Regret Analysis of Policy Gradient Algorithm for Infinite Horizon Average Reward Markov Decision Processes

Qinbo Bai*, Washim Uddin Mondal*, Vaneet Aggarwal

Purdue University, USA
{bai113, wmondal, vaneet}@purdue.edu

Abstract

In this paper, we consider an infinite horizon average reward Markov Decision Process (MDP). Distinguishing itself from existing works within this context, our approach harnesses the power of the general policy gradient-based algorithm, liberating it from the constraints of assuming a linear MDP structure. We propose a policy gradient-based algorithm and show its global convergence property. We then prove that the proposed algorithm has $\tilde{\mathcal{O}}(T^{3/4})$ regret. Remarkably, this paper marks a pioneering effort by presenting the first exploration into regret-bound computation for the general parameterized policy gradient algorithm in the context of average reward scenarios.

Introduction

Reinforcement Learning (RL) describes a class of problems where a learner repeatedly interacts with an unknown environment with the intention of maximizing the cumulative sum of rewards. This model has found its application in a wide array of areas, ranging from networking to transportation to epidemic control (Geng et al. 2020; Al-Abbasi, Ghosh, and Aggarwal 2019; Ling, Mondal, and Ukkusuri 2023). RL problems are typically analysed via three distinct setups − episodic, infinite horizon discounted reward, and infinite horizon average reward. Among these, the infinite horizon average reward setup holds particular significance in real-world applications (including those mentioned above) due to its alignment with many practical scenarios and its ability to capture essential long-term behaviors. However, scalable algorithms in this setup have not been widely studied. This paper provides the first algorithm in the infinite horizon average reward setup with general parametrization (which helps scale this to large state spaces), for which sublinear regret guarantees are provided.

There are two major approaches to solving an RL problem. The first one, known as the model-based approach, involves constructing an estimate of the transition probabilities of the underlying Markov Decision Process (MDP). This estimate is subsequently leveraged to derive policies (Auer, Jaksch, and Ortner 2008; Agrawal and Jia 2017; Ouyang et al. 2017; Fruit et al. 2018). It is worth noting that model-based techniques encounter a significant challenge – these algorithms demand a substantial memory to house the model parameters. Consequently, their practical application is hindered when dealing with large state spaces. An alternative strategy is referred to as model-free algorithms. These methods either directly estimate the policy function or maintain an estimate of the Q function, which are subsequently employed for policy generation (Mnih et al. 2015; Schulman et al. 2015; Mnih et al. 2016). The advantage of these algorithms lies in their adaptability to handle large state spaces.

In the average reward MDP, which is the setting considered in our paper, one of the key performance indicators of an algorithm is the expected regret. It has been theoretically demonstrated in (Auer, Jaksch, and Ortner 2008) that the expected regret of any algorithm for a broad class of MDPs is lower bounded by $\Omega(\sqrt{T})$ where T denotes the length of the time horizon. Many model-based algorithms, such as, (Auer, Jaksch, and Ortner 2008; Agrawal and Jia 2017) achieve this bound. However, these algorithms are applicable for the tabular setup, and hence not practical for large state spaces. Recently, (Wei et al. 2021) proposed a model-based algorithm for the linear MDP setup that is shown to achieve the optimal regret bound. On the other hand, (Wei et al. 2020) proposed a model-free Q-estimation-based algorithm that achieves the optimal regret in the tabular setup.

One way to allow algorithms to handle large state space is via policy parameterization. Here, the policies are indexed by parameters (via, for example, neural networks), and the learning process is manifested by updating these parameters using some update rule (such as gradient descent). Such algorithms are referred to as policy gradient (PG) algorithms. Interestingly, the analysis of PG algorithms is typically restricted within the discounted reward setup. For example, (Agarwal et al. 2021) characterized the sample complexity of PG and Natural PG (NPG) with softmax and tabular parameterization. Sample complexity results for general parameterization are given by (Liu et al. 2020; Ding et al. 2020). However, the sub-linear regret analysis of a PG-based algorithm with general parameterization in the average reward setup, to the best of our knowledge, has not been studied in the literature. This paper aims to bridge this gap by addressing this crucial problem.

*These authors contributed equally.

Challenges and Contribution

We propose a PG-based algorithm with general parameterization in the average reward setup and establish a sublinear regret of the proposed algorithm. In particular, within the class of ergodic MDPs, we first show that our PG-based algorithm achieves an average optimality error of $\tilde{\mathcal{O}}(T^{-\frac{1}{4}})$. Utilizing this convergence result, we establish that the algorithm achieves a regret bound of $\tilde{O}(T^{3/4})$. The regret does not depend on the size of the state space, indicating that the result is also applicable for large state space.

Despite the availability of sample complexity analysis of PG algorithms in the discounted reward setup, obtaining a sublinear regret bound for their average reward counterpart is quite difficult. This is because in the average reward case, the value function estimators, which are crucial to estimate the gradient, can become unbounded, unlike their discounted reward counterparts. Indeed, the sample complexity results in the discounted MDPs are often associated with a $\frac{1}{1-\gamma}$ factor (where γ denotes the discount factor which is 1 for the average reward case), indicating that a naive adaptation of these estimators will not work for the average reward setup. Also, discounted setups typically assume access to a simulator to generate unbiased value estimates. On the contrary, our paper deals with a single sample trajectory and does not assume the availability of a simulator. To obtain a good estimator of the gradient, we design an epoch-based algorithm where the length of each epoch is H. The algorithm estimates the value functions within a given epoch by sampling rewards of sub-trajectories of length N that are at least N distance apart. The separation between these sub-trajectories ensures that their reward samples are sufficiently independent. The key challenge of this paper is to bound a second-order term which is related to the variance of the estimated gradient and the true gradient. We show that by judiciously controlling the growth rate of H and N with T, it is possible to obtain a gradient estimator that has an asymptotically decreasing variance.

Related Works

As discussed in the introduction, the reinforcement learning problem has been widely studied recently for infinite horizon discounted reward cases or the episodic setting. For example, (Jin et al. 2018) proposed the model-free UCB-Q learning and showed a $\mathcal{O}(\sqrt{T})$ regret in the episodic setting. In the discounted reward setting, (Ding et al. 2020) achieved $\mathcal{O}(\epsilon^{-2})$ sample complexity for the softmax parametrization using the Natural Policy Gradient algorithm whereas (Mondal and Aggarwal 2023; Fatkhullin et al. 2023) exhibited the same complexity for the general parameterization. However, the regret analysis or the global convergence of the average reward infinite horizon case is much less investigated.

