Convergence for Natural Policy Gradient on Infinite-State Average-Reward Markov Decision Processes

ISAAC GROSOF, Georgia Institute of Technology
SIVA THEJA MAGULURI, Georgia Institute of Technology
R. SRIKANT, University of Illinois, Urbana-Champaign

Infinite-state Markov Decision Processes (MDPs) are essential in modeling and optimizing a wide variety of engineering problems. In the reinforcement learning (RL) context, a variety of algorithms have been developed to learn and optimize these MDPs. At the heart of many popular policy-gradient based learning algorithms, such as natural actor-critic, TRPO, and PPO, lies the Natural Policy Gradient (NPG) algorithm. Convergence results for these RL algorithms rest on convergence results for the NPG algorithm. However, all existing results on the convergence of the NPG algorithm are limited to finite-state settings.

We prove the first convergence rate bound for the NPG algorithm for infinite-state average-reward MDPs, proving a $O(1/\sqrt{T})$ convergence rate, if the NPG algorithm is initialized with a good initial policy. Moreover, we show that in the context of a large class of queueing MDPs, the MaxWeight policy suffices to satisfy our initial-policy requirement and achieve a $O(1/\sqrt{T})$ convergence rate. Key to our result are state-dependent bounds on the relative value function achieved by the iterate policies of the NPG algorithm.

1 INTRODUCTION

Infinite-state Markov Decision Processes (MDPs) are used to model many engineering problems. A state might include the number of orders for a product, the time since an event occurred, or the number of people waiting in an unbounded queue. While artificial truncation could be imposed, often the cleanest mathematical description of a problem will have infinitely many states.

In various disciplines, people have tried to study and optimize these MDPs using a variety of tools, including methods based off of dynamic programming, such as value iteration and policy iteration methods [5, 25], as well as gradient-based methods drawing on continuous optimization.

In the reinforcement learning (RL) context, algorithms have been developed to simultaneously learn the structure of the MDPs and optimally solve them. While most standard learning theory focuses on the settings with finite state spaces, some recent work has empirically established that learning-based approaches can be used in the infinite-state setting as well [12].

An important and popular class of learning algorithms are policy-gradient-based methods, including the actor-critic algorithm [30], the natural actor-critic algorithm [24], and more recent methods such as the Trust Region Policy Optimization (TRPO) [26] and Proximal Policy Optimization (PPO) algorithms [12, 27].

In this paper, we study the natural policy gradient (NPG) algorithm,which lies at the heart of policy-gradient methods such as natural actor-critic, TRPO, and PPO [20], Algorithm 1. While the interpretation of the original NPG algorithm is in terms of the gradient of the objective of the MDP with respect to a value function parameterization, it can also be thought as a “softer" version of policy iteration. In this viewpoint, instead of finding the optimal policy with respect to the current estimate of the value function as in policy iteration, we find a randomized approximation to the optimal policy. This encourages exploration of all states and actions, which is important in the RL context. In the RL algorithm, the policy evaluation part of policy iteration is further replaced by a learning algorithm such as TD learning, Using this viewpoint, the convergence of the overall algorithm can be decomposed into an analysis of NPG with perfect policy evaluation and the

Authors’ addresses: Isaac Grosof, isaac.grosof@northwestern.edu, Georgia Institute of Technology; Siva Theja Maguluri, siva.theja@gatech.edu, Georgia Institute of Technology; R. Srikant, University of Illinois, Urbana-Champaign, rsrikant@illinois.edu.

analysis of TD learning; see, for example, [1]. In this paper, we focus on the analysis of NPG with perfect policy evaluation. Understanding its behavior is critical to understanding the behavior of the overall RL algorithm [1].

In recent years, progress has been made into understanding the convergence rate of these learning-based algorithms, building upon convergence-rate results for the NPG algorithm. However, existing results on the convergence of the NPG algorithm are limited to settings with finite (but potentially large) state spaces [3, 13, 23].

We want to investigate the convergence rate of the NPG algorithm in infinite-state, average-reward MDPs, inspired by the MDPs which arise when optimizing queueing systems. Specifically, our goal is to establish theoretical bounds on the convergence rate of the NPG algorithm to the globally optimal policy.

Main contributions:

- In Theorem 1, we prove that for any infinite-state average-reward MDP, given mild assumptions on the structure of the MDP and a good initial policy, the NPG algorithm converges to the global optimum policy. Moreover, we prove an $O(1/\sqrt{T})$ convergence rate bound. This result is the first convergence result for the NPG algorithm in the infinite-state setting. Prior work in the finite-state setting required strong assumptions on all policies and over all states, as we discuss in Section 4.1 [13, 23]. In contrast, our result only makes assumptions on the initial policy.
- In Theorem 2, we examine a large class MDPs arising from queueing systems, and show that the MaxWeight policy satisfies our quality requirement for the initial policy. Therefore, initializing the NPG algorithm with MaxWeight will lead to an $O(1/\sqrt{T})$ convergence rate.
- Methodologically, the key step in our proof is to establish a bound the growth rate of the relative value function, relative to the reward available in a given state. We do this by showing that the relative value function only mildly changes between the iterations of NPG, in conjunction with our assumption on the growth rate of the initial policy. Armed with the growth rate bound, we make a novel alteration to the NPG algorithm by setting a corresponding state-dependent step size. We prove that our state-dependent step size adjusts for the growing relative value function, allowing us to prove our convergence result. We discuss our approach in more detail in Section 4.1.

This paper focuses on infinite-state average-reward MDPs, which are the MDPs of most interest in a variety of engineering contexts. While infinite-state discounted-reward MDPs have rarely been explicitly studied, results on the NPG algorithm in such settings follow from existing finite-state NPG results in the discounted setting. This is in contrast to infinite-state average-reward results, for which new techniques were needed. We discuss infinite-state discounted-reward MDPs in Section 7.

The paper is organized as follows:

- Section 2: We discuss prior work on the NPG algorithm.
- Section 3: We define the MDP model, give our mild assumptions on the MDP, and define the Natural Policy Gradient algorithm.
- Section 4: We state our convergence result for the NPG algorithm in the infinite-state, average-reward setting, and discuss the key challenges and a sketch of our proof.
- Section 5: We prove our convergence result for the NPG in the infinite-state, average-reward setting.
- Section 6: We demonstrate that our results are applicable to a natural class of infinite-state average-reward MDPs arising out of queueing theory.
- Section 7: We briefly discuss the infinite-state discounted-reward setting.

2 PRIOR WORK

We overview prior results on the NPG algorithm in the finite-state setting in Section 2.1, connections between the NPG algorithm and a variety of reinforcement learning algorithms in Section 2.2, results on the NPG algorithm in highly-structured infinite-state settings in Section 2.3, and prior work applying reinforcement learning to queueing problems in Section 2.4.

2.1 Natural Policy Gradient with finite state-space

The Natural Policy Gradient (NPG) algorithm for MDP optimization utilizes the Fisher information associated with a policy to choose a gradient descent direction to update a MDP policy [20]. The NPG algorithm has been shown to have attractive properties, including global convergence in the tabular setting with finite state space [3].

In the discounted-reward tabular setting with finite-state space, NPG has been shown to converge to the optimal policy at a rate of $O(1/T)$: The expected discounted reward of the policy after T iterations of NPG is within $O(1/T)$ of the optimal expected discounted reward [3, 15].

As we discuss in Section 7, this result straightforwardly generalizes from the finite-state-space setting with discounted reward to the infinite-state-space setting with discounted reward, given mild assumptions on the structure of the infinite-state MDP, proving the same $O(1/T)$ convergence result.

In the average-reward tabular setting with finite-state space, NPG has long been known to converge to the optimal policy at a rate of $O(1/\sqrt{T})$ [13]. Recently, an improved convergence rate result of $O(1/T)$ has been proven for the average-reward finite-state setting [23].

However, as we discuss in Section 4.1, these average-reward results do not generalize from the finite-state-space setting to the infinite-state-space setting. Each of the finite-state-space results makes crucial assumptions bounding the behavior of a worst-case state in the MDP. If the state space is finite, these assumptions are reasonable, but in an infinite-state space setting, the assumptions fail. Proving convergence in this infinite-state average-reward setting is the focus of this paper.

2.2 NPG and Reinforcement Learning

The Natural Policy Gradient algorithm can be thought of as applying the Mirror Descent framework, using a Kullback-Leiber divergence penalty to regularize the basic Policy Gradient algorithm [15].

Many reinforcement learning algorithms have been built off of this core idea. The first such algorithm was the Natural Actor-Critic algorithm [24], which explicitly builds off of the NPG algorithm, and which also generalizes previously existing reinforcement learning algorithms such as the original Actor-Critic algorithm [30] and Linear Quadratic Q-Learning [7].

More recent, practically used reinforcement learning algorithms, such as the Trust Region Policy Optimization (TRPO) [26] and Proximal Policy Optimization (PPO) [27], also build off the core NPG algorithm. TRPO replaces the NPG algorithm's KL divergence regularization term with a KL divergence *constraint*, allowing the algorithm to take larger steps towards an improved policy, and achieve better empirical performance. PPO further tweaks the handling of the KL divergence, introducing a clipped surrogate objective based off of the KL divergence.

Further extensions of the PPO algorithm have recently been proposed and empirically studied, including in the setting of infinite-state average-reward reinforcement learning [12]. These extensions reduce the variance of the value estimation process, thereby improving learning performance.

The theoretical analysis of the NPG algorithm underpins the theoretical motivation for these reinforcement learning algorithms. We prove the first convergence result for NPG in the infinite-state average-reward setting, giving theoretical backing for the use of NPG-based reinforcement learning algorithms in this setting.

2.3 Natural Policy Gradient for specialized settings

The policy gradient algorithm has been studied in certain specialized settings with infinite state-spaces and average-reward objective. These results rely heavily on the specific details of those settings, and do not generalize beyond those specific contexts.

Fazel et al. [14] study the Linear Quadratic Regulator (LQR), an important problem in control theory which can be thought of as an MDP optimization problem with an uncountably infinite state-space and average reward. The paper proves that in this setting, the naive policy gradient algorithm converges to the optimal policy, and the NPG algorithm converges with a faster rate guarantee. Kunnumkal and Topaloglu [21] study the base-stock inventory control problem, and prove that an algorithm using the policy gradient framework achieves convergence to the optimal policy, with a bound on the convergence rate.

Follow-up study of these settings has recently demonstrated that these settings exhibit additional structure which allows these convergence results to hold [6]. This additional structure relates to the interplay between policy *gradient* algorithms, which the above results focus on, and policy *improvement* algorithms. In particular, it has been shown that if the standard policy improvement algorithm exhibits no suboptimal stationary points, then policy gradient algorithms will track the standard policy iteration algorithm and thereby converge to the global optimum. This mechanism is the root cause behind the prior convergence results in the LQR and base-stock settings.

In contrast, such properties do not hold for general infinite-state MDPs, such as the ones we study, which have much less structure than these previously-studied settings.

2.4 Queueing and Reinforcement Learning

In this paper, we study infinite-state average-reward MDP optimization, and use MDPs arrising from queueing theory as an example application of our results (See Section 6).

A variety of papers have studied applying reinforcement learning to queueing problems, including using learning algorithms which build off of the NPG algorithm.

