

A Policy Gradient Approach for Finite Horizon Constrained Markov Decision Processes

Soumyajit Guin¹ and Shalabh Bhatnagar¹

Abstract—The infinite horizon setting is widely adopted for problems of reinforcement learning (RL). These invariably result in stationary policies that are optimal. In many situations, finite horizon control problems are of interest and for such problems, the optimal policies are time-varying in general. Another setting that has become popular in recent times is of Constrained Reinforcement Learning, where the agent maximizes its rewards while it also aims to satisfy some given constraint criteria. However, this setting has only been studied in the context of infinite horizon MDPs where stationary policies are optimal. We present an algorithm for constrained RL in the Finite Horizon Setting where the horizon terminates after a fixed (finite) time. We use function approximation in our algorithm which is essential when the state and action spaces are large or continuous and use the policy gradient method to find the optimal policy. The optimal policy that we obtain depends on the stage and so is non-stationary in general. To the best of our knowledge, our paper presents the first policy gradient algorithm for the finite horizon setting with constraints. We show the convergence of our algorithm to a constrained optimal policy. We also compare and analyze the performance of our algorithm through experiments and show that our algorithm performs better than some other well known algorithms.

I. INTRODUCTION

The Constrained Markov Decision Process (C-MDP) setting has recently received significant attention in the reinforcement learning (RL) literature due to its natural application in safe RL problems [1], [2]. A textbook treatment of C-MDP can be found in [3]. In the C-MDP framework, in addition to the long-term objective specified via single-stage rewards (that are associated with state transitions), there are also long-term constraint functions specified via additional single-stage rewards or costs. The goal then is to find an optimal policy that maximizes a long-term reward objective while satisfying prescribed constraints. RL algorithms for infinite horizon discounted C-MDP have been studied in [4]. For the long-run average cost C-MDP, [5] has developed the first actor-critic RL algorithm in the full state setting. RL Algorithms with function approximation have also been developed for infinite horizon discounted cost C-MDP [6] as well as the long-run average cost C-MDP [7].

In this paper, we present an RL algorithm for C-MDP in the finite horizon setting. Finite Horizon problems [8], [9] deal with situations where the agent needs to choose a finite number of actions depending on the states of the environment

in order to maximize the expected sum of single-stage and the terminal reward. An optimal policy in this setting would in general be non-stationary as the choice of an action at an instant would depend not just on the state at that instant but also on the number of actions remaining to be chosen from then on so as to maximize a long-term objective. RL techniques for Finite Horizon (regular) MDP in the full state case have been discussed in [10]. They give two algorithms for the tabular case: $Q_{\mathcal{H}}$ -Learning and $R_{\mathcal{H}}$ -Learning, and do a learning-rate analysis between them. Actor-critic type algorithms for Finite-Horizon (regular) MDPs are discussed in [11]. They give four algorithms for Finite Horizon, among which three are for tabular setting, and one algorithm uses function approximation. Their algorithms for tabular setting are not scalable to large state spaces. The algorithm which uses function approximation uses zeroth order gradient instead of first order gradient, like us and also doesn't consider inequality constraints, which we do. Our RL algorithm is devised for finite-horizon C-MDP, uses function approximation, and involves actor-critic type updates. Temporal difference learning algorithms for a finite horizon setting have also recently been studied in [12]. They show convergence for Q-Learning with Linear function-approximation and some general function approximations for Finite Horizon. With their motivation to make infinite horizon Q-Learning stable, they do not consider time varying transition probability and reward functions, which we do, which is a more complicated setting.

We prove the convergence of our algorithm under standard assumptions. Our convergence guarantees gives one the power to mimic our algorithm and make algorithms using neural networks for solving complex tasks. One such task could be a portfolio management system [13], where a person wants to invest in the stock market for a finite amount of time. The person here needs to decide the ratio of the money they will invest in different stocks. The stock values naturally change with time, and the decisions made are time critical in nature. We also show in this paper, empirical results on a two-dimensional grid world problem using our algorithms where we observe that our algorithms clearly meet the constraint cost performance while giving a good reward performance. The other algorithms in the literature do not meet the constraint objective.

Our contributions:

- 1) We present and analyze both theoretically and experimentally the first policy gradient reinforcement learning algorithm with function approximation for Finite

*This work was supported by a J. C. Bose Fellowship, Project No. DFTM/02/3125/M/04/AIR-04 from DRDO under DIA-RCOE, a project from DST-ICPS, and the RBCCPS, IISc.

¹ The authors are with the Department of Computer Science and Automation, Indian Institute of Science, Bangalore 560012, India gsoumyajit@iisc.ac.in; shalabh@iisc.ac.in.

Horizon Constrained Markov Decision Processes.

- 2) This setting differs significantly from infinite horizon problems since in the latter stationary policies are optimal unlike the finite horizon setting where invariably non-stationary policies are optimal. This is because knowledge of the time remaining for termination of the horizon often has a profound bearing on action selection in the finite horizon case.
- 3) We prove that our proposed algorithm converges almost surely to a constrained optimum over a tuple of parameters, one corresponding to each instant in the horizon.
- 4) We show a comparison of the empirical performance of our algorithm in relation to well known algorithms in the literature originally designed for infinite horizon problems.
- 5) Our key observation here is that our algorithm gives a good reward performance while strictly meeting the constraint criterion at every time instant unlike the other algorithms that do meet the constraint criterion and are therefore unsuitable for Constrained Finite Horizon problems.

