Actor-Critic Algorithms with Online Feature Adaptation

K. J. PRABUCHANDRAN and SHALABH BHATNAGAR, Indian Institute of Science, Bangalore
VIVEK S. BORKAR, Indian Institute of Technology, Mumbai

We develop two new online actor-critic control algorithms with adaptive feature tuning for Markov Decision Processes (MDPs). One of our algorithms is proposed for the long-run average cost objective, while the other works for discounted cost MDPs. Our actor-critic architecture incorporates parameterization both in the policy and the value function. A gradient search in the policy parameters is performed to improve the performance of the actor. The computation of the aforementioned gradient, however, requires an estimate of the value function of the policy corresponding to the current actor parameter. The value function, on the other hand, is approximated using linear function approximation and obtained from the critic. The error in approximation of the value function, however, results in suboptimal policies. In our article, we also update the features by performing a gradient descent on the Grassmannian of features to minimize a mean square Bellman error objective in order to find the best features. The aim is to obtain a good approximation of the value function and thereby ensure convergence of the actor to locally optimal policies. In order to estimate the gradient of the objective in the case of the average cost criterion, we utilize the policy gradient theorem, while in the case of the discounted cost objective, we utilize the simultaneous perturbation stochastic approximation (SPSA) scheme. We prove that our actor-critic algorithms converge to locally optimal policies. Experiments on two different settings show performance improvements resulting from our feature adaptation scheme.

Categories and Subject Descriptors: L.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

General Terms: Design, Algorithms

Additional Key Words and Phrases: Markov decision processes, actor-critic algorithms, function approximation, feature adaptation, online learning, residual gradient scheme, temporal difference learning, stochastic approximation, Grassmann manifold, SPSA, policy gradients

ACM Reference Format:
K. J. Prabuchandran, Shalabh Bhatnagar, and Vivek S. Borkar. 2016. Actor-critic algorithms with online feature adaptation. ACM Trans. Model. Comput. Simul. 26, 4, Article 24 (February 2016), 26 pages.
DOI: http://dx.doi.org/10.1145/2868723

The work of K. J. Prabuchandran has been supported through a fellowship from Tata Consultancy Services, India. S. Bhatnagar acknowledges support from projects supported by the Department of Science and Technology, Xerox Corporation, USA, and the Robert Bosch Centre, IISc. V. S. Borkar acknowledges support from a J.C. Bose fellowship & a grant from the Department of Science and Technology for a project titled "Distributed Computation over Large Networks and High-Dimensional Data Analysis."
Authors' addresses: K. J. Prabuchandran and S. Bhatnagar, Department of Computer Science & Automation, Indian Institute of Science, Bangalore 560012; emails: {prabu.kj, shalabh}@csa.iisc.ernet.in; V. S. Borkar, Department of Electrical Engineering, Indian Institute of Technology, Bombay, Powai, Mumbai 400076; email: borkar.vs@gmail.com.

DOI: http://dx.doi.org/10.1145/2868723

1. INTRODUCTION

We begin with a discussion on reinforcement learning (RL) approaches.

1.1. RL Preliminaries

Optimal sequential decision-making problems under uncertainty are often formulated as Markov Decision Processes (MDPs). An MDP is a 4-tuple consisting of a state space, an action space, a probabilistic transition mechanism, and a cost structure. In an MDP setting, the goal is to find the optimal sequence of actions that minimizes a certain long-term cost objective. In infinite horizon problems, based on the nature of the application, quite often one minimizes either the long-term discounted cost or the average cost. In our article, we develop algorithms for solving MDPs to minimize a weighted long-term discounted cost objective as well as the average cost objective.

Classical approaches for solving MDP such as value iteration or policy iteration, (see Bertsekas [2011] and Barto [1998]) involve solving the Bellman equation of optimality and are based on Dynamic Programming (DP). These methods require complete model information in the form of transition probabilities and the cost structure. Also, such approaches are suitable only for MDPs with small state-action spaces. However, in most practical problems of interest, the model of the MDP is unknown, and further, the MDP has large state and action spaces. RL methods alleviate the aforementioned difficulties by using a combination of simulated outcomes and function approximation. Most RL methods are based on policy or value iteration schemes. The RL algorithms perform parameter updates (most often) using only a sample trajectory obtained from simulation. So we can solve the MDP even when the model is unknown; however, the outcomes can be simulated. The use of function approximation helps obtain solutions for large MDPs, albeit at the cost of convergence to suboptimal policies. RL algorithms involve parameterizing the value function (critic-only methods, Lagoudakis and Parr [2003]) or policy (actor-only methods, Marbach and Tsitsiklis [2001]) or both (actor-critic methods, Konda and Tsitsiklis [2003] and Bhatnagar et al. [2009]).