For infinite horizon average reward MDPs, (Auer, Jaksch, and Ortner 2008) proposed a model-based Upper confidence Reinforcement learning (UCRL2) algorithm and established that it obeys a $\tilde{\mathcal{O}}(\sqrt{T})$ regret bound. (Agrawal and Jia 2017) proposed posterior sampling-based approaches for average reward MDPs. (Wei et al. 2020) proposed the optimistic-Q learning algorithm which connects the discounted reward and average reward setting together to show $\mathcal{O}(T^{3/4})$ regret in weakly communicating average reward case and another online mirror descent algorithm which achieves $\mathcal{O}(\sqrt{T})$ regret in the ergodic setting. For the linear MDP setting, (Wei et al. 2021) proposed three algorithms, including the MDP-EXP2 algorithm which achieves $O(\sqrt{T})$ regret under the ergodicity assumption. These works have been summarized in Table 1. We note that the assumption of weakly communicating MDP is the minimum assumption needed to have sublinear regret results. However, it is much more challenging to work with this assumption in the general parametrized setting because of the following reasons. Firstly, there is no guarantee that the state distribution will converge to the steady distribution exponentially fast which is the required property to show that the value functions are bounded by the mixing time. Secondly, it is unclear how to obtain an asymptotically unbiased estimate of the policy gradient. Thus, we assume ergodic MDP in this work following other works in the literature (Pesquerel and Maillard 2022; Gong and Wang 2020). MDPs with constraints have also been recently studied for model-based (Agarwal, Bai, and Aggarwal 2022b,a), model-free tabular (Wei, Liu, and Ying 2022; Chen, Jain, and Luo 2022), and linear MDP setup (Ghosh, Zhou, and Shroff 2022).

However, all of the above algorithms are designed for the tabular setting or the linear MDP assumption, and none of them uses a PG algorithm with the general parametrization setting. In this paper, we propose a PG algorithm for ergodic MDPs with general parametrization and analyze its regret. Our algorithm can be applied to large state space even without assuming the linear MDP structure.

Formulation

In this paper, we consider an infinite horizon reinforcement learning problem with an average reward criterion, which is modeled by the Markov Decision Process (MDP) written as a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, r, P, \rho)$ where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space of size A, $r : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is the reward function, $P : \mathcal{S} \times \mathcal{A} \to \Delta^{|\mathcal{S}|}$ is the state transition function where $\Delta^{|\mathcal{S}|}$ denotes the probability simplex with dimension $|\mathcal{S}|$, and $\rho : \mathcal{S} \to [0, 1]$ is the initial distribution of states. A policy $\pi : \mathcal{S} \to \Delta^{|\mathcal{A}|}$ decides the distribution of the action to be taken given the current state. For a given policy, π we define the long-term average reward as follows.

$$J_\rho^\pi \triangleq \lim_{T\to\infty} \frac{1}{T}\mathbf{E}\left[\sum_{t=0}^{T-1} r(s_t, a_t)\middle| s_0 \sim \rho\right] \tag{1}$$

where the expectation is taken over all state-action trajectories that are generated by following the action execution process, $a_t \sim \pi(\cdot|s_t)$ and the state transition rule, $s_{t+1} \sim P(\cdot|s_t, a_t), \forall t \in \{0, 1, \cdots\}$. To simplify notations, we shall drop the dependence on ρ whenever there is no confusion. We consider a parametrized class of policies, Π whose each element is indexed by a d-dimensional parameter, $\theta \in \Theta$ where $\Theta \subset \mathbb{R}^{\mathrm{d}}$. Our goal is to solve the following optimization problem.

$$\max_{\theta\in\Theta} J^{\pi_\theta} \triangleq J(\theta) \tag{2}$$

Algorithm	Regret	Ergodic	Model-free	Setting
UCRL2 (Auer, Jaksch, and Ortner 2008)	$\tilde{O}\left(DS\sqrt{AT}\right)$	No	No	Tabular
PSRL (Agrawal and Jia 2017)	$\tilde{O}\left(DS\sqrt{AT}\right)$	No	No	Tabular
OPTIMISTIC Q-LEARNING (Wei et al. 2020)	$\tilde{O}\left(T^{2/3}\right)$	No	Yes	Tabular
MDP-OOMD (Wei et al. 2020)	$\tilde{O}\left(\sqrt{T}\right)$	Yes	Yes	Tabular
FOPO (Wei et al. 2021)[1]	$\tilde{O}\left(\sqrt{T}\right)$	No	No	Linear MDP
OLSVI.FH (Wei et al. 2021)	$\tilde{O}\left(T^{3/4}\right)$	No	No	Linear MDP
MDP-EXP2 (Wei et al. 2021)	$\tilde{O}\left(\sqrt{T}\right)$	Yes	No	Linear MDP
This paper	$\tilde{O}\left(T^{\frac{3}{4}}\right)$	Yes	Yes	General parametrization
Lower bound (Auer, Jaksch, and Ortner 2008)	$\Omega\left(\sqrt{DSAT}\right)$	N/A	N/A	N/A

Table 1: This table summarizes the different model-based and mode-free state-of-the-art algorithms available in the literature for average reward MDPs. We note that the proposed algorithm is the first paper to analyze the regret for average reward MDP with general parametrization.

A policy π_θ induces a transition function $P^{\pi_\theta} : \mathcal{S} \to \Delta^{|\mathcal{S}|}$ as $P^{\pi_\theta}(s, s') = \sum_{a\in\mathcal{A}} P(s'|s, a)\pi_\theta(a|s)$, $\forall s, s' \in \mathcal{S}$. If $\mathcal{M}$ is such that for every policy π, the induced function, P^π is irreducible, and aperiodic, then $\mathcal{M}$ is called ergodic.

Assumption 1. *The MDP $\mathcal{M}$ is ergodic.*

Ergodicity is commonly applied in the analysis of MDPs (Pesquerel and Maillard 2022; Gong and Wang 2020). It is well known that if $\mathcal{M}$ is ergodic, then $\forall\theta \in \Theta$, there exists a unique stationary distribution, $d^{\pi_\theta} \in \Delta^{|\mathcal{S}|}$ defined as,

$$d^{\pi_\theta}(s) = \lim_{T\to\infty} \frac{1}{T}\left[\sum_{t=0}^{T-1} \Pr(s_t = s|s_0 \sim \rho, \pi_\theta)\right] \tag{3}$$

Note that under the assumption of ergodicity, d^{π_θ} is independent of the initial distribution, ρ, and satisfies $P^{\pi_\theta} d^{\pi_\theta} = d^{\pi_\theta}$. In this case, we can write the average reward as follows.

$$\begin{aligned} &J(\theta) = \mathbf{E}_{s\sim d^{\pi_\theta}, a\sim\pi_\theta(\cdot|s)}[r(s, a)] = (d^{\pi_\theta})^T r^{\pi_\theta} \\ &\text{where } r^{\pi_\theta}(s) \triangleq \sum_{a\in\mathcal{A}} r(s, a)\pi_\theta(a|s),\ \forall s \in \mathcal{S} \end{aligned} \tag{4}$$