Dai and Gluzman [12] study variance reduction techniques for the PPO algorithm in the context of queueing models, and demonstrate empirically strong performance, converging towards an optimal policy in several queueing settings. Che et al. [10] further employ a differentiable simulation-based modeling technique, allowing additional approximate gradient information to be derived, which is then used to improve the empirical reinforcement learning performance in a queueing context.

Wei et al. [32] have demonstrated that a "sample augmentation" technique can reduce the amount of sample data necessary to converge towards an optimal policy for pseudo-stochastic reinforcement learning settings, including queueing models. However, the paper focuses on discounted-cost finite-state queueing MDPs, rather than the infinite-state average-reward queueing models which are preferred elsewhere in queueing theory.

In preprint work, Adler and Subramanian [2] study an infinite-state, average-reward MDP optimization setting, where the underlying MDP is parameterized by an unknown parameter θ. Queueing models with unknown dynamics are used as the motivating models. The paper assumes that an optimal policy π^*_θ is known for the MDP under any given parameter θ, and focuses on learning the true parameter value θ^*, and on minimizing the regret experienced while learning that parameter. Our result complements this paper, as we show that the NPG algorithm can be used to find optimal policies for MDPs in a setting where the underlying parameters are exactly known, providing the necessary input π^*_θ for the above result.

In work currently under submission, Chen et al. [11] study a primal-dual optimization technique for optimizing constrained MDPs, such as those encountered in queueing problems. They consider a Langrangian relaxation of the original MDP, and simultaneously optimize the policy and the dual weights of the relaxation. In the special case of an unconstrained MDP, the underlying primal algorithm they study is the NPG algorithm.

While they consider an infinite-state, average-reward setting, they make a key assumption, [11, Assumption 3], assuming that the state-action relative value function $Q_\pi(s, a)$ is bounded. This is true in the finite-state average-reward setting, and in the infinite-state discounted-reward setting with bounded reward, but fails to hold in the general infinite-state average-reward setting which we study. Correspondingly, in their empirical evaluation, they study queueing MDPs where the queue length has been truncated, allowing their assumption to hold. Using that assumption, they prove an $O(1/\sqrt{T})$ convergence rate to the optimal policy. In contrast, our result is the first to study the NPG algorithm in a setting of unbounded relative value $Q_\pi(s, a)$, allowing our results to apply to a much more general class of infinite-state, average-reward MDPs, including queueing models with unbounded queue length.

3 MODEL, ASSUMPTIONS, AND ALGORITHM

We introduce our MDP model in Section 3.1, our mild assumptions on the structure of the MDP in Section 3.2, and the NPG algorithm in Section 3.3.

3.1 Markov Decision Processes

We consider Markov Decision Processes (MDPs) with infinite horizon, infinite state space S, and finite action space A, and some class of randomized policies $\pi \in \Pi$. Each policy $\pi \in \Pi$ is a function from states to distributions over actions.

The environment is captured by a transition function $\mathbb{P}$ which maps a state-action pair (s, a) to a distribution over states $s' \in S$, denoted by $\mathbb{P}\{s' \mid s, a\}$. For convenience, we will write $\mathbb{P}\{s' \mid s, \pi\}$ to denote the probability of transition from s to s' under policy π:

$$\mathbb{P}_\pi\{s' \mid s\} := \mathbb{P}\{s' \mid s, \pi\} := \sum\nolimits_{a\in A} \pi(a \mid s)\mathbb{P}\{s' \mid s, a\}.$$

The reward is given by a reward function $r(s, a)$ which maps a state-action pair to a real-valued reward. We likewise write $r(s, \pi)$ to denote the expected single-step reward associated with policy π and state s. We define $r_{\max}(s)$ to be the maximum reward achievable in state s, $r_{\max}(s) := \max_a r(s, a)$.

If π gives rise to a stable Markov chain, let J_π denote the average reward associated with policy π, defined as

$$J_\pi = \lim_{T\to\infty} \frac{1}{T}\mathbb{E}_\pi \left[\sum\nolimits_{i=0}^{T-1} r(s_i, \pi)\right],$$

where s_i is the state at time i, and where the expectation $\mathbb{E}_\pi$ is taken with respect to the transition probability $\mathbb{P}_\pi$.

Let d_π denote the stationary distribution over S under the policy π. We can express J_π as

$$J_\pi = \mathbb{E}_{s\sim d_\pi}[r(s, \pi)].$$

The state relative value function $V_\pi(s)$ is defined, up to an additive constant C, to be the additive transient effect of the initial state on the total reward:

$$V_\pi(s) = C + \lim_{T\to\infty} \left(\mathbb{E}_\pi \left[\sum\nolimits_{i=0}^{T-1} r(s_i, \pi) \mid s_0 = s\right] - J_\pi T\right)$$

This is also the solution, up to an additive constant, of the Poisson equation:

$$J_\pi + V_\pi(s) = r(s, \pi) + \mathbb{E}_{s'\sim\mathbb{P}_\pi(s)}[V_\pi(s')]$$

We also define the state-action relative value function $Q_\pi(s,a)$ and its associated Poisson equation:

$$Q_\pi(s,a) = C + \lim_{T\to\infty}\left(\mathbb{E}_\pi\left[\sum_{i=0}^{T-1} r(s_i,\pi) \mid s_0 = s, a_0 = a\right] - J_\pi T\right)$$
$$J_\pi + Q_\pi(s,a) = r(s,a) + \mathbb{E}_{s'\sim\mathbb{P}_\pi(s)}[Q_\pi(s',\pi)].$$

To uniquely specify the additive constant C, we will adopt the convention that $V_\pi(\vec{0}) = 0$, for a specially identified state $\vec{0}$ defined in Assumption 1. In the queueing setting discussed in Section 6, $\vec{0}$ is the state with no jobs in the system.

Let $\tau_\pi(s)$ denote the expected time under policy π to hit the specially identified state $\vec{0}$, starting from state s.

Let $\pi_0, \pi_1, \ldots, \pi_k, \ldots$ denote the iterate policies of the NPG algorithm defined in Section 3.3. For concision, we will write J_k, τ_k, Q_k, etc. as shorthand for $J_{\pi_k}, \tau_{\pi_k}, Q_{\pi_k}$, etc., to denote functions of the iterate policy π_k.

3.2 Assumptions

We make two assumptions on the structure of our MDP: One on the rewards in the MDP, and one on the connectedness of high-reward states. We verify these assumptions for a large class of queueing MDPs in Theorem 2.

First, we make mild assumptions on the reward structure of the MDP:

Assumption 1 (Reward structure).

(a) *We assume that the MDP has bounded positive reward, though it may have unlimited negative reward (e.g. unlimited cost). Specifically, we assume that $c_{\max} := \sup_{s,a} r(s,a) < \infty$.*

It will be useful in some cases to normalize this upper bound to 0. Let us define the reduced reward $\hat{r}(s,a) := r(s,a) - c_{\max}$.

(b) *We assume that there are finitely many high-reward states. Specifically, for any z, we assume that there are finitely many states s such that $r_{\max}(s) \geq z$.*

From (a) and (b), it follows that there must exist a state s^* which achieves the maximum reward $r_{\max}(s^*) = c_{\max}$. Let $\vec{0}$ be a specific maximum-reward state.

(c) *We assume the reward $\hat{r}(s,a)$ is not overwhelmingly dominated by the action a, as opposed to the state s. Specifically, we assume that there exist constants $R_1, R_2 \geq 0$ such that for all states s and all actions a, $\hat{r}_{\max}(s) - \hat{r}(s,a) \leq R_1 \hat{r}_{\max}(s)^2 + R_2$.*

(d) *We assume that the reward $r_{\max}(s)$ does not change very quickly between neighboring states. Specifically, we assume that there exist constants $R_3 \geq 1, R_4 \geq 0$, such that, if s, s' are a pair of states such that $\mathbb{P}\{s \mid s,a\} > 0$ for some action a, then $\hat{r}_{\max}(s') \geq R_3 \hat{r}_{\max}(s) - R_4$.*

Second, we assume that the high-reward states are uniformly connected, under an arbitrary policy:

Assumption 2 (Uniform connectedness of high-reward states). *Given a reward threshold z, we assume that there exists a number of steps x_z and a probability $p_z > 0$ such that for all states s, s' where $r_{\max}(s) \geq z$, $r_{\max}(s') \geq z$, and for all policies π, the probability that the MDP initialized at state s and transitioning under policy π reaches state s' in at most x_z steps is at least p_z.*

3.3 NPG Algorithm

This paper studies the Natural Policy Gradient algorithm, given in Algorithm 1.

Note that the NPG algorithm is an MDP optimization algorithm, rather than a learning algorithm. In particular, we assume that the relative value function $Q_\pi(s,a)$ can be exactly computed.

Algorithm 1 The Natural Policy Gradient algorithm

Initialize: A learning rate function, mapping states s to learning rates β_s.
Initialize: An initial policy π_0.
for each iteration $k = 0$ to $T - 1$ **do**
  for all state-action pairs s, a **do**
    Compute the value function $Q_k(s, a)$.
    Compute the weighted update $\pi_k(a \mid s)\beta_s^{Q_k(s,a)}$.
    Set the new policy probability $\pi_{k+1}(a \mid s) = \pi_k(a \mid s)\beta_s^{Q_k(s,a)}/Z_{s,k}$,
    where $Z_{s,k} = \sum_{a'} \pi_k(a' \mid s)\beta_s^{Q_k(s,a')}$.

4 RESULTS

We prove the first convergence result for the Natural Policy Gradient algorithm in an MDP setting with infinite state space and average-reward objective.

Theorem 1. *For any average-reward MDP satisfying Assumptions 1 and 2, given an initial policy π_0 such that there exist constants $c_0 > 0, c_1 \geq 0$ such that*

$$V_0(s) \geq -c_0 \hat{r}_{\max}(s)^2 - c_1, \tag{1}$$

the NPG algorithm with learning rate parameterization β_s given in (5) *achieves the convergence rate*

$$J_* - J_T \leq \frac{c_*}{\sqrt{T}},$$

where c_* is a constant depending on the MDP parameters and on c_0 and c_1.

Proof deferred to Section 5.

To give a concrete example of an infinite-state average-reward MDP satisfying Assumptions 1 and 2 and an initial policy satisfying (1), we examine the "Generalized Switch with Static Environment" (GSSE) in Section 6. This is a highly general queueing model which can capture a wide variety of commonly studied queueing systems, including the N-system, the switch, and the multiserver-job system. We study the MaxWeight initial policy in this setting.

In Theorem 2, we prove that in the GSSE setting, with the MaxWeight initial policy, the NPG algorithm converges to the policy with optimal mean queue length with convergence rate $O(1/\sqrt{T})$.

4.1 Challenges

To prove a convergence rate result for the NPG algorithm in the infinite-state-space, average-reward, setting, a natural approach would be to try to generalize finite-state average-reward results to the infinite-reward setting. There are two relevant approaches to consider: the approach of Even-Dar et al. [13], and of Murthy and Srikant [23]. Unfortunately, each of these approaches makes crucial use of assumptions that are only plausible for MDPs with finite state-spaces.

