)}(\mu)$ to get $\theta^{(t+1)}$. Else, we first add the time index t to the "bad set" $\mathcal{N}$. Next, we find the violated constraint $V_{c^{(t)}}^{\pi^{(t)}}(\mu) - u_{y^{(t)}} > \eta$, and perform a step of update with a policy optimization subroutine to minimize the value of cost $V_{c^{(t)}}^{\pi^{(t)}}(\mu)$ to get $\theta^{(t+1)}$. After T iterations, we draw $\hat{\theta}$ uniformly from the set $\{\theta^{(t)}, t \in \mathcal{B}\}$, as return the policy $\hat{\pi} = \pi_{\hat{\theta}}$. The procedure of SI-CPO is summarized in Algorithm 2.

We can get different instances of the SI-CPO algorithms by making different choices of the subroutines aforementioned. Specifically, the policy optimization subroutine can be any policy optimization algorithm like policy gradient (PG) [57], natural policy gradient (NPG) [29], trust-region policy gradient (TRPO) [48], or proximal policy optimization (PPO) [49]. The policy evaluation subroutine can be chosen as Monte-Carlo policy evaluation algorithms [16] or various TD-learning algorithms [17], [56]. We may also integrate the policy optimization subroutine and the policy evaluation subroutine into actor-critic-type algorithms [32]. The optimization subroutine can be any optimization algorithm suitable for the problem instance, like the case in Algorithm 1.

V. Theoretical Analysis

A. Theoretical Analysis of SI-CMBRL

We give PAC-type bounds for SI-CMBRL under different settings. The error of SI-CMBRL is decomposed into two parts: the optimization error from the inexactness of the solution of (6) obtained by the dual exchange method, and the statistical error from approximating Problem (M) with Problem (6). On the optimization side, we show that if the inner maximization problem w.r.t. y is solved via random search or projected subgradient ascent, the dual exchange method would produce an ϵ-optimal solution (see Definition 2) when the number of iterations $T = O([\frac{\text{diam}(Y)|\mathcal{S}|^2|\mathcal{A}|}{(1-\gamma)\epsilon}]^m)$.

On the statistical side, our goal is to determine how many samples are required to make SI-CMBRL (ϵ, δ)-optimal (see Definition 1) when Problem (6) can be solved exactly, i.e., we want to find the sample complexity of SI-CMBRL (see Definition 1). We show that the sample complexity of SI-CMBRL is

Algorithm 2: SI-CPO.

```
Input: state space $\mathcal{S}$, action space $\mathcal{A}$, reward function $r$,
  a continuum of cost function $c$, index set $Y$, value for
  constraints $u$, discount factor $\gamma$, learning rate $\alpha$,
  tolerance $\eta$, maximum iteration number $T$.
Initialize $\mathcal{B} = \emptyset$, $\mathcal{N} = \emptyset$, $\theta^{(0)} = \theta_0 \in \Theta$.
for $t = 0, \ldots, T-1$ do
  Obtain $\widehat{V}_{c_y}^{\pi^{(t)}}(\mu)$ via a policy evaluation subroutine.
  Use an optimization subroutine to solve
  $\max_y \ \widehat{V}_{c_y}^{\pi^{(t)}}(\mu) - u_y$, and set
  $y^{(t)} \approx \mathrm{argmax}_y \widehat{V}_{c_y}^{\pi^{(t)}}(\mu) - u_y$, $c^{(t)} = c_{y^{(t)}}$.
  if $\widehat{V}_{c^{(t)}}^{\pi^{(t)}}(\mu) - u_{y^{(t)}} \leq \eta$ then
    Perform a step of policy update to maximize $V_r^{\pi^{(t)}}(\mu)$
    to get $\pi^{(t+1)}$. Specifically, $\theta^{(t+1)} = \theta^{(t)} + \alpha\hat{w}^{(t)}$.
    Add $t$ to $\mathcal{B}$
  else
    Perform a step of policy update to minimize $V_{c^{(t)}}^{\pi^{(t)}}(\mu)$
    to get $\pi^{(t+1)}$. Specifically, $\theta^{(t+1)} = \theta^{(t)} - \alpha\hat{w}^{(t)}$.
    Add $t$ to $\mathcal{N}$
  end if
end for
RETURN $\hat{\pi} = \pi_{\hat{\theta}}$, where $\hat{\theta} \sim \mathrm{Unif}(\{\theta^{(t)}, t \in \mathcal{B}\})$.
```

$\widetilde{O}(\frac{|\mathcal{S}|^2|\mathcal{A}|^2}{\epsilon^2(1-\gamma)^3})$ if the dataset we use is generated by a generative model, and $\widetilde{O}(\frac{|\mathcal{S}||\mathcal{A}|}{\nu_{\min}\epsilon^2(1-\gamma)^3})$ if the dataset we use is generated by a probability measure ν defined on the space $\mathcal{S} \times \mathcal{A}$ and $P(\cdot|s,a)$ as considered in [14]. Here $\widetilde{O}$ means that all logarithm terms are discarded, and $\nu_{\min} := \min_{\nu(s,a)>0} \nu(s,a)$. We will present our theoretical analysis in more detail in the remainder of this section.

1) Preliminaries: Let π^* denote the optimal policy. An (ϵ,δ)-optimal policy is defined as follows.

Definition 1: An RL algorithm is called (ϵ,δ)-optimal for $\epsilon, \delta > 0$ if, with probability at least $1-\delta$, it returns a policy π such that

$$V_r^{\pi^*}(\mu) - V_r^{\pi}(\mu) \leq \epsilon; \quad V_{c_y}^{\pi}(\mu) - u_y \leq \epsilon, \forall y \in Y.$$

An ϵ-optimal solution of Problem (6) is defined as

Definition 2: A stationary policy $\hat{\pi}$ is called an ϵ-optimal solution of Problem (6) for $\epsilon > 0$ if

$$|V_r^{\hat{\pi}}(\mu) - V_r^{\tilde{\pi}}(\mu)| \leq \epsilon \quad \text{and} \quad |V_{c_y}^{\hat{\pi}}(\mu) - u_y| \leq \epsilon, \forall y \in Y$$

hold simultaneously. Here $\tilde{\pi}$ is the optimal solution of Problem (6).

Unless otherwise specified, we assume that $\forall (s,a) \in \mathcal{S} \times \mathcal{A}$, $c_y(s,a)$ is L_y-Lipschitz in y w.r.t. $\|\cdot\|_\infty$. We also assume that u_y is L_y-Lipschitz in y w.r.t. $\|\cdot\|_\infty$. The assumptions can be formally stated as:

Assumption 2: $c_y(s,a)$ and u_y are Lipschitz in y w.r.t. $\|\cdot\|_\infty$, i.e., $\exists L_y > 0$ s.t. $\forall y, y' \in Y, (s,a) \in \mathcal{S} \times \mathcal{A}, |c_y(s,a) - c_{y'}(s,a)| \leq L_y\|y - y'\|_\infty, |u_y - u_{y'}| \leq L_y\|y - y'\|_\infty$.

The Lipschitz assumption is usually necessary when dealing with a semi-infinitely constrained problem [27], [55]. And this assumption is indeed quite mild because Y is a compact set.

We say an offline dataset $\{(s_i, a_i, s_i')|i = 1, 2, \ldots, n\}$ to be generated by a generative model if we sample according to $P(\cdot|s,a)$ for each (s,a)-pair $n_0 = n/|\mathcal{S}||\mathcal{A}|$ times and record the results in the dataset. We say an offline dataset to be generated by probability measure ν and $P(\cdot|s,a)$ if $(s_i, a_i) \overset{i.i.d.}{\sim} \nu$ and $s_i' \sim P(\cdot|s_i, a_i)$.