Hence, the average reward $J(\theta)$ is also independent of the initial distribution, ρ. Furthermore, $\forall\theta \in \Theta$, there exist a function $Q^{\pi_\theta} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ such that the following Bellman equation is satisfied $\forall(s, a) \in \mathcal{S} \times \mathcal{A}$.

$$Q^{\pi_\theta}(s, a) = r(s, a) - J(\theta) + \mathbf{E}_{s'\sim P(\cdot|s,a)}\left[V^{\pi_\theta}(s')\right] \tag{5}$$

where the state value function, $V^{\pi_\theta} : \mathcal{S} \to \mathbb{R}$ is defined as,

$$V^{\pi_\theta}(s) = \sum_{a\in\mathcal{A}} \pi_\theta(a|s)Q^{\pi_\theta}(s, a),\ \forall s \in \mathcal{S} \tag{6}$$

Note that if (5) is satisfied by Q^{π_θ}, then it is also satisfied by $Q^{\pi\theta} + c$ for any arbitrary constant, c. To define these functions uniquely, we assume that $\sum_{s\in\mathcal{S}} d^{\pi_\theta}(s)V^{\pi_\theta}(s) = 0$. In this case, $V^{\pi_\theta}(s)$ can be written as follows $\forall s \in \mathcal{S}$.

$$\begin{aligned} V^{\pi_\theta}(s) &= \sum_{t=0}^{\infty}\sum_{s'\in\mathcal{S}} \left[(P^{\pi_\theta})^t(s, s') - d^{\pi_\theta}(s')\right] r^{\pi_\theta}(s') \\ &= \mathbf{E}_\theta\left[\sum_{t=0}^{\infty} r(s_t, a_t) - J(\theta)\middle| s_0 = s\right] \end{aligned} \tag{7}$$

where $\mathbf{E}_\theta[\cdot]$ denotes expectation over all trajectories induced by the policy π_θ. Similarly, $\forall(s, a) \in \mathcal{S} \times \mathcal{A}$, $Q^{\pi_\theta}(s, a)$ can be uniquely written as,

$$Q^{\pi_\theta}(s, a) = \mathbf{E}_\theta\left[\sum_{t=0}^{\infty} r(s_t, a_t) - J(\theta)\middle| s_0 = s, a_0 = a\right] \tag{8}$$

Additionally, we define the advantage function $A^{\pi_\theta} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ such that $\forall(s, a) \in \mathcal{S} \times \mathcal{A}$,

$$A^{\pi_\theta}(s, a) \triangleq Q^{\pi_\theta}(s, a) - V^{\pi_\theta}(s) \tag{9}$$

Ergodicity also implies the existence of a finite mixing time. In particular, if $\mathcal{M}$ is ergodic, then the mixing time is defined as follows.

Definition 1. *The mixing time of an MDP $\mathcal{M}$ with respect to a policy parameter θ is defined as,*

$$t^\theta_{\text{mix}} \triangleq \min\left\{t \geq 1\middle|\ \left\|(P^{\pi_\theta})^t(s, \cdot) - d^{\pi_\theta}\right\| \leq \frac{1}{4}, \forall s \in \mathcal{S}\right\} \tag{10}$$

We also define $t_{\text{mix}} \triangleq \sup_{\theta\in\Theta} t^\theta_{\text{mix}}$ as the the overall mixing time. In this paper, t_{mix} is finite due to ergodicity.

Mixing time is a measure of how fast the MDP reaches close to its stationary distribution if the same policy is kept on being executed repeatedly. We also define the hitting time as follows.

[1] FOPO is computationally inefficient.

Definition 2. *The hitting time of an MDP $\mathcal{M}$ with respect to a policy parameter, θ is defined as,*

$$t_{\text{hit}}^{\theta} \triangleq \max_{s \in \mathcal{S}} \frac{1}{d^{\pi_\theta}(s)} \tag{11}$$

We also define $t_{\text{hit}} \triangleq \sup_{\theta \in \Theta} t_{\text{hit}}^{\theta}$ as the the overall hitting time. In this paper, t_{hit} is finite due to ergodicity.

Let, $J^* \triangleq \sup_{\boldsymbol{\theta} \in \Theta} J(\theta)$. For a given MDP $\mathcal{M}$ and a time horizon T, the regret of an algorithm $\mathbb{A}$ is defined as follows.

$$\text{Reg}_T(\mathbb{A}, \mathcal{M}) \triangleq \sum_{t=0}^{T-1} (J^* - r(s_t, a_t)) \tag{12}$$

where the action, a_t, $t \in \{0, 1, \cdots\}$ is chosen by following the algorithm, $\mathbb{A}$ based on the trajectory up to time, t, and the state, s_{t+1} is obtained by following the state transition function, P. Wherever there is no confusion, we shall simplify the notation of regret to Reg_T. The goal of maximizing $J(\cdot)$ can be accomplished by designing an algorithm that minimizes the regret.

Algorithm

In this section, we discuss a policy-gradient-based algorithm in the average reward RL settings. For simplicity, we assume that the set of all policy parameters is $\Theta = \mathbb{R}^{\text{d}}$. The standard policy gradient algorithm iterates the policy parameter θ as follows $\forall k \in \{1, 2, \cdots\}$ starting with an initial guess θ_1.

$$\theta_{k+1} = \theta_k + \alpha \nabla_\theta J(\theta_k) \tag{13}$$

where α is the parameter learning rate. The following result is well-known in the literature (Sutton et al. 1999).

Lemma 1. *The gradient of the long-term average reward can be expressed as follows.*

$$\nabla_\theta J(\theta) = \mathbf{E}_{s \sim d^{\pi_\theta}, a \sim \pi_\theta(\cdot|s)} \left[A^{\pi_\theta}(s, a) \nabla_\theta \log \pi_\theta(a|s) \right] \tag{14}$$

Typically we have access neither to P, the state transition function to compute the required expectation nor to the functions $V^{\pi_\theta}, Q^{\pi_\theta}$. In the absence of this knowledge, computation of gradient, therefore, becomes a difficult job. In the subsequent discussion, we shall demonstrate how the gradient can be estimated using sampled trajectories. Our policy gradient-based algorithm is described in Algorithm 1.

