Reinforcement Learning for Cost-Aware Markov Decision Processes

Wesley A. Suttle [1] **Kaiqing Zhang** [2] **Zhuoran Yang** [3] **David N. Kraemer** [1] **Ji Liu** [4]

Abstract

Ratio maximization has applications in areas as diverse as finance, reward shaping for reinforcement learning (RL), and the development of safe artificial intelligence, yet there has been very little exploration of RL algorithms for ratio maximization. This paper addresses this deficiency by introducing two new, model-free RL algorithms for solving cost-aware Markov decision processes, where the goal is to maximize the ratio of long-run average reward to long-run average cost. The first algorithm is a two-timescale scheme based on relative value iteration (RVI) Q-learning and the second is an actor-critic scheme. The paper proves almost sure convergence of the former to the globally optimal solution in the tabular case and almost sure convergence of the latter under linear function approximation for the critic. Unlike previous methods, the two algorithms provably converge for general reward and cost functions under suitable conditions. The paper also provides empirical results demonstrating promising performance and lending strong support to the theoretical results.

1. Introduction

Many reinforcement learning (RL) algorithms have been advocated in the literature for solving Markov decision processes (MDPs) and related sequential decision-making problems. The classic objective functions, including expected discounted reward, total reward, and long-run average reward, form the basis for most RL algorithms and are frequently used in practice. Nevertheless, alternative objectives have seen increasing interest, as researchers seek to extend RL techniques to larger classes of problems and incorporate *a priori* knowledge to accelerate learning. In particular, a great deal of effort has gone into extending RL techniques to constrained MDPs (Altman, 1999), where safety and risk considerations impose limits on agent behavior, and various schemes have been proposed for reward shaping, such as including entropy regularization to improve exploration (Geist et al., 2019; Grill et al., 2019; Vieillard et al., 2020) and general methods for incorporating domain knowledge (Mataric, 1994; Ng et al., 1999).

One area where little progress has been made, however, is in the application of RL techniques to *ratio maximization*. In finance, a wide range of portfolio optimization problems are explicitly formulated via ratio maximization, for example when maximizing the Sharpe, Calmar, Sortino, and Omega ratios (Sharpe, 1966; Young, 1991; Sortino & Price, 1994; Keating & Shadwick, 2002) of a financial portfolio. The subfield of reward shaping in RL studies methods for incorporating domain knowledge and expert guidance into the rewards an agent receives. Such techniques can be used to accelerate learning and improve solution quality (Mataric, 1994; Ng et al., 1999). Though currently unexploited in the reward shaping literature, the ability to simultaneously specify both rewards and costs has clear potential as a powerful tool for the reward shaping toolkit. Finally, safe RL, a subfield of safe AI, has seen a dramatic surge of interest in recent years (García et al., 2015; Berkenkamp et al., 2017; Cheng et al., 2019; Yu et al., 2019; Ding et al., 2020), driven by safety-critical applications such as autonomous vehicles (Kiran et al., 2021); incorporating ratio maximization techniques would complement and enhance existing methods for safe RL, and potentially have a significant impact on the development of safe AI.

Contribution. Our primary contribution is to propose two tractable new algorithms with theoretical guarantees and lay sound theoretical foundations for future study of this previously under-studied type of problem. Our experiments validate and supplement our theoretical contributions.

First, motivated by the lack of RL methods for solving ratio maximization problems, we develop two new, model-free reinforcement learning algorithms for provably solving cost-

[1]Department of Applied Mathematics and Statistics, Stony Brook University, Stony Brook, New York, USA. [2]Department of Electrical and Computer Engineering, Coordinated Science Laboratory, University of Illinois Urbana-Champaign, Urbana, Illinois, USA. [3]Department of Operations Research and Financial Engineering, Princeton University, Princeton, New Jersey, USA. [4]Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, New York, USA. Correspondence to: Ji Liu <ji.liu@stonybrook.edu>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021.

aware MDPs, where the goal is to maximize the ratio of long-run average reward to long-run average cost. The first, cost-aware relative value iteration (CARVI) Q-learning, is a two-timescale algorithm with a novel structure: a running estimate of the optimal ratio is updated at the slower timescale, while RVI Q-learning is used to solve certain auxiliary MDPs, to be defined later, at the faster timescale. The second, cost-aware actor-critic (CAAC), is based on a policy gradient theorem for the cost-aware setting.

We subsequently provide theoretical convergence guarantees for our algorithms. Difficulties arising from the problem structure, described in detail in Section 4, prevent a standard convergence analysis for CARVI Q-learning. We nonetheless prove almost sure (a.s.) convergence to the globally optimal solution in the tabular case by leveraging certain properties of the algorithm and innovating the standard two-timescale analysis. This result may be of independent interest for the theory of two-timescale stochastic approximation. We furthermore provide a cost-aware policy gradient theorem and prove a.s. convergence of CAAC under linear function approximation for the critic. Finally, we present numerical results illustrating and supporting our theory, demonstrating promising empirical performance, and motivating future empirical exploration of our algorithms.

Related Work. RL has a rich literature stretching back several decades; see (Sutton & Barto, 1998) for a comprehensive introduction. Among value iteration-inspired techniques are Q-learning for the discounted reward setting (Watkins & Dayan, 1992) and RVI Q-learning (Abounadi et al., 2001) for the average reward setting, for which convergence is established in the tabular case. Another mainstream class of RL algorithms optimizes the policy directly (Sutton et al., 2000). Typical examples include the actor-critic (Konda & Tsitsiklis, 2000) and natural actor-critic algorithms (Peters & Schaal, 2008; Bhatnagar et al., 2009), for which convergence has been shown with linear function approximation for the critic.

The literature concerning RL methods for ratio maximization problems is sparse. The theory of MDPs with fractional cost is first proposed in (Ren & Krogh, 2005), which provided an alternative but equivalent formulation of the cost-aware MDPs considered in our work. (Ren & Krogh, 2005) rigorously analyzed algorithms for the fractional cost problem, but they are either model-based, computation- and memory-intensive, or do not readily admit the use of function approximation. In contrast, our algorithms are model-free, have natural function approximation versions, and are of the same per-timestep computational and memory complexity as standard actor-critic and Q-learning algorithms. More recently, (Tanaka, 2017; 2019) elaborated notions of optimality and duality for extensions of MDPs with fractional rewards. There is also some empirical work applying RL to ratio maximization problems in finance, such as (Moody & Saffell, 2001), which studied RL-based methods for optimizing the Sharpe ratio of a financial portfolio.

Related to (but distinct from) the ratio maximization setting, constrained MDPs (CMDPs) introduce constraints on long-term reward (Altman, 1999). It is important to note that RL methods for solving CMDPs typically assume a priori knowledge of the constraints, while the cost-aware formulation makes no such assumptions. It is also difficult to formulate ratio maximization within the CMDP framework. A well-known model-free approach to solving CMDPs with convergence guarantees is the Lagrangian method (Altman, 1998; Borkar, 2005; Bhatnagar, 2010). Risk-sensitive MDPs that consider variance-related constraints have also been addressed by actor-critic algorithms (Prashanth & Ghavamzadeh, 2016; Chow et al., 2017) under the Lagrangian formulation.

As an initial step towards the development of a robust theory of RL for ratio optimization problems, in this paper we focus on the asymptotic convergence analysis and leave finite-time convergence analysis as an important future work. Existing finite-time works, such as (Gupta et al., 2019; Hong et al., 2020; Wu et al., 2020; Li et al., 2020), do not apply, since our algorithms are either too structurally dissimilar from the algorithms these works consider or do not satisfy the necessary assumptions imposed in these works. See the supplementary material for a more detailed review of the two-timescale finite-time literature.

2. Background and Model

In this section, we first introduce the relevant RL background, and then discuss cost-aware MDPs. In this paper we restrict our attention to finite state and action spaces.