With respect to Even-Dar et al. [13], combining that paper's Algorithm 4 (MDP Experts) with its Algorithm 1 (Weighted Majority) results in exactly the Natural Policy Gradient algorithm, and the paper's Theorem 4.1 proves that NPG converges to the optimal policy in the finite-state average-reward setting, with convergence rate $O(1/\sqrt{T})$. To prove this theorem, the paper assumes that there exists some finite mixing time τ for all policies π. Intuitively, this assumption states that every policy takes the system state to a distribution becomes near the policy's stationary distribution in at most $O(\tau)$ steps. In a finite-state MDP, this is a reasonable assumption: For reasonable policies, τ might be exponential in $|S|$ at worst. In contrast, in an infinite-state MDP with infinite diameter, such as the queueing systems we consider in Section 6, it is plainly false. No policy can achieve a finite mixing time, much less all policies.

Murthy and Srikant [23] prove that the NPG algorithm converges to the optimal policy in the finite-state average-reward setting at rate $O(1/T)$. To do so, the paper assumes that there exists some uniform lower bound Δ on the relative state probability of an arbitrary policy and of the optimal policy, in stationarity:

$$\Delta = \inf_{\pi,s} \frac{d_\pi(s)}{d_*(s)} > 0$$

In a finite-state MDP, this is a reasonable assumption: As long as d_π places nonzero probability on every state s, the assumption will be satisfied. In contrast, in an infinite-state MDP, $d_\pi(s)/d_*(s)$ will approach zero as we look at more and more states s, for almost all policies π. Restricting to a class of policies π where $d_\pi(s)/d_*(s)$ remains bounded away from zero would require advance knowledge of the optimal policy π^*.

Another natural approach would be to try to use results from the discounted-reward setting to prove results in the average-reward setting. A standard result states that a policy π's average reward J_π can be related to its discounted reward $V^\pi_{\eta,\alpha}$ via the formula $J_\pi = \lim_{\alpha\to 1}(1-\alpha)V^\pi_{\eta,\alpha}$, where $V^\pi_{\eta,\alpha}$ denotes a policy π's expected reward, starting from distribution η, with discount factor α [5]. Unfortunately, existing convergence-rate bounds on NPG in the discounted-reward setting, such as [3, Theorem 16], have a $\theta(\frac{1}{(1-\alpha)^2})$ dependency on α, so discounted-reward results do not prove that the NPG algorithm converges in the average-reward setting, much less bound its convergence rate.

These proof approaches do not straightforwardly generalize to our infinite-state average-reward setting.

4.2 Proof sketch

Delving deeper into the Even-Dar et al. [13] approach, their key insight is that one can think of the MDP optimization problem as consisting of many instances of the expert advice problem [9], and think of the NPG algorithm as running a separate instance of the weighted majority algorithm for each state s, where the reward function at time step k is $Q_k(s, a)$.

In the expert advice problem, an agent has a set of possible actions, each with a secret reward. After an action a_k is chosen on time-step k, the reward function $\widetilde{r}_k(\cdot)$ is revealed to the agent. The goal of the expert advice problem is to choose a sequence of actions $\{a_k\}$ whose total reward is close to that of the optimal single action, a_*.

The weighted majority algorithm maintains a distribution π over actions, sampling an action at random from its distribution at each time step. At each time step, the action distribution is updated according to the rule:

$$\pi_{k+1}(a) = \beta^{\widetilde{r}_k(a)}/Z_k, \text{ where } Z_k = \sum\nolimits_{a'} \pi_k(a')\beta^{\widetilde{r}_k(a)}.$$

[9, Theorem 4.4.3], restated as Lemma 4, states that, as long as the reward $\widetilde{r}_k(a)$ is bounded, there exists a choice of β such that the weighted majority algorithm achieves $O(\sqrt{T})$ regret when compared to the optimal action.

Thus, to bound the convergence rate of the NPG algorithm, it is crucial to bound the relative value function $V_\pi(s)$, and thereby bound the state-action relative value $Q_k(s, a)$. In the finite-state setting, Even-Dar et al. [13] use their bounded-mixing time assumption to prove a universal bound on $V_\pi(s)$ over all policies π, states s, and actions a, which allows the NPG convergence rate proof to be completed.

In our infinite-state setting, we also prove bounds on $V_\pi(s)$, but we prove policy- and state-dependent bounds. Specifically, we prove strong enough bounds to complete the proof in a similar fashion to Even-Dar et al. [13], transferring bounds on the relative value function to bounds on the convergence rate. These relative-value bounds form the key novelty of this paper.

We now outline our lemmas, which primarily focus on bounding $V_\pi(s)$, towards our goal of proving Theorem 1, our convergence rate bound.

Step 1: Bounding change in relative value under NPG update The presence of unstable policies π shows that for general policies π, $V_\pi(s)$ may be unbounded. However, we don't need to bound $V_\pi(s)$ over all policies π. Instead, we focus on the specific policies π_k visited by the NPG algorithm, and bound the relative value functions $V_k(s)$ of only those policies.

We start with Lemma 5 by bounding how much worse $V_{k+1}(s)$ can be than $V_k(s)$, relative to the policy's hitting time $\tau_k(s)$ from states s to the special state $\vec{0}$. This lemma builds on Lemma 1, a standard monotonicity result for the NPG policy.

Step 2: Bounding high-reward states Next, in Lemma 6, we bound the time $\tau_\pi(s)$ for the system to hit $\vec{0}$, starting from a high-reward state s under policies π with high average reward J_π. Here, we make use of the fact that the NPG algorithm is known to monotonically increase the average reward at each iteration (Lemma 2), so this result applies to all iterates of the algorithm.

Step 3: Bounding all states Putting it all together, in Lemma 7, we bound the relative value $V_k(s)$ of the iterate policies in comparison to the relative value $V_0(s)$ of the initial policy, for arbitrary states s. Thus, if the initial policy has a well-behaved relative value function, then each iterate policy will also have a well-behaved relative value function, and we can prove fast convergence.

Step 4: Main result, using the quadratic assumption In Theorem 1, we specifically assume that the initial policy's relative value function $V_0(s)$ grows at most quadratically relative to the reward $r(s,a)$, an assumption that we show in Section 6 is satisfied by the MaxWeight policy in the setting of queueing MDPs.

As a result, we now have a bound on the relative value function $V_k(s)$ for each NPG iterate π_k which depends on the state s – the bound is not uniform over all states. We specify this bound in Lemma 8. Correspondingly, we set our learning rate β as a function of the state s, based on the bound on the relative value function that we are able to prove in that particular state. In states with more negative reward, our bound on $V_\pi(s)$ is weaker, so we use a slower learning rate, to improve convergence.

Now, with our state-dependent bounds on the relative value function for the NPG iterate policies, we are ready to employ the approach outlined at the beginning of this section: Thinking of the MDP optimization problem as many instances of the expert advice problem, and thinking of the NPG algorithm has many instances of the weighted majority algorithm for that problem.

In Theorem 1, we prove that by starting with a initial policy π_0 whose relative value function $V_0(s)$ grows at most quadratically with respect to reward $r(s,a)$, and by selecting the right β_s function, the NPG algorithm is guaranteed to achieve a $O(1/\sqrt{T})$ convergence rate. Our quadratic assumption on the initial policy is exactly strong enough to prove this convergence rate.

5 PROOFS

We start by stating background lemmas from the literature in Section 5.1. Then we proceed with our proof:

(1) In Lemma 5, we bound the change in relative value between two policies which are consecutive iterates of the NPG algorithm.
(2) In Lemma 6, we bound the hitting time $\tau_k(s)$ for high-reward states s under high-average-reward policies, such as the policies which are iterates of the NPG algorithm.
(3) In Lemma 7, we bound the relative value of all NPG iterates in all states s, in comparison to the initial policy π_0.

(4) In Section 5.5, we prove our main result, Theorem 1, making use of our quadratic-relative-value assumption on our initial policy π_0.

5.1 Background lemmas

We start by restating results from the literature. Note that Lemmas 1 and 2 were each originally proven in a setting with fixed β, but the proofs still hold unchanged in our setting with variable β_s.

First, we state two monotonicity results for the NPG algorithm, both proven in [23, Lemma 2].

LEMMA 1. *Policy π_{k+1} improves upon policy π_k relative to π_k's relative-value function Q_k:*

$$Q_k(s, \pi_{k+1}) \geq Q_k(s, \pi_k) = V_k(s)$$

PROOF. In [23, Lemma 2]. We restate the proof in Appendix A for added clarity. □

LEMMA 2. *Average reward increases monotonically: $J_{k+1} \geq J_k$.*

Lemma 2 follows immediately from Lemma 1 and the Performance Difference Lemma, Lemma 3, which relates the difference in average reward between two policies to their relative value functions:

LEMMA 3. *[8, (10)] For any pair of policies π, π',*

$$J_\pi - J_{\pi'} = E_{s \sim d_\pi}[Q_{\pi'}(s, \pi) - V_{\pi'}(s)].$$

In particular, Lemma 3 holds in our infinite-state setting. The proof of Lemma 3 given by Cao [8] requires only that the underlying Markov chains are ergodic, ensuring that the relevant quantities are well-defined.

Finally, we state a result on the weighted majority algorithm for the expert advice problem. The problem and algorithm are described in Section 4.2. Note that this paper operates in a reward maximization framework rather than the loss minimization framework of [9], so we invert $g(z)$ relative to that paper.

LEMMA 4. *[9, Theorem 4.4.3] Consider an instance of the expert advice problem, where for any step k and any pair of actions a, a', M is an upper bound on the value of $\widetilde{r_k}(a) - r_k(a')$. By selecting*

$$\beta = g\left(\sqrt{\frac{\ln |A|}{TM}}\right), \text{ where } g(z) = 1 + 2z + z^2/\ln 2, \tag{2}$$

the weighted majority algorithm achieves the following regret guarantee:

$$\sum_{k=1}^{T} r_k(a_*) - r_k(a_k) \leq \sqrt{TM \ln |A|} + \log_2(|A|)/2.$$

5.2 Step 1: Bounding change in relative value under NPG update

First, we bound the amount by which the relative value function of an NPG iterate $V_{k+1}(s)$ can be lower (i.e. worse) than $V_k(s)$, the previous policy iterate, relative to $\tau_{k+1}(s)$, the time to hit $\vec{0}$ from state s under policy π_{k+1}:

LEMMA 5. *For any two NPG policy iterates π_k and π_{k+1}, and any state s, we can lower bound $V_{k+1}(s)$:*

$$V_{k+1}(s) \geq V_k(s) - \tau_{k+1}(s)(J_{k+1} - J_k).$$

PROOF. To compare V_{k+1} and V_k, we will examine non-stationary policies which perform policy π_{k+1} for some number of steps, and then perform π_k afterwards. Note that such policies have an average reward of J_k, so their relative values are directly comparable to $V_k(s)$.

Let $V_k^n(s)$ denote the relative value of the policy which performs π_{k+1} for n steps before switching to π_k. Note that $V_k^0(s) = V_k(s)$, and that $V_k^1(s) = Q_k(s, \pi_{k+1})$.

Applying Lemma 1, we know that $V_k^1(s) \geq V_k^0(s)$. Furthermore, let us compare $V_k^2(s)$ and $V_k^1(s)$. Both policies start by applying π_{k+1} for one step, accruing the same reward and transitioning to the same distribution over states. From that point onward, Lemma 1 again tells us that $V_k^2(s) \geq V_k^1(s)$. In general, $V_k^n(s) \geq V_k^0(s)$ for any number of steps n. Even if n is chosen to be a stopping time, rather than a constant number of steps, $V_k^n(s) \geq V_k^0(s)$.