The actor-critic methods [Barto et al. 1983] combine ideas from actor-only and critic-only methods by parameterizing both value function and policy. The critic uses an approximation architecture involving state or state-action features and simulation to learn the value function of the policy for the "current" actor policy parameter (denoted θ). The parameter θ is then updated along the negative gradient of the objective criterion (see Konda and Tsitsiklis [2003], Sutton et al. [2000], and Bhatnagar et al. [2009]). In Konda and Tsitsiklis [2003], the critic estimates the state-action value function using either the TD(1) or TD(λ) update rule, while in Bhatnagar et al. [2009], the critic computes the estimate of the state-value function using the TD(0) critic. Convergence to near-optimal policies for such schemes can be ensured only when the approximated value function is close to the original. The error in approximation depends on the choice of the features used in the critic to approximate the value function. In most RL algorithms, the features are fixed a priori, as a result of which the approximation may be highly inaccurate and the resulting policy from the actor may perform poorly. To overcome this problem, we propose to adaptively tune these features in the critic so as to obtain an "optimal" set of features, resulting in a good approximation to the value function and thereby resulting in an optimal policy. The schematic of this actor-critic architecture incorporating the feature adaptation is given in Figure 1.

1.2. Our Contributions

In this article, we present two online actor-critic control algorithms, one for the average cost MDP and another for the weighted discounted cost MDP that incorporates the basis feature adaptation. The feature search is performed using gradient descent on the

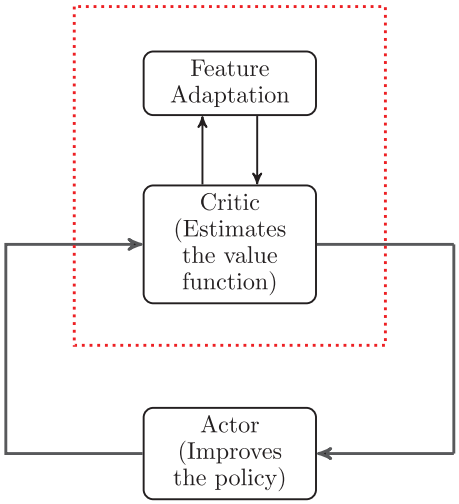


Fig. 1. New actor-critic architecture.

Grassmann manifold of features. Our algorithms address the problem of control, unlike many other feature adaptation algorithms, which are primarily policy evaluation procedures. Our algorithms are in fact the first control algorithms that perform gradient search on the Grassmannian for feature tuning. We provide proofs of convergence of our control algorithms in both the settings (average and weighted discounted cost, respectively) to locally optimal policies. Our algorithms exhibit good empirical performance on two MDP settings. We also show the results of experiments using similar algorithms with temporal difference (TD) run on the faster timescale in place of the residual gradient scheme. It may be noted that unlike the scheme in Castro and Mannor [2010], we do not consider parameterized basis functions and instead our algorithms search in the entire Grassmann manifold of features to find the optimal bases. We provide a schema for obtaining the sample complexity estimates for our algorithms in the Online Appendix. A short version of this article containing only the weighted discounted cost setting and without the convergence proofs is available in Prabuchandran et al. [2014].

In the weighted discounted cost setting, we extend the PE-based algorithm in Bhatnagar et al. [2013a] to the problem of control by performing gradient search on the weighted long-term discounted cost using simultaneous perturbation stochastic approximation (SPSA). In the average cost setting, we develop a feature adaptation algorithm as in Bhatnagar et al. [2013a] to estimate the differential value function of the policy and then use the obtained estimate to develop an RL control algorithm by performing gradient search on the long-run average cost using policy gradient estimates (cf. Sutton et al. [2000] and Konda and Tsitsiklis [2003]). We note here that the policy gradient theorem is applicable even for the discounted cost objective. However, obtaining a policy gradient estimate using simulation is difficult in the discounted cost case. Thus, we resort to gradient estimates based on the SPSA technique (see Spall [1992] and Bhatnagar et al. [2013b]) to update the policy parameters for the discounted cost objective. We perform feature adaptation using gradient search on the Grassmann manifold of features for both the cost objectives. As we consider the model-free setting, we use simulation samples to estimate this gradient and incorporate multiple timescale stochastic approximations in our procedure. The critic uses two timescales to estimate the value function. For any given feature value, both our algorithms perform