II. THE FRAMEWORK AND PROBLEM FORMULATION

By a Markov Decision Process (MDP), we mean a stochastic process $\{X_h\}$ taking values in a set (called the state space) S , and governed by a control sequence $\{Z_h\}$ taking values in a set A (called the action space), that satisfies the following controlled Markov property: $P(X_{h+1} = s' | X_h, Z_h, s) = p_h(X_h, Z_h, s')$ a.s. Also, associated with any state transition is a single-stage reward. We assume here that S and A are finite sets. Actions are chosen by the decision maker at instants $h = 0, 1, \dots, H-1$, and the process terminates at instant H , where H is a given positive integer. For simplicity, we consider all actions in A to be feasible in every state in S . A non-stationary randomized policy (NRP) is a set of H distributions defined by $\pi := \{\mu_0, \mu_1, \dots, \mu_{H-1}\}$, where $\mu_h(s, \cdot)$ is a distribution over A , $s \in S$, $h = 0, 1, \dots, H-1$. Here, the action $Z_h \sim \mu_h(X_h, \cdot)$, $\forall h$. It is easy to see under a NRP, $\{X_h\}$ is a non-homogeneous Markov chain.

Let $r_h(s, a, s')$ (resp. $g_h^{(1)}(s, a, s'), \dots, g_h^{(M)}(s, a, s')$) be the single-stage reward (resp. the set of single-stage costs) at instant $h = 0, 1, \dots, H-1$ when the state is $s \in S$, the action chosen is $a \in A$ and the next state is $s' \in S$. Let $r_H(s)$ (resp. $g_H^{(1)}(s), \dots, g_H^{(M)}(s)$) likewise denote the terminal reward (resp. set of terminal costs) at instant H when the terminal state is $s \in S$. Our aim is to find a NRP π^* that maximizes the following over all NRP π :

$$J(\pi) = \mathbb{E} \left[r_H^{(k)}(X_H) + \sum_{h=0}^{H-1} r_h^{(k)}(X_h, Z_h, X_{h+1}) \right], \quad (1)$$

subject to the constraints

$$S^{(k)}(\pi) = \mathbb{E} \left[g_H^{(k)}(X_H) + \sum_{h=0}^{H-1} g_h^{(k)}(X_h, Z_h, X_{h+1}) \right] \leq \alpha^{(k)}, \quad (2)$$

$k = 1, \dots, M$. Here $\alpha^{(1)}, \dots, \alpha^{(M)}$ are certain prescribed threshold values. We assume there exists at least one NRP π for which all the inequality constraints in (2) are satisfied. Let $\lambda = (\lambda^{(1)}, \dots, \lambda^{(M)})^T$ denote the vector of Lagrange multipliers $\lambda^{(1)}, \dots, \lambda^{(M)} \in \mathbb{R}^+ \cup \{0\}$ and let $L(\pi, \lambda)$ denote the Lagrangian:

$$L(\pi, \lambda) = J(\pi) + \sum_{k=1}^M \lambda^{(k)} (S^{(k)}(\pi) - \alpha^{(k)}) \\ = \mathbb{E} \left[c_H^\lambda(X_H) + \sum_{h=0}^{H-1} c_h^\lambda(X_h, Z_h, X_{h+1}) \right], \quad (3)$$

where $c_H^\lambda(s) := r_H(s) + \sum_{k=1}^M \lambda^{(k)} (g_H^{(k)}(s) - \alpha^{(k)})$ and $c_h^\lambda(s, a, s') := r_h(s, a, s') + \sum_{k=1}^M \lambda^{(k)} g_h^{(k)}(s, a, s')$ respectively. Let $V_h^{\pi, \lambda}(s)$ be the value function at instant $h = 0, 1, \dots, H-1$ for the relaxed MDP problem.

$$V_h^{\pi, \lambda}(s) = \mathbb{E} \left[c_H^\lambda(X_H) + \sum_{t=h}^{H-1} c_t^\lambda(X_t, Z_t, X_{t+1}) \right] \\ |X_h = s] \quad (4)$$

and similarly,

$$V_H^{\pi, \lambda}(s) = c_H^\lambda(s). \quad (5)$$

Note that the foregoing can be written in terms of the Q-value function as

$$V_h^{\pi, \lambda}(s) = \sum_{a \in A} \mu_h(s, a) Q_h^{\pi, \lambda}(s, a) \quad (6)$$

where,

$$Q_h^{\pi, \lambda}(s, a) = \sum_{s' \in S} p_h(s, a, s') \left[c_h^\lambda(s, a, s') + V_{h+1}^{\pi, \lambda}(s') \right].$$

Similarly we can define value function for the constraints. The value function of the k th constraint, $k = 1, \dots, M$, is defined as:

$$W_h^{\pi, (k)}(s, a) = \sum_{a \in A} \mu_h(s, a) \sum_{s' \in S} p_h(s, a, s') \\ \times [g_h(s, a, s') + W_{h+1}^{\pi, (k)}(s')] \quad (7)$$

and,

$$W_H^{\pi, (k)}(s) = g_H^{(k)}(s) - \alpha^{(k)}. \quad (8)$$

Let $\beta(s_0)$, $s_0 \in S$ be the initial state distribution. For an NRP π , let $Pr(s_0 \rightarrow s, h, \pi)$ be the probability of transitioning from initial state s_0 to state s in h steps under policy π .

Also, under π , let $d_h^\pi(s)$ be the probability of reaching state s in h steps, $h = 0, 1, \dots, H$.

$$\begin{aligned}
d_h^\pi(s) &:= \sum_{s_0 \in S} \beta(s_0) Pr(s_0 \rightarrow s, h, \pi) \\
&= \sum_{s_0 \in S} \beta(s_0) \sum_{a_0 \in A} \mu_0(s_0, a_0) \sum_{s_1 \in S} p_0(s_0, a_0, s_1) \\
&\quad \sum_{a_1 \in A} \mu_1(s_1, a_1) \sum_{s_2 \in S} p_1(s_1, a_1, s_2) \dots \\
&\quad \sum_{a_{h-1} \in A} \mu_{h-1}(s_{h-1}, a_{h-1}) p_{h-1}(s_{h-1}, a_{h-1}, s).
\end{aligned} \tag{9}$$

We consider here a parameterized class of NRP π where μ_h depends on parameter $\theta_h \in \mathbb{R}^{y_h}$, $h = 0, 1, \dots, H-1$. Let $\theta \triangleq (\theta_0, \theta_1, \dots, \theta_{H-1})^T$. Let $\pi_\theta = \{\mu_{\theta_h}(s, a), s \in S, a \in A, \theta_h \in \mathbb{R}^{y_h}, h = 0, \dots, H-1\}$ denote the parameterized class of NRP. Our goal then is to find an optimal θ^* that maximizes (1) while satisfying the constraints (2). We will use π and θ interchangeably henceforth.