Reinforcement Learning. The goal in RL is to learn an optimal decision rule via interacting with the environment. The environment is modeled by an MDP, denoted by $(\mathcal{S}, \mathcal{A}, p, r)$, where $\mathcal{S}$ is a finite set of states, $\mathcal{A}$ is a finite set of actions, $p\colon \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ is the Markov transition kernel, where $\mathcal{P}(\mathcal{S})$ denotes the set of all probability distributions over $\mathcal{S}$, and $r\colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function. At each time step $n \geq 0$, the agent finds itself in state $s_n \in \mathcal{S}$, chooses an action $a_n \in \mathcal{A}$, obtains an immediate reward $r(s_n, a_n)$, and the environment transitions into a new state $s_{n+1} \in \mathcal{S}$ according to distribution $p(\,\cdot \mid s_n, a_n)$.

A policy $\pi : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ maps states $s \in S$ to distributions $\pi(\,\cdot \mid s) \in \mathcal{P}(\mathcal{A})$, so that, at a given state $s \in \mathcal{S}$, the probability of selecting action $a \in \mathcal{A}$ is given by $\pi(a \mid s)$. Note that deterministic policies can be recovered from this definition by assigning probability one to the desired action.

We focus on the average reward setting, where the goal of the agent is to find a policy π maximizing the long-run average reward, defined as

$$J(\pi) = \lim_{N\to\infty} \frac{1}{N} \mathbb{E}_\pi \left[\sum_{n=0}^{N-1} r(s_n, a_n) \right],$$

where $a_n \sim \pi(\,\cdot\,|\, s_n)$ for all $n \geq 0$. Note that π induces a Markov chain on $\mathcal{S}$. Assuming this Markov chain has a stationary distribution d^π, then it holds that $J(\pi) = \mathbb{E}_{s\sim d^\pi}\mathbb{E}_{a\sim\pi(\,\cdot\,|\,s)}[r(s,a)]$. In addition, the relative state and action value functions of policy π are defined as

$$V^\pi(s) = \mathbb{E}_\pi \left[\sum_{n=0}^{\infty} [r_n - J(\pi)] \mid s_0 = s \right],$$

$$Q^\pi(s,a) = \mathbb{E}_\pi \left[\sum_{n=0}^{\infty} [r_n - J(\pi)] \mid s_0 = s, a_0 = a \right],$$

respectively. We hereafter omit the word "relative" when referring to value functions. Moreover, V^π satisfies the Poisson equation (Puterman, 2014)

$$V^\pi(s) + J(\pi) = \mathbb{E}_\pi \left[r(s,a) + V^\pi(s') \right], \tag{1}$$

for any $(s,a) \in \mathcal{S}\times\mathcal{A}$, where s' is the next state given (s,a). An optimal policy π^* satisfies $V^* = V^{\pi^*}$, where V^* is the unique solution modulo an additive constant (Abounadi et al., 2001) to the dynamic programming equation

$$V^*(s) + J(\pi^*) = \max_a \{ r(s,a) + \mathbb{E}_{\pi^*}[V^*(s')] \},$$

for any $s \in \mathcal{S}$. The action-value functions Q^π and $Q^* = Q^{\pi^*}$ also satisfy similar Poisson and dynamic programming equations, respectively. Note that in the theory of average-reward MDPs, which is subtler than the discounted setting, optimality of value functions is typically modulo an additive constant; this does not affect the recovery of optimal policies from the family of optimal value functions, since, for any scalar β, $\mathrm{argmax}_a\, Q(s,a) = \mathrm{argmax}_a (Q(s,a) + \beta)$, for all s. See (Puterman, 2014; Abounadi et al., 2001; Bertsekas, 2012) for details. When there is no risk of confusion, we will often refer to *the* optimal value function when the optimal value function *modulo an additive constant* is meant.

Q-learning-based RL methods can be used to learn the optimal action value function Q^*. From this function, an optimal policy $\pi^*(s) = \mathrm{argmax}_a Q^*(s,a)$ can be immediately extracted. The policy gradient approach to RL instead focuses on a parametrized policy class $\{\pi_\theta\}_{\theta\in\Theta}$ and attempts to maximize $J(\pi_\theta)$ over the parameter space $\Theta \subset \mathbb{R}^m$ via stochastic gradient methods. Such approaches are typically based on the classic policy gradient theorem (Sutton et al., 2000), which states that $\nabla_\theta J(\pi_\theta) = \mathbb{E}\left[\nabla_\theta \log \pi_\theta(a\,|\,s) \cdot A^\theta(s,a)\right]$, where $s \sim d^\theta(\cdot)$, the stationary distribution induced on $\mathcal{S}$ by π_θ, $a \sim \pi_\theta(\,\cdot\,|\,s)$, and $A^\theta(s,a) = Q^{\pi_\theta}(s,a) - V^{\pi_\theta}(s)$ is the *advantage function*.

Cost-Aware MDP. Motivated by the task of extending RL methods to ratio maximization problems, we define the cost-aware MDP as follows.

Definition 1 (Cost-aware MDP)**.** *A cost-aware Markov decision process (CAMDP) is a sequential decision-making problem specified by the tuple $(\mathcal{S}, \mathcal{A}, p, r, c)$, where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ denotes the action space, $p : \mathcal{S}\times\mathcal{A} \to \mathcal{P}(\mathcal{S})$ denotes the Markov transition kernel, $r : \mathcal{S}\times\mathcal{A} \to \mathcal{R}$ is the reward function, $c : \mathcal{S}\times\mathcal{A} \to \mathcal{R}$ is the cost function, and the goal of the agent is to find a policy π maximizing*

$$\rho(\pi) := \frac{\lim_{N\to\infty} \frac{1}{N}\mathbb{E}_\pi[\sum_{n=0}^{N-1} r(s_n,a_n)]}{\lim_{N\to\infty} \frac{1}{N}\mathbb{E}_\pi[\sum_{n=0}^{N-1} c(s_n,a_n)]}. \tag{2}$$

The CAMDP extends the MDP in the previous section by augmenting it with a cost function: at each step, the agent receives both a reward and cost associated with its current state and action and must learn to maximize long-run average reward over long-run average cost. As noted above, we assume throughout that $\mathcal{S}$ and $\mathcal{A}$ are finite.

Following the definitions for MDPs, we denote by $J_r(\pi)$ and $J_c(\pi)$ the long-term average reward and cost, respectively, under policy π. We assume throughout that these limits exist, which is reasonable given the standard ergodicity conditions on the induced Markov chains. We also assume $r(s,a)$ and $c(s,a)$ are always strictly positive. Note that this implies $J_r(\pi) > 0$ and $J_c(\pi) > 0$, for all policies π. This mild assumption stipulates that some non-negligible reward and cost are incurred at each step. Moreover, we denote by V_r^π, V_c^π, Q_r^π, Q_c^π, A_r^π, and A_c^π, the value, action-value, and advantage functions of policy π based on the reward r and cost c, respectively. We also write ρ as shorthand for $\rho(\pi)$. It is interesting to note that if the cost function is some positive constant, solving $(\mathcal{S}, \mathcal{A}, p, r, c)$ also solves the MDP $(\mathcal{S}, \mathcal{A}, p, r)$.

For an interesting example of how the CAMDP model can be applied to maximizing the Omega ratio of a financial portfolio, see the supplementary material.

3. Algorithms

In this section, we propose two RL algorithms for CAMDPs, based on the RVI Q-learning and actor-critic algorithms, respectively.