We solve the inner-loop problem in Algorithm 1 with random search or projected gradient ascent. The idea of random search is simple. For an objective $f(y)$ defined on domain Y, we form a random grid of Y consisting of M grid points and select the grid point with the largest objective value. The projected subgradient ascent is defined in a standard way [12]. The precise definitions can be found in Algorithm 4 and Algorithm 5 in Appendix F, available online in our supplementary material.

2) Iteration Complexity of SI-CMBRL: We give the iteration complexity of SI-CMBRL, i.e., how many iterations are required to output an ϵ-optimal solution of Problem (6) when near-optimal solutions of the inner-loop optimization problems can be obtained. Our result is similar to Theorem 4 in [27]. Specifically, we consider two different cases: 1) we make no assumption of the constraint and use random search to solve the inner-loop problem; 2) we assume the constraint is concave and use projected subgradient ascent to solve the inner-loop problem.

Before we give the iteration complexity in the case of random search, we make the following assumption to ensure technical rigor.

Assumption 3: For any weight $v \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$, let $y_0 \in \arg\max_{y\in Y}(v^\top c_y - u_y)$. Then $\exists\epsilon_0$ such that

$$\{y : \|y - y_0\|_\infty \leq \epsilon_0\} \subset Y.$$

Assumption 3 guarantees any possible solution of the inner-loop problem lies in the interior of Y.

Theorem 2: Suppose we use random search to solve the inner-loop problem of the SI-CMBRL algorithm. If Assumption 3 holds and we set the size of random grid $M = O(\frac{\log(\delta/T)}{\log(1-((1-\gamma)\epsilon/|\mathcal{S}|^2|\mathcal{A}|\mathrm{diam}(Y))^m)})$, $\quad T = O([\frac{\mathrm{diam}(Y)|\mathcal{S}|^2|\mathcal{A}|}{(1-\gamma)\epsilon}]^m)$, SI-CMBRL would output a ϵ-optimal solution of Problem (7) with probability at least $1-\delta$. Here we require $\epsilon \leq \frac{2|\mathcal{S}|^2|\mathcal{A}|L_y\epsilon_0}{1-\gamma}$.

The proof can be found in Appendix B, available online in our supplementary material. To derive theoretical guarantees for the case of projected subgradient ascent, we need the following assumption of concavity.

Assumption 4: For any $(s,a) \in \mathcal{S} \times \mathcal{A}$, $c_y(s,a)$ is concave in y. In addition, u_y is convex in y.

Theorem 3: Suppose we use projected gradient ascent to solve the inner-loop problem of the SI-CMBRL algorithm. If Assumption 4 is true and we set the iteration number of the projected subgradient ascent $T_{PGA} = O(\frac{|\mathcal{S}|^4|\mathcal{A}|^2\mathrm{diam}(Y)^2}{(1-\gamma)^2\epsilon^2})$, $T = O([\frac{\mathrm{diam}(Y)|\mathcal{S}|^2|\mathcal{A}|}{(1-\gamma)\epsilon}]^m)$, SI-CMBRL would output a ϵ-optimal solution of Problem (7).

One may refer to Appendix B, available online in our supplementary material for the detailed proofs. The most crucial part of our proofs is an ϵ-packing argument. Suppose we can get a $\epsilon/2$-optimal solution to the inner-loop problem by either random search of projected subgradient ascent and set the tolerance $\eta = \epsilon/2$. By the assumption of Lipschitzness and the construction of the SI-CMBRL algorithm, for any $t \leq T$, either the SI-CMBRL algorithm has already terminated and we obtain a ϵ-optimal solution to Problem 7, or $\{B^{(t')}, t' = 1, \ldots, t\}$ forms a packing of Y. Here $B^{(t')} := \{y : \|y - y^{(t')}\|_\infty \leq \epsilon/2\beta\}$, and β is some Lipschitz coefficient. Then we may draw the conclusion by noting that the maximum iteration number of SI-CMBRL is no larger than the $\epsilon/2\beta$-packing number of Y. We find that [27] also used similar techniques to derive their convergence rate, although they assume the inner-loop problem can always be solved exactly.

Remark 2: The iteration complexity of the SI-CMBRL algorithm grows with m in an exponential manner. Thus from a theoretical viewpoint, the SI-CMBRL algorithm is no better than the naive discretization method mentioned in Remark 1. However, we find SI-CMBRL is far more efficient than the naive method in empirical evaluations. Perhaps it is because our bound of iteration complexity is obtained by the packing argument and not tight enough. Hopefully, the bound can be tightened by a refined analysis of the dynamics of $\{(y^{(t)}, z^{(t)}), t = 1, \ldots, T\}$.

3) Sample Complexity of SI-CMBRL: We consider the case where the offline dataset we use is generated by a generative model. First, we consider a restricted setting as in [34] where for each (s, a)-pair in the true SICMDP there are at most two possible next-states and provide the sample complexity bound. Then we will drop Assumption 5 using the same strategy as in [34] and derive the sample complexity bound of the general case.

Assumption 5: The true unknown SICMDP M satisfies $P(s'|s, a) = 0$ for all but two $s' \in \mathcal{S}$ denoted as sa^+ and $sa^- \in \mathcal{S}$.

Although Assumption 5 seems quite restrictive, we argue that it is necessary to establish sharp sample complexity bound, as shown in [34]. Specifically, without this assumption the "quasi-Bernstein bound" will not hold, thus we may not be able to get the $\widetilde{O}((1-\gamma)^{-3})$ bound.

Lemma V.1: Suppose Assumption 5 holds, and the dataset we use is generated by a generative model with $n/|\mathcal{S}||\mathcal{A}| = n_0 > \max\{\frac{36 \log 4/\delta}{(1-\gamma)^2}, \frac{4 \log 4/\delta}{(1-\gamma)^3}\}$. Then with probability $1 - 2|\mathcal{S}|^2|\mathcal{A}|\delta$, we have that

$$V_r^{\pi^*}(\mu) - V_r^{\tilde{\pi}}(\mu) \leq 24\sqrt{\frac{\log 4/\delta}{n_0(1-\gamma)^3}};$$

$$V_{c_y}^{\tilde{\pi}}(\mu) - u_y \leq 12\sqrt{\frac{\log 4/\delta}{n_0(1-\gamma)^3}}, \ \forall y \in Y.$$

Here $\tilde{\pi}$ is the exact solution of Problem (6).

Theorem 4: Suppose Assumption 5 holds, the dataset we use is generated by a generative model and Problem 6 can be solved exactly. Then when $n = O(\frac{|\mathcal{S}||\mathcal{A}| \log(8|\mathcal{S}|^2|\mathcal{A}|/\delta)}{\epsilon^2(1-\gamma)^3})$, SI-CMBRL is (ϵ, δ)-optimal.

Proof: Theorem 4 is a direct consequence of Lemma V.1.

Theorem 5: Suppose the dataset we use is generated by a generative model and Problem 6 can be solved exactly. Then when $n = O(\frac{|\mathcal{S}|^2|\mathcal{A}|^2(\log|\mathcal{S}|)^3 \log(8|\mathcal{S}|^4|\mathcal{A}|^3/\delta)}{\epsilon^2(1-\gamma)^3})$, a modification of SI-CMBRL is (ϵ, δ)-optimal.

The detailed proof can be found in Appendix B, available online in our supplementary material. Our proof strategy is similar to [34]. However, to get a $\widetilde{O}((1-\gamma)^{-3})$ bound, [34] used a tedious recursion argument. We greatly simplify the proof and achieve improvements in log terms (by a factor of $(\log(\frac{|\mathcal{S}|}{\epsilon(1-\gamma)}))^2$) using sharper bounds on local variances of MDPs developed in [3].

Remark 3: It can be noted that our sample complexity bound does not rely on the constraint set Y. This is because we consider the setting where r and c_y are known deterministic functions and the only source of randomness comes from estimating the unknown transition dynamic using an offline dataset. In other words, the constraints do not make the problem more difficult in the statistical sense.