Assumption 1: The policies $\pi_\theta = \{\mu_{\theta_h}, h = 0, 1, \dots, H-1\}$ are twice continuously differentiable functions of $\theta = (\theta_0, \theta_1, \dots, \theta_{H-1})^T$. Moreover, $\mu_{\theta_h}(s, a) > 0, \forall h = 0, 1, \dots, H-1, s \in S, a \in A$.

Remark 1: Assumption 1 is seen to be satisfied for example by the Gibbs distribution, see Proposition 11 of [14]. From the expression of $d_h^\pi(s)$ in (9), under the setting of parameterized policies, it is easy to see under Assumption 1 that $d_h^\theta(s)$ is continuously differentiable, $\forall h = 0, 1, \dots, H$.

We have the following preliminary result along the lines of the policy gradient theorem for infinite horizon problems (cf. Chapter 13 of [15]).

Theorem 1: (The Policy Gradient Theorem for Finite Horizon Constrained MDPs). Under Assumption 1, for any baseline function $b: S \rightarrow \mathbb{R}$, we have

$$\nabla_\theta L(\pi, \lambda) = \left(\sum_{s \in S} d_h^\pi(s) \sum_{a \in A} \nabla_{\theta_h} \mu_h(s, a) (Q_h^{\pi, \lambda}(s, a) - b(s)), \right. \\
\left. h = 0, 1, \dots, H-1 \right)^T.$$

Proof: Notice first that $\nabla_\theta L(\pi, \lambda) = (\nabla_{\theta_h} L(\pi, \lambda), h = 0, 1, \dots, H-1)^T$. Now observe that

$$\begin{aligned}
\nabla_{\theta_h} V_0^{\pi, \lambda}(s_0) &= \nabla_{\theta_h} \left(\sum_{a_0 \in A} \mu_0(s_0, a_0) Q_0^{\pi, \lambda}(s_0, a_0) \right) \\
&= \sum_{a_0 \in A} [\nabla_{\theta_h} \mu_0(s_0, a_0) Q_0^{\pi, \lambda}(s_0, a_0) + \mu_0(s_0, a_0) \nabla_{\theta_h} \\
&\quad \times Q_0^{\pi, \lambda}(s_0, a_0)] \\
&= \sum_{a_0 \in A} [\nabla_{\theta_h} \mu_0(s_0, a_0) Q_0^{\pi, \lambda}(s_0, a_0) + \mu_0(s_0, a_0) \\
&\quad \times \nabla_{\theta_h} \sum_{s_1 \in S} p_0(s_0, a_0, s_1) (c_0^\lambda(s_0, a_0, s_1) + V_1^{\pi, \lambda}(s_1))] \\
&= \sum_{a_0 \in A} [\nabla_{\theta_h} \mu_0(s_0, a_0) Q_0^{\pi, \lambda}(s_0, a_0) + \mu_0(s_0, a_0) \\
&\quad \times \sum_{s_1 \in S} p_0(s_0, a_0, s_1) \nabla_{\theta_h} V_1^{\pi, \lambda}(s_1)].
\end{aligned}$$

Unrolling the above expression repeatedly we get:

$$\begin{aligned}
\nabla_{\theta_h} V_0^{\pi, \lambda}(s_0) &= \sum_{s \in S} Pr(s_0 \rightarrow s, h, \pi) \\
&\quad \times \sum_{a \in A} \nabla_{\theta_h} \mu_h(s, a) Q_h^{\pi, \lambda}(s, a).
\end{aligned} \tag{10}$$

Now the gradient of the Lagrangian $L(\pi, \lambda)$ w.r.t θ_h can be seen to be

$$\begin{aligned}
\nabla_{\theta_h} L(\pi, \lambda) &= \sum_{s \in S} \beta(s) \nabla_{\theta_h} V_0^{\pi, \lambda}(s) \\
&= \sum_{s \in S} d_h^\pi(s) \sum_{a \in A} \nabla_{\theta_h} \mu_h(s, a) Q_h^{\pi, \lambda}(s, a).
\end{aligned} \tag{11}$$

For any baseline $b(s), s \in S$, it follows that:

$$\begin{aligned}
\sum_{s \in S} d_h^\pi(s) \sum_{a \in A} \nabla_{\theta_h} \mu_h(s, a) (Q_h^{\pi, \lambda}(s, a) - b(s)) \\
= \nabla_{\theta_h} L(\pi, \lambda).
\end{aligned}$$

We approximate all the value functions for the Lagrangian for $h = 0, 1, \dots, H$ as $V_h^{\pi, \lambda}(s) \approx v_h^{\pi, \lambda T} \phi_h(s)$ and all the value functions for the constraint costs $W_h^{\pi, (k)}(s) \approx W_h^{\pi, (k) T} \phi_h(s)$ where $v_h^{\pi, \lambda} := (v_h^{\pi, \lambda}(l), l = 1, \dots, x_h)^T$ and $w_h^{\pi, (k)} := (w_h^{\pi, (k)}(l), l = 1, \dots, x_h)^T$ are the weight vectors and $\phi_h(s) \in \mathbb{R}^{x_h}$ is the feature vector $\phi_h(s) := (\phi_h(s)(1), \dots, \phi_h(s)(x_h))^T$ associated with state s and time instant h . Note that in a given time instant h , it is not possible to reach every state in S . Hence we define the subset

$$S_h^\pi = \{s \in S | d_h^\pi(s) > 0\},$$

in other words S_h^π is the set of states that can be visited at time instant h under policy π . Since from Assumption 1, $\mu_h(s, a) > 0, \forall (s, a)$, S_h^π will be the same under any policy π . Hence we will refer to it as S_h . Let $\Phi_h \in \mathbb{R}^{|S_h| \times x_h}$ denote the feature matrix with k th column being $\phi_h(k) = (\phi_h(s)(k), s \in S_h)^T, h = 0, 1, \dots, H$.