Cost-Aware RVI Q-learning. Our cost-aware RVI Q-learning algorithm follows a two-timescale stochastic approximation scheme. An estimate ρ of the optimal ratio ρ^* is iteratively updated on the slower timescale, while an auxiliary MDP, whose objective function depends on the current value of ρ, is solved using RVI Q-learning on the faster timescale. In what follows we provide a derivation of the algorithm that starts with the CAMDP we wish to solve, then proceeds by looking at a related bi-level optimization problem. The bi-level program we consider suggests a tractable, provably effective solution approach, namely, CARVI Q-learning. We conclude by giving a concrete formulation of this algorithm.

To see where the bi-level program and two-timescale algorithm originate, consider the CAMDP $(\mathcal{S}, \mathcal{A}, p, r, c)$ as split into two parts: an outer estimate ρ of the optimal ratio, and an inner, auxiliary MDP which depends on ρ and is defined as $(\mathcal{S}, \mathcal{A}, p, \eta^\rho)$, with reward function $\eta^\rho(s,a) = r(s,a) - \rho c(s,a)$. Then, $J_{\eta^\rho}(\pi)$ is the long-run average reward of policy π for $(\mathcal{S}, \mathcal{A}, p, \eta^\rho)$, and $J_{\eta^\rho}(\pi) = J_r(\pi) - \rho J_c(\pi)$. Splitting the CAMDP into outer and inner parts in this way is useful for the following reason:

Lemma 1. *Given the optimal ratio ρ^* of the CAMDP $(\mathcal{S}, \mathcal{A}, p, r, c)$, any optimal policy for $(\mathcal{S}, \mathcal{A}, p, \eta^{\rho^*})$ is an optimal policy for $(\mathcal{S}, \mathcal{A}, p, r, c)$.*

We next develop a search procedure for finding the optimal (ρ^*, π^*). Consider the bi-level program:

$$\begin{aligned} \underset{\rho,\,\pi}{\text{minimize}} \quad & [J_r(\pi) - \rho J_c(\pi)]^2 \\ \text{subject to} \quad & \pi \in \underset{\pi'}{\operatorname{argmax}}\{J_r(\pi') - \rho J_c(\pi')\}, \end{aligned} \tag{3}$$

where $\operatorname{argmax}_{\pi'}\{J_r(\pi') - \rho J_c(\pi')\} = \{\pi \mid J_r(\pi) - \rho J_c(\pi) = \max_{\pi'}\{J_r(\pi') - \rho J_c(\pi')\}\}$. This problem has the following useful property:

Lemma 2. *Solving (3) yields the optimal ratio and an optimal policy for CAMDP $(\mathcal{S}, \mathcal{A}, p, r, c)$.*

See supplementary material for proofs of Lemmas 1 and 2. Lemmas 1 and 2 present a new way to solve ratio-maximization problems as bi-level optimization problems that is potentially promising beyond the CAMDP setting.

Assume that, for a given ρ, we can efficiently obtain the value $J_r(\pi^\rho) - \rho J_c(\pi^\rho)$ corresponding to some optimal policy $\pi^\rho \in \operatorname{argmax}_\pi\{J_r(\pi) - \rho J_c(\pi)\}$. Then, in this situation (3) reduces to an unconstrained problem, which we can solve using gradient-based methods. In particular, we can update ρ via update steps of the form $\rho \leftarrow \rho + \beta\left[J_r(\pi^\rho) - \rho J_c(\pi^\rho)\right]$, with learning rate β. To see why this update scheme is justified, consider the following:

Lemma 3. *If $\rho > \rho^*$, then $J_r(\pi^\rho) - \rho J_c(\pi^\rho) < 0$. If $\rho < \rho^*$, then $J_r(\pi^\rho) - \rho J_c(\pi^\rho) > 0$. If $\rho = \rho^*$, then $J_r(\pi^\rho) - \rho J_c(\pi^\rho) = 0$.*

With the above in mind, our search procedure for finding (ρ^*, π^*) is within reach. The overall goal is to perform gradient-descent-type updates in ρ on the objective in (3) on the slower timescale, while using RVI Q-learning at the faster timescale to solve the inner optimization problem in π, given the current ρ. Once we have solved (3), Lemmas 1 and 2 apply to show that we have in fact solved the original CAMDP.

To perform the ρ updates, our algorithm must approximately find the optimal action-value function, Q^ρ, for the auxiliary MDP corresponding to the current ρ, from which an estimate of $J_r(\pi^\rho) - \rho J_c(\pi^\rho)$ can be obtained. Given ρ, the RVI Q-learning update is as follows:

$$\begin{aligned} Q_{n+1}(s_n, a_n) = \ & Q_n(s_n, a_n) + \alpha_n\Big[r(s_n, a_n) - \rho c(s_n, a_n) \\ & + \max_a Q_n(s_{n+1}, a) - V_n(s_{\text{ref}}) - Q_n(s_n, a_n)\Big], \end{aligned} \tag{4}$$

where $V_n(s_{\text{ref}}) = \max_a Q_n(s_{\text{ref}}, a)$ for a fixed reference state s_{ref}.

Assuming at each step that the faster timescale update has converged to the optimal Q^{ρ_n} for the current ρ_n, we want to perform updates of the form

$$\rho_{n+1} = \rho_n + \beta_n[J_r(\pi^{\rho_n}) - \rho_n J_c(\pi^{\rho_n})],$$

where $\pi^{\rho_n}(s) = \operatorname{argmax}_a Q^{\rho_n}(s,a)$, with ties broken arbitrarily. For a given ρ, we can obtain an optimal policy π^ρ directly from Q^ρ, so, with a slight abuse of notation, the above update can be rewritten as

$$\rho_{n+1} = \rho_n + \beta_n[J_r(Q^{\rho_n}) - \rho_n J_c(Q^{\rho_n})].$$

We do not have direct access to the quantity $J_r(Q^{\rho_n}) - \rho_n J_c(Q^{\rho_n})$, however, so we must find an approximation. Given that the faster timescale update has approximately converged at time n, we have $Q_n \approx Q^{\rho_n}$. As demonstrated in (Abounadi et al., 2001), under standard ergodicity conditions discussed in Chapter 5 of (Bertsekas, 2012), $\lim_{n\to\infty} V_n(s_{\text{ref}}) = \kappa_\rho$, where κ_ρ is the optimal average cost for the auxiliary MDP $(\mathcal{S}, \mathcal{A}, p, \eta^\rho)$. This implies that $V_n(s_{\text{ref}})$ provides an estimate of $J_r(Q^{\rho_n}) - \rho_n J_c(Q^{\rho_n})$. Since $V_n(s_{\text{ref}}) = \max_a Q_n(s_{\text{ref}}, a)$, we can thus use our current estimate Q_n to approximate $J_r(Q^{\rho_n}) - \rho_n J_c(Q^{\rho_n})$ at each timestep n.

Putting all these pieces together, we can finally write our CARVI Q-learning algorithm for solving the CAMDP:

$$\begin{aligned} Q_{n+1}(s_n, a_n) &= Q_n(s_n, a_n) + \alpha_n[r(s_n, a_n) - \rho_n\, c(s_n, a_n) \\ &\quad + V_n(s_{n+1}) - V_n(s_{\text{ref}}) - Q_n(s_n, a_n)], \end{aligned} \tag{5}$$

$$\rho_{n+1} = \rho_n + \beta_n V_n(s_{\text{ref}}). \tag{6}$$

To handle large state and action spaces, it is often necessary to use a function approximator Q_ω parameterized by a vector ω, such as a neural network, in place of the true Q-function. In this setting the gradient update (4) is carried out with respect to the parameter ω rather than the entire Q-table. This more general form of the algorithm is summarized in Algorithm 1. It should be noted that the theoretical analysis in this paper is for the tabular case.