Let us consider the specific case where the number of steps n is the time to hit $\vec{0}$ from state s under the policy π_{k+1}. This policy has relative value $V_k^{\tau_{k+1}(s)}(s)$.

Because we define $V_\pi(\vec{0}) = 0$, for any policy π, to compute a policy's relative value, we need only examine its relative value over the first $\tau_\pi(s)$ steps. In particular,

$$V_\pi(s) = \mathbb{E}\left[\sum_{i=0}^{\tau_\pi(s)} (r(s_\pi(i), a_\pi(i)) - J_\pi)\right],$$

where $s_\pi(i), a_\pi(i)$ are the state and action taken by policy π on iteration i.

Applying this formula for $V_k^{\tau_{k+1}(s)}(s)$, we find that

$$V_k^{\tau_{k+1}(s)}(s) = \mathbb{E}\left[\sum_{i=0}^{\tau_{k+1}(s)} (r(s_{k+1}(i), a_{k+1}(i)) - J_k)\right] = V_{k+1}(s) + \tau_{k+1}(s)(J_{k+1} - J_k)$$

Applying the fact that $V_k^{\tau_{k+1}(s)}(s) \geq V_k(s)$, we find that

$$V_{k+1}(s) + \tau_{k+1}(s)(J_{k+1} - J_k) = V_k^{\tau_{k+1}(s)}(s) \geq V_k(s) \implies V_{k+1}(s) \geq V_k(s) - \tau_{k+1}(s)(J_{k+1} - J_k). \quad \square$$

5.3 Step 2: Bounding the time to hit the highest-reward state $\vec{0}$.

Now, we bound the time $\tau_\pi(s)$ for the system to hit the highest reward state $\vec{0}$, starting from any state s and under any policy π. Later, we will apply this bound when the initial state s has high reward $r_{\max}(s)$ and for policies π with high average reward $J_\pi(s)$. In doing so, we will make key use of Assumption 2, our assumption of uniform connectedness of states with high reward $r_{\max}(s)$.

Lemma 6. *Let y, z be two reward thresholds, $y > z$. For all states s with $r_{\max}(s) \geq z$ and all policies π such that $J_\pi \geq y$, the hitting time is bounded:*

$$\tau_\pi(s) \leq \tau_{y,z}^{bound} := \frac{x_z(c_{\max} - y)}{p_z^2(y - z)} + x_z$$

where x_z and p_z are the constants from Assumption 2.

Proof. Let $\mathbb{S}_z$ be the set of states s such that $r_{\max}(s) \geq z$. For each state $s \in \mathbb{S}_z$, let $t_{\pi,z}(s)$ be the expected amount of time until policy π next reaches a state $s' \in \mathbb{S}_z$, starting in state s. Let $t_{\pi,z}^{\max}$ be the maximum over $s \in \mathbb{S}_z$ of $t_{\pi,z}(s)$. Let s^* be the state in $\mathbb{S}_z$ which achieves this maximum. By Assumption 2, every state $s \in \mathbb{S}_z$ has at least p_z probability of moving to state s^* in the next x_z steps, and therefore spending at most x_z of the next $t_{\pi,z}^{\max} + x_z$ steps inside $\mathbb{S}_z$.

Let us use a renewal-reward argument, where the renewal moments are steps on which the system visits state s^*. Let us subdivide this renewal period into three segments:

(1) Starting at s^*, the time until a state $s \in S_z$ is next visited.
(2) Starting from a state $s \in S_z$, the next x_z steps, during which there is at least a p_z probability of reaching the state s^*.
(3) Any time not in either of these two segments, which must be spent outside of S_z.

From these segments, we can upper-bound the fraction of time during the renewal period that the system spends in $\mathbb{S}_z$: The system can only be in $\mathbb{S}_z$ during Item 2, and the system can spend at most

$\frac{1}{p_z}$ periods of length at most x_z in that segment. The fraction of time spent in $\mathbb{S}_z$ is maximized if no time is spent in Item 3, resulting in the following bound:

$$\mathbb{P}\{s_t \in \mathbb{S}_z\} \leq \frac{x_z/p_z}{t^{\max}_{\pi,z} + x_z/p_z}$$

Recalling that the system accrues at most $c_{\max}$ reward inside of $\mathbb{S}_z$ and at most z reward outside of z, we can conclude that the long-term reward of the policy π is at most

$$J_\pi \leq c_{\max}\frac{x_z/p_z}{t^{\max}_{\pi,z} + x_z/p_z} + z\frac{t^{\max}_{\pi,z}}{t^{\max}_{\pi,z} + x_z/p_z} = \frac{x_z c_{\max} + p_z t^{\max}_{\pi,z} z}{x_z + p_z t^{\max}_{\pi,z}} \tag{3}$$

However, we know that $J_\pi \geq y$. We can therefore use (3) to bound $t^{\max}_{\pi,z}$:

$$y \leq \frac{x_z c_{\max} + p_z t^{\max}_{\pi,z} z}{x_z + p_z t^{\max}_{\pi,z}} \implies t^{\max}_{\pi,z} \leq \frac{x_z(c_{\max} - y)}{p_z(y-z)}$$

Now, we can bound the time until we reach $\vec{0}$, starting from s. Whenever the system is in a state in $\mathbb{S}_z$, there is a p_z chance of reaching $\vec{0}$ in the next x_z steps, and a $t^{\max}_{\pi,z}$ expected time until we next get the opportunity.

Let $p_{\pi,z}(s)$ be the probability that the system reaches $\vec{0}$ from s in at most x_z steps. Let $t'_{\pi,z}(s)$ be the expected time until the system re-enters $\mathbb{S}_z$, conditional on not reaching $\vec{0}$ in at most x_z steps. Note that by Markov's inequality, $t'_{\pi,z}(s) \leq \frac{t_{\pi,z}(s)}{1-p_{\pi,z}(s)}$. We can now start to bound $\tau_\pi(s)$ as follows:

$$\tau_\pi(s) \leq p_{\pi,z}(s)x_z + (1 - p_{\pi,z}(s))(t'_{\pi,z}(s) + \tau_\pi(s'))$$

where s' is the state at which we re-enter $\mathbb{S}_z$. Thus,

$$\tau_\pi(s) \leq p_{\pi,z}(s)x_z + (1-p_{\pi,z}(s))\left(\frac{t_{\pi,z}(s)}{1-p_{\pi,z}(s)} + \tau_\pi(s')\right) = x_z + t_{\pi,z}(s) + (1-p_{\pi,z}(s))(\tau_\pi(s') - x_z)$$

In particular, letting $\tau^{\max}_\pi$ be the maximum over $s \in \mathbb{S}_z$ of $\tau_\pi(s)$, and letting s^*_τ be the state in which that maximum occurs, we have

$$\tau^{\max}_\pi \leq x_z + t_{\pi,z}(s^*_\tau) + (1-p_{\pi,z}(s^*_\tau))(\tau^{\max}_\pi - x_z)$$
$$\tau^{\max}_\pi \leq x_z + \frac{t_{\pi,z}(s^*_\tau)}{p_{\pi,z}(s^*_\tau)} \leq x_z + \frac{t^{\max}_{\pi,z}}{p_z} = \frac{x_z(c_{\max}-y)}{p_z^2(y-z)} + x_z. \qquad \square$$

5.4 Step 3: Bounding relative value $V_k(s)$ for all states s

We are now ready to bound the relative value function V_k for all iterates π_k of the Natural Policy Gradient algorithm, and for all states s. Lemma 6 covered high-reward states, so this lemma focuses on low-reward states, building off of Lemma 5 to do so.

Lemma 7. *For any reward threshold $z < J_0$, and for each iterate π_k of the NPG algorithm, the relative value function is lower and upper bounded as follows:*

$$\forall s,k, \quad V_k(s) \geq \min\left(V_0(s)\frac{J_*-z}{J_0-z}, V_0(s)\right) - \tau^{bound}_{J_0,z}\frac{(J_*-J_0)(c_{\max}-z)(J_*-z)}{(J_0-z)^2}$$
$$\forall s,k, \quad V_k(s) \leq \tau^{bound}_{J_0,z}(c_{\max} - J_0),$$

where $\tau^{bound}_{y,z}$ is the hitting-time bound from Lemma 6. Note that these bounds do not depend on the iteration k, and they only depend on the state s via the initial policy's relative value $V_0(s)$.

Proof. To lower bound $V_k(s)$, let us start by using the following formula for $V_k(s)$:

$$V_k(s) = -J_k \tau_k(s) + \bar{r}_k(s)\tau_k(s)$$

where $\bar{r}_k(s)$ is the average reward that policy π_k accrues over the interval until it first enters state $\vec{0}$.

Let us subdivide the time $\tau_k(s)$ into two periods: Let τ_k^{out} be the time until the policy π_k first enters $\mathbb{S}_z$, and let τ_k^{in} be the time from then until the policy first enters state $\vec{0}$. We define $\bar{r}_k^{out}$ and $\bar{r}_k^{in}$ similarly.

Similarly, let us define $V_k^{out}(s)$ and $V_k^{in}(s)$:

$$V_k^{out}(s) = -J_k \tau_k^{out}(s) + \bar{r}_k^{out}(s)\tau_k^{out}(s), V_k^{in}(s) = -J_k \tau_k^{in}(s) + \bar{r}_k^{in}(s)\tau_k^{in}(s)$$

Note that $V_k(s) = V_k^{out}(s) + V_k^{in}(s)$

We can upper bound each of these quantities. Note that $\bar{r}_k^{out}(s) \leq z$, and $\bar{r}_k^{in}(s) \leq c_{\max}$. Note that $\tau_k^{in}(s) \leq \tau_{J_0,z}^{bound}$, by Lemma 6. Together, we can upper bound $V_k(s)$ relative to $\tau_k(s)$, or equivalently upper bound $\tau_k(s)$ relative to $V_k(s)$:

$$V_k(s) \leq -J_k \tau_k(s) + z\tau_k(s) + (c_{\max} - z)\tau_{J_0,z}^{bound}$$

$$\frac{V_k(s) - (c_{\max} - z)\tau_{J_0,z}^{bound}}{-J_k + z} \geq \tau_k(s)$$

Note that $-J_k + z < 0$, so the direction of the inequality flips.

The key fact relating $V_k(s)$ and $V_{k+1}(s)$ is Lemma 5: $V_{k+1}(s) \geq V_k(s) - \tau_{k+1}(s)(J_{k+1} - J_k)$. Combining our bounds, we find that

$$V_{k+1}(s) \geq V_k(s) - \frac{V_{k+1}(s) - (c_{\max} - z)\tau_{J_0,z}^{bound}}{-J_{k+1} + z}(J_{k+1} - J_k)$$

$$V_{k+1}(s) \geq V_k(s)\frac{J_{k+1} - z}{J_k - z} - \tau_{J_0,z}^{bound}(J_{k+1} - J_k)\frac{(c_{\max} - z)}{J_k - z}$$

Now, we apply this bound telescopically, for all $\pi_i \in [0, k]$. By doing so, we find that

$$V_k(s) \geq V_0(s)\frac{J_k - z}{J_0 - z} - \tau_{J_0,z}^{bound}\sum_{i=0}^{k-1}(J_{i+1} - J_i)\frac{c_{\max} - z}{J_i - z}\frac{J_k - z}{J_{i+1} - z}$$

Applying the monotonicity bound $J_0 \leq J_k$ (Lemma 2) and the optimality bound $J_k \leq J_*$,

$$V_k(s) \geq \min(V_0(s)\frac{J_* - z}{J_0 - z}, V_0(s)) - \tau_{J_0,z}^{bound}\frac{(J_* - J_0)(c_{\max} - z)(J_* - z)}{(J_0 - z)^2}$$

That completes the proof of the lower bound.