Remark 4: Here "a modification of SI-CMBRL" stands for the following procedure: first we transform the original SICMDP to a new SICMDP satisfying Assumption 5, then we run SI-CMBRL to solve the new SICMDP. One may refer to the proof in Appendix B, available online in our supplementary material for more details.

Now we generalize our results to the case where the offline dataset is generated by a probability measure.

Theorem 6: Suppose the dataset we use is generated by a probability measure ν and Problem 6 can be solved exactly. Then when $m = O(\frac{|\mathcal{S}||\mathcal{A}|(\log|\mathcal{S}|)^3 \log(8|\mathcal{S}|^4|\mathcal{A}|^3/\delta)}{\nu_{\min}\epsilon^2(1-\gamma)^3})$, a modification of SI-CMBRL is (ϵ, δ)-optimal.

One may find the proof in Appendix B, available online in our supplementary material.

B. Theoretical Analysis of SI-CPO

In this section, we present theoretical guarantees of SI-CPO. We consider a version of the SI-CPO algorithm where we use sample-based NPG [4] as the policy optimization subroutine, a finite-horizon Monte-Carlo estimator as the policy evaluation subroutine, and either random search or projected subgradient ascent as the optimization subroutine. It is shown that when the function approximation error ϵ_{bias} is of the same order with ϵ, our proposed algorithm takes $\widetilde{O}(\frac{1}{\epsilon^2(1-\gamma)^6})$ iterations and make $\widetilde{O}(\frac{1}{\epsilon^4(1-\gamma)^{10}})$ interactions with the environment to achieve an ϵ-level of averaged suboptimality with high probability. This corresponds to a $\widetilde{O}(1/\sqrt{T})$ global convergence rate, which is typical for NPG-based policy optimization algorithms. We give a detailed description of the considered version of the SI-CPO algorithm as well as our technical assumptions in Section V-B1 and present the theoretical results in Sections V-B2 and V-B3.

1) Preliminaries: Recall the policy π is parameterized by $\theta \in \Theta \subset \mathbb{R}^d$ (denoted by π_θ). We make the following assumptions about the parameterized policy class.

Assumption 6 (Differentiable policy class): Π can be parametrized as $\Pi_\theta = \{\pi_\theta | \theta \in \mathbb{R}^d\}$ such that, for all $s \in \mathcal{S}, a \in \mathcal{A}$, $\log_\theta \pi(a|s)$ is a differentiable function of θ.

Assumption 7 (Lipschitz policy class): For all $s \in \mathcal{S}$, $a \in \mathcal{A}$, $\log \pi_\theta(a|s)$ is a L_π-Lipschitz function of θ, i.e.,

$$\|\nabla_\theta \log \pi_\theta(s|a)\|_2 \leq L_\pi, \forall s \in \mathcal{S}, a \in \mathcal{A}, \theta \in \mathbb{R}^d.$$

Assumption 8 (Smooth policy class): For all $s \in \mathcal{S}$, $a \in \mathcal{A}$, $\log \pi_\theta(a|s)$ is a β-smooth function of θ, i.e.,

$$\|\nabla_\theta \log \pi_\theta(a|s) - \nabla_\theta \log \pi_{\theta'}(a|s)\|_2 \leq \beta\|\theta - \theta'\|_2,$$
$$\forall s \in \mathcal{S}, a \in \mathcal{A}, \theta, \theta' \in \mathbb{R}^d.$$

Assumption 9 (Positive semidefinite Fisher information): For all $\theta \in \mathbb{R}^d$,

$$F(\theta) := \mathbb{E}_{(s,a)\sim\nu_\theta}[\nabla_\theta \log \pi_\theta(a|s) \nabla_\theta \log \pi_\theta(a|s)^\top] \succeq \mu_F I_d.$$

The assumptions above are standard in the literature of policy optimizations [4]. Assumption 6–8 require that $\log \pi_\theta(a|s)$ is a smooth function of θ, and can be fufilled by most parametrized policy classes. Assumption 9 states that $F(\theta)$ is a good preconditioner in the NPG update. This is a common requirement for the convergence of preconditioned algorithms, for example, the quasi-Newton algorithms [50]. We also assume the parametrization realizes good function approximation in terms of transferred compatible function approximation errors, which was first introduced by [4]. The error term can be close to zero if the policy class is rich [64] or the underlying MDP has low-rank structures [28].

Assumption 10 (Bounded function approximation error): The transferred compatible function approximation errors satisfy that $\forall t \in \{1, \ldots, T\}$

$$\min_w E^{\nu^{(t)}}(r, \theta^{(t)}, w) \leq \epsilon_{\text{bias}}$$
$$\min_w E^{\nu^{(t)}}(c_y, \theta^{(t)}, w) \leq \epsilon_{\text{bias}} \; \forall y \in Y,$$

where $\nu^{(t)}$ denotes the state-action occupancy measure induced by policy $\pi^{(t)}$. The transferred compatible function approximation errors are defined as:

$$E^\nu(\diamond, \theta, w) := \mathbb{E}_{(s,a)\sim\nu}(A_\diamond^{\pi_\theta}(s,a) - w^\top \nabla_\theta \log \pi_\theta(a|s))^2.$$

Besides, we also assume the weights to minimize the transferred compatible function approximation errors are bounded.

Assumption 11 (Bounded Weight): For any $t \in \{1, \ldots, T\}$, $\forall y \in Y$,

$$\left\| \underset{w}{\operatorname{arg}min} E^{\nu^{(t)}}(r, \theta^{(t)}, w) \right\|_2^2 \leq W^2,$$
$$\left\| \underset{w}{\operatorname{argmin}} E^{\nu^{(t)}}(c_y, \theta^{(t)}, w) \right\|_2^2 \leq W^2.$$

In the theoretical analysis of SI-CPO, we consider an instance of SI-CPO where we use a sample-based version of NPG [4] as the policy optimization subroutine, a fixed-horizon Monte-Carlo estimator as the policy evaluation subroutine, and either random search or projected subgradient ascent as the optimization subroutine. In the NPG algorithm, we use the following natural policy gradient $w^{(t)}$ to update the policy parameters:

$$w^{(t)} := F(\theta^{(t)})^\dagger \mathbb{E}_{(s,a)\sim\nu^{(t)}}(A_\diamond^{\pi^{(t)}}(s,a) \nabla_\theta \pi_\theta(a|s)).$$

Here $\diamond$ can be either the reward r or some cost function c_y. However, for most RL problems it is computationally prohibitive to evaluate $F(\theta)^\dagger$, and $\mathbb{E}_{(s,a)\sim\nu^{(t)}}(A_\diamond^{\pi^{(t)}}(s,a) \nabla_\theta \pi_\theta(a|s))$ are usually unknown to the algorithm. Therefore, we instead use a sample-based estimate of $w^{(t)}$, which can be obtained by solving the following optimization problem by running K_{sgd} steps of stochastic gradient descent:

$$\hat{w}^{(t)} \approx \arg\min_w E^{\nu^{(t)}}(b, \theta^{(t)}, w),$$

recall that $E^{\nu^{(t)}}(\diamond, \theta^{(t)}, w)$ is the transferred function approximation error defined in Assumption 10. The precise definition of sample-based NPG can be found in Appendix F, available online in our supplementary material.

As for policy evaluation, we choose to use a Monte-Carlo estimator with a fixed horizon H. The idea is very simple, in each episode we run the target policy π for H steps, and record the return

$$G_i = \sum_{k=0}^{H-1} \gamma^k c_y(s_k, a_k).$$

The procedure is repeated for K_{eval} times and we take the average as an estimate of $V_{c_y}^{\pi^{(t)}}(\mu)$. Compared with the more commonly used unbiased Monte-Carlo estimate

$$\widetilde{G}_i = \sum_{k=0}^{H'-1} c_y(s_k, a_k)$$

where H' is no longer fixed and drawn from an exponential distribution $\text{Exp}(\lambda)$, G_i does introduce bias, but it also has the advantage of being sub-Gaussian. Moreover, the bias term is always bounded by $\frac{\gamma^H}{1-\gamma}$, which decays exponentially as we choose larger Hs.