Cost-Aware Actor-Critic. We next develop the cost-aware actor-critic algorithm. Let $\{\pi_\theta\}_{\theta\in\Theta}$ be a family of parametrized policies. To simplify the notation, we denote $J_r(\pi_\theta)$ and $J_c(\pi_\theta)$ by $J_r(\theta)$ and $J_c(\theta)$, respectively. As the limits in (2) exist, the limiting ratio can be written as

Algorithm 1 CARVI Q-learning

Initialization: Randomly generate ω_0 and ρ_0; fix an arbitrary start state s_0; specify learning rates $\{\alpha_n, \beta_n\}$; set $n \leftarrow 0$.
repeat
$a_n \sim \epsilon$-greedy$[Q_{\omega_n}(s_n, \cdot)]$, observe r_n, c_n, s_{n+1}
$\delta_n^\eta = r_n - \rho_n \cdot c_n + \max_a Q_{\omega_n}(s_{n+1}, a) - V_{\omega_n}(s_{\text{ref}}) - Q_{\omega_n}(s_n, a_n)$
$\omega_{n+1} = \omega_n + \alpha_n \cdot \delta_n^\eta \cdot \nabla Q_{\omega_n}(s_n, a_n)$
$\rho_{n+1} = \rho_n + \beta_n \cdot V_{\omega_n}(s_{\text{ref}})$
until convergence

$L(\theta) = J_r(\theta)/J_c(\theta)$. The policy gradient theorem in (Sutton et al., 2000) yields

$$\nabla_\theta L(\theta) = \frac{J_r(\theta)}{J_c(\theta)} \cdot \left(\frac{\mathbb{E}\left[\nabla_\theta \log \pi_\theta(a \mid s) \cdot A_r^\theta(s,a)\right]}{J_r(\theta)} - \frac{\mathbb{E}\left[\nabla_\theta \log \pi_\theta(a \mid s) \cdot A_c^\theta(s,a)\right]}{J_c(\theta)} \right). \tag{7}$$

By the Poisson equation (1), one can estimate the value functions, and thus the advantage function, via (Sutton, 1988; Tsitsiklis & Van Roy, 1999). Specifically, let $\{V_\nu \colon \mathcal{S} \to \mathbb{R}\}_{\nu \in \Omega}$ be a parametrized function class, where Ω is the parameter space. We use V_{ν^r} and V_{ν^c} to estimate the value functions V_r^θ and V_c^θ, respectively. In addition, we use the sequence $\{\mu_n^r\}$ and $\{\mu_n^c\}$ to track the values of $J_r(\theta)$ and $J_c(\theta)$. Then, in the n-th iteration, given the observation $(s_n, a_n, r_n, c_n, s_{n+1})$, the agent updates the sequences $\{\mu_n^r\}$ and $\{\mu_n^c\}$ via

$$\mu_{n+1}^r = \mu_n^r + \alpha_n \cdot (r_n - \mu_n^r), \qquad \mu_{n+1}^c = \mu_n^c + \alpha_n \cdot (c_n - \mu_n^c).$$

As a result, the TD-errors of value functions, denoted by δ_n^r and δ_n^c, can be calculated as

$$\delta_n^r = r_n + V_{\nu_n^r}(s_{n+1}) - \mu_n^r - V_{\nu_n^r}(s_n),$$
$$\delta_n^c = c_n + V_{\nu_n^c}(s_{n+1}) - \mu_n^c - V_{\nu_n^c}(s_n),$$

where ν_n^r and ν_n^c are the values of parameters ν^r and ν^c at iteration n. The TD-learning updates for the critic are then given by

$$\nu_{n+1}^r = \nu_n^r + \alpha_n \cdot \delta_n^r \cdot \nabla V_{\nu_n^r}(s_n),$$
$$\nu_{n+1}^c = \nu_n^c + \alpha_n \cdot \delta_n^c \cdot \nabla V_{\nu_n^c}(s_n),$$

where $\alpha_n > 0$ is the stepsize for the critic.

Hence, from (7), the actor update can be written as

$$\theta_{n+1} = \theta_n + \beta_n \cdot \frac{\mu_n^r}{\mu_n^c} \cdot \nabla_\theta \log \pi_{\theta_n}(a_n \mid s_n) \cdot \left(\frac{\delta_n^r}{\mu_n^r} - \frac{\delta_n^c}{\mu_n^c} \right),$$

where $\beta_n > 0$ is the stepsize and the TD-errors δ_n^r, δ_n^c are used to estimate the advantage functions $A_r^{\theta_n}, A_c^{\theta_n}$, respectively. Notice that, so long as we initialize $\mu_0^r, \mu_0^c > 0$, the fact that rewards and costs are always strictly positive ensures $\mu_n^r, \mu_n^c > 0$, for all n. The details of the algorithm are summarized in Algorithm 2.

Algorithm 2 Cost-Aware Actor-Critic (CAAC)

Initialization: Randomly generate $\mu_0^c > 0$ and $\mu_0^r > 0$, as well as ν_0^c, ν_0^r, and θ_0; fix an arbitrary start state s_0; specify learning rates $\{\alpha_n, \beta_n\}$; set $n \leftarrow 0$.
repeat
$a_n \sim \pi_{\theta_n}(\cdot \mid s_n)$, observe r_n, c_n, s_{n+1}
Critic step:
$\mu_{n+1}^r \leftarrow \mu_n^r + \alpha_n \cdot (r_n - \mu_n^r)$
$\mu_{n+1}^c \leftarrow \mu_n^c + \alpha_n \cdot (c_n - \mu_n^c)$
$\delta_n^r \leftarrow r_n + V_{\nu_n^r}(s_{n+1}) - \mu_n^r - V_{\nu_n^r}(s_n)$
$\delta_n^c \leftarrow c_n + V_{\nu_n^c}(s_{n+1}) - \mu_n^c - V_{\nu_n^c}(s_n)$
$\nu_{n+1}^r \leftarrow \nu_n^r + \alpha_n \cdot \delta_n^r \cdot \nabla V_{\nu_n^r}(s_n)$
$\nu_{n+1}^c \leftarrow \nu_n^c + \alpha_n \cdot \delta_n^c \cdot \nabla V_{\nu_n^c}(s_n)$
Actor step:
$\psi_n = \frac{\mu_n^r}{\mu_n^c}\left(\frac{\delta_n^r}{\mu_n^r} - \frac{\delta_n^c}{\mu_n^c}\right)$
$\theta_{n+1} = \theta_n + \beta_n \cdot \psi_n \cdot \nabla_\theta \log \pi_{\theta_n}(a_n \mid s_n)$
until convergence

Remark. The CAAC algorithm and the actor-critic algorithm for optimizing the Sharpe ratio given in (Prashanth & Ghavamzadeh, 2016) share similarities in that they both seek to maximize a ratio. The Sharpe ratio maximization scheme can also likely be extended to accommodate more general objectives. CAAC uses general rewards, however, while the other algorithm is specific to the Sharpe ratio; furthermore, the Sharpe ratio denominator is the *square root* of an expectation, while CAAC's is an expectation, so the gradient expressions used are different.

4. Convergence Analysis

In this section, we provide theoretical convergence guarantees for the algorithms developed in the last section. All proofs are given in the supplementary material.

CARVI Q-learning Convergence. We prove almost sure convergence of Algorithm 1 in the tabular setting to the globally optimal action value function and corresponding maximal ratio, (Q^*, ρ^*), by leveraging the RVI Q-learning convergence results in (Abounadi et al., 2001) and generalizing the classic machinery of two timescale stochastic approximation (Borkar, 2008). The central result of this section is Theorem 2. For ease of presentation, our analysis is given for the synchronous case, where every entry of the Q function is updated at each timestep. Extension to the asynchronous case, where only one state-action pair entry is updated at each timestep, follows exactly as in (Abounadi et al., 2001).