To upper bound $V_k(s)$, let's again focus on $V_k^{out}(s)$ and $V_k^{in}(s)$.

$$V_k^{out}(s) = -J_k \tau_k^{out}(s) + \bar{r}_k^{out}(s)\tau_k^{out}(s) \leq -J_k \tau_k^{out}(s) + z\tau_k^{out}(s) = (-J_k + z)\tau_k^{out}(s) \leq 0$$

$$V_k^{in}(s) = -J_k \tau_k^{in}(s) + \bar{r}_k^{in}(s)\tau_k^{in}(s) \leq (c_{\max} - J_k)\tau_k^{in}(s) \leq (c_{\max} - J_0)\tau^b ound_{J_0,z}$$

$$V_k(s) = V_k^{out}(s) + V_k^{in}(s) \leq (c_{\max} - J_0)\tau^b ound_{J_0,z}$$

□

Now, we combine Lemma 7 with an assumption on the initial policy and our mild structural assumption on the MDP, Assumption 1, to give a state-dependent bound on $Q_k(s, a)$:

Lemma 8. *Given an MDP satisfying Assumption 1, and given an initial policy π_0 such that there exist constants $c_0 > 0, c_1 \geq 0$ such that*

$$V_0(s) \geq -c_0 \hat{r}_{\max}(s)^2 - c_1,$$

there exists a uniform bound M_s:

$$M_s := c_2 |\hat{r}_{\max}(s)|^2 + c_3 |\hat{r}_{\max}(s)| + c_4,$$

for constants $c_2, c_3, c_4 \geq 0$ depending on c_0, c_1, and the MDP parameters, such that for any NPG iterate π_k, and any pair of actions a, a',

$$Q_k(s,a) - Q_k(s,a') \leq M_s.$$

Proof. Deferred to Appendix B. □

5.5 Step 4: Proof of main result

With our bound Lemma 8 on the relative value function $V_k(s)$ of the iterates of the NPG algorithm, we are now ready to prove our main result on the convergence of the NPG algorithm, Theorem 1.

Theorem 1. *For any average-reward MDP satisfying Assumptions 1 and 2, given an initial policy π_0 such that there exist constants $c_0 > 0, c_1 \geq 0$ such that*

$$V_0(s) \geq -c_0 \hat{r}_{\max}(s)^2 - c_1, \tag{4}$$

the NPG algorithm with learning rate parameterization β_s given in (5) *achieves the convergence rate $J_* - J_T \leq \frac{c_*}{\sqrt{T}}$, where c_* is a constant depending on the MDP parameters and on c_0 and c_1.*

Proof. As outlined in Section 4.2, we will think of the MDP optimization problem as many instances of the expert advice problem with reward $Q_k(s,a)$. Note that the NPG algorithm, Algorithm 1, is exactly identical to the weighted majority algorithm, executing a parallel instance of the weighted majority algorithm in every state s.

Moreover, the MDP objective of maximizing the average reward J_π is closely related to the objective of maximizing total reward in the expert advice problem. Recall the Performance Difference Lemma, Lemma 3, which states that for any pair of policies π, π':

$$J_\pi - J_{\pi'} = E_{s \sim d_\pi}[Q_{\pi'}(s,\pi) - V_{\pi'}(s)].$$

Let us apply this lemma for the iterates π_k of our algorithm, in comparison to the optimal policy π_*, summing over all iterates $k \in [1, T]$:

$$\sum_{k=1}^T J_* - J_k = E_{s \sim \pi_*}\left[\sum_{k=1}^T Q_k(s,\pi_*) - V_k(s)\right]$$

Note that for any specific state s, $\sum_{k=1}^T Q_k(s,\pi_*) - V_k(s)$ is exactly the difference in total reward between a specific fixed policy $\pi_*(a \mid s)$ and the weighted majority policy $\pi_k(a \mid s)$ of the expert advice problem with with reward function at time step k of $Q_k(s,a)$.

Thus, we can apply Lemma 4, a known regret bound on the performance of the weighted majority algorithm for the expert advice problem, to bound the convergence rate of the NPG algorithm. To do so, we use Lemma 8, which states that for all NPG intermediate policies π_k and actions a, a',

$$Q_k(s,a) - Q_k(s,a') \leq M_s$$
$$\text{where } M_s := c_2 |\hat{r}_{\max}(s)|^2 + c_3 |\hat{r}_{\max}(s)| + c_4,$$

for some positive constants c_2, c_3, c_4 given in the proof of Lemma 8. In particular, M_s can be computed ahead of time, given only the structure of the MDP and the initial policy π_0.

We select the learning rate β_s for state s as given in Lemma 4:

$$\beta_s := g\left(\sqrt{\frac{\ln |A|}{TM_s}}\right), \text{where } g(z) = 1 + 2z + z^2/\ln 2. \tag{5}$$

Thus, we may apply Lemma 4, and thereby obtain the following regret guarantee for the NPG algorithm:

$$\begin{aligned}\sum_{k=1}^{T} Q_k(s,\pi_*) - V_k(s) &\leq \sqrt{TM_s \ln |A|} + \log_2(|A|)\\ &= \sqrt{T\ln|A|}\sqrt{c_2|\hat{r}_{\max}(s)^2| + c_3|\hat{r}_{\max}(s)| + c_4} + \log_2(|A|)/2\\ &\leq \sqrt{T\ln|A|}(c_5|\hat{r}_{\max}(s)| + c_6) + \log_2(|A|)/2\end{aligned}$$
$$\text{where } c_5 := \sqrt{c_2} + \sqrt{c_3}, \text{ and } c_6 := \sqrt{c_3}/4 + \sqrt{c_4}.$$

Now, let's bound the difference in average reward between the iterates and the optimal policy:

$$\sum_{k=1}^{T} J_* - J_k \leq \sqrt{T\ln|A|}(c_5 E_{s\sim\pi_*}[|\hat{r}_{\max}(s)|] + c_6) + \log_2(|A|)/2$$

Here, we see the importance of our quadratic assumption on π_0, (4). We know that $J_* = E_{s\sim\pi_*}[r(s,\pi_*(s))]$ is finite and small. This is essentially the only property we know about the optimal policy π_*. Because of our quadratic assumption, we have shown that J_k depends crucially on $E_{s\sim\pi_*}[|\hat{r}_{\max}(s)|]$. If π_0 had a faster-growing relative value function, we would only be able to relate J_k to $E_{s\sim\pi_*}[|\hat{r}_{\max}(s)|^\alpha]$ for some $\alpha > 1$, which we would not be able to bound.

Now that we have shown that, under our assumption, the NPG algorithm's regret depends on $E_{s\sim\pi_*}[|\hat{r}_{\max}(s)|]$, note that $|\hat{r}_{\max}(s)| \leq |\hat{r}(s,a)| = c_{\max} - r(s,a)$ for any action a. As a result,

$$\begin{aligned}\sum_{k=1}^{T} J_* - J_k &\leq \sqrt{T\ln|A|}(c_5 E_{s\sim\pi_*}[c_{\max} - r(s,\pi_*)] + c_6) + \log_2(|A|)/2\\ &= \sqrt{T\ln|A|}(c_5(c_{\max} - J_*) + c_6) + \log_2(|A|)/2\end{aligned}$$

Thus, because $T \geq 1$, we have

$$\sum_{k=1}^{T} J_* - J_k \leq c_*\sqrt{T}, \text{ where } c_* := \sqrt{\ln|A|}(c_5(c_{\max} - J_*) + c_6) + \log_2(|A|)/2.$$

Finally, by Lemma 2, we know that $J_T \geq J_k$ for all iterates k. As a result,

$$J_* - J_T \leq \frac{c_*}{\sqrt{T}}. \qquad \square$$

6 QUEUEING MDPS

We have proven Theorem 1, which gives a convergence rate bound for the Natural Policy Gradient (NPG) algorithm in infinite-state, average-reward MDPs that satisfy Assumptions 1 and 2, given an assumption on the quality of the initial policy. In this section, we demonstrate that a broad, natural family of MDPs arising from queueing theory, known as the generalized switch with static environment (GSSE) setting "[29], satisfy these assumptions. Moreover, we prove that a natural and well-studied policy, the MaxWeight policy, satisfies the requirement to be used as an initial policy in Theorem 1. We therefore demonstrate that NPG efficiently converges to the optimal policy in this broad natural class, when initialized with the MaxWeight policy.

We define the GSSE setting in Section 6.1. This setting is broad enough to capture a wide variety of natural queueing problems, including the N-system, the input-queued switch, and the multiserver-job setting. We primarily focus on the optimization goal of minimizing mean queue length $E[q]$, which corresponds to the reward function equal to the negative total queue length.

Our results also apply to generalizations such as expected weighted queue length $E[cq]$ or the αth moment of queue length $E[q^\alpha]$.

In Section 6.2, we discuss prior work on MaxWeight and the Generalized Switch setting. In Section 6.5, we prove that NPG with MaxWeight initialization meets the requirements for Theorem 1 and converges rapidly, using lemmas built up in Sections 6.3 and 6.4.

6.1 Definition of Generalized Switch with Static Environment

The Generalized Switch with Static Environment (GSSE) is a queueing model with n classes of jobs, and the state of the system is a size-n vector $q = \{q_i\}$ specifying the number of jobs of each class present in the system. This model is a special-case of the Generalized Switch model [29].

The system evolves in discrete time: At each time step, v_i jobs of class i arrive, where v_i is a bounded, i.i.d. random variable. Jobs of each class arrive independently.

There are m service options. At each time step, prior to new jobs arriving, the scheduling policy selects a service option j. If the policy selects service option j, then w_i^j jobs of class i complete, where w_i^j is a bounded, i.i.d. random variable. Jobs of each class complete independently. If on some time step t, $w_i^j(t)$ exceeds $q_i(t)$, the number of jobs of class i present in the system, then all of the class i jobs complete.

We use the notation v, w instead of the more common a, s to avoid collision with the action and state notation defined in Section 3.

Let ℓ be the maximum number of jobs of any class that can arrive or depart in one time slot: The maximum over all v_i and all w_i^j. Let $\lambda_i = E[v_i]$ be the average arrival rate of class i jobs, and let $\mu_i^j = E[w_i^j]$ be the average completion rate of class i jobs under service option j.

We make certain non-triviality assumptions to ensure that the state space is not disconnected under any scheduling policy, namely:

- For each class i, it is possible for no class i jobs to arrive ($P(v_i = 0) > 0$).
- For each class i, it is possible for a class i job to arrive ($P(v_i > 0) > 0$).
- For each pair (i, j), it is possible for more jobs arrive than depart ($P(w_i^j > v_i) > 0$).
- For each pair (i, j), it is possible for equally many jobs to arrive as depart ($P(w_i^j = v_i) > 0$).