2) Iteration Complexity of SI-CPO: The following two theorems give the iteration complexity of the SI-CPO algorithm when we use either random search or projected subgradient ascent to solve the inner-loop problem.

Theorem 7: Suppose we use random search to solve the inner-loop problem and Assumption 3 holds. If we set

$$\alpha = 1/\sqrt{T},$$
$$\eta = \epsilon + \frac{1}{(1-\gamma)^{3/2}} \sqrt{\left\| \frac{\nu^*}{\nu_0} \right\|_\infty \epsilon_{bias}},$$
$$K_{sgd} = \widetilde{O}\left(\frac{1}{\epsilon_{bias}^2 (1-\gamma)^4} \right),$$
$$K_{eval} = \widetilde{O}\left(\frac{1}{\epsilon^2 (1-\gamma)^2} \right),$$

$$H = O\left(\frac{\log(1-\gamma) + \min\{\log(\epsilon_{bias}), \log(\epsilon)\}}{\log\gamma}\right),$$

$$M = \widetilde{O}\left(\frac{(\text{diam}(Y))^m}{\epsilon^m(1-\gamma)^m}\right),$$

and $T = \widetilde{O}(\frac{1}{\epsilon^2(1-\gamma)^2})$, then we have with probability $1-2\delta$,

$$\frac{1}{|\mathcal{B}|}\sum_{t\in\mathcal{B}}(V_r^*(\mu) - V_r^{(t)}(\mu)) \le \epsilon + \frac{1}{(1-\gamma)^{3/2}}\sqrt{\left\|\frac{\nu^*}{\nu_0}\right\|_\infty \epsilon_{bias}},$$

and $\forall t \in \mathcal{B}$

$$\sup_{y\in Y}\left[V_{c_y}^{(t)}(\mu) - u_y\right] \le 2\epsilon + \frac{1}{(1-\gamma)^{3/2}}\sqrt{\left\|\frac{\nu^*}{\nu_0}\right\|_\infty \epsilon_{bias}}.$$

for any $\epsilon < 2\epsilon_0 L_y/(1-\gamma)$.

Theorem 8: Suppose we use projected subgradient ascent to solve the inner-loop problem and Assumption 4 holds. If we set

$$\alpha = 1/\sqrt{T},$$

$$\eta = \epsilon + \frac{1}{(1-\gamma)^{3/2}}\sqrt{\left\|\frac{\nu^*}{\nu_0}\right\|_\infty \epsilon_{bias}},$$

$$K_{sgd} = \widetilde{O}\left(\frac{1}{\epsilon_{bias}^2(1-\gamma)^4}\right),$$

$$K_{eval} = \widetilde{O}\left(\frac{1}{\epsilon^2(1-\gamma)^2}\right),$$

$$H = O\left(\frac{\log(1-\gamma) + \min\{\log(\epsilon_{bias}), \log(\epsilon)\}}{\log\gamma}\right),$$

$$T_{PGA} = O\left(\frac{[\text{diam}(Y)]^2}{\epsilon^2(1-\gamma)^2}\right),$$

and $T = \widetilde{O}(\frac{1}{\epsilon^2(1-\gamma)^2})$, then we have with probability $1-\delta$,

$$\frac{1}{|\mathcal{B}|}\sum_{t\in\mathcal{B}}(V_r^*(\mu) - V_r^{(t)}(\mu)) \le \epsilon + \frac{1}{(1-\gamma)^{3/2}}\sqrt{\left\|\frac{\nu^*}{\nu_0}\right\|_\infty \epsilon_{bias}},$$

and $\forall t \in \mathcal{B}$

$$\sup_{y\in Y}\left[V_{c_y}^{(t)}(\mu) - u_y\right] \le 2\epsilon + \frac{1}{(1-\gamma)^{3/2}}\sqrt{\left\|\frac{\nu^*}{\nu_0}\right\|_\infty \epsilon_{bias}}.$$

One may refer to Appendix C, available online in our supplementary material for the detailed proof. In our proof, we focus on the event that the policy evaluation subroutine returns accurate estimates of $V_{c_y}^{(t)}(\mu)$ and the sample-based NPG generates a near-optimal solution of $\min_w E^{\nu^{(t)}}(\diamond, \theta^{(t)}, w)$. We show that this event happens with high probability. When it happens, with carefully chosen tolerance threshold η, either the "good set" $\mathcal{B}$ is large or the policies in $\mathcal{B}$ perform as well as the optimal policy π^* on average, i.e. $\sum_{t\in\mathcal{B}}(V_r^{(t)}(\mu) - V_r^*(\mu)) \ge 0$. As long as $\mathcal{B}$ is large enough, we may further conclude that $\frac{1}{|\mathcal{B}|}\sum_{t\in\mathcal{B}}|V_r^{(t)}(\mu) - V_r^*(\mu)|$ is small by typical analysis techniques of NPG [4]. Recalling that the constraint violations of policies in $\mathcal{B}$ are small as long as the inner-loop optimization problems are effectively solved, we complete our proof.

Our ideas of proof are similar to [65], [66]. However, [65] focused on the semi-infinitely constrained convex problems and we focus on the semi-infinitely constrained RL problems. Moreover, their theoretical results are in the form of bounds on expectations, while ours are in the form of high probability bounds. Our work is different from [66] in the sense that they address finitely constrained RL problems where the sub-problem of cut generation can be exactly solved and we address semi-infinite RL problems where the sub-problem of cut generation can only be approximately solved. Also, [66] restrict their analysis to two specific forms of policy parametrizations, whereas we consider general policy parametrizations.

Remark 5: The error terms of SI-CPO can be attributed to three sources: the function approximation error, the statistical error, and the optimization error. When we say SI-CPO converges to the globally optimal policy π^* at a $\widetilde{O}(1/\sqrt{T})$ rate, we mean that if we use a near-perfect parameterized policy class, estimate $V_{c_y}^{(t)}(\mu)$ and the natural policy gradient with adequate data, and solve the inner-loop problem with sufficient accuracy, then the averaged error term of SI-CPO has a $\widetilde{O}(1/\sqrt{T})$ order with high probability.

Remark 6: When solving the inner-loop problem, an alternative approach to random search is to search over a fixed grid of Y. This is equivalent to a version of naive discretization: we first transform the SICMDP to a finitely constrained MDP by discretizing Y, and then solve the resulting problem with CRPO [66]. From a theoretical viewpoint, random search is no better than the grid search since both need to search over a $\widetilde{O}((\text{diam}(Y)/\epsilon)^m)$-sized grid to ensure ϵ-optimality. However, in numerical experiments we find that the approach based on random search is far more efficient than the approach based on grid search. The reasons can be two-fold: 1) in the theoretical analysis we must give guarantees for the hardest problem instances, but real-world problem settings may contain structures that can be exploited by random search [10]; 2) in random search, the random grids are generated in an independent way in each iteration, which can reduce the bias introduced by replacing the constraint set Y with a fixed finite grid.