Assumption 2: The basis functions $\{\phi_h(k), k = 1, \dots, x_h\}$ are linearly independent for $h = 0, \dots, H$. Further $x_h \leq |S_h|$, and Φ_h has full rank.

Remark 2: It appears desirable to have a higher dimension size for the feature matrix. However, one must be careful about Assumption 2, as it is crucial for the stability of the critic recursion (see Section IV). Note that Assumption 2 can be satisfied by slowly increasing the dimension size of both the weight vector and the features as more states are visited over episodes for a time instant h instead of having a large dimension right from the beginning. This is unlike the setting of infinite horizon problems, where the dimension size remains a constant (see [16]).

Note that in infinite horizon problems, under an ergodic policy, the Markov chain settles into a stationary distribution. This is however not the case with finite horizon problems where, in general, only a certain subset of the states will ever be visited depending on the starting state because of the finite nature of the horizon length and the Markov chain does not enter into a stationary distribution. Hence, the probability

Algorithm 1 Constraint Estimation using critic

Input: step-sizes

- 1: **for** each episode $n \geq 0$ **do**
 - 2: Collect the states, actions and rewards from the episode.
 - 3: **for** $h = 0, 1, \dots, H - 1$ **do**
 - 4: Perform main critic update (12)
 - 5: Perform main actor update (14)
 - 6: **end for**
 - 7: Perform main critic update (13)
 - 8: **for** $k = 1, \dots, M$ **do**
 - 9: **for** $h = 0, 1, \dots, H - 1$ **do**
 - 10: Perform constraint critic update (15)
 - 11: **end for**
 - 12: Perform constraint critic update (16)
 - 13: Perform Lagrange multiplier update (17)
 - 14: **end for**
 - 15: **end for**
-

of visit to many of the states starting from a given state is actually zero in the finite horizon setting.

III. ACTOR-CRITIC ALGORITHM FOR FINITE HORIZON CONSTRAINED MDPs

We present an algorithm based on multi-timescale stochastic approximation (MTSA) for our problem. Let $a(n), b(n), c(n), n \geq 0$, be three positive step-size schedules that satisfy the following assumption:

Assumption 3: The step-size sequences $\{a(n)\}, \{b(n)\}, \{c(n)\}$ satisfy the following properties:

$$\begin{aligned} \sum_n a(n) &= \sum_n b(n) = \sum_n c(n) = \infty, \\ \sum_n (a(n)^2 + b(n)^2 + c(n)^2) &< \infty, \\ \lim_{n \rightarrow \infty} \frac{b(n)}{a(n)} &= \lim_{n \rightarrow \infty} \frac{c(n)}{b(n)} = 0. \end{aligned}$$

Our algorithm performs updates once after each episode of length H . Note that in our setting, H is finite and deterministic. For each episode $n \geq 0$, let $\theta_h(n), h = 0, \dots, H - 1$ denote the running updates of the policy parameter θ_h and let $\lambda^{(k)}(n), k = 1, \dots, M$ denote the running updates of the Lagrange multiplier $\lambda^{(k)}$. Let $\theta(n) \triangleq (\theta_0(n), \dots, \theta_{H-1}(n))^T$ and $\lambda(n) = (\lambda^{(1)}(n), \dots, \lambda^{(M)}(n))^T$. At every instant $h = 0, 1, \dots, H - 1$, action $a_h(n)$ is sampled from the distribution $\mu_{\theta_h(n)}(s_h(n), \cdot)$ when the state is $s_h(n)$. The episode ends in state $s_H(n)$.

We define temporal difference error $\delta_h(n), h = 0, \dots, H - 1$ using the Lagrangian as the single-stage reward. The value function weights $v_h^{\pi, \lambda}$ estimated using $v_h(n), h = 0, 1, \dots, H$ are updated as in (12)-(13) while the policy parameter is updated according to (14).

$$\begin{aligned} \delta_h(n) &= c_h^{\lambda(n)}(s_h(n), a_h(n), s_{h+1}(n)) \\ &\quad + v_{h+1}(n)^T \phi_{h+1}(s_{h+1}(n)) - v_h(n)^T \phi_h(s_h(n)), \end{aligned}$$

$$v_h(n+1) = v_h(n) + a(n)\delta_h(n)\phi_h(s_h(n)), \quad (12)$$

$$\begin{aligned} v_H(n+1) &= v_H(n) + a(n)[c_H^{\lambda(n)}(s_H(n)) \\ &\quad - v_H(n)^T \phi_H(s_H(n))] \phi_H(s_H(n)). \end{aligned} \quad (13)$$

In most settings involving parameterized policies, the parameter naturally takes values in a prescribed compact set. Hence, we define a projection operator Γ , which projects the iterates to a given compact set. The projection has the additional advantage that it ensures stability of the policy iterates. Let $\psi_{\theta_h}(s, a) := \nabla_{\theta_h} \log \mu_{\theta_h}(s, a), s \in S, a \in A$ denote the compatible features (cf. [17]). The actor update for $h = 0, 1, \dots, H - 1$ is then given by

$$\begin{aligned} \theta_h(n+1) &= \Gamma[\theta_h(n) + b(n)\psi_{\theta_h(n)}(s_h(n), a_h(n)) \\ &\quad \times \delta_h(n)]. \end{aligned} \quad (14)$$