Less restrictive assumptions are possible – we make these assumptions for simplicity.

Finally, we assume that the scheduling policy π is non-idling: If there are any jobs in the system, we assume that the policy selects a service option j with a nonzero chance of completing a job. Subject to that restriction, the scheduling policy may be arbitrary.

Our primary reward function $r(s, \cdot)$ of state $s = \{q_i\}$ under any action is the negative total queue length $r(s, \cdot) = -\sum_i q_i$. We will also consider the setting where the reward is the negative α-th moment of the total queue length, $-(\sum_i q_i)^\alpha$, for a generic $\alpha \geq 1$.

Many commonly-studied queueing models fall within the GSSE framework, with appropriate choices of arrival distributions and service options. For instance, the input-queued switch [22], the multiserver-job model [16], and parallel server systems [18] such as the N-system [17], all fit within the GSSE framework.

6.2 Prior work on MaxWeight and Generalized Switch

The Generalized Switch is an expansive queueing model, simultaneously generalizing the input-queued switch and the parallel server system [29]. An important scheduling policy in the Generalized Switch model is the MaxWeight policy [31], which chooses the service option which maximizes the inner product of the queue lengths and the service rates:

$$\text{MaxWeight}(q) := \arg\max_j \sum\nolimits_i q_i \mu_i^j \tag{6}$$

MaxWeight is known to have many advantageous properties in the *heavy traffic* limit. The heavy traffic limit is the limit in which the arrival rates approach the boundary of the capacity region. In the GSSE setting, the capacity region is the convex hull of the available service rates μ_i^j.

In particular, MaxWeight is known to be throughput optimal, meaning that it keeps the system stable for all arrival rates within the capacity region [19, 29]. Moreover, MaxWeight asymptotically minimizes the total queue length $\sum_i q_i$ in the heavy traffic limit [29].

Outside of the heavy-traffic limit, in non-asymptotic regimes, it is known that MaxWeight can be significantly outperformed, especially by methods based on MDP optimization. For instance, in the N-system, a threshold-based policy is known to significantly outperform MaxWeight at nonasymptotic arrival rates [4], and optimization-based methods are needed to find the optimal policy in the non-asymptotic arrival-rate regime [12].

6.3 Framework for bounding $V_\pi(s)$

First, we show how to bound $V_\pi(s)$ using a Lyanpunov function argument.

Lemma 9. *Under a given policy π, suppose that there exists a function $f(s) \geq 0$ and constants $c_1 > 0, c_2 \geq 0$ such that*

$$\forall s, \quad E_\pi[f(s_{t+1}) - f(s_t) \mid s_t = s] \leq c_1 r(s, \pi) + c_2.$$

Then there exist explicit constants $c_3, c_4 \geq 0$ such that $V_\pi(s) \geq -c_3 f(s) - c_4$.

Proof. We will start by rescaling the function $f(s)$ to more closely relate it to the value function. Let $\widetilde{f}(s) = \frac{2}{c_1} f(s)$. Then we have

$$E_\pi[\widetilde{f}(s_{t+1}) - \widetilde{f}(s_t) \mid s_t = s] \leq 2r(s, \pi) + 2c_2/c_1$$

Next, let c_5 be the reward threshold $c_5 = -\frac{2c_2}{c_1} - J_\pi$. For states s for which $r_{\max}(s) \leq c_5$, we have

$$E_\pi[\widetilde{f}(s_{t+1}) - \widetilde{f}(s_t) \mid s_t = s] \leq r(s, \pi) - J_\pi$$

Next, letting $s_0 = s$, let τ^{c_5} be the first positive time when the system enters $\mathbb{S}_{c_5}$, the set of states for which $r_{\max}(s) \geq c_5$. Let $V^{c_5}(s)$ be the expected relative value after this point in time:

$$\sum\nolimits_{t=0}^{\tau^{c_5}} E_\pi[\widetilde{f}(s_{t+1}) - \widetilde{f}(s_t)] \leq \sum\nolimits_{t=0}^{\tau^{c_5}} r(s_t) - J_\pi$$
$$\widetilde{f}(s_{\tau^{c_5}}) - \widetilde{f}(s) \leq V_\pi(s) - V^{c_5}(s)$$
$$V_\pi(s) \geq V^{c_5}(s) + \widetilde{f}(s_{\tau^{c_5}}) - \widetilde{f}(s).$$

Because $f(s) \geq 0$, we know that $\widetilde{f}(s) \geq 0$. Because $\mathbb{S}_{c_5}$ is a finite set, V^{c_5} is bounded below:

$$V^{c_5}(s) \geq V_{\min} := \min_{s \in \mathbb{S}_{c_5}} V(s)$$
$$V_\pi(s) \geq V_{\min} - \widetilde{f}(s) = -\frac{2}{c_1} f(s) + V_{\min}.$$

Setting $c_3 = \frac{2}{c_1}$ and $c_4 = (-V_{\min})^+$ completes the proof. □

6.4 Bounding $V_{MW}(s)$ for MaxWeight

Now, we use the framework established in Section 6.3 to bound the relative value function for the MaxWeight policy.

The stability region of a queueing system is defined to be the set of average arrival rates λ_i for which there exists a scheduling policy such that the system is stable, i.e. the system state converges to a stationary distribution. The GSSE stability region is the convex hull of the mean service rates μ_i^j. The optimal stability region is achieved by the MaxWeight policy, defined in (6).

The GSSE stability region is defined by the following constraints: For an arrival rate vector λ to be stable, there must exist a nonnegative service vector $\gamma \geq 0$ and a slack margin $\epsilon > 0$ such that

$$\sum_j \gamma_j = 1 \quad \text{and} \quad \forall i, (1+\epsilon)\lambda_i \leq \sum_j \gamma_j \mu_i^j. \tag{7}$$

We can bound MaxWeight's relative value $V_{MW}(q)$ if we can prove a Lyapunov function result of the form given by Lemma 9. We prove such a result using the Lyapunov function $f(q) = \sum_i q_i^2$, which implies that $V_{MW}(q)$ is $O(q^2)$.

Lemma 10. *For any GSSE queueing system,*

$$E_{MW}[f(q(t+1)) - f(q(t)) \mid q(t) = q] \leq c_1 r(q(t)) + c_2,$$

where MW denotes the MaxWeight policy, where $f(q) := \sum_i q_i^2$, and where c_1 and c_2 are:

$$c_1 = 2\epsilon \min_i \lambda_i, \qquad c_2 = \ell^2 n$$

Proof. This result is proven in [28, Theorem 4.2.4], for the classic switch model. However, the proof applies unchanged to any GSSE system. For completeness, we reprove the result in the GSSE setting in Appendix C.1. □

6.5 Rapid convergence for NPG on queueing MDPs

Theorem 2. *For any GSSE queueing system, with the objective of minimizing mean queue length starting with the initial policy π_0 =MaxWeight, the NPG algorithm with learning rate parameterization β_s given in Theorem 1 achieves the convergence rate $E[q_{\pi_T}] \geq E[q_*] - \frac{c_*}{\sqrt{T}}$ for a constant c_* given in the proof of Theorem 1.*

Proof. To prove this result, we will apply Theorem 1. To do so, we will verify Assumptions 1 and 2 and prove the MaxWeight achieves the required bound (4) on the relative value function.

Let us start by verifying Assumption 1.

- Assumption 1(a): The maximum possible reward is 0, achieved by the all-zeros state $q_i = 0$.
- Assumption 1(b): The set of states with total reward at least z is the set of states with total queue length at most $|z|$. As a generous bound, there are at most $n^{|z|}$ such states, recalling that n is the number of job classes.
- Assumption 1(c): In each state, the reward under all actions is equal.
- Assumption 1(d): If it is possible to transition from state s and s', then s and s' have similar numbers of jobs, differing by at most ℓn, where n is the number of job classes and ℓ is the maximum arrivals or departures per class in a step. As a result, the maximum reward in those states also differs by at most ℓn.

Now, let's turn to Assumption 2, the assumption that the set of states with at most $|z|$ jobs present is uniformly connected under an arbitrary policy π.

In particular, we will show that there is a nonzero probability that, for any pair of states q, q' with at most $|z|$ jobs present, the system moves from q to q' in at most $2|z|$ time steps. More specifically, there is a nonzero probability that the system moves from q to $\vec{0}$ in at most $|z|$ time steps, and then moves from $\vec{0}$ to q' in at most $|z|$ time steps.

To see why, first recall from Section 6.1 that we require that the policy π is non-idling: If the system state is not the all-zero state, the π selects a service option which has a nonzero probability of completing a job. Recall also that on each time step, there's a nonzero probability that no new job arrives. Thus, on each of the first $|z|$ time steps, if the system is not yet empty there is a nonzero probability that at least one job completes and no new job arrives. At some point during those steps, we will reach the all-zeros state.

Recall also that we require that for any service option j, for each class i there is a nonzero chance that more jobs arrive than complete, and a nonzero chance that the same number of jobs arrive as complete. Thus, over the next $|z|$ time steps, regardless of the policy selected, there is a nonzero chance that the number of jobs present of class i rises from 0 to q'_i, and then stays at q'_i, allowing the system to reach the state q. This verifies Assumption 2.

Finally, let us verify that the MaxWeight initial policy satisfies the desired relationship,

$$V_{MW}(s) \geq -c_0 \hat{r}_{\max}(s)^2 - c_1,$$

for some constants c_0 and c_1.

To do so, we simply apply the Lyapunov function argument from Lemma 9, using the Lyapunov function $f(q) = \sum_i q_i^2$, which we verified is a valid Lyapunov function in Lemma 10.

Thus, Theorem 1 applies to any GSSE queueing MDP, given the MaxWeight initial policy. □

This result can be generalized to other reward functions, including a weighted queue length reward, as well as the α-moment reward function $-(\sum_i q_i)^\alpha$, for any $\alpha \geq 1$. For these generalizations of the reward function, we can use weighted-MaxWeight and α-MaxWeight as the initial policies. These policies replace the q_i term in the MaxWeight definition (6) with the weighted queue length or αth power of queue length. The proof that weighted-MaxWeight has the desired relative value property (4) is essentially immediate, and for α-MaxWeight one can use the Lyapunov function $f_\alpha(q) = \sum_i q_i^{\alpha+1}$. Assumptions 1 and 2 are straightforward to verify for the weighted queue length setting, and Assumption 1(a-c) and Assumption 2 are straightforward to verify for the αth moment reward function. Assumption 1(d) is slightly more involved, but we verify that it holds in Appendix C.2.

7 INFINITE STATE, DISCOUNTED REWARD

In this section, we discuss the discounted setting: Rather than seeking to maximize the average reward, we seek to maximize the γ-discounted reward, for some $\gamma < 1$, in expectation over some starting distribution η. In this setting, one can generalize existing finite-state results to the infinite-state setting. In particular, we can state the following theorem:

Theorem 3. *In the infinite-state discounted-reward setting, if the discounted relative-value function V_η^π is uniformly bounded over all policies π for a given initial distribution η, then the NPG algorithm converges to the globally optimum discounted reward with $O(1/T)$ convergence rate.*

Proof. The proof of [3, Theorem 16] goes through nearly unchanged. The assumption of a finite state space is only to justify a bounded reward assumption, which in turn is only used in [3, Lemma 17], to prove that V_η^π, the expected discounted reward starting in initial-state-distribution η, is bounded. In the infinite-state setting, if V_η^π is bounded, the proof still holds. □

This standard proof only uses its finite-state assumption to justify its bounded reward assumption. In particular, even when the state space is infinite, if the reward is bounded, the proof goes through unchanged. This is due in part to the fact that in the discounted-reward setting, a policy need not be stable to achieve finite reward.