3) Sample Complexity of SI-CPO: Corollary 1: SI-CPO needs to perform $\widetilde{O}(\frac{1}{\epsilon^2\min\{\epsilon^2,\epsilon_{bias}^2\}(1-\gamma)^6})$ interactions with the environment to ensure with high probability

$$\frac{1}{|\mathcal{B}|}\sum_{t\in\mathcal{B}}(V_r^*(\mu) - V_r^{(t)}(\mu)) \le \epsilon + \frac{1}{(1-\gamma)^{3/2}}\sqrt{\left\|\frac{\nu^*}{\nu_0}\right\|_\infty \epsilon_{bias}},$$

and $\forall t \in \mathcal{B}$

$$\sup_{y\in Y}\left[V_{c_y}^{(t)}(\mu) - u_y\right] \le 2\epsilon + \frac{1}{(1-\gamma)^{3/2}}\sqrt{\left\|\frac{\nu^*}{\nu_0}\right\|_\infty \epsilon_{bias}}.$$

Proof: This corollary is a direct consequence of Theorem 7 as the sample complexity is of the order $T\cdot H\cdot(K_{eval}+K_{sgd})$. Note that the sample complexity bound is independent of how we solve the inner-loop problem. □

Our sample complexity bound is of the order $\widetilde{O}(\frac{1}{\epsilon^4(1-\gamma)^6})$. This is better than typical sample complexity bounds for sample-based NPG such as the $\widetilde{O}(\frac{1}{\epsilon^4(1-\gamma)^{10}})$ in [4]. The difference comes from that when optimizing $E^{v^{(t)}}(\diamond, \theta^{(t)}, w)$, the noise of our gradient estimate has a sub-gaussian tail due to the use of a biased finite-horizon Monte-Carlo estimator for policy evaluation. Therefore, in our algorithm, we need to run fewer SGD iterations (corresponding to smaller K_{sgd}) to find a good estimate of the natural gradient.

VI. Numerical Experiments

We design two numerical examples: discharge of sewage and ship route planning. Through a set of numerical experiments, we illustrate the SICMDP model and validate the efficacy of our proposed algorithms. In particular, we find that the SICMDP framework greatly outperforms the CMDP baseline obtained by naively discretizing the original problem in modeling problems like Examples 1, 2. We highlight that in the example of ship route planning the SI-CPO algorithm is can efficiently solving complex reinforcement learning tasks using modern deep reinforcement learning approaches.

A. Discharge of Sewage

We consider a tabular sequential decision-making problem called discharge of sewage that is adapted from the literature of environmental science [24]. Assume there are $|\mathcal{S}|$ sewage outfalls in a region $[0,1]^2$, and at each time point only one single outfall is active. The active outfall would cause pollution in nearby areas, and the impact would decrease with euclidean distance. Hence our state is the current active outfall. Given the current active outfall, the available actions are to switch to one of $|\mathcal{A}|$ neighboring outfalls or do nothing. Each switch would receive a negative reward representing the switching cost. We need to figure out a switching policy to avoid over-pollution at each location of the region while minimizing the switching cost. Clearly, this problem can be formulated as a SICMDP model with $Y = [0,1]^2$ and corresponding c_y and u_y. Specifically, we use $c_y(s,a) = c_y(s) = 1/(1 + \|y - s\|_2^2)$, where s represents the position of the state (outfall). Given a target state-occupancy measure d we define $u_y = (1 + \Delta) \sum_{s \in S} d(s) c_y(s)$, where Δ is a small positive number. The SICMDP would be nontrivial if we choose a suitable Δ. In the following numerical experiments, we assume that an offline dataset generated by a generative model is available.

First, we compare our SI-CMBRL algorithm with a naive discretization baseline 1. In the baseline method, we only consider the constraints on a grid of Y containing N_{baseline} points, which allows us to model Discharge of Sewage as a standard CMDP problem with N_{baseline} constraints. The CMDP problem is then solved by the algorithm proposed in [20]. Details of our implementation can be found in Appendix E, available online in our supplementary material. We visualize the quality of solutions of our proposed method and baseline method in Fig. 3. It can be found that when $T = N_{\text{baseline}}$, the policy obtained by our proposed methods is of far better quality than the policy obtained by the baseline methods.

A counter-intuitive phenomenon is that although in our method we need to deal with multiple LP problems and in the baseline we only solve one single LP problem, our method is still more time-efficient than the CMDP baseline. Fig. 5 indicates that our method takes less time to get a solution of given accuracy, which is evaluated by the error term $\max\{V_r^{\pi^*}(\mu) - V_r^{\hat{\pi}}(\mu), \sup_{y \in Y} V_{c_y}^{\hat{\pi}}(\mu) - u_y\}$. The reason is that in SI-CMBRL we can solve LP problems with a dual simplex method, thus re-optimization after adding a new constraint is much faster than re-solving the LP problem from scratch [31]. And our method needs far fewer active constraints to attain the same accuracy as the baseline methods, see Fig. 4.

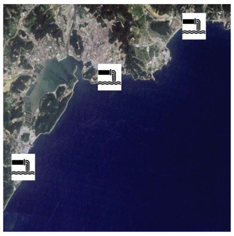


Fig. 1. (Discharge of Sewage) The icons represent locations of the sewage outfalls. The satellite image is from NASA and only for illustrative purpose.

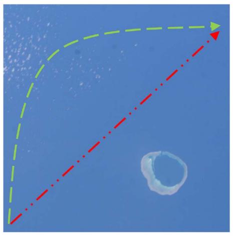


Fig. 2. (Ship Route Planning) The island represents the ecological critical point. The green dashed line represents a feasible route, while the red dash-dot-dot line represents a more efficient but ecologically infeasible route. The satellite image is from NASA and only for illustrative purpose.

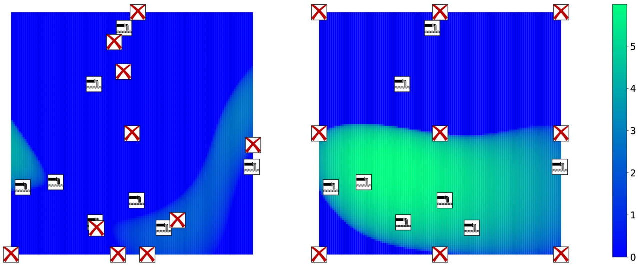


Fig. 3. (Discharge of Sewage) Visualization of violation of constraints using SI-CMBRL (left) and baseline (right). The heat refers to the number $\log((V_{c_y}^{\hat{\pi}}(\mu) - u_y)_+ + 5 \times 10^{-6}) - \log(5 \times 10^{-6})$. Larger numbers mean a more serious violation of constraints. The red cross icons in the left two subfigures represent the $T = N_{\text{baseline}} = 9$ checkpoints selected by the algorithms.

We also compare our SI-CPO algorithm with the pre-mentioned discretization baseline. In this numerical experiment, we use softmax policy parametrization. Our SI-CPO algorithm is instantiated with sample-based NPG as the policy optimization subroutine, a finite-horizon Monte-Carlo estimator as the policy evaluation subroutine, and random search with a grid of 100 points as the optimization subroutine. Here the size of

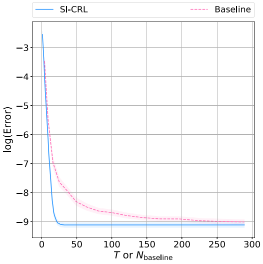


Fig. 4. (Discharge of Sewage) Averaged error term of our proposed method and the baseline method over 100 seeds when T and N_{baseline} vary. ($\delta = \frac{0.005}{|S|^2|A|}$, m sufficiently large).

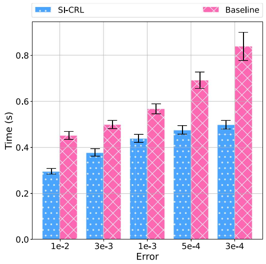


Fig. 5. (Discharge of Sewage) Averaged time consumption of our method and the CMDP baseline to get a solution of given accuracy over 100 seeds. ($\delta = \frac{0.005}{|S|^2|A|}$, m sufficiently large).