The updates are justified by the approximate policy gradient equation (22) and the statements that follow. We define the temporal difference error $\xi_h^{(k)}(n), h = 0, \dots, H - 1$ using single stage constraint cost for the k th constraint $k = 1, \dots, M$. The value function weights $w_h^{\pi, (k)}$ estimated using $w_h^{(k)}(n), h = 0, 1, \dots, H$ are updated as in (15)-(16).

$$\begin{aligned} \xi_h^{(k)}(n) &= g_h^{(k)}(s_h(n), a_h(n), s_{h+1}(n)) \\ &\quad + w_{h+1}^{(k)}(n)^T \phi_{h+1}(s_{h+1}(n)) \\ &\quad - w_h^{(k)}(n)^T \phi_h(s_h(n)), \end{aligned}$$

$$w_h^{(k)}(n+1) = w_h^{(k)}(n) + a(n)\xi_h^{(k)}(n)\phi_h(s_h(n)), \quad (15)$$

$$\begin{aligned} w_H^{(k)}(n+1) &= w_H^{(k)}(n) + a(n)[g_H^{(k)}(s_H(n)) \\ &\quad - w_H^{(k)}(n)^T \phi_H(s_H(n))] \phi_H(s_H(n)). \end{aligned} \quad (16)$$

We define a projection operator $(\cdot)^- : \mathbb{R} \rightarrow [P, 0]$ as $(x)^- = \min(0, \max(x, P))$, where $x \in \mathbb{R}$ and $- \infty < P$ is a large negative constant. We provide the Lagrange parameter update below.

$$\lambda^{(k)}(n+1) = (\lambda^{(k)}(n) - c(n)w_0^{(k)}(n)^T \phi_0(s_0(n)))^-. \quad (17)$$

IV. CONVERGENCE ANALYSIS

Both recursions (14) and (17) clearly satisfy the stability requirement, i.e., $\sup_n \|\theta_h(n)\| < \infty, h = 0, \dots, H - 1$ and $\sup_n \|\lambda^{(k)}(n)\| < \infty, k = 1, \dots, M$ almost surely, since the projection operators $\Gamma(\cdot)$ and $(\cdot)^-$ force the iterates to evolve within a compact set. Further, using arguments as in Chapter 6 of [18], one may let $\lambda(n) \equiv \lambda$ when analyzing the actor update (14) and $\theta(n) \equiv \theta$ when analyzing the critic update (12)-(13) and (15)-(16) since $c(n) = o(a(n))$ and moreover $b(n) = o(a(n))$ from Assumption 3.

Let P_h^θ be the probability transition matrix for $h = 0, 1, \dots, H - 1$ with elements $P_h^\theta(s, s') = \sum_{a \in A} \mu_{\theta_h}(s, a) p_h(s, a, s'), s \in S_h, s' \in S_{h+1}$. Let D_h^θ denote a diagonal matrix for $h = 0, 1, \dots, H$ with elements $d_h^\theta(s), s \in S_h$ along the diagonal. This is unlike the infinite horizon case, where the stationary distribution of all the states in S are taken (see [16]). Let us define the vectors

$C_H^{\theta,\lambda}$, $C_h^{\theta,\lambda}$, $G_H^{\theta,(k)}$ and $G_h^{\theta,(k)}$, $h = 0, \dots, H-1$, $k = 1, \dots, M$, as

$$C_H^{\theta,\lambda} = (c_H^\lambda(s), s \in S)^T,$$

$$C_h^{\theta,\lambda} = \left(\sum_{a \in A} \mu_{\theta_h}(s, a) \sum_{s' \in S} p_h(s, a, s') c_h^\lambda(s, a, s'), s \in S \right)^T.$$

$$G_H^{\theta,(k)} = (g_H^{(k)}(s), s \in S)^T,$$

$$G_h^{\theta,(k)} = \left(\sum_{a \in A} \mu_{\theta_h}(s, a) \sum_{s' \in S} p_h(s, a, s') g_h^{(k)}(s, a, s'), s \in S \right)^T.$$

Let us define the following points, which is used in Theorem 2:

$$\Lambda_H = (\Phi_H^T D_H^\theta \Phi)^{-1} \Phi_H^T D_H^\theta C_H^{\theta,\lambda},$$

$$\Lambda_h = (\Phi_h^T D_h^\theta \Phi_h)^{-1} (\Phi_h^T D_h^\theta P_h^\theta \Phi_{h+1} \Lambda_{h+1} + \Phi_h^T D_h^\theta C_h^{\theta,\lambda}),$$

$$\Xi_H^{(k)} = (\Phi_H^T D_H^\theta \Phi_H)^{-1} \Phi_H^T D_H^\theta G_H^{\theta,(k)},$$

$$\Xi_h^{(k)} = (\Phi_h^T D_h^\theta \Phi_h)^{-1} (\Phi_h^T D_h^\theta P_h^\theta \Phi_{h+1} \Xi_{h+1}^{(k)} + \Phi_h^T D_h^\theta G_h^{\theta,(k)}).$$

Theorem 2: For $\lambda(n) \equiv \lambda$ and $\theta(n) \equiv \theta$, $v_h(n)$, $h = 0, \dots, H-1$ and $v_H(n)$ converge to points Λ_h , $h = 0, \dots, H-1$ and Λ_H respectively. Similarly $w_h^{(k)}(n)$, $h = 0, \dots, H-1$ and $w_H^{(k)}(n)$ converge to points $\Xi_h^{(k)}$, $h = 0, \dots, H-1$ and $\Xi_H^{(k)}$, respectively, for $k = 1, \dots, M$ almost surely. Also the points Λ_h , $h = 0, \dots, H-1$, Λ_H , $\Xi_h^{(k)}$, $h = 0, \dots, H-1$ and $\Xi_H^{(k)}$ for $k = 1, \dots, M$ are Lipschitz continuous in θ and λ .