The algorithm proceeds in multiple epochs with the length of each epoch being $H = 16 t_{\text{hit}} t_{\text{mix}} \sqrt{T} (\log T)^2$. Observe that the algorithm is assumed to be aware of T. This assumption can be easily relaxed invoking the well-known doubling trick (Lattimore and Szepesvári 2020). We also assume that the values of t_{mix}, and t_{hit} are known to the algorithm. Similar presumptions have been used in the previous literature (Wei et al. 2020). In the kth epoch, the algorithm generates a trajectory of length H, denoted as $\mathcal{T}_k = \{(s_t, a_t)\}_{t=(k-1)H}^{kH-1}$, by following the policy π_{θ_k}. We utilise the policy parameter θ_k and the trajectory $\mathcal{T}_k$ in Algorithm 2 to compute the estimates $\hat{V}^{\pi_{\theta_k}}(s)$, and $\hat{Q}^{\pi_{\theta_k}}(s, a)$ for a given state-action pair (s, a). The algorithm searches the trajectory $\mathcal{T}_k$ to locate disjoint sub-trajectories of length $N = 4 t_{\text{mix}} (\log T)$ that start with the given state s and are at least N distance apart. Let i be the number of such sub-trajectories and the sum of rewards in the jth such sub-trajectory be y_j. Then $\hat{V}^{\pi_{\theta_k}}(s)$ is computed as,

$$\hat{V}^{\pi_{\theta_k}}(s) = \frac{1}{i} \sum_{j=1}^{i} y_j \tag{15}$$

The sub-trajectories are kept at least N distance apart to ensure that the samples $\{y_j\}_{j=1}^{i}$ are fairly independent. The estimate $\hat{Q}^{\pi_\theta}(s, a)$, on the other hand, is given as,

$$\hat{Q}^{\pi_\theta}(s, a) = \frac{1}{\pi_{\theta_k}(a|s)} \left[\frac{1}{i} \sum_{j=1}^{i} y_j 1(a_{\tau_j} = a) \right] \tag{16}$$

where τ_j is the starting time of the jth chosen sub-trajectory. Finally, the advantage value is estimated as,

$$\hat{A}^{\pi_{\theta_k}}(s, a) = \hat{Q}^{\pi_{\theta_k}}(s, a) - \hat{V}^{\pi_{\theta_k}}(s) \tag{17}$$

This allows us to compute an estimate of the policy gradient as follows.

$$\omega_k \triangleq \hat{\nabla}_\theta J(\theta_k) = \frac{1}{H} \sum_{t=t_k}^{t_{k+1}-1} \hat{A}^{\pi_\theta}(s_t, a_t) \nabla_\theta \log \pi_{\theta_k}(a_t|s_t) \tag{18}$$

where $t_k = (k-1)H$ is the starting time of the kth epoch. The policy parameters are updated via (20). In the following lemma, we show that $\hat{A}^{\pi_{\theta_k}}(s, a)$ is a good estimator of $A^{\pi_{\theta_k}}(s, a)$.

Lemma 2. *The following inequalities hold $\forall k$, $\forall (s, a)$ and sufficiently large T.*

$$\begin{aligned} &\mathbf{E}\left[\left(\hat{A}^{\pi_{\theta_k}}(s, a) - A^{\pi_{\theta_k}}(s, a) \right)^2 \right] \\ &\quad \leq \mathcal{O}\left(\frac{t_{\text{hit}} N^3 \log T}{H \pi_{\theta_k}(a|s)} \right) = \mathcal{O}\left(\frac{t_{\text{mix}}^2 (\log T)^2}{\sqrt{T} \pi_{\theta_k}(a|s)} \right) \end{aligned} \tag{19}$$

Lemma 2 establishes that the L_2 error of our proposed estimator can be bounded above as $\tilde{\mathcal{O}}(1/\sqrt{T})$. As we shall see later, this result can be used to bound the estimation error of the gradient. It is worthwhile to point out that $\hat{V}^{\pi_{\theta_k}}(s)$ and $\hat{Q}^{\pi_{\theta_k}}(s, a)$ defined in (15), (16) respectively, may not themselves be good estimators of their target quantities although their difference is one. We would also like to mention that our Algorithm 2 is inspired by Algorithm 2 of (Wei et al. 2020). The main difference is that we take the episode length to be $H = \tilde{\mathcal{O}}(\sqrt{T})$ while in (Wei et al. 2020), it was chosen to be $\tilde{\mathcal{O}}(1)$. This extra $\sqrt{T}$ factor makes the estimation error a decreasing function of T.

Global Convergence Analysis

In this section, we show that our proposed Algorithm 1 converges globally. This essentially means that the parameters

Algorithm 1: Parameterized Policy Gradient

Input: Initial parameter θ_1, learning rate α, initial state $s_0 \sim \rho(\cdot)$, episode length H
$K = T/H$
for $k \in \{1, \cdots, K\}$ **do**
  $\mathcal{T}_k \leftarrow \phi$
  for $t \in \{(k-1)H, \cdots, kH-1\}$ **do**
    Execute $a_t \sim \pi_{\theta_k}(\cdot|s_t)$, receive reward $r(s_t, a_t)$ and observe s_{t+1}
    $\mathcal{T}_k \leftarrow \mathcal{T}_k \cup \{(s_t, a_t)\}$
  end for
  for $t \in \{(k-1)H, \cdots, kH-1\}$ **do**
    Using Algorithm 2, and $\mathcal{T}_k$, compute $\hat{A}^{\pi_{\theta_k}}(s_t, a_t)$
  end for
  Using (18), compute ω_k
  Update parameters as

$$\theta_{k+1} = \theta_k + \alpha\omega_k \tag{20}$$

end for

$\{\theta_k\}_{k=1}^{\infty}$ are such that the sequence $\{J(\theta_k)\}_{k=1}^{\infty}$, in certain sense, approaches the optimal average reward, J^*. Such convergence will be later useful in bounding the regret of our algorithm. Before delving into the analysis, we would like to first point out a few assumptions that are needed to establish the results.

Assumption 2. *The log-likelihood function is G-Lipschitz and B-smooth. Formally,* $\forall\theta, \theta_1, \theta_2 \in \Theta, \forall(s,a) \in \mathcal{S} \times \mathcal{A}$

$$\begin{aligned} &\|\nabla_\theta \log \pi_\theta(a|s)\| \leq G \quad \forall\theta \in \Theta, \forall(s,a) \in \mathcal{S} \times \mathcal{A} \\ &\|\nabla_\theta \log \pi_{\theta_1}(a|s) - \nabla_\theta \log \pi_{\theta_2}(a|s)\| \leq B\|\theta_1 - \theta_2\| \end{aligned} \tag{21}$$

Remark 1. *The Lipschitz and smoothness properties for the log-likelihood are quite common in the field of policy gradient algorithm (Agarwal et al. 2020; Zhang et al. 2021; Liu et al. 2020). Such properties can also be verified for simple parameterization such as Gaussian policy.*

One can immediately see that by combining Assumption 2 with Lemma 2 and using the definition of the gradient estimator as given in (18), we arrive at the following important result.