Given ρ, let Q^ρ be the optimal action-value function for the auxiliary MDP $(\mathcal{S}, \mathcal{A}, p, \eta^\rho)$ obtained by applying the RVI Q-learning algorithm. See Appendix A.2 of the supplementary materials for details on this Q^ρ. Let $\mathcal{F}_n = \sigma(\rho_k, Q_k, s_k, a_k; k \le n)$ be the σ-field generated by the iterates and trajectory up to time n. Our goal is to rewrite the updates (5) and (6) as

$$Q_{n+1} = Q_n + \alpha_n[h(Q_n, \rho_n) + M_{n+1}], \tag{8}$$

$$\rho_{n+1} = \rho_n + \beta_n[g(Q_n, \rho_n) + \epsilon_{n+1}], \tag{9}$$

where $\{M_n\}$ is an appropriate martingale difference sequence conditioned on $\mathcal{F}_n$, $\{\epsilon_n\}$ is a suitable error sequence, h and g are appropriate Lipschitz functions that satisfy the conditions needed for our ordinary differential equation (ODE) analysis, and the stepsizes α_n, β_n satisfy Assumption 2 below. We will proceed by first identifying the terms in (8) and studying the corresponding ODEs

$$\dot{\rho}(t) = 0, \tag{10}$$

$$\dot{Q}(t) = h(Q(t), \rho(t)), \tag{11}$$

using the analysis of RVI Q-learning given in (Abounadi et al., 2001) to simultaneously obtain a.s. convergence of (8) and (9) to the set $\{(Q^\rho, \rho) \mid \rho \in \mathbb{R}\}$ and show that the function $\lambda(\rho) := Q^\rho$ is the unique globally asymptotically stable equilibrium point of (10) and (11) for each $\rho \in \mathbb{R}$. Finally, we will study the slower timescale ODE

$$\dot{\rho}(t) = g(\lambda(\rho(t)), \rho(t)), \tag{12}$$

and use our analysis of it to prove a.s. convergence of our algorithm to the globally optimal pair (Q^*, ρ^*), where $Q^* = Q^{\rho^*}$. In what follows we will occasionally use $\lambda(\rho)$ instead of Q^ρ to emphasize the fact that Q^ρ is a function of ρ. We make the following assumptions.

Assumption 1. *The action value function Q and state value functions V provide tabular representation, i.e., $Q \in \mathbb{R}^{|S|\cdot|A|}$ and $V \in \mathbb{R}^{|S|}$.*

Assumption 2. *The stepsizes α_n and β_n satisfy $\sum_n \alpha_n = \sum_n \beta_n = \infty,\ \sum_n \alpha_n^2 + \beta_n^2 < \infty, \lim_n \frac{\beta_n}{\alpha_n} = 0$.*

Assumption 3. *For any policy π, the Markov chain it induces on $\mathcal{S}$ is ergodic.*

Assumption 1 is key in the analysis of RVI Q-learning (Abounadi et al., 2001) and is needed in Theorem 1. Assumption 2 is standard in the stochastic approximation literature (Borkar, 2008) and is needed in Lemma 5 and Theorems 1 and 2. Assumption 3, adapted from (Abounadi et al., 2001; Bertsekas, 2012), helps ensure that (2) is well-defined and that, for fixed ρ, $V_n(s_{\text{ref}})$ converges to κ_ρ, the optimal average reward for the auxiliary MDP $(\mathcal{S}, \mathcal{A}, p, \eta^\rho)$; it is essential for Theorems 1 and 2.

We begin our two-timescale analysis with convergence of the faster timescale. Define $g : \mathbb{R}^{|S|\cdot|A|} \times \mathbb{R} \to \mathbb{R}$ by $g(Q, \rho) = \max_a Q(s_{\text{ref}}, a) = V(s_{\text{ref}})$, and let $\epsilon_{n+1} = V_{n+1}(s_{\text{ref}}) - V_n(s_{\text{ref}})$, where $\{\epsilon_n\}$ is the error sequence in (9). These definitions will be important throughout. Note that, since the dependence of g on ρ is vacuous and the max operator over a vector is Lipschitz, we have by Assumption 1 that g is Lipschitz in both Q and ρ. We need two preliminary lemmas for Theorem 1.

Lemma 4. *The function $\hat{g}(\rho) := g(\lambda(\rho), \rho) = g(Q^\rho, \rho) = V^\rho(s_{\text{ref}})$ is strictly decreasing and piecewise linear (and thus Lipschitz) in ρ.*

For the next lemma, the following remarks on notation will be needed. Each vector $Q \in \mathbb{R}^{|S|\cdot|A|}$, regarded as an action value function, induces at least one deterministic policy $\pi_Q(s) = \text{argmax}_a Q(s, a)$ for the auxiliary MDP $(\mathcal{S}, \mathcal{A}, p, \eta^\rho)$. There may be multiple maximizing actions and thus multiple distinct policies, however, so π_Q may not be well-defined. Nonetheless, all policies induced by a given Q will have identical long-run average rewards and costs. We will therefore slightly abuse notation in what follows by writing $J_r(Q)$ and $J_c(Q)$ to denote the long-run average reward and cost, respectively, of any policy induced by Q. As in the previous lemma, we write $\hat{g}(\rho)$ as shorthand for $g(\lambda(\rho), \rho)$.

Lemma 5. *$\{\rho_n\}$ is a.s. bounded.*

With Lemmas 4 and 5, we can show convergence of the faster timescale:

Theorem 1. *$(Q_n, \rho_n) \to \{(Q^\rho, \rho) \mid \rho \in \mathbb{R}\}$ a.s. as $n \to \infty$.*

To complete our analysis of CARVI Q-learning, the following corollary and lemma are needed. Theorem 1 implies that $\|Q_n - Q^{\rho_n}\| \to 0$ a.s., and, as a consequence, we immediately have the following:

Corollary 1. *$|g(Q_n, \rho_n) - g(Q^{\rho_n}, \rho_n)| = |V_n(s_{\text{ref}}) - V^{\rho_n}(s_{\text{ref}})| \to 0$ a.s. as $n \to \infty$.*

This corollary allows us to bound the noise introduced by using $V_n(s_{\text{ref}})$ to estimate $J_r(Q^{\rho_n}) - \rho_n J_c(Q^{\rho_n})$. The next lemma shows that the ODE (12), which the ρ updates (6) asymptotically track as shown in Theorem 2, has an important limit point.

Lemma 6. *ρ^* is the unique globally asymptotically stable equilibrium point of (12).*

The next theorem is the main result of this subsection and provides a.s. convergence of CARVI Q-learning to the globally optimal $(Q^*, \rho^*) = (\lambda(\rho^*), \rho^*)$. Its proof relies on Theorem 1, Corollary 1, Lemma 6, and the two-timescale stochastic approximation results in (Borkar, 2008), but requires a key modification of the latter that exploits the special structure of g to accommodate the fact that $\lambda(\rho)$ is potentially not Lipschitz or even continuous in ρ.

Theorem 2. *$(Q_n, \rho_n) \to (\lambda(\rho^*), \rho^*)$ a.s. as $n \to \infty$.*

In other words, CARVI Q-learning a.s. solves the bi-level optimization problem (3), and, by Lemmas 1 and 2, it therefore solves the CAMDP $(\mathcal{S}, \mathcal{A}, p, r, c)$.

Remark. Due to the special structure of g and $\hat{g}$ described in Lemma 4 and Corollary 1, the proof of Theorem 2 did not require $\lambda(\rho)$ to be Lipschitz. This contrasts with the standard conditions assumed when proving a.s. convergence of a two-timescale stochastic approximation scheme. In the standard setting, the limit point of the faster timescale ODE is assumed to be Lipschitz in the slower timescale variable, viewed as a quasi-static external parameter. The aforementioned special structure is important in our case, as $\lambda(\rho) = Q^\rho$ may not in general be continuous, let alone Lipschitz. Interestingly, the fact that we can relax the Lipschitz condition on λ in our case suggests the possibility of potentially useful generalizations of the classic convergence conditions for two-timescale stochastic approximation. We leave this as future work.