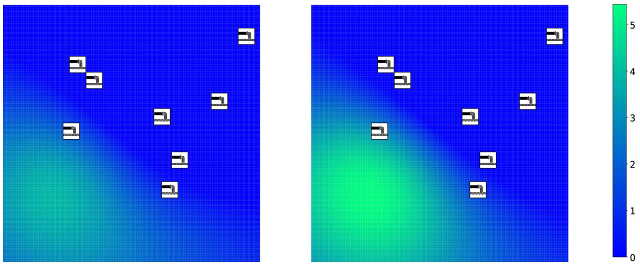


Fig. 6. (Discharge of Sewage) Visualization of violation of constraints using SI-CPO (left) and the naive discretization baseline with $N_{\text{baseline}} = 500$ solved by CRPO (right). The heat refers to the number $18(V_{c_y}^{\hat{\pi}}(\mu) - u_y)_+$. Larger numbers mean a more serious violation of constraints.

the random grid in each iteration is 100. For a fairer comparison, here the CMDP resulting from discretizing Y is solved by CRPO [66]. This is also equivalent to a naive version of our SI-CPO algorithm where the inner problem is solved by searching over a fixed grid. (See Remark 6). One may find the details of the implementation of our methods as well as the baselines in Appendix E, available online in our supplementary material.

The visualization of the solutions' quality can be found in Fig. 6, which shows that the policy obtained by SI-CPO is better than the policy obtained by the baseline solved by CRPO. In Fig. 7 we compare the convergence performance of SI-CPO to baselines that naively discretize Y into grids with different sizes (different N_{baseline}s). The resulting CMDPs are also solved by CRPO. We may observe that the SI-CPO algorithm achieves a more rapid convergence measured by the number of iterations

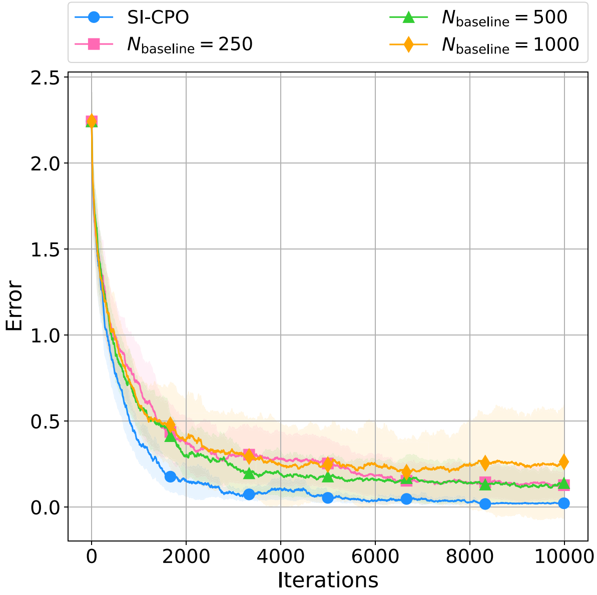


Fig. 7. (Discharge of Sewage) Error term of SI-CPO and baselines versus the number of iterations. The solid line is the error term averaged over 20 random seeds. And we also provide the according error bars.

TABLE I
(Discharge of Sewage) Time Consumption of Each Iteration in SI-CPO and Baselines

	time per iteration (s)
SI-CPO	0.19 ± 0.02
baseline, $N_{\text{baseline}} = 250$	0.23 ± 0.02
baseline, $N_{\text{baseline}} = 500$	0.30 ± 0.03
baseline, $N_{\text{baseline}} = 1000$	0.45 ± 0.04

than all the naive discretization baselines no matter how large the grid is. Also, Table I suggests that the time consumption of a single iteration of SI-CPO is comparable to baseline methods.

B. Ship Route Planning

To demonstrate the power of the SICMDP model and our proposed algorithms, we design a more complex continuous control problem with continuous state space named ship route planning. This numerical example tackles a challenging task in maritime science [62], [63]: planning ship routes while ensuring their negative environmental impact remains under an adaptive threshold. Consider a ship sailing in a 2-dimensional area represented by the unit square $[0, 1]^2$. At each time step t, the state of the ship is represented by its current position $s_t \in [0, 1]^2$ and the action it takes is represented by the next heading angle $a_t \in [0, 2\pi)$. Given an outset $O \in [0, 1]^2$ and a destination $D \in [0, 1]^2$, at each time step t, we receive a negative reward $r(s_t) = -0.1 \times (\|s_t - D\|_2 + 1)$, and after we arrive at D we will receive a large positive reward 5. The most efficient route is apparently a straight line. However, we must take into account additional environmental concerns. Specifically, the ship positioned at s would cause pollution $c_y(s) = e^{-20\|y-s\|_2}$ to position y. c_y is designed to account for the greater pollution impact on areas closer to the ship. The adaptive threshold

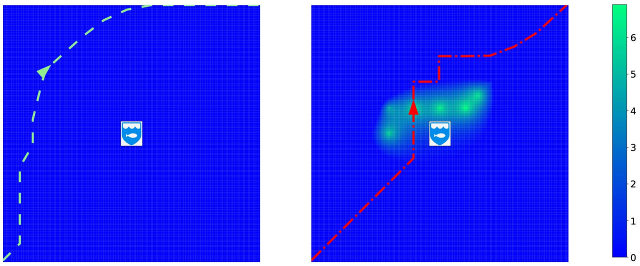


Fig. 8. (Ship Route Planning) Visualization of routes and violation of constraints using SI-CPPO (left) and naive discretization with $N_{\text{baseline}} = 1000$ (right). The heat refers to the number $5(V^{\hat{\pi}}_{c_y}(\mu) - u_y)_+$. Larger numbers mean a more serious violation of constraints. The green dashed line represents a feasible route induced by the SI-CPPO policy, while the red dash-dot line represents an infeasible route induced by the baseline policy. The blue icons in the center represent the ecologically critical points.

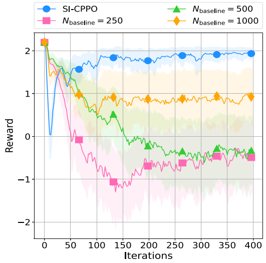


Fig. 9. (Ship Route Planning) Cumulative reward of SI-CPPO and baselines versus the number of iterations. The solid line is the cumulative reward averaged over 20 random seeds. And we also provide the according error bars.

of pollution is defined by $u_y = 0.015 + 0.005 \times e^{20\|y-\text{MPA}\|_2}$, where MPA $\in [0,1]^2$ is an environmentally critical point that has special ecological significance, such as a habitat of endangered species or a natural heritage priority site. The design of u_y reflects the principle that we implement more strict pollution restrictions for nearer positions from the environmentally critical point MPA. We would like to complement that due to the existence of a terminal state (the ship's destination), we set the discount factor $\gamma = 1$. Fig. 2 provides a visual explanation of this numerical example.

We study the performance of an actor-critic version of SI-CPO called SI-CPPO in this example. In SI-CPPO, the policy optimization subroutine is PPO, the policy evaluation subroutine is TD-learning and the optimization subroutine is a trust-region method [15]. Both the policy and the value estimator are parametrized by deep neural networks. We still consider the naive discretization baseline where the CMDPs resulting from discretization are solved by CRPO. The implementation details of SI-CPPO and the baseline can be found in Appendix E, available online in the supplementary material. Fig. 8 is a visualization of the solutions attained via SI-CPPO and the discretization baseline. While the baseline fails to generate a feasible route, SI-CPPO manages to plan a route that is both feasible and efficient. We demonstrate the convergence performance of SI-CPPO and baselines with various N_{baseline} in Figs. 9 and 10. It is shown that the convergence of baselines is very slow and the curves oscillate a lot. And simply increasing N_{baseline} does not help. In contrast, our SI-CPPO algorithm rapidly converges to the optimal solution. Also, Table II shows that a single iteration of SI-CPPO consumes a similar amount of time compared with baseline methods.