Lemma 3. *The following relation holds* $\forall k$.

$$\mathbf{E}\left[\|\omega_k - \nabla_\theta J(\theta_k)\|^2\right] \leq \mathcal{O}\left(\frac{AG^2 t_{\text{mix}}^2 (\log T)^2}{t_{\text{hit}}\sqrt{T}}\right) \tag{22}$$

Lemma 3 claims that the error in estimating the gradient can be bounded above as $\tilde{\mathcal{O}}(1/\sqrt{T})$. This result will be used in proving the global convergence of our algorithm.

Assumption 3. *Define the transferred function approximation error*

$$L_{d_\rho^{\pi^*}, \pi^*}(\omega_\theta^*, \theta) = \mathbf{E}_{s\sim d_\rho^{\pi^*}} \mathbf{E}_{a\sim\pi^*(\cdot|s)}\Bigg[\left(\nabla_\theta \log \pi_\theta(a|s) \cdot \omega_\theta^* - A^{\pi_\theta}(s,a)\right)^2\Bigg] \tag{23}$$

Algorithm 2: Advantage Estimation

Input: Trajectory $(s_{t_1}, a_{t_1}, \ldots, s_{t_2}, a_{t_2})$, state s, action a, and policy parameter θ
Initialize: $i \leftarrow 0$, $\tau \leftarrow t_1$
Define: $N = 4t_{\text{mix}} \log_2 T$.
while $\tau \leq t_2 - N$ **do**
  if $s_\tau = s$ **then**
    $i \leftarrow i + 1$.
    $\tau_i \leftarrow \tau$
    $y_i = \sum_{t=\tau}^{\tau+N-1} r(s_t, a_t)$.
    $\tau \leftarrow \tau + 2N$.
  else
    $\tau \leftarrow \tau + 1$.
  end if
end while
if $i > 0$ **then**
  $\hat{V}(s) = \frac{1}{i}\sum_{j=1}^{i} y_j$,
  $\hat{Q}(s,a) = \frac{1}{\pi_\theta(a|s)}\left[\frac{1}{i}\sum_{j=1}^{i} y_j 1(a_{\tau_j} = a)\right]$
else
  $\hat{V}(s) = 0, \hat{Q}(s,a) = 0$
end if
return $\hat{Q}(s,a) - \hat{V}(s)$

where π^* *is the optimal policy and* ω_θ^* *is given as*

$$\omega_\theta^* = \arg\min_{\omega\in\mathbb{R}^{\text{d}}} \mathbf{E}_{s\sim d_\rho^{\pi_\theta}} \mathbf{E}_{a\sim\pi_\theta(\cdot|s)}\Bigg[\left(\nabla_\theta \log \pi_\theta(a|s) \cdot \omega - A^{\pi_\theta}(s,a)\right)^2\Bigg] \tag{24}$$

We assume that the error satisfies $L_{d_\rho^{\pi^*}, \pi^*}(\omega_\theta^*, \theta) \leq \epsilon_{\text{bias}}$ *for any* $\theta \in \Theta$ *where* ϵ_{bias} *is a positive constant.*

Remark 2. *The transferred function approximation error, defined by* (23) *and* (24)*, quantifies the expressivity of the policy class in consideration. It has been shown that the softmax parameterization (Agarwal et al. 2021) or linear MDP structure (Jin et al. 2020) admits* $\epsilon_{\text{bias}} = 0$*. When parameterized by the restricted policy class that does not contain all the policies,* ϵ_{bias} *turns out to be strictly positive. However, for a rich neural network parameterization, the* ϵ_{bias} *is small (Wang et al. 2019). A similar assumption has been adopted in (Liu et al. 2020) and (Agarwal et al. 2021).*

Remark 3. *It is to be mentioned that* ω_θ^* *defined in* (24) *can be alternatively written as,*

$$\omega_\theta^* = F(\theta)^\dagger \mathbf{E}_{s\sim d^{\pi_\theta}} \mathbf{E}_{a\sim\pi_\theta(\cdot|s)} \left[\nabla_\theta \log \pi_\theta(a|s) A^{\pi_\theta}(s,a)\right]$$

where $\dagger$ *symbolizes the Moore-Penrose pseudoinverse operation and* $F(\theta)$ *is the Fisher information matrix as defined below.*

$$F(\theta) = \mathbf{E}_{s\sim d^{\pi_\theta}} \mathbf{E}_{a\sim\pi_\theta(\cdot|s)} \big[\nabla_\theta \log \pi_\theta(a|s) (\nabla_\theta \log \pi_\theta(a|s))^T\big] \tag{25}$$

Assumption 4. *There exists a constant* $\mu_F > 0$ *such that* $F(\theta) - \mu_F I_{\text{d}}$ *is positive semidefinite where* I_{d} *denotes an identity matrix.*

Assumption 4 is also commonly used in the policy gradient analysis (Liu et al. 2020). This is satisfied by the Gaussian policy with a linearly parameterized mean.

In the discounted reward setup, one key result is the performance difference lemma. In the averaged reward setting, this is derived as stated below.

Lemma 4. *The difference in the performance for any policies π_θ and $\pi_{\theta'}$ is bounded as follows*

$$J(\theta) - J(\theta') = \mathbf{E}_{s\sim d^{\pi_\theta}} \mathbf{E}_{a\sim\pi_\theta(\cdot|s)} \left[A^{\pi_{\theta'}}(s,a)\right] \tag{26}$$

Using Lemma 4, we present a general framework for convergence analysis of the policy gradient algorithm in the averaged reward case as dictated below. This is inspired by the convergence analysis of (Liu et al. 2020) for the discounted reward MDPs.

Lemma 5. *Suppose a general gradient ascent algorithm updates the policy parameter in the following way.*

$$\theta_{k+1} = \theta_k + \alpha\omega_k \tag{27}$$

When Assumptions 2, 3, and 4 hold, we have the following inequality for any K.

$$J^* - \frac{1}{K}\sum_{k=1}^{K} J(\theta_k) \leq \sqrt{\epsilon_{\text{bias}}} + \frac{G}{K}\sum_{k}^{K} \|(\omega_k - \omega_k^*)\| + \frac{B\alpha}{2K}\sum_{k=1}^{K}\|\omega_k\|^2 + \frac{1}{\alpha K}\mathbf{E}_{s\sim d^{\pi^*}}\left[KL(\pi^*(\cdot|s)\|\pi_{\theta_1}(\cdot|s))\right] \tag{28}$$

where $\omega_k^* := \omega_{\theta_k}^*$ and $\omega_{\theta_k}^*$ is defined in (24)*, $J^* = J(\theta^*)$, and $\pi^* = \pi_{\theta^*}$ where θ^* is the optimal parameter.*