CAAC Convergence. This subsection provides convergence guarantees for Algorithm 2. For these results we make Assumption 2 above, as well as the following:

Assumption 4. *The value functions in Algorithm 2 are parameterized as* $V_\nu(s) = \nu^\top \phi(s)$, *where* $\phi(s) = [\phi_1(s) \ \cdots \ \phi_K(s)]^\top \in \mathbb{R}^K$ *is the feature vector associated with* $s \in \mathcal{S}$. *The feature vectors* $\phi(s)$ *are uniformly bounded for any* $s \in \mathcal{S}$, *and the feature matrix* $\Phi = [\phi(s)]^\top_{s\in\mathcal{S}} \in \mathbb{R}^{|\mathcal{S}|\times K}$ *has full column rank. For any* $u \in \mathbb{R}^K$, $\Phi u \neq \mathbb{1}$, *where* $\mathbb{1}$ *is the vector of all ones.*

Assumption 5. *The update of the policy parameter* θ_n *includes a projection operator,* $\Gamma : \mathbb{R}^d \to \Theta \subset \mathbb{R}^d$, *that projects any* θ_n *onto a compact set* Θ.

Assumption 6. *For any* $\theta \in \Theta$, π_θ *is continuously differentiable with respect to* θ, *and the Markov chain under* π_θ *is ergodic.*

Assumptions 4 and 5 are standard in convergence analyses for two-timescale actor-critic algorithms (Tsitsiklis & Van Roy, 1999; Bhatnagar et al., 2009). Assumption 4 is needed to guarantee convergence of the critic in Lemma 7, while Assumption 5 is needed to ensure boundedness of the actor parameters. Note that the projection in Assumption 5 is merely for technical reasons, and is usually not required in practice. Furthermore, so long as Θ is taken to be large enough, it will contain at least one local optimum of $L(\theta)$. Finally, Assumption 6 is required to ensure the existence of the gradients in Lemma 8 and guarantee that the ODEs considered in Theorem 3 are well-posed.

Now we are ready to establish the convergence of Algorithm 2, again using the machinery of two-timescale stochastic approximation. For notational convenience, let $D^\theta = \mathrm{diag}\{d^{\pi_\theta}\} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$, where d^{π_θ} is the stationary distribution of the Markov chain induced by policy π_θ, and $r^\theta = [r^\theta(s)]^\top_{s\in\mathcal{S}} \in \mathbb{R}^{|\mathcal{S}|}$, where $r^\theta(s) = \sum_{a\in\mathcal{A}} \pi_\theta(a \,|\, s) r(s,a)$. Moreover, let $P^\theta \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ be the transition probability matrix of states under policy π_θ, i.e., $P^\theta(s' \,|\, s) = \sum_{a\in\mathcal{A}} \pi_\theta(a \,|\, s) p(s' \,|\, s,a)$ for any $s, s' \in \mathcal{S}$. We first show the convergence of the critic.

Lemma 7. *For a given policy* π_θ *and for both* $i = r, c$, *with* $\{\mu_n^i\}$ *and* $\{\nu_n^i\}$ *generated from the critic step in Algorithm 2, we have* $\lim_{n\to\infty} \mu_n^i = J_i(\theta)$ *and* $\lim_{n\to\infty} \nu_n^i = \nu_\theta^i$ *a.s., where* ν_θ^r *and* ν_θ^c *are the unique solutions to*

$$\Phi^\top D^\theta \big[r^\theta - J_r(\theta)\cdot \mathbb{1} + P^\theta(\Phi\nu_\theta^r) - \Phi\nu_\theta^r\big] = \mathbf{0},$$
$$\Phi^\top D^\theta \big[c^\theta - J_c(\theta)\cdot \mathbb{1} + P^\theta(\Phi\nu_\theta^c) - \Phi\nu_\theta^c\big] = \mathbf{0}.$$

Lemma 7 shows that the sequences $\{\nu_n^r\}$ and $\{\nu_n^c\}$ both converge to the limiting point of the TD(0) algorithm with linear function approximation, i.e., ν_θ^r and ν_θ^c. We note that the resulting ν_θ^r and ν_θ^c, and thus the estimates $V_{\nu_\theta^r}$ and $V_{\nu_\theta^c}$, do not provide an unbiased estimate of the policy gradient given by (7), in general. However, the bias of policy gradient estimates based on the critic step can be characterized as follows, which is an analog of Lemma 4 in (Bhatnagar et al., 2009).

Lemma 8. *For any* $\theta \in \Theta$, *let*

$$\delta_n^{\theta,r} = r_n - J_r(\theta) + [\phi(s_{n+1})]^\top \nu_\theta^r - [\phi(s_n)]^\top \nu_\theta^r,$$
$$\delta_n^{\theta,c} = c_n - J_c(\theta) + [\phi(s_{n+1})]^\top \nu_\theta^c - [\phi(s_n)]^\top \nu_\theta^c,$$

denote the stationary estimates of the TD-errors, let

$$\bar{V}_r^\theta(s) = \mathbb{E}\Big\{r(s,a) - J_r(\theta) + [\phi(s')]^\top \nu_\theta^r\Big\},$$
$$\bar{V}_c^\theta(s) = \mathbb{E}\Big\{c(s,a) - J_c(\theta) + [\phi(s')]^\top \nu_\theta^c\Big\},$$

where the expectation is taken over $a \sim \pi_\theta(\cdot \,|\, s)$ *and* $s' \sim p(\cdot \,|\, s,a)$, *and let* $e_i^\theta = \sum_{s\in\mathcal{S}} d^{\pi_\theta}(s)\big[\nabla_\theta \bar{V}_i^\theta(s) - [\phi(s)]^\top \nabla_\theta \nu_\theta^i\big]$ *for* $i = r, c$. *Then,*

$$\mathbb{E}\left[\frac{J_r(\theta)}{J_c(\theta)} \cdot \nabla_\theta \log \pi_\theta(a_n \,|\, s_n) \cdot \left(\frac{\delta_n^{\theta,r}}{J_r(\theta)} - \frac{\delta_n^{\theta,c}}{J_c(\theta)}\right) \Big|\, \theta\right]$$
$$= \nabla_\theta L(\theta) + \frac{J_r(\theta)}{J_c(\theta)} \cdot \left[\frac{e_r^\theta}{J_r(\theta)} - \frac{e_c^\theta}{J_c(\theta)}\right].$$

Alternatively, it may be possible to use compatible features to obtain unbiased gradient estimates (Sutton et al., 2000; Bhatnagar et al., 2009).

Now we are ready to establish the convergence of the actor step, and thus the actor-critic algorithm. Given any continuous function $f : \Theta \to \mathbb{R}^d$, we define the function $\hat{\Gamma}(\cdot)$ using the projection operator Γ in Assumption 5 to be

$$\hat{\Gamma}(f(\theta)) = \lim_{\eta\to 0^+} \left[\Gamma(\theta + \eta \cdot f(\theta)) - \theta\right]/\eta.$$

Define $e^\theta = \big[J_r(\theta)/J_c(\theta)\big] \cdot \big[e_r^\theta/J_r(\theta) - e_c^\theta/J_c(\theta)\big]$, and consider the ODE

$$\dot{\theta} = \hat{\Gamma}(-\nabla_\theta L(\theta) - e^\theta),$$

with the set of asymptotically stable equilibria $\mathcal{Z}$. In addition, define the ϵ-neighborhood of $\mathcal{Z}$ as $\mathcal{Z}^\epsilon = \{x \,|\, \|x - z\| \le \epsilon, z \in \mathcal{Z}\}$. We then have the following theorem.