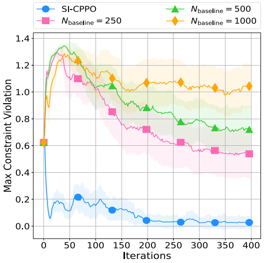


Fig. 10. (Ship Route Planning) Maximum constraint violation of SI-CPPO and baselines versus the number of iterations. The solid line is the maximum constraint violation averaged over 20 random seeds. And we also provide the according error bars.

TABLE II
(Ship Route Planning) Time Consumption of Each Iteration in SI-CPPO and Baselines

	time per iteration (s)
SI-CPPO	11.39 ± 1.43
baseline, $N_{\text{baseline}} = 250$	8.91 ± 1.50
baseline, $N_{\text{baseline}} = 500$	10.05 ± 1.72
baseline, $N_{\text{baseline}} = 1000$	11.78 ± 1.74

VII. Conclusion

We have studied a novel generalization of CMDP that we have called SICMDP. In particular, we have considered a continuum of constraints rather than a finite number of constraints. We have devised two reinforcement learning algorithms, SI-CMBRL and SI-CPO, to solve SICMDP problems. Furthermore, we have presented a theoretical analysis for our proposed algorithms, establishing the iteration complexity bounds as well as the sample complexity bounds. We have also performed extensive numerical experiments to show the efficacy of our proposed methods and their advantage over traditional CMDPs.

Acknowledgments

The authors would like to thank the anonymous reviewers, the Associate Editor, and the Editor for their detailed and constructive comments that improved the quality of this paper. The authors would also like to thank Hao Jin for helpful discussions.

References

[1] N. Abe et al., “Optimizing debt collections using constrained reinforcement learning,” in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2010, pp. 75–84.

[2] J. Achiam, D. Held, A. Tamar, and P. Abbeel, “Constrained policy optimization,” in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2017, pp. 22–31.

[3] A. Agarwal, S. Kakade, and L. F. Yang, “Model-based reinforcement learning with a generative model is minimax optimal,” in *Proc. 33rd Conf. Learn. Theory*, PMLR, 2020, pp. 67–83.

[4] A. Agarwal, S. M. Kakade, J. D. Lee, and G. Mahajan, “On the theory of policy gradient methods: Optimality, approximation, and distribution shift,” *J. Mach. Learn. Res.*, vol. 22, no. 98, pp. 1–76, 2021.

[5] E. Altman, “Constrained Markov decision processes with total cost criteria: Lagrangian approach and dual linear program,” *Math. Methods operations Res.*, vol. 48, no. 3, pp. 387–417, 1998.

[6] E. Altman, *Constrained Markov Decision Processes*, vol. 7. Boca Raton, FL, USA: CRC, 1999.

[7] S. Amani, C. Thrampoulidis, and L. F. Yang, “Safe reinforcement learning with linear function approximation,” 2021, *arXiv:2106.06239*.

[8] S. Andradóttir, “A review of random search methods,” in *Handbook of Simulation Optimization*, Berlin, Germany: Springer, 2015, pp. 277–292.

[9] A. Basu and V. S. Borkar, “Stochastic control with imperfect models,” *SIAM J. Control Optim.*, vol. 47, no. 3, pp. 1274–1300, 2008.

[10] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” *J. Mach. Learn. Res.*, vol. 13, no. 2, pp. 281–305, 2012.

[11] B. Betrò, “An accelerated central cutting plane algorithm for linear semi-infinite programming,” *Mathmatical Program.*, vol. 101, no. 3, pp. 479–495, Dec. 2004.

[12] S. Bubeck et al., “Convex optimization: Algorithms and complexity,” *Found. Trends Mach. Learn.*, vol. 8, no. 3/4, pp. 231–357, 2015.

[13] T.M. Carrijo, W.B. Cardoso, and A.T. Avelar, “Linear semi-infinite programming approach for entanglement quantification,” *Phys. Rev. A*, vol. 104, no. 2, Aug. 2021, Art. no. 022413. [Online]. Available: https://journals.aps.org/pra/abstract/10.1103/PhysRevA.104.022413

[14] J. Chen and N. Jiang, “Information-theoretic considerations in batch reinforcement learning,” in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2019, pp. 1042–1051.

[15] A. R. Conn, N. IM Gould, and P. L. Toint, *Trust Region Methods*, Philadelphia, PA, USA: SIAM, 2000.

[16] J. H. Curtiss, “A Theoretical Comparison of the Efficiencies of Two Classical Methods and a Monte Carlo Method for Computing One Component of the Solution of a Set of Linear Algebraic Equations,” New York, NY, USA: Courant Institute of Mathematical Sciences, New York University, 1954.

[17] C. Dann et al., “Policy evaluation with temporal differences: A survey and comparison,” *J. Mach. Learn. Res.*, vol. 15, pp. 809–883, 2014.

[18] S. Daum and R. Werner, “A novel feasible discretization method for linear semi-infinite programming applied to basket option pricing,” *Optimization*, vol. 60, no. 10/11, pp. 1379–1398, 2011.

[19] D. Ding, K. Zhang, T. Basar, and M. Jovanovic, “Natural policy gradient primal-dual method for constrained Markov decision processes. in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 8378–8390.

[20] Y. Efroni, S. Mannor, and M. Pirotta, “Exploration-exploitation in constrained MDPs,” 2020, *arXiv:2003.02189*.

[21] J. Garcıa and F. Fernández, “A comprehensive survey on safe reinforcement learning,” *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 1437–1480, 2015.

[22] M. A. Goberna and M. A. López, “Linear semi-infinite programming theory: An updated survey,” *Eur. J. Oper. Res.*, vol. 143, no. 2, pp. 390–405, 2002.

[23] M. A. Goberna and M. A. López, “Recent contributions to linear semi-infinite optimization: An update,” *Ann. Operations Res.*, vol. 271, pp. 237–278, 2018.

[24] W. L. Gorr, S.-Å. Gustafson, and K. O. Kortanek, “Optimal control strategies for air quality standards and regulatory policy,” *Environ. Plan. A*, vol. 4, no. 2, pp. 183–192, 1972.

[25] A. HasanzadeZonuzy, D. Kalathil, and S. Shakkottai, “Model-based reinforcement learning for infinite-horizon discounted constrained markov decision processes,” in *Proc. 13th Int. Joint Conf. Artif. Intell.*, 2021, pp. 2519–2525.

[26] R. Hettich and K. O. Kortanek, “Semi-infinite programming: Theory, methods, and applications,” *SIAM Rev.*, vol. 35, no. 3, pp. 380–429, 1993.

[27] H. Hu, “A. one-phase algorithm for semi-infinite linear programming,” *Math. Prog.*, vol. 46, no. 1, pp. 85–103, Jan. 1990.

[28] N. Jiang, A. Krishnamurthy, A. Agarwal, J. Langford, and R. E. Schapire, “Contextual decision processes with low Bellman rank are PAC-learnable,” in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2017, pp. 1704–1713.

[29] M. ShamKakade, “A natural policy gradient,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 1531–1538.

[30] J. Kober, J. A. Bagnell, and J. Peters, “Reinforcement learning in robotics: A survey,” *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1238–1274, 2013.

[31] A. Koberstein, “The dual simplex method, techniques for a fast and stable implementation,” Unpublished doctoral thesis, Dept. fakultät für wirtschaftswissenschaften (Faculty of Economics), Universität Paderborn, Paderborn, Germany, 2005.

[32] V. Konda and J. Tsitsiklis, “Actor-critic algorithms,” in *Proc. Adv. Neural Inf. Process. Syst.*, 1999, pp. 1008–1014.

[33] G. Lan and Z. Zhou, “Algorithms for stochastic optimization with function or expectation constraints,” *Comput. Optim. Appl.*, vol. 76, no. 2, pp. 461–498, 2020.