Lemma 5 bounds the optimality error of any gradient ascent algorithm as a function of intermediate gradient norms. Note the presence of ϵ_{bias} in the upper bound. Clearly, for a severely restricted policy class where ϵ_{bias} is significant, the optimality bound becomes poor. Consider the expectation of the second term in (28). Note that,

$$\begin{aligned}
&\left(\frac{1}{K}\sum_{k=1}^{K}\mathbf{E}\|\omega_k - \omega_k^*\|\right)^2 \leq \frac{1}{K}\sum_{k=1}^{K}\mathbf{E}\left[\|\omega_k - \omega_k^*\|^2\right] \\
&= \frac{1}{K}\sum_{k=1}^{K}\mathbf{E}\left[\|\omega_k - F(\theta_k)^\dagger\nabla_\theta J(\theta_k)\|^2\right] \\
&\leq \frac{2}{K}\sum_{k=1}^{K}\mathbf{E}\left[\|\omega_k - \nabla_\theta J(\theta_k)\|^2\right] \\
&\qquad + \frac{2}{K}\sum_{k=1}^{K}\mathbf{E}\left[\|\nabla_\theta J(\theta_k) - F(\theta_k)^\dagger\nabla_\theta J(\theta_k)\|^2\right] \\
&\overset{(a)}{\leq} \frac{2}{K}\sum_{k=1}^{K}\mathbf{E}\left[\|\omega_k - \nabla_\theta J(\theta_k)\|^2\right] \\
&\qquad + \frac{2}{K}\sum_{k=1}^{K}\left(1 + \frac{1}{\mu_F^2}\right)\mathbf{E}\left[\|\nabla_\theta J(\theta_k)\|^2\right]
\end{aligned} \tag{29}$$

where (a) uses Assumption 4. The expectation of the third term in (28) can be bounded as follows.

$$\begin{aligned}
\frac{1}{K}\sum_{k=1}^{K}\mathbf{E}\left[\|\omega_k\|^2\right] &\leq \frac{1}{K}\sum_{k=1}^{K}\mathbf{E}\left[\|\omega_k - \nabla_\theta J(\theta_k)\|^2\right] \\
&\quad + \frac{1}{K}\sum_{k=1}^{K}\mathbf{E}\left[\|\nabla_\theta J(\theta_k)\|^2\right]
\end{aligned} \tag{30}$$

In both (29), (30), the terms related to $\|\omega_k - \nabla_\theta J(\theta_k)\|^2$ can be bounded by Lemma 3. We have,

$$\mathbf{E}\left[\|\omega_k - \nabla_\theta J(\theta_k)\|^2\right] \leq \mathcal{O}\left(\frac{AG^2 t_{\text{mix}}^2 (\log T)^2}{t_{\text{hit}}\sqrt{T}}\right) \tag{31}$$

To bound the $\|\nabla_\theta J(\theta_k)\|^2$ related terms, we use the following lemma.

Lemma 6. *Let $J(\cdot)$ be L-smooth and $\alpha = \frac{1}{4L}$. Then the following inequality holds.*

$$\frac{1}{K}\sum_{k=1}^{K}\|\nabla J(\theta_k)\|^2 \leq \frac{16L}{K} + \frac{16}{3K}\sum_{k=1}^{K}\|\nabla J(\theta_k) - \omega_k\|^2 \tag{32}$$

Using Lemma 3, we obtain the following inequality.

$$\frac{1}{K}\sum_{k=1}^{K}\|\nabla J(\theta_k)\|^2 \leq \mathcal{O}\left(\frac{AG^2 t_{\text{mix}}^2 (\log T)^2}{t_{\text{hit}}\sqrt{T}}\right) \tag{33}$$

Applying (31) and (33) in (30), we finally arrive at,

$$\frac{1}{K}\sum_{k=1}^{K}\mathbf{E}\left[\|\omega_k\|^2\right] \leq \mathcal{O}\left(\frac{AG^2 t_{\text{mix}}^2 (\log T)^2}{t_{\text{hit}}\sqrt{T}}\right) \tag{34}$$

Similarly, using (29), we deduce the following.

$$\frac{1}{K}\sum_{k=1}^{K}\mathbf{E}\|\omega_k - \omega_k^*\| \leq \mathcal{O}\left(\frac{\sqrt{A}Gt_{\text{mix}}}{\sqrt{t_{\text{hit}}}}\left(1 + \frac{1}{\mu_F}\right)\frac{\log T}{T^{\frac{1}{4}}}\right) \tag{35}$$

Inequalities (34) and (35) lead to the following result.

Theorem 1. *Let $\{\theta_k\}_{k=1}^K$ be defined as in Lemma 5. If assumptions 1, 2, 3, 4 hold, $J(\cdot)$ is L-smooth and $\alpha = \frac{1}{4L}$, then the following inequality holds for $K = T/H$ where T is sufficiently large and $H = 16 t_{\text{mix}} t_{\text{hit}}\sqrt{T}(\log_2 T)^2$.*

$$\begin{aligned}
J^* - \frac{1}{K}\sum_{k=1}^{K}\mathbf{E}\left[J(\theta_k)\right] &\leq \mathcal{O}\left(\frac{ABG^2 t_{\text{mix}}^2}{t_{\text{hit}}}\frac{(\log T)^2}{L\sqrt{T}}\right) \\
&\quad + \mathcal{O}\left(\frac{\sqrt{A}G^2 t_{\text{mix}}}{\sqrt{t_{\text{hit}}}}\left(1 + \frac{1}{\mu_F}\right)\frac{\log T}{T^{\frac{1}{4}}}\right) + \sqrt{\epsilon_{\text{bias}}}
\end{aligned} \tag{36}$$

Theorem 1 dictates that the sequence $\{J(\theta_k)\}_{k=1}^K$ generated by Algorithm 1 converges to J^* with a convergence rate of $\mathcal{O}(T^{-\frac{1}{4}} + \sqrt{\epsilon_{\text{bias}}})$. Alternatively, one can say that in order to achieve an optimality error of $\epsilon + \sqrt{\epsilon_{\text{bias}}}$, it is sufficient to choose $T = \mathcal{O}\left(\epsilon^{-4}\right)$. It matches the state-of-the-art sample complexity bound of the policy gradient algorithm with general parameterization in the discounted reward setup (Liu et al. 2020).