Theorem 3. *Under Assumptions 2 and 4–6, given any* $\epsilon > 0$, *there exists* $\delta > 0$ *such that, for* $\{\theta_n\}$ *obtained from Algorithm 2, if* $\sup_{\theta_n} \|e^{\theta_n}\| < \delta$, *then* $\theta_n \to \mathcal{Z}^\epsilon$ *a.s. as* $n \to \infty$.

Theorem 3 establishes the almost sure convergence of the actor-critic algorithm to a neighborhood of an equilibrium point when linear function approximation is used for the critic. Note that if the linear function class is expressive enough, i.e., both the error terms e_r^θ and e_c^θ are small, then the neighborhood will also be small.

5. Empirical Evaluation

In this section, we present numerical experiments that illustrate the convergence results obtained in the preceding. In addition to providing strong support for our theory, our simulations suggest both CARVI Q-learning and CAAC enjoy promising performance and merit further study. We evaluate tabular CARVI Q-learning and linear critic CAAC on eight discrete domains. These experiments are provided to illustrate our theoretical results.

Experiment Setup. We considered two different sizes of CAMDP for our experiments: $|\mathcal{S}| = |\mathcal{A}| = 5$ and $|\mathcal{S}| = |\mathcal{A}| = 10$. For simplicity, we set $\mathcal{S} = \{0, 1, \dots, |\mathcal{S}| - 1\}, \mathcal{A} = \{0, 1, \dots, |\mathcal{A}| - 1\}$. We chose four different reward and cost function combinations of varying complexity, which can be found at the top of Figure 1. Our choice of reward and cost functions was ultimately arbitrary, but led to experiments that exhibited instructive behavior. For each size and reward/cost combination, we randomly generated a transition kernel $P(\cdot|s, a)$ that satisfies Assumption 3, completing the specification of the corresponding CAMDP.

We implemented the algorithms almost exactly as in Algorithms 1 and 2, with two key differences: we used fixed stepsizes $\alpha_t = \alpha, \beta_t = \beta$, with $\beta \leq \alpha$, and we also introduced an additional fixed learning rate μ_{lr} for the μ^r, μ^c updates in Algorithm 2. Though this violates Assumption 2 and has the potential to lead to instability around optima, constant stepsizes are widely adopted in practice and did not greatly affect average performance in our experiments. The Q function for Algorithm 1 contained an entry for each state-action pair, providing a tabular representation satisfying Assumption 1. The policy for Algorithm 2 was chosen to be the softmax function

$$\pi_\theta(a_i|s) = \frac{\exp(\theta^T \psi(s, a_i))}{\sum_j \exp(\theta^T \psi(s, a_j))},$$

where $\theta \in \mathbb{R}^{|\mathcal{S}|\cdot|\mathcal{A}|}$ and $\psi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{|\mathcal{S}|\cdot|\mathcal{A}|}$ maps each state-action pair to a unique standard basis vector $e_k \in \mathbb{R}^{|\mathcal{S}|\cdot|\mathcal{A}|}$, where e_k has a 1 in its kth entry and 0 everywhere else. Note that this choice of policy satisfies Assumption 6. We did not use a projection operation to satisfy Assumption 5, but this is also common in practice and, in any case, the CAAC algorithm's policy parameter iterates converged in all our tests. Finally, since $\mathcal{S} = \{0, 1, \dots, |\mathcal{S}| - 1\}$, we used simple linear function approximators of the form $V_\nu(s) = \nu^T \phi(s) = \nu_0 s + \nu_1$ for the value functions in Algorithm 2, satisfying Assumption 4. To produce the data used in the figures below, we ran 15 independent replications of each algorithm on each of the eight synthetic CAMDPs. Hyperparameters were determined through experimentation and are included in the supplementary material.

Discussion. The empirical results presented in Figure 1 demonstrate clear convergence of Algorithms 1 and 2 on a variety of different CAMDP environments and illustrate important features of our theoretical analysis. Recall from the convergence results of Section 4 that Algorithm 1 is guaranteed to converge to the globally optimal (ρ^*, Q^*), while Algorithm 2 converges to a neighborhood of a local optimum. This implies that the optimal ratio obtained by the RVI Q-learning algorithm should always provide an upper bound on the ratio obtained by the actor-critic algorithm. This relationship clearly holds in Figure 1, as Algorithm 1 does as well as or better than Algorithm 2 in all cases. Interestingly, our implementation of Algorithm 2 manages to achieve performance comparable to that of Algorithm 1 on several problems, indicating that our actor-critic algorithm is capable of achieving near-optimal and even optimal performance.

Deep CARVI Q-learning. For this paper we also implemented a version of CARVI Q-learning using neural networks for the Q function approximators and tested it on a cost-aware modification of the classic MountainCar control environment (Moore, 1990) provided by OpenAI's Gym RL testbed (Brockman et al., 2016). In these experiments we augmented the Gym environment's reward with a cost function providing additional information about the state space. In the best trials, our CARVI Q-learning agent successfully learned to solve the problem after training for only a small number of episodes. A more detailed discussion of these experiments can be found in the supplementary material.

The empirical results in the supplementary materials motivate that deep RL algorithms based on our theory are worth further study by showing promising performance on a novel, cost-aware version of the familiar MountainCar problem. The results are *not* intended to show that CARVI Q-learning outperforms existing state-of-the-art algorithms on standard benchmarks like MountainCar, which do not take costs into account. To our knowledge, benchmark problems do not exist for our cost-aware setting. Due to the presence of costs, the cost-aware MountainCar environment that we developed is distinct from classic MountainCar, and solving the cost-aware version does not guarantee a solution to the original MountainCar. As described in the supplementary material, including costs alters the problem, since the agent's objective is now to maximize expected average reward divided by expected average cost. Figure 4 of the supplementary materials shows that CARVI Q-learning succeeds in improving this objective. Nonetheless, Figure 3 also motivates further study of applying our algorithms to reward shaping: though it is solving a cost-aware problem, deep CARVI

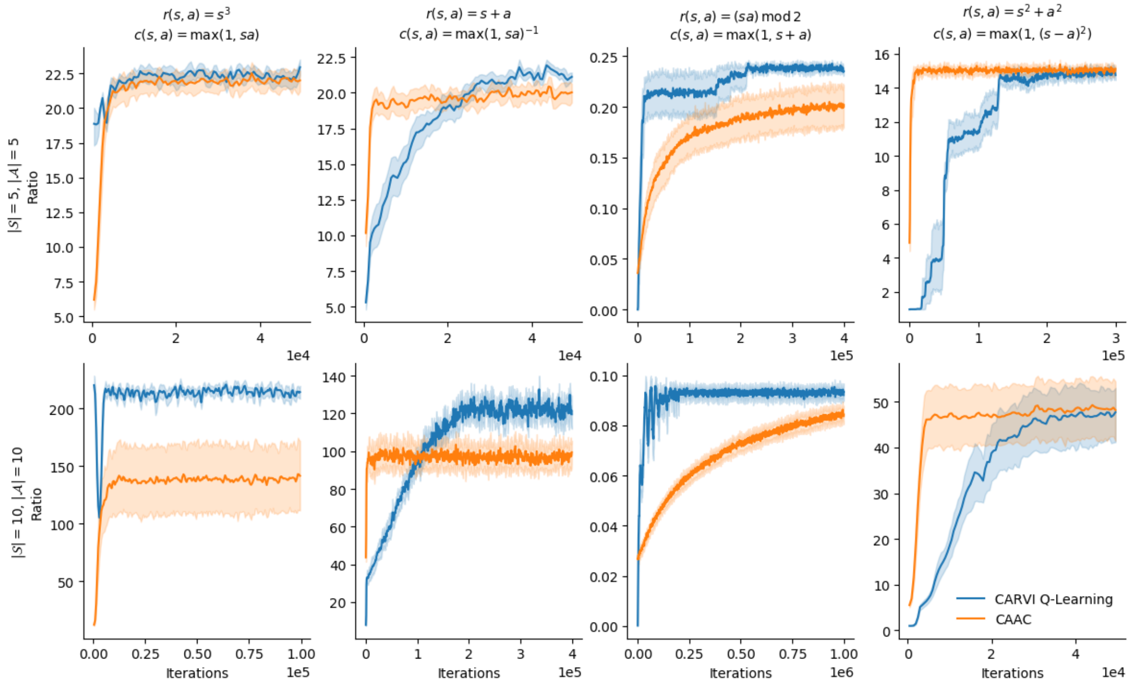

Figure 1. Comparison of tabular CARVI Q-learning and CAAC with a linear critic on an array of synthetic CAMDP environments. Ratios are computed by taking the average over a moving window of 1000 timesteps. Learning curves show the mean and 90% confidence intervals over 15 independent replications.