Proof: We prove the convergence of $v_h(n)$, $h = 0, \dots, H-1$ and $v_H(n)$. The convergence of $w_h^{(k)}(n)$, $h = 0, \dots, H-1$ and $w_H^{(k)}(n)$ for $k = 1, \dots, M$ follows similarly. The recursions (12)-(13) can be written as,

$$v_h(n+1) = v_h(n) + a(n) [f_h(v_h(n), v_{h+1}(n)) + N_h^1(n+1)], \quad (18)$$

$$v_H(n+1) = v_H(n) + a(n) [f_H(v_H(n)) + N_H^1(n+1)] \quad (19)$$

$h = 0, 1, \dots, H-1$. In the above,

$$f_h(v_h, v_{h+1}) = \sum_{s \in S} d_h^\theta(s) \sum_{a \in A} \mu_{\theta_h}(s, a) \sum_{s' \in S} p_h(s, a, s') \times [c_h^\lambda(s, a, s') + v_{h+1}^T \phi_h(s') - v_h^T \phi_h(s)] \phi_h(s),$$

$$f_H(v_H, \theta) = \sum_{s \in S} d_H^\theta(s) [c_H^\lambda(s) - v_H^T \phi_H(s)] \phi_H(s),$$

The noise terms can be derived from above. f_h is linear in v_h and v_{h+1} and f_H is linear in v_H , therefore they are Lipschitz continuous. Let $\mathcal{F}(n)$, $n \geq 0$ be the increasing sequence of sigma fields

$$\mathcal{F}(n) = \sigma(v_h(m), v_H(m), N_h^1(m), N_H^1(m), m \leq n, h = 0, 1, \dots, H-1).$$

For some $0 < C < \infty$, it is easy to see from Assumption 2, $\mathbb{E} [\|N_h^1(n+1)\|^2 | \mathcal{F}(n)] \leq C(1 + \|v_h(n)\|^2 + \|v_{h+1}(n)\|^2)$, $h = 0, 1, \dots, H-1$,

$$\mathbb{E} [\|N_H^1(n+1)\|^2 | \mathcal{F}(n)] \leq C(1 + \|v_H(n)\|^2).$$

Let us define:

$$f_h^\infty(v_h, v_{h+1}) = \lim_{c \rightarrow \infty} \frac{f_h(cv_h, cv_{h+1})}{c} = \Phi_h^T D_h^\theta P_h^\theta \Phi_{h+1} v_{h+1} - \Phi_h^T D_h^\theta \Phi_h v_h, \quad (20)$$

$$f_H^\infty(v_H) = \lim_{c \rightarrow \infty} \frac{f_H(cv_H)}{c} = -\Phi_H^T D_H^\theta \Phi_H v_H. \quad (21)$$

Clearly D_h^θ is positive definite. Since from Assumption 2, Φ_h has full rank then $-\Phi_h^T D_h^\theta \Phi_h$ is negative definite for $h = 0, \dots, H$. Hence the ODE $\dot{v}_H(t) = f_H^\infty(v_H(t))$ has the origin in \mathbb{R}^{x_H} as its unique asymptotically stable equilibrium. Now plugging $v_{h+1}(t) = 0$, one can see going backwards $h = H-1, \dots, 0$, each ODE $\dot{v}_h(t) = f_h^\infty(v_h(t), v_{h+1}(t))$ has the origin in \mathbb{R}^{x_h} as its unique asymptotically stable equilibrium. Hence (A1) and (A2) of [19] are satisfied and the first claim follows from Theorem 2.2 of [19]. The iterates $v_h(n)$ converges where the approximate TD error f_h is 0, for $h = 0, \dots, H$.

From Assumption 1, D_h^θ and $C_h^{\theta,\lambda}$ are continuously differentiable in θ and $C_h^{\theta,\lambda}$ is linear in λ for $h = 0, \dots, H$. Since θ and λ lie in a compact space, $d_h^\theta(s) \geq B_1$, $B_1 > 0$, therefore $\nabla_\theta(\Phi_h^T D_h^\theta \Phi_h)^{-1}$, $\nabla_\theta D_h^\theta$ and $\nabla_\theta C_h^{\theta,\lambda}$ are bounded. Therefore, $\nabla_\theta \Lambda_h$ is bounded. Similarly one can show $\nabla_\theta \Xi_h$ is bounded for $h = 0, \dots, H$. The second claim follows. ■ Let the approximate policy gradient be defined as,

$$g_h(\theta) = \sum_{s \in S} d_h^\theta(s) \sum_{a \in A} \mu_{\theta_h}(s, a) \nabla_{\theta_h} \log \mu_{\theta_h}(s, a) \times \sum_{s' \in S} p_h(s, a, s') (c_h^\lambda(s, a, s') + \Lambda_{h+1}^T \phi_{s'} - \Lambda_h^T \phi_s), \quad (22)$$

This is the gradient given in (11), with baseline $b(s) := V_h^{\pi,\lambda}(s)$, $s \in S$, baring that approximate value functions $V_h^{\pi,\lambda}(s) \approx \Lambda_h^T \phi_h(s)$ are used here. The directional derivative of $\Gamma(\cdot)$ at point x , along the vector y is defined as,

$$\Gamma(x, y)' = \lim_{\eta \downarrow 0} \frac{\Gamma(x + \eta y) - x}{\eta} \quad (23)$$

Let us define the set: $\kappa_h = \{\theta \in \mathbb{R}^{y_h} | \Gamma(\theta, g_h(\theta))' = 0\}$ for $h = 0, \dots, H-1$. This is the set of θ such that the approximate policy gradient is 0, which means θ is at local optimum.

Theorem 3: For $\lambda(n) \equiv \lambda$, $\theta_h(n)$, $h = 0, \dots, H-1$ converges to a set κ_h almost surely.