In particular, if the reward is unbounded, but only grows polynomially from state to state, such as in the queueing setting, one can still prove that V_η^π is bounded, and still prove a convergence-rate bound. More generally, it is sufficient for the worst-case discounted sum of rewards to be finite. The dependence on the discount rate γ degrades in the $\gamma \to 1$ limit, depending on the growth rate of the reward, but the dependence on the number of iterations T is unchanged.

Thus, in the discounted-reward setting, the standard proof of NPG's convergence does generalize from the finite-state setting to the infinite-state setting, in contrast to the average-reward setting studied in this paper.

8 CONCLUSION

We give the first proof of convergence for the Natural Policy Gradient algorithm in the infinite-state average-reward setting. In particular, we demonstrate that by setting the learning rate function β_s appropriately, we can guarantee an $O(1/\sqrt{T})$ convergence rate to the optimal policy, as long as the initial policy has a well-behaved relative value function. Moreover, we demonstrate that in infinite-state average-reward MDPs arising out of queueing theory, the MaxWeight policy satisfies our requirement on the initial policy, allowing our results on the NPG algorithm to apply.

One potential direction for future work would be to tighten the dependence of our convergence rate result on the structural parameters of the MDP, which we did not seek to optimize in this result. Another potential direction would be to extend our result to an uncountably-infinite-state setting, rather than the countably-infinite setting that we focused on, by tweaking the assumptions made in Section 3.2.

In the reinforcement learning (RL) setting, an important direction for future work would be to build off of our MDP optimization result to prove convergence-rate results for policy-gradient-based RL algorithms in the infinite-state setting.

REFERENCES

[1] Yasin Abbasi-Yadkori, Peter Bartlett, Kush Bhatia, Nevena Lazic, Csaba Szepesvari, and Gellért Weisz. Politex: Regret bounds for policy iteration using expert prediction. In *International Conference on Machine Learning*, pages 3692–3702. PMLR, 2019.

[2] Saghar Adler and Vijay Subramanian. Bayesian learning of optimal policies in markov decision processes with countably infinite state-space. *arXiv preprint arXiv:2306.02574*, 2023.

[3] Alekh Agarwal, Sham M. Kakade, Jason D. Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *J. Mach. Learn. Res.*, 22(1), jan 2021. ISSN 1532-4435.

[4] S. L. Bell and R. J. Williams. Dynamic scheduling of a system with two parallel servers in heavy traffic with resource pooling: asymptotic optimality of a threshold policy. *The Annals of Applied Probability*, 11(3):608 – 649, 2001. doi: 10.1214/aoap/1015345343. URL https://doi.org/10.1214/aoap/1015345343.

[5] Dimitri Bertsekas. *Dynamic programming and optimal control: Volume I*, volume 4. Athena scientific, 2012.

[6] Jalaj Bhandari and Daniel Russo. Global optimality guarantees for policy gradient methods. *arXiv preprint arXiv:1906.01786*, 2019.

[7] S.J. Bradtke, B.E. Ydstie, and A.G. Barto. Adaptive linear quadratic control using policy iteration. In *Proceedings of 1994 American Control Conference - ACC '94*, volume 3, pages 3475–3479 vol.3, 1994. doi: 10.1109/ACC.1994.735224.

[8] Xi-Ren Cao. Single sample path-based optimization of markov chains. *Journal of optimization theory and applications*, 100:527–548, 1999.

[9] Nicolò Cesa-Bianchi, Yoav Freund, David Haussler, David P. Helmbold, Robert E. Schapire, and Manfred K. Warmuth. How to use expert advice. *J. ACM*, 44(3):427–485, may 1997. ISSN 0004-5411. doi: 10.1145/258128.258179. URL https://doi.org/10.1145/258128.258179.

[10] Ethan Che, Hongseok Namkoong, and Jing Dong. Dynamic control of queuing networks via differentiable discrete-event simulation. In *ICML 2023 Workshop on Differentiable Almost Everything: Differentiable Relaxations, Algorithms, Operators, and Simulators*, 2023.

[11] Yi Chen, Jing Dong, and Zhaoran Wang. A primal-dual approach to constrained markov decision processes with applications to queue scheduling and inventory management. URL http://www.columbia.edu/~jd2736/publication/CMDP.pdf. Working paper.

[12] J. G. Dai and Mark Gluzman. Queueing network controls via deep reinforcement learning. *Stochastic Systems*, 12(1): 30–67, 2022. doi: 10.1287/stsy.2021.0081.

[13] Eyal Even-Dar, Sham. M. Kakade, and Yishay Mansour. Online markov decision processes. *Mathematics of Operations Research*, 34(3):726–736, 2009. doi: 10.1287/moor.1090.0396.

[14] Maryam Fazel, Rong Ge, Sham Kakade, and Mehran Mesbahi. Global convergence of policy gradient methods for the linear quadratic regulator. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1467–1476. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/fazel18a.html.

[15] Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized Markov decision processes. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2160–2169. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/geist19a.html.

[16] Isaac Grosof, Mor Harchol-Balter, and Alan Scheller-Wolf. WCFS: A new framework for analyzing multiserver systems. *Queueing Systems*, 102(1-2):143–174, 2022.

[17] J. Michael Harrison. Heavy traffic analysis of a system with parallel servers: asymptotic optimality of discrete-review policies. *The Annals of Applied Probability*, 8(3):822 – 848, 1998. doi: 10.1214/aoap/1028903452. URL https://doi.org/10.1214/aoap/1028903452.

[18] J Michael Harrison and Marcel J López. Heavy traffic resource pooling in parallel-server systems. *Queueing systems*, 33:339–368, 1999.

[19] Daniela Andrea Hurtado Lange and Siva Theja Maguluri. Heavy-traffic analysis of queueing systems with no complete resource pooling. *Mathematics of Operations Research*, 47(4):3129–3155, 2022. doi: 10.1287/moor.2021.1248.

[20] Sham M Kakade. A natural policy gradient. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001. URL https://proceedings.neurips.cc/paper_files/paper/2001/file/4b86abe48d358ecf194c56c69108433e-Paper.pdf.

[21] Sumit Kunnumkal and Huseyin Topaloglu. Using stochastic approximation methods to compute optimal base-stock levels in inventory control problems. *Operations Research*, 56(3):646–664, 2008. doi: 10.1287/opre.1070.0477.

[22] Adisak Mekkittikul and Nick McKeown. A starvation-free algorithm for achieving 100% throughput in an input-queued switch. In *Proc. of the IEEE International Conference on Communication Networks*, volume 96, pages 226–231, 1996.

[23] Yashaswini Murthy and R. Srikant. On the convergence of natural policy gradient and mirror descent-like policy methods for average-reward MDPs, 2023.

[24] Jan Peters and Stefan Schaal. Natural actor-critic. *Neurocomputing*, 71(7):1180–1190, 2008. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2007.11.026. URL https://www.sciencedirect.com/science/article/pii/S0925231208000532. Progress in Modeling, Theory, and Application of Computational Intelligence.

[25] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[26] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

[27] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[28] R. Srikant and Lei Ying. *Communication networks: an optimization, control, and stochastic networks perspective*. Cambridge University Press, 2013.

[29] Alexander L. Stolyar. MaxWeight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic. *The Annals of Applied Probability*, 14(1):1 – 53, 2004. doi: 10.1214/aoap/1075828046. URL https://doi.org/10.1214/aoap/1075828046.

[30] Richard S Sutton and Andrew G Barto. *Reinforcement learning: an introduction*. MIT Press, Cambridge, MA, 1998.

[31] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. In *29th IEEE Conference on Decision and Control*, pages 2130–2132 vol.4, 1990. doi: 10.1109/CDC.1990.204000.

[32] Honghao Wei, Xin Liu, Weina Wang, and Lei Ying. Sample efficient reinforcement learning in mixed systems through augmented samples and its applications to queueing networks. *NeurIPS*, 2023.

A BACKGROUND LEMMAS

While this lemma is contained within [23, Lemma 2], we reprove it here in a standalone fashion.

Lemma 1. *Policy π_{k+1} improves upon policy π_k relative to π_k's relative-value function Q_k:*

$$Q_k(s, \pi_{k+1}) \geq Q_k(s, \pi_k) = V_k(s)$$

Proof. Recall the definition of the NPG algorithm:

$$\pi_{k+1}(a \mid s) = \pi_k(a \mid s)\beta_s^{Q_k(s,a)}/Z_{s,k}, \text{ where } Z_{s,k} = \sum_{a'} \pi_k(a' \mid s)\beta_s^{Q_k(s,a')}.$$

Rearranging to solve for $Q_k(s, a)$, we find that

$$Q_k(s, a) = \frac{1}{\log \beta_s} \log \left(\frac{Z_{s,k}\pi_{k+1}(a \mid s)}{\pi_k(a \mid s)} \right)$$

Now, let's start manipulating $Q_k(s, \pi_{k+1})$:

$$Q_k(s, \pi_{k+1}) = \sum_a \pi_{k+1}(a \mid s)Q_k(s, a) = \sum_a \pi_{k+1}(a \mid s)\frac{1}{\log \beta_s} \log \left(\frac{Z_{s,k}\pi_{k+1}(a \mid s)}{\pi_k(a \mid s)} \right)$$
$$= \frac{1}{\log \beta_s} \sum_a \pi_{k+1}(a \mid s) \log \left(\frac{\pi_{k+1}(a \mid s)}{\pi_k(a \mid s)} \right) + \frac{1}{\log \beta_s} \sum_a \pi_{k+1}(a \mid s) \log Z_{s,k} \tag{8}$$

Note that the left-hand term in (8) is simply the *KL* divergence $D_{KL}(\pi_{k+1}(\cdot|s)||\pi_k(\cdot|s))$. As a result, it is positive. Thus,

$$Q_k(s, \pi_{k+1}) \geq \frac{1}{\log \beta_s} \sum_a \pi_{k+1}(a \mid s) \log Z_{s,k} = \frac{1}{\log \beta_s} \log Z_{s,k} = \frac{1}{\log \beta_s} \log \sum_{a'} \pi_k(a' \mid s)\beta_s^{Q_k(s,a')}$$

Note that the log function is concave, so we can apply Jensen's inequality:

$$Q_k(s, \pi_{k+1}) \geq \frac{1}{\log \beta_s} \sum_{a'} \pi_k(a' \mid s)Q_k(s, a') \log \beta_s = \sum_{a'} \pi_k(a' \mid s)Q_k(s, a') = Q_k(s, \pi_k) = V_k(s). \quad \square$$

B BOUNDING RELATIVE VALUE

Lemma 8. *Given an MDP satisfying Assumption 1, and given an initial policy π_0 such that there exist constants $c_0 > 0, c_1 \geq 0$ such that*

$$V_0(s) \geq -c_0\hat{r}_{\max}(s)^2 - c_1, \tag{9}$$

there exists a uniform bound M_s:

$$M_s := c_2|\hat{r}_{\max}(s)|^2 + c_3|\hat{r}_{\max}(s)| + c_4,$$

for constants $c_2, c_3, c_4 \geq 0$ depending on c_0, c_1, and the MDP parameters, such that for any NPG iterate π_k, and any pair of actions a, a',

$$Q_k(s, a) - Q_k(s, a') \leq M_s.$$

Proof. The difference $Q_k(s, a) - Q_k(s, a')$ can be separated into the following terms:

$$Q_k(s, a) - Q_k(s, a') = E_{s' \sim P(s,a)}[V_k(s')] - E_{s'' \sim P(s,a')}[V_k(s'')] + r(s, a) - r(s, a')$$

We will bound these terms relative to $|\hat{r}_{\max}(s)|$, recalling that rewards are negative.