Regret Analysis

In this section, we demonstrate how the convergence analysis in the previous section can be used to bind the expected regret of our proposed algorithm. Note that the regret can be decomposed as follows.

$$\begin{aligned}\mathrm{Reg}_T &= \sum_{t=0}^{T-1}\left(J^* - r(s_t,a_t)\right)\\ &= H\sum_{k=1}^{K}\left(J^* - J(\theta_k)\right) + \sum_{k=1}^{K}\sum_{t\in\mathcal{I}_k}\left(J(\theta_k) - r(s_t,a_t)\right)\end{aligned} \tag{37}$$

where $\mathcal{I}_k \triangleq \{(k-1)H, \cdots, kH-1\}$. Note that the expectation of the first term in (37) can be bounded using Theorem 36. The expectation of the second term can be expressed as follows,

$$\begin{aligned}&\mathbf{E}\left[\sum_{k=1}^{K}\sum_{t\in\mathcal{I}_k}\left(J(\theta_k) - r(s_t,a_t)\right)\right]\\ &\overset{(a)}{=} \mathbf{E}\left[\sum_{k=1}^{K}\sum_{t\in\mathcal{I}_k}\mathbf{E}_{s'\sim P(\cdot|s_t,a_t)}[V^{\pi_{\theta_k}}(s')] - Q^{\pi_{\theta_k}}(s_t,a_t)\right]\\ &\overset{(b)}{=} \mathbf{E}\left[\sum_{k=}^{K}\sum_{t\in\mathcal{I}_k}V^{\pi_{\theta_k}}(s_{t+1}) - V^{\pi_{\theta_k}}(s_t)\right]\\ &= \mathbf{E}\left[\sum_{k=1}^{K}V^{\pi_{\theta_k}}(s_{kH}) - V^{\pi_{\theta_k}}(s_{(k-1)H})\right]\\ &= \underbrace{\mathbf{E}\left[\sum_{k=1}^{K-1}V^{\pi_{\theta_{k+1}}}(s_{kH}) - V^{\pi_{\theta_k}}(s_{kH})\right]}_{\triangleq P}\\ &+ \underbrace{\mathbf{E}\left[V^{\pi_{\theta_K}}(s_T) - V^{\pi_{\theta_0}}(s_0)\right]}_{\triangleq Q}\end{aligned} \tag{38}$$

where (a) follows from Bellman equation and (b) utilises the following facts: $\mathbf{E}[V^{\pi_{\theta_k}}(s_{t+1})] = \mathbf{E}_{s'\sim P(\cdot|s_t,a_t)}[V^{\pi_{\theta_k}}(s')]$ and $\mathbf{E}[V^{\pi_{\theta_k}}(s_t)] = \mathbf{E}[Q^{\pi_{\theta_k}}(s_t,a_t)]$. The term, P in (38) can be bounded using Lemma 7 (stated below). Moreover, the term Q can be upper bounded as $\mathcal{O}(t_{\mathrm{mix}})$ as clarified in (Bai, Mondal, and Aggarwal 2023).

Lemma 7. *If assumptions 1 and 2 hold, then for $K = T/H$ where $H = 16t_{\mathrm{mix}}t_{\mathrm{hit}}\sqrt{T}(\log_2 T)^2$, the following inequalities are true $\forall k$, $\forall(s,a) \in \mathcal{S}\times\mathcal{A}$ and sufficiently large T.*

(a) $|\pi_{\theta_{k+1}}(a|s) - \pi_{\theta_k}(a|s)| \le G\pi_{\bar{\theta}_k}(a|s)\,\|\theta_{k+1} - \theta_k\|$

(b) $\sum_{k=1}^{K-1}\mathbf{E}|J(\theta_{k+1}) - J(\theta_k)| \le \mathcal{O}\left(\alpha\sqrt{A}G^2\dfrac{t_{\mathrm{mix}}T^{\frac{1}{4}}}{t_{\mathrm{hit}}^{3/2}\log T}\right)$

(c) $\sum_{k=1}^{K}\mathbf{E}|V^{\pi_{\theta_{k+1}}}(s) - V^{\pi_{\theta_k}}(s)| \le \mathcal{O}\left(\alpha\sqrt{A}G^2\dfrac{t_{\mathrm{mix}}}{t_{\mathrm{hit}}^{3/2}}T^{\frac{1}{4}}\right)$
$+ \mathcal{O}\left(\alpha\sqrt{A}G^2\dfrac{t_{\mathrm{mix}}T^{\frac{1}{4}}}{t_{\mathrm{hit}}^{3/2}\log T}\right) + \mathcal{O}\left(\alpha\sqrt{A}G^2\dfrac{t_{\mathrm{mix}}^2T^{\frac{1}{4}}\log T}{t_{\mathrm{hit}}^{3/2}}\right)$

where $\bar{\theta}_k$ is some convex combination of θ_k and θ_{k+1}.

Lemma 7 can be interpreted as the stability results of our algorithm. It essentially states that the policy parameters are updated such that the average difference between consecutive average reward and value functions decreases with the horizon, T. Using the above result, we now prove our regret guarantee.

Theorem 2. *If assumptions 1, 2, 3, and 4 hold, $J(\cdot)$ is L-smooth, and T is sufficiently large, then our proposed Algorithm 1 achieves the following expected regret bound with learning rate $\alpha = \frac{1}{4L}$.*

$$\begin{aligned}\mathbf{E}\left[\mathrm{Reg}_T\right] &\le T\sqrt{\epsilon_{\mathrm{bias}}} + \tilde{\mathcal{O}}\left(\frac{ABG^2t_{\mathrm{mix}}^2}{t_{\mathrm{hit}}}\frac{\sqrt{T}}{L}\right)\\ &+ \tilde{\mathcal{O}}\left(\frac{\sqrt{A}G^2t_{\mathrm{mix}}}{\sqrt{t_{\mathrm{hit}}}}\left(1+\frac{1}{\mu_F}\right)T^{\frac{3}{4}}\right) + \mathcal{O}(t_{\mathrm{mix}})\\ &+ \tilde{\mathcal{O}}\left(\frac{\sqrt{A}G^2t_{\mathrm{mix}}}{t_{\mathrm{hit}}^{3/2}L}T^{\frac{1}{4}}\right) + \tilde{\mathcal{O}}\left(\sqrt{A}\frac{G^2}{L}\frac{t_{\mathrm{mix}}^2}{t_{\mathrm{hit}}^{3/2}}T^{\frac{1}{4}}\right)\end{aligned} \tag{39}$$

Theorem 2 shows that the expected regret of Algorithm 1 is bounded by $\tilde{\mathcal{O}}(T^{\frac{3}{4}} + T\sqrt{\epsilon_{\mathrm{bias}}})$. It also shows how other parameters such as t_{mix}, and t_{hit} influence the regret value. Note that the expected regret does not depend on the size of the state space. Hence, our algorithm can be applied even if the state space is large without compromising the regret performance.

Conclusion

In this paper, we proposed an algorithm based on the vanilla policy gradient for reinforcement learning in an infinite horizon average reward setting. Unlike the recent works on this setting which require the MDP to be tabular or have a linear structure, we adopt the general parametrization to the policy to ensure the algorithm can be applied even with large state space. We show that the proposed algorithm converges to the neighborhood of the global optimum with rate $\mathcal{O}(T^{-1/4})$, which matches the result of vanilla policy gradient with general parametrization in discounted reward setting. We use this convergence result to further show that our algorithm achieves a regret of $\tilde{\mathcal{O}}(T^{\frac{3}{4}})$.