Proof: The recursion (14) can be written as,

$$\theta_h(n+1) = \Gamma(\theta_h(n) + b(n)(g_h(\theta(n)) + N_h^2(n+1) + e^{\theta(n)})), \quad (24)$$

where,

$$N_h^2(n+1) = \nabla_{\theta_h} \log \mu_{\theta_h}(s, a) (c_h^\lambda(s, a, s') + \Lambda_{h+1}^T \phi_{s'} - \Lambda_h^T \phi_s) - g_h(\theta(n)).$$

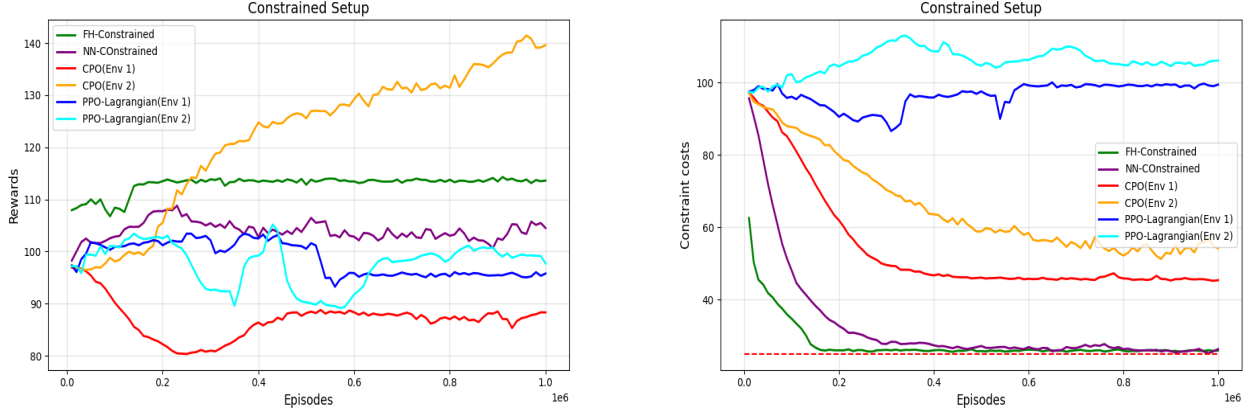


Fig. 1. Comparison of rewards and constraint costs achieved by different algorithms.

The error $e^{\theta(n)}$ can be derived from above and $e^{\theta(n)} \rightarrow 0$ as $n \rightarrow \infty$ (from Theorem 2). It is easy to see $N_h^2(n+1)$ is uniformly bounded, since Λ_h and Λ_{h+1} is uniformly bounded from Theorem 2. Therefore the martingale $M_1(n) = \sum_{r=0}^{n-1} b(r)N_h^2(r+1)$ converges almost surely (see Appendix C of [18]). Therefore the tail sum $M_1' = \sum_{r=n}^{\infty} b(r)N_h^2(r+1) \rightarrow 0$ as $n \rightarrow \infty$. Now from Theorem 2, Λ_h and Λ_{h+1} are both Lipschitz continuous in θ . Since θ lies in a compact space, from Assumption 1, $\nabla_{\theta}^2 \mu_{\theta_h}(s, a)$ is bounded. Therefore from Theorem 5.3.1 of [20], θ_h converges to the set κ_h . ■

Let $(\cdot)^{-'}$ be the directional derivative of $(\cdot)^{-}$ defined as (23). Let us define the set: $F = \{\lambda \in \mathbb{R}^M \mid (\lambda, \sum_{s_0 \in S} \beta(s_0) \Xi_0^{(k)T} \phi_0(s_0))^{-'} = 0, \forall k = 1, \dots, M\}$. This set contains all the Lagrange parameter λ for which the approximate constraint violation is 0. Note that the point $\Xi_0^{(k)}$ implicitly depends on λ .

Theorem 4: As $n \rightarrow \infty$, $\lambda(n) = (\lambda^{(1)}(n), \dots, \lambda^{(M)}(n))^T$ obtained from (17) converges to a set F almost surely.

Proof: The Lagrange parameter update (17) can be rewritten as,

$$\lambda^{(k)}(n+1) = (\lambda^{(k)}(n) - c(n)(\Xi_0^{(k)T} \phi_0(s_0(n)) + \gamma^{(k)}(n))^{-}) \quad (25)$$

where $\gamma^{(k)}(n)$ can be derived from above and $\gamma^{(k)}(n) \rightarrow 0$ as $n \rightarrow \infty$ (from Theorem 2). Thus (25) is an Euler discretization with (non-uniform) step-sizes $c(n)$ of the ODE,

$$\dot{\lambda}^{(k)}(t) = \left(\sum_{s_0 \in S} \beta(s_0) \Xi_0^{(k)T} \phi_0(s_0) \right)^{-'}$$

Now from an application of Theorem 2, Chapter 2 of [18] (see the third extension described on pp. 17, Chapter 2, of [18]), $\lambda(n)$ converges to F . ■

Hence by Theorem 2-4, the algorithm converges asymptotically.

V. EXPERIMENTS

Now that we have proved the convergence of our algorithm, we study the empirical performance of our algorithm on a randomly generated 2-dimensional Grid World Problem. We also introduce a deep learning version of our algorithm, where the value and policy function are parameterized using a neural network architecture. Along each dimension, at any instant, the agent can go one step forward or one step back, or else stay in the same place. So for a 2-dimensional grid, there are available $3 \times 3 = 9$ actions in each state. As far as state transitions are concerned, there is a 90% chance that the agent will go one step in the direction suggested by the action and a 10% chance that it will randomly move one step in any of the other directions. The agent collects a reward if it goes to a particular state. We have a single constraint. There are bad states, which the agent must avoid and there is a constraint cost associated with those states. The position of the rewards and constraints change with time, so the agent needs to make decisions at each time instant accordingly. This kind of setup can be applied in applications such as taxi services, where a driver needs to maximize his/her fare in a day comprising of say 10 hours and the grid is of the city in which he/she is driving. The rewards would then correspond to the fares and the constraints could be on the level of traffic congestion encountered on the different routes. The congestion levels are clearly dynamic in nature as they would naturally change with time.