Q-learning demonstrates significant learning on the original MountainCar, nearly solving it in the best trials.

6. Conclusion

In this paper, we have developed and studied two new RL algorithms with convergence guarantees for CAMDPs. We have also presented numerical results supporting our theory and indicating promising performance. Important future directions include finite-time analysis and practical applications of our algorithms.

Acknowledgements

The research of W. Suttle and J. Liu was supported in part by the US Army Research Laboratory (ARL) Cooperative Agreement W911NF-21-2-0098. Z. Yang acknowledges Simons Institute (Theory of Reinforcement Learning); part of this work took place while he was a VMWare Research Fellow at the Simons Institute. The authors wish to thank the anonymous reviewers for their helpful comments.

References

Abounadi, J., Bertsekas, D., and Borkar, V. S. Learning algorithms for Markov Decision Processes with average cost. *SIAM Journal on Control and Optimization*, 40(3): 681–698, 2001.

Altman, E. Constrained Markov Decision Processes with total cost criteria: Lagrangian approach and dual linear program. *Mathematical Methods of Operations Research*, 48(3):387–417, 1998.

Altman, E. *Constrained Markov Decision Processes*, volume 7. CRC Press, 1999.

Berkenkamp, F., Turchetta, M., Schoellig, A., and Krause, A. Safe model-based reinforcement learning with stability guarantees. In *Advances in Neural Information Processing Systems*, pp. 908–918, 2017.

Bertsekas, D. P. *Dynamic Programming and Optimal Control, Vol. II, 4E*. Athena scientific Belmont, MA, 2012.

Bhatnagar, S. An actor–critic algorithm with function approximation for discounted cost constrained markov decision processes. *Systems & Control Letters*, 59(12): 760–766, 2010.

Bhatnagar, S., Sutton, R., Ghavamzadeh, M., and Lee, M. Natural actor-critic algorithms. *Automatica*, 45(11):2471–2482, 2009.

Borkar, V. S. An actor-critic algorithm for constrained Markov Decision Processes. *Systems & Control Letters*, 54(3):207–213, 2005.

Borkar, V. S. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press, 2008.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. OpenAI Gym. *arXiv preprint arXiv:1606.01540*, 2016.

Cheng, R., Orosz, G., Murray, R. M., and Burdick, J. W. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3387–3395, 2019.

Chow, Y., Ghavamzadeh, M., Janson, L., and Pavone, M. Risk-constrained reinforcement learning with percentile risk criteria. *The Journal of Machine Learning Research*, 18(1):6070–6120, 2017.

Ding, D., Zhang, K., Basar, T., and Jovanovic, M. Natural policy gradient primal-dual method for constrained markov decision processes. In *Advances in Neural Information Processing Systems*, volume 33, pp. 8378–8390, 2020.

García, J. et al. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(42):1437–1480, 2015.

Geist, M., Scherrer, B., and Pietquin, O. A theory of regularized markov decision processes. In *International Conference on Machine Learning*, pp. 2160–2169. PMLR, 2019.

Grill, J.-B., Domingues, O. D., Ménard, P., Munos, R., and Valko, M. Planning in entropy-regularized markov decision processes and games. In *Advances in Neural Information Processing Systems*, pp. 12404–12413, 2019.

Gupta, H., Srikant, R., and Ying, L. Finite-time performance bounds and adaptive learning rate selection for two time-scale reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 4706–4715, 2019.

Hong, M., Wai, H.-T., Wang, Z., and Yang, Z. A two-timescale framework for bilevel optimization: Complexity analysis and application to actor-critic. *arXiv preprint arXiv:2007.05170*, 2020.

Keating, C. and Shadwick, W. F. A universal performance measure. *Journal of Performance Measurement*, 6(3): 59–84, 2002.

Kiran, B. R., Sobh, I., Talpaert, V., Mannion, P., Al Sallab, A. A., Yogamani, S., and Pérez, P. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

Konda, V. R. and Tsitsiklis, J. N. Actor-critic algorithms. In *Advances in Neural Information Processing Systems*, pp. 1008–1014, 2000.

Li, G., Wei, Y., Chi, Y., Gu, Y., and Chen, Y. Sample complexity of asynchronous q-learning: Sharper analysis and variance reduction. In *Advances in Neural Information Processing Systems*, volume 33, pp. 7031–7043, 2020.

Mataric, M. J. Reward functions for accelerated learning. In *Machine Learning Proceedings*, pp. 181–189. Elsevier, 1994.

Moody, J. and Saffell, M. Learning to trade via direct reinforcement. *IEEE Transactions on Neural Networks*, 12(4):875–889, 2001.

Moore, A. W. Efficient memory-based learning for robot control. 1990.

Ng, A. Y., Harada, D., and Russell, S. Policy invariance under reward transformations: Theory and application to reward shaping. In *International Conference on Machine Learning*, volume 99, pp. 278–287, 1999.

Peters, J. and Schaal, S. Natural actor-critic. *Neurocomputing*, 71(7):1180–1190, 2008.

Prashanth, L. and Ghavamzadeh, M. Variance-constrained actor-critic algorithms for discounted and average reward MDPs. *Machine Learning*, 105(3):367–417, 2016.

Puterman, M. L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

Ren, Z. and Krogh, B. H. Markov Decision Processes with fractional costs. *IEEE Transactions on Automatic Control*, 50(5):646–650, 2005.

Sharpe, W. F. Mutual fund performance. *The Journal of Business*, 39(1):119–138, 1966.

Sortino, F. A. and Price, L. N. Performance measurement in a downside risk framework. *The Journal of Investing*, 3(3):59–64, 1994.

Sutton, R. S. Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44, 1988.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. Cambridge: MIT press, 1998.

Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, pp. 1057–1063, 2000.

Tanaka, T. A partially observable discrete time markov decision process with a fractional discounted reward. *Journal of Information and Optimization Sciences*, 38(1):21–37, 2017.

Tanaka, T. Saddle–point type optimality criteria and dualities in fractional markov decision processes with constraints. *Journal of Information and Optimization Sciences*, 40(4):957–972, 2019.

Tsitsiklis, J. N. and Van Roy, B. Average cost temporal-difference learning. *Automatica*, 35(11):1799–1808, 1999.

Vieillard, N., Kozuno, T., Scherrer, B., Pietquin, O., Munos, R., and Geist, M. Leverage the average: an analysis of kl regularization in reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 12163–12174, 2020.

Watkins, C. J. and Dayan, P. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

Wu, Y., Zhang, W., Xu, P., and Gu, Q. A finite time analysis of two time-scale actor critic methods. *arXiv preprint arXiv:2005.01350*, 2020.

Young, T. W. Calmar ratio: A smoother tool. *Futures*, 20 (1):40, 1991.

Yu, M., Yang, Z., Kolar, M., and Wang, Z. Convergent policy optimization for safe reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 3127–3139, 2019.