We note that this paper unveils numerous promising directions for future research. These avenues encompass exploring the possibility of relaxing the assumption of ergodic MDPs to weakly communicating MDPs, refining regret bounds for enhanced performance, deriving more robust lower bounds for the general parametrization, and extending the problem domain to incorporate constraints.

Acknowledgements

This work was supported in part by the U.S. National Science Foundation under Grant CCF-2149588 and Cisco Inc.

References

Agarwal, A.; Kakade, S. M.; Lee, J. D.; and Mahajan, G. 2020. Optimality and Approximation with Policy Gradient

Methods in Markov Decision Processes. In *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, 64–66. PMLR.

Agarwal, A.; Kakade, S. M.; Lee, J. D.; and Mahajan, G. 2021. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *The Journal of Machine Learning Research*, 22(1): 4431–4506.

Agarwal, M.; Bai, Q.; and Aggarwal, V. 2022a. Concave Utility Reinforcement Learning with Zero-Constraint Violations. *Transactions on Machine Learning Research*.

Agarwal, M.; Bai, Q.; and Aggarwal, V. 2022b. Regret guarantees for model-based reinforcement learning with long-term average constraints. In *Uncertainty in Artificial Intelligence*, 22–31. PMLR.

Agrawal, S.; and Jia, R. 2017. Optimistic posterior sampling for reinforcement learning: worst-case regret bounds. *Advances in Neural Information Processing Systems*, 30.

Al-Abbasi, A. O.; Ghosh, A.; and Aggarwal, V. 2019. Deeppool: Distributed model-free algorithm for ride-sharing using deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 20(12): 4714–4727.

Auer, P.; Jaksch, T.; and Ortner, R. 2008. Near-optimal regret bounds for reinforcement learning. *Advances in neural information processing systems*, 21.

Bai, Q.; Mondal, W. U.; and Aggarwal, V. 2023. Regret analysis of policy gradient algorithm for infinite horizon average reward Markov decision processes. *arXiv:2309.01922*.

Chen, L.; Jain, R.; and Luo, H. 2022. Learning infinite-horizon average-reward Markov decision process with constraints. In *International Conference on Machine Learning*, 3246–3270. PMLR.

Ding, D.; Zhang, K.; Basar, T.; and Jovanovic, M. 2020. Natural Policy Gradient Primal-Dual Method for Constrained Markov Decision Processes. In *Advances in Neural Information Processing Systems*, volume 33, 8378–8390. Curran Associates, Inc.

Fatkhullin, I.; Barakat, A.; Kireeva, A.; and He, N. 2023. Stochastic policy gradient methods: Improved sample complexity for fisher-non-degenerate policies. *arXiv:2302.01734*.

Fruit, R.; Pirotta, M.; Lazaric, A.; and Ortner, R. 2018. Efficient bias-span-constrained exploration-exploitation in reinforcement learning. In *International Conference on Machine Learning*, 1578–1586. PMLR.

Geng, N.; Lan, T.; Aggarwal, V.; Yang, Y.; and Xu, M. 2020. A multi-agent reinforcement learning perspective on distributed traffic engineering. In *2020 IEEE 28th International Conference on Network Protocols (ICNP)*, 1–11. IEEE.

Ghosh, A.; Zhou, X.; and Shroff, N. 2022. Achieving Sub-linear Regret in Infinite Horizon Average Reward Constrained MDP with Linear Function Approximation. In *The Eleventh International Conference on Learning Representations*.

Gong, H.; and Wang, M. 2020. A Duality Approach for Regret Minimization in Average-Award Ergodic Markov Decision Processes. In *Learning for Dynamics and Control*, 862–883. PMLR.

Jin, C.; Allen-Zhu, Z.; Bubeck, S.; and Jordan, M. I. 2018. Is Q-Learning Provably Efficient? In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Jin, C.; Yang, Z.; Wang, Z.; and Jordan, M. I. 2020. Provably efficient reinforcement learning with linear function approximation. In *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, 2137–2143. PMLR.

Lattimore, T.; and Szepesvári, C. 2020. *Bandit algorithms*. Cambridge University Press.

Ling, L.; Mondal, W. U.; and Ukkusuri, S. V. 2023. Cooperating Graph Neural Networks with Deep Reinforcement Learning for Vaccine Prioritization. *arXiv preprint arXiv:2305.05163*.

Liu, Y.; Zhang, K.; Basar, T.; and Yin, W. 2020. An improved analysis of (variance-reduced) policy gradient and natural policy gradient methods. *Advances in Neural Information Processing Systems*, 33: 7624–7636.

Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, 1928–1937. PMLR.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533.

Mondal, W. U.; and Aggarwal, V. 2023. Improved Sample Complexity Analysis of Natural Policy Gradient Algorithm with General Parameterization for Infinite Horizon Discounted Reward Markov Decision Processes. *arXiv preprint arXiv:2310.11677*.

Ouyang, Y.; Gagrani, M.; Nayyar, A.; and Jain, R. 2017. Learning unknown markov decision processes: A thompson sampling approach. *Advances in neural information processing systems*, 30.

Pesquerel, F.; and Maillard, O.-A. 2022. IMED-RL: Regret optimal learning of ergodic Markov decision processes. In *NeurIPS 2022-Thirty-sixth Conference on Neural Information Processing Systems*.

Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; and Moritz, P. 2015. Trust region policy optimization. In *International conference on machine learning*, 1889–1897. PMLR.

Sutton, R. S.; McAllester, D.; Singh, S.; and Mansour, Y. 1999. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12.

Wang, L.; Cai, Q.; Yang, Z.; and Wang, Z. 2019. Neural Policy Gradient Methods: Global Optimality and Rates of Convergence. arXiv:1909.01150.

Wei, C.-Y.; Jahromi, M. J.; Luo, H.; and Jain, R. 2021. Learning infinite-horizon average-reward mdps with linear function approximation. In *International Conference on Artificial Intelligence and Statistics*, 3007–3015. PMLR.

Wei, C.-Y.; Jahromi, M. J.; Luo, H.; Sharma, H.; and Jain, R. 2020. Model-free reinforcement learning in infinite-horizon average-reward markov decision processes. In *International conference on machine learning*, 10170–10180. PMLR.

Wei, H.; Liu, X.; and Ying, L. 2022. A provably-efficient model-free algorithm for infinite-horizon average-reward constrained Markov decision processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 3868–3876.

Zhang, J.; Ni, C.; Yu, Z.; Szepesvari, C.; and Wang, M. 2021. On the Convergence and Sample Efficiency of Variance-Reduced Policy Gradient Method. *Advances in Neural Information Processing Systems*, 34: 2228–2240.