We let a horizon length of $H = 100$ for our experiments. Because of our constrained setup, our objective is to maximize the rewards while satisfying the inequality constraint. We name our finite horizon constrained algorithms as FH-Constrained. We also name our corresponding deep learning algorithms as NN-Constrained. We compare our algorithm with Constrained Policy Optimization (CPO) [1] and PPO-Lagrangian [21], which are the best known algorithms for Constrained RL. Each comparison is done on two environments, one with 2-dimensional state space as input and another with 3-dimensional input (two dimensions for the state variable and one dimension for the time instant). The

first plot in Fig.1 is for the aggregate reward while the second is for the constraint cost. We take the total reward/constraint cost of each episode and average the same over the last 10,000 episodes. Each setting is run 5 times with independent seeds. The red dotted line in the second plot denotes the constraint threshold of $\alpha = 25$.

We can see that our algorithm performs better than both CPO and PPO-Lagrangian as it nearly satisfies the constraint every time while others do not. Also note that CPO (Env 2) gets more rewards than our algorithms but that comes at a cost of it violating the constraints by a wide margin. It is clear from the experiments that constrained algorithms for infinite horizon problems are not appropriate for finite horizon settings. Also feeding the time instant with state as input does not help in these algorithms. In fact, they perform poorly when the reward and the constraint cost change with time, unlike our algorithm that easily adapts to such changes.

VI. CONCLUSIONS

We presented the first policy gradient reinforcement learning algorithm for Finite Horizon Constrained MDPs. One must adhere to our algorithm when the agent needs to take time critical decisions, which is evident from our empirical comparisons with other well known algorithms. Our algorithm involves three timescale schedules and is of the actor-critic type. We provided the full asymptotic convergence analysis for our algorithm. The power of our algorithm comes from using separate parametric functions for each time instant, and our convergence analysis shows that they can successfully interact with each other to learn a time critical task. It will be of interest to analyze the sample complexity of our constrained three-timescale finite horizon MDP algorithm. Further, it will be interesting to perform a finite sample analysis of Actor critic algorithms for Finite Horizon CMDPs as future work. One may also come up with sophisticated policy optimization methods such as CPO [1], for finite horizon MDPs.

REFERENCES

- [1] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 22–31.
- [2] J. Garcia and F. Fernández, "A comprehensive survey on safe reinforcement learning," *Journal of Machine Learning Research*, vol. 16, no. 1, pp. 1437–1480, 2015.
- [3] E. Altman, *Constrained Markov Decision Processes*. CRC Press, 1999, vol. 7.
- [4] P. Geibel, "Reinforcement learning for MDPs with constraints," in *Machine Learning: ECML 2006*, J. Fürnkranz, T. Scheffer, and M. Spiliopoulou, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 646–653.
- [5] V. Borkar, "An actor-critic algorithm for constrained Markov decision processes," *Systems & Control Letters*, vol. 54, no. 3, pp. 207–213, 2005. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0167691104001276>
- [6] S. Bhatnagar, "An actor-critic algorithm with function approximation for discounted cost constrained Markov decision processes," *Systems & Control Letters*, vol. 59, no. 12, pp. 760–766, 2010.

- [7] S. Bhatnagar and K. Lakshmanan, "An online actor-critic algorithm with function approximation for constrained Markov decision processes," *Journal of Optimization Theory and Applications*, vol. 153, no. 3, pp. 688–708, Jun 2012.
- [8] D. Bertsekas, *Dynamic Programming and Optimal Control, Vol. I*. Athena Scientific, 2005.
- [9] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*, 1st ed. Athena Scientific, 1996.
- [10] F. Garcia and S. M. Ndiaye, "A learning rate analysis of reinforcement learning algorithms in finite-horizon," in *Proceedings of the 15th International Conference on Machine Learning*. Morgan Kaufmann, 1998, pp. 215–223.
- [11] S. Bhatnagar and M. S. Abdulla, "Simulation-based optimization algorithms for finite-horizon Markov decision processes," *SIMULATION*, vol. 84, no. 12, pp. 577–600, 2008.
- [12] K. De Asis, A. Chan, S. Pitis, R. Sutton, and D. Graves, "Fixed-horizon temporal difference methods for stable reinforcement learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, pp. 3741–3748, Apr. 2020.
- [13] Z. Jiang, D. Xu, and J. Liang, "A deep reinforcement learning framework for the financial portfolio management problem," 2017.
- [14] C. Lakshminarayanan and S. Bhatnagar, "A stability criterion for two timescale stochastic approximation schemes," *Automatica*, vol. 79, pp. 108–114, 2017.
- [15] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: The MIT Press, 2018.
- [16] J. Tsitsiklis and B. Van Roy, "An analysis of temporal-difference learning with function approximation," *IEEE Transactions on Automatic Control*, vol. 42, no. 5, pp. 674–690, 1997.
- [17] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in Neural Information Processing Systems*, S. Solla, T. Leen, and K. Müller, Eds., vol. 12. MIT Press, 1999.
- [18] V. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press, 2008.
- [19] V. Borkar and S. P. Meyn, "The o.d.e. method for convergence of stochastic approximation and reinforcement learning," *SIAM J. Control. Optim.*, vol. 38, pp. 447–469, 2000.
- [20] H. J. Kushner and D. S. Clark, *Stochastic approximation methods for constrained and unconstrained systems*. Springer Science & Business Media, 2012, vol. 26.
- [21] J. Achiam and D. Amodei, "Benchmarking safe exploration in deep reinforcement learning," 2019.