[34] T. Lattimore and M. Hutter, “Near-optimal PAC bounds for discounted MDPs,” *Theor. Comput. Sci.*, vol. 558, pp. 125–143, 2014.

[35] Y. Liu, J. Ding, and X. Liu, “IPO: Interior-point policy optimization under constraints,” in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 4940–4947.

[36] N. Mastronarde and M. van der Schaar, “Fast reinforcement learning for energy-efficient wireless communication,” *IEEE Trans. Signal Process.*, vol. 59, no. 12, pp. 6262–6266, Dec. 2011.

[37] S. Meyn, *Control Systems and Reinforcement Learning*, Cambridge, U.K.: Cambridge Univ. Press, 2022.

[38] P. Moulin, M. Anitescu, K. O. Kortanek, and F. A. Potra, “The role of linear semi-infinite programming in signal-adapted QMF bank design,” *IEEE Trans. Signal Process.*, vol. 45, no. 9, pp. 2160–2174, Sep. 1997.

[39] Q. Ni, C. Ling, L. Qi, and K. Lay Teo, “A truncated projected newton-type algorithm for large-scale semi-infinite programming,” *SIAM J. Optim.*, vol. 16, no. 4, pp. 1137–1154, 2006.

[40] S. Nordebo, Z. Zang, and I. Claesson, “A semi-infinite quadratic programming algorithm with applications to array pattern synthesis,” *IEEE Trans. Circuits Syst. II: Analog Digit. Signal Process.*, vol. 48, no. 3, pp. 225–232, Mar. 2001.

[41] M. Ono, M. Pavone, Y. Kuwata, and J. Balaram, “Chance-constrained dynamic programming with application to risk-aware robotic space exploration,” *Auton. Robots*, vol. 39, no. 4, pp. 555–571, 2015.

[42] OpenAI, Gpt-4 technical report, 2023, *arXiv:2303.08774*.

[43] L. Ouyang et al., “Training language models to follow instructions with human feedback,” 2022, *arXiv:2203.02155*.

[44] E. Polak, D. Q. Mayne, and J. E. Higgins, “On the extension of Newton’s method to semi-infinite minimax problems,” *SIAM J. Control Optim.*, vol. 30, no. 2, pp. 367–389, 1992.

[45] E. Polak and L. He, “Rate-preserving discretization strategies for semi-infinite programming and optimal control,” *SIAM J. Control Optim.*, vol. 30, no. 3, pp. 548–572, 1992.

[46] R. Reemtsen and S. Görner, “Numerical methods for semi-infinite programming: A survey,” in *Semi-Infinite Programming*, Berlin, Germany: Springer, 1998, pp. 195–275.

[47] E. Y. Remez, “Sur la détermination des polynômes d’approximation de degré donnée,” *Commun. Soc. Math. Kharkov*, vol. 10, no. 196, pp. 41–63, 1934.

[48] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, “Trust region policy optimization,” in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2015, pp. 1889–1897.

[49] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” 2017, *arXiv: 1707.06347*.

[50] D. F. Shanno, “Conditioning of quasi-Newton methods for function minimization,” *Math. Comput.*, vol. 24, no. 111, pp. 647–656, 1970.

[51] D. Silver et al., “A general reinforcement learning algorithm that masters chess, shogi, and go through self-play,” *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.

[52] R. Singh and P. R. Kumar, “Throughput optimal decentralized scheduling of multihop networks with end-to-end deadline constraints: Unreliable links,” *IEEE Trans. Autom. Control*, vol. 64, no. 1, pp. 127–142, Jan. 2019.

[53] J. F. Solis and J.-B. RogerWets, “Minimization by random search techniques,” *Math. Operations Res.*, vol. 6, no. 1, pp. 19–30, 1981.

[54] O. Stein and G. Still, “Solving semi-infinite optimization problems with interior point techniques,” *SIAM J. Control Optim.*, vol. 42, no. 3, pp. 769–788, 2003.

[55] G. Still, “Discretization in semi-infinite programming: The rate of convergence,” *Math. Program.*, vol. 91, no. 1, pp. 53–69, 2001.

[56] R. S. Sutton, “Learning to predict by the methods of temporal differences,” *Mach. Learn.*, vol. 3, pp. 9–44, 1988.

[57] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, “Policy gradient methods for reinforcement learning with function approximation,” in *Proc. Adv. Neural Inf. Process. Syst.*, 1999, pp. 1057–1063.

[58] C. Tessler, Daniel J. Mankowitz, and S. Mannor, “Reward constrained policy optimization,” 2018, *arXiv:1805.11074*.

[59] S. Vaswani, L. F. Yang, and C. Szepesvári, “Near-optimal sample complexity bounds for constrained MDPs,” 2022, *arXiv:2206.06270*.

[60] O. Vinyals et al., “Grandmaster level in StarCraft II using multi-agent reinforcement learning,” *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[61] A. Wachi and Y. Sui, “Safe reinforcement learning in constrained markov decision processes,” in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2020, pp. 9797–9806.

[62] Z. Wan, J. Chen, A. El Makhloufi, D. Sperling, and Y. Chen, “Four routes to better maritime governance,” *Nature*, vol. 540, no. 7631, pp. 27–29, 2016.

[63] Z. Wan, M. Zhu, S. Chen, and D. Sperling, “Pollution: Three steps to a green shipping industry,” *Nature*, vol. 530, no. 7590, pp. 275–277, 2016.

[64] L. Wang, Q. Cai, Z. Yang, and Z. Wang, “Neural policy gradient methods: Global optimality and rates of convergence,” 2019, *arXiv: 1909.01150*.

[65] B. Wei, W. B. Haskell, and S. Zhao, “The comirror algorithm with random constraint sampling for convex semi-infinite programming,” *Ann. Operations Res.*, vol. 295, no. 2, pp. 809–841, 2020.

[66] T. Xu, Y. Liang, and G. Lan, “CRPO: A new approach for safe reinforcement learning with convergence guarantee,” in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2021, pp. 11480–11491.

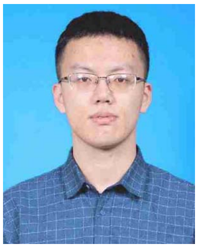
[67] T.-Y. Yang, J. Rosca, K. Narasimhan, and P. J. Ramadge, “Projection-based constrained policy optimization,” 2020, *arXiv: 2010.03152*.

[68] K. Zhang, B. Hu, and T. Basar, “Policy optimization for h_2 linear control with h_∞ robustness guarantee: Implicit regularization and global convergence,” *SIAM J. Control Optim.*, vol. 59, no. 6, pp. 4081–4109, 2021.

[69] L. Zhang, Y. Peng, W. Yang, and Z. Zhang, “Semi-infinitely constrained markov decision processes,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 16808–16820.

[70] L. Zhang, S.-Y. Wu, and M. A. López, “A new exchange method for convex semi-infinite programming,” *SIAM J. Optim.*, vol. 20, no. 6, pp. 2959–2977, Oct. 2010.

[71] L. Zheng and L. Ratliff, “Constrained upper confidence reinforcement learning,” in *Proc. 2nd Annu. Learn. Dyn. Control*, PMLR, 2020, pp. 620–629.



Liangyu Zhang is working toward the PhD degree with the Academy for Advanced Interdisciplinary Studies of Peking University, China. His research interests include reinforcement learning theory and non-parametric statistics.

Yang Peng is working toward the PhD degree with the School of Mathematical Sciences of Peking University, China. His research interests include reinforcement learning and combinatorial optimization.

Wenhao Yang received the PhD degree from Peking University, in 2023. He is a postdoctoral researcher with the Department of Management Science and Engineering (MS&E), Stanford University, USA. His research interests include reinforcement learning theory and distributionally robust optimization.

Zhihua Zhang is a professor in statistics with Peking University, China. His research interests include machine learning, artificial intelligence, high-dimensional statistics, and high-dimensional probability.