Let us start by applying the upper and lower bounds from Lemma 7. Define the following constants, where z is an arbitrary reward threshold below J_0:

$$c_5 := \frac{J_* - z}{J_0 - z}, c_6 := \tau_{J_0,z}^{bound} \frac{(J_* - J_0)(c_{\max} - z)(J_* - z)}{(J_0 - z)^2}, c_7 := \tau_{J_0,z}^{bound}(c_{\max} - J_0)$$

Note that c_5, c_6, c_7 are all positive. Our results hold for all values of $z < J_0$.

Lemma 7 states that $\min(c_5 V_0(s), V_0(s)) - c_6 \leq V_k(s) \leq c_7$. Our lower bound on $V_0(s)$, (9), is negative, so we can simplify the bound to $c_5\widetilde{V_0}(s) - c_6 \leq V_k(s) \leq c_7$, where $\widetilde{V_0}$ is our lower bound on V_0. As a result,

$$c_5 E_{s''\sim P(s,a')}[\widetilde{V_0}(s'')] - c_6 \leq E_{s''\sim P(s,a')}[V_k(s'')]$$
$$E_{s'\sim P(s,a)}[V_k(s')] \leq c_7.$$

To relate $\widetilde{V_0}(s'')$ to $\hat{r}_{\max}(s)$, we will combine (9), our assumption on $V_0(s)$, with Assumption 1(d):

$$\widetilde{V_0}(s'') \geq -c_0\hat{r}_{\max}(s'')^2 - c_1 \geq -c_0(R_3\hat{r}_{\max}(s) - R_4)^2 - c_1 = -c_0(R_3|\hat{r}_{\max}(s)| + R_4)^2 - c_1$$
$$E_{s''\sim P(s,a')}[V_k(s'')] \geq -c_0c_5(R_3|\hat{r}_{\max}(s)| + R_4)^2 - c_1c_5 - c_6$$

Finally, we can bound $r(s,a) - r(s,a')$ using Assumption 1(c): $r(s,a) - r(s,a') \leq R_1\hat{r}_{\max}(s)^2 + R_2$.
Combining these bounds together, we find that

$$\begin{aligned}
Q_k(s,a) - Q_k(s,a') &\leq c_0c_5(R_3|\hat{r}_{\max}(s)| + R_4)^2 + c_1c_5 + c_6 + c_7 + R_1|\hat{r}_{\max}(s)|^2 + R_2 \\
&= (c_0c_5R_3^2 + R_1)|\hat{r}_{\max}(s)|^2 + 2c_0c_5R_3R_4|\hat{r}_{\max}(s)| + c_0c_5R_4^2 + c_1c_5 + c_6 + c_7 + R_2 \\
&= c_2|\hat{r}_{\max}(s)|^2 + c_3|\hat{r}_{\max}(s)| + c_4
\end{aligned}$$
$$\text{where } c_2 := c_0c_5R_3^2 + R_1, c_3 := 2c_0c_5R_3R_4, \text{ and } c_4 := c_0c_5R_4^2 + c_1c_5 + c_6 + c_7 + R_2.$$

Note that c_2, c_3, and c_4 are all nonnegative. □

C QUEUEING LEMMAS

C.1 Bound on MaxWeight

This result is proven in [28, Theorem 4.2.4], for the classic switch model. For completeness, we now reprove the result for the general GSSE setting.

Lemma 10. *For any GSSE queueing system,*

$$E_{MW}[f(q(t+1)) - f(q(t)) \mid q(t) = q] \leq c_1 r(q(t)) + c_2,$$

where MW denotes the MaxWeight policy, where $f(q) := \sum_i q_i^2$, and where c_1 and c_2 are the following constants:

$$c_1 = 2\epsilon \min_i \lambda_i, \qquad c_2 = \ell^2 n$$

Proof. Let $w_i^{MW}(t)$ denote MaxWeight's service on step t. We proceed as follows:

$$\begin{aligned}
f(q(t+1)) - f(q(t)) &= \sum_i \left((q_i(t) + v_i(t) - w_i^{MW}(t))^+\right)^2 - \sum_i q_i^2(t) \\
&\leq \sum_i \left(q_i(t) + v_i(t) - w_i^{MW}(t)\right)^2 - \sum_i q_i^2(t) \\
&= \sum_i (2q_i(t) + v_i(t) - w_i^{MW}(t))(v_i(t) - w_i^{MW}(t)) \\
&= \sum_i 2q_i(t)(v_i(t) - w_i^{MW}(t)) + \sum_i (v_i(t) - w_i^{MW}(t))^2
\end{aligned}$$

Note that the second term is bounded. Recall that ℓ is the maximum number of jobs that can arrive or depart from a class in one time step. As a result, $(v_i(t) - w_i^{MW}(t))^2 \leq \ell^2$, so the second summation is bounded by $\ell^2 n$.

Taking expectations, we find that

$$E[f(q(t+1)) - f(q(t)) \mid q(t) = q] \leq 2\sum_i q_i(\lambda_i - \mu_i^{MW}) + \ell^2 n \tag{10}$$

Let us now lower bound $\sum_i q_i \mu_i^{MW}$. Recall that Maxweight selects the service option j which maximizes $\sum_i q_i \mu_i^j$. Thus, for all j,

$$\sum_i q_i \mu_i^{MW} \geq \sum_i q_i \mu_i^j. \tag{11}$$

Recall that we assumed that λ lies within the stability region. In particular, by (7), there exists a vector $\gamma \geq 0$ of service-option weights such that

$$\sum_j \gamma_j = 1 \text{ and } \forall i, (1+\epsilon)\lambda_i \leq \sum_j \gamma_j \mu_i^j$$

Let us multiply this inequality by q_i and sum over i. We then find that

$$\sum_{i,j} q_i \gamma_j \mu_i^j \geq (1+\epsilon)\sum_i q_i \lambda_i. \tag{12}$$

On the other hand, let us take the MaxWeight inequality (11), multiply by γ_j and sum over j. We then find that

$$\sum_i q_i \mu_i^{MW} \sum_j \gamma_j \geq \sum_{i,j} q_i \gamma_j \mu_i^j$$

$$\sum_i q_i \mu_i^{MW} \geq \sum_{i,j} q_i \gamma_j \mu_i^j \tag{13}$$

Combining (12) and (13), we find the desired lower bound on $\sum_i q_i \mu_i^{MW}$:

$$\sum_i q_i \mu_i^{MW} \geq (1+\epsilon)\sum_i q_i \lambda_i \tag{14}$$

Substituting this bound into (10), we find that

$$E[f(q(t+1)) - f(q(t)) \mid q(t) = q] \leq -2\epsilon \sum_i q_i \lambda_i + \ell^2 n$$

Recall that $r(q) = -\sum_i q_i$. Note that $\sum_i q_i \lambda_i \geq -r(q) \min_i \lambda_i$. As a result, taking $c_1 = 2\epsilon \min_i \lambda_i$ and taking $c_2 = n$, the proof is complete.

□

The above argument straightforwardly generalizes to the setting where the reward is the α-th moment of the total queue length, using the Lyapunov function $f(q) = \sum_i q^{\alpha+1}$.

C.2 Verifying Assumption 1(d) for α-moment reward

We want to verify Assumption 1(d) for the GSSE MDP with reward $-(\sum_i q_i)^\alpha$, for $\alpha \geq 1$.

LEMMA 11. *In any GSSE MDP with the α-moment reward function, there exists a pair of constants $R_3 \geq 1$ and $R_4 \geq 0$ such that for any pair of states s, s' such that there is a nonzero probability of transitioning from s to s' under some action a,*

$$\hat{r}_{\max}(s') \geq R_3 \hat{r}_{\max}(s) - R_4$$

Proof. Let q be the total queue length of state s. Note that the total queue length of states s' is at most $q + \ell n$, because at most ℓ jobs of each of the n job classes can arrive in a given time step. Let $n' \leq \ell n$ denote this maximum possible number of arriving jobs.

We have

$$|r_{\max}(s)| = q^\alpha, \quad |r_{\max}(s')| \leq (q + n')^\alpha$$

Let us set $R_3 = 2$. We want to find some R_4 such that

$$(q + n')^\alpha \leq 2q^\alpha + R_4, \quad \forall n', \alpha \geq 1$$

First, let's apply the mean value theorem to the quantity $(q + n')^\alpha - q^\alpha$. The function x^α has derivative $\alpha x^{\alpha-1}$, which is increasing for positive x. Thus,

$$(q + n)^\alpha - q^\alpha \leq n'\alpha(q + n')^{\alpha-1}$$

We will bound $n'\alpha(q+n')^{\alpha-1}$ in two parts: For large q, we will show $n'\alpha(q+n')^{\alpha-1} \leq q^\alpha$. For small q, we will select R_4 such that $n'\alpha(q + n')^{\alpha-1} \leq R_4$. Our split between these bounds is the value $q = 2n'\alpha$, where W is the Lambert W function.

First, for large q,

$$\begin{aligned}
\forall q \geq 2n'\alpha, \quad q^\alpha &\geq 2n'\alpha q^{\alpha-1} \\
&= 2n'\alpha(q + n')^{\alpha-1}\left(\frac{q + n'}{q}\right)^{-\alpha+1} \\
&= 2n'\alpha(q + n')^{\alpha-1}\left(1 + \frac{n'}{q}\right)^{-\alpha+1} \\
&\geq 2n'\alpha(q + n')^{\alpha-1}\left(1 + \frac{n'}{2n'\alpha}\right)^{-\alpha+1} \\
&\geq 2n'\alpha(q + n')^{\alpha-1}\left(1 + \frac{1}{2\alpha}\right)^{-\alpha+1} \\
&\geq 2n'\alpha(q + n')^{\alpha-1}e^{-1/2} \\
&\geq n'\alpha(q + n')^{\alpha-1}
\end{aligned}$$

To cover small q, we simply set

$$R_4 = n'\alpha(2\alpha n' + n')^{\alpha-1} \geq n'\alpha(q + n')^{\alpha-1} \quad \forall q \leq 2n'\alpha$$

Now we can conclude that

$$\begin{aligned}
\forall q, \quad n'\alpha(q + n')^{\alpha-1} &\leq q^\alpha + R_4 \\
(q + n')^\alpha - q^\alpha &\leq q^\alpha + R_4 \\
(q + n')^\alpha &\leq 2q^\alpha + R_4
\end{aligned}$$

□