gradient search in the parameter space via suitable residual gradient schemes on the faster timescale and, on a medium timescale, perform gradient search in the Grassmann manifold of features. The actor then on the slowest timescale uses the estimate of the value function to update the policy parameters using SPSA for the discounted cost case and policy gradient for the average cost setting, respectively.

1.3. Related Work

We begin with a brief review of literature related to optimization on manifolds. The optimization of the scalar function on the manifold has been extensively studied in the optimization and machine-learning literature. In Smith [1993], intrinsic approaches for optimization on an arbitrary Riemann manifold analogous to optimization on the Euclidean space have been developed. They establish theory of large-step manifold methods and apply the same to the subspace tracking problem. For learning methods that treat each data point as a linear subspace, distance metrics such as the Projection metric and Binet-Cauchy metric have been explored, and a kernel for the corresponding metrics has been developed in Hamm and Lee [2008]. They state that if the learning methods perform feature extraction in the Euclidean space and classification in the non-Euclidean space, the inconsistency results in weak guarantees. So they integrate both the feature extraction and classification around a Grassmann kernel to develop a Grassmann Discriminant Analysis for classification problems with real image databases.

In Meyer et al. [2011], novel gradient-based algorithms for learning a parametric model on the space of fixed-rank positive semidefinite matrices has been developed. Their algorithms intrinsically maintain the gradient updates to belong to the nonlinear search space. In Wolf and Shashua [2003], a kernel $k(A, B)$ is defined for a pair of matrices (A, B) based on the principal angle between two linear subspaces generated by A and B. This kernel is then utilized to perform a classification task on a video sequence.

Next, we discuss work in the literature related to feature adaptation approaches. Various feature adaptation methods to approximate the value function have been studied in the literature (cf. Menache et al. [2005], Keller et al. [2006], Sun et al. [2011], and Mahadevan and Liu [2010]). In Menache et al. [2005], radial basis functions (RBFs) are considered with parameterization as the feature vectors. The parameters of RBF are then tuned using two methods: gradient descent and the cross-entropy method. A general framework for studying adaptive bases as an extension of Menache et al. [2005] is presented in Yu and Bertsekas [2009]. An automatic basis function construction has been proposed in Keller et al. [2006], where a high-dimensional state space is mapped to a low-dimensional space using neighborhood component analysis and state aggregation is used in the lower-dimensional space to construct the basis functions. In Mahadevan and Liu [2010], Laurent series expansion of the discounted value function is utilized to construct a Drazin basis representation. An approach based on state aggregation and linear function approximation is presented in Baras and Borkar [2000], where the feature matrix is kept fixed, but state aggregation is done adaptively using estimates of the approximate value function. An incremental procedure for expanding the available set of basis functions by using the TD error from the "current" set to obtain a new basis function is proposed in Sun et al. [2011]. Another approach of expanding the set of basis functions using the Bellman error is provided in Parr et al. [2007], where the basis functions are not considered parameterized and the Bellman error is included as an additional basis in each iteration.

In Bhatnagar et al. [2012], a nonparameterized adaptive scheme for basis selection is proposed in conjunction with TD. Two columns of the feature matrix corresponding to the two smallest components of the weight vector parameter are respectively replaced

by the most recent estimate of the value function obtained using TD and a randomly generated column, without modifying the remaining columns. In Bhatnagar et al. [2013a], a policy evaluation (PE)-based RL algorithm incorporating adaptive feature tuning has been developed to estimate the value function for a discounted cost MDP. In Konda and Tsitsiklis [2003], actor-critic algorithms with function approximation are proposed and analyzed. The actor-critic algorithms in Konda and Tsitsiklis [2003] do not give useful information about the gradient at certain parameterized values if the critic feature vectors are either close to zero or almost linearly dependent, which can make the algorithm unstable. To alleviate this difficulty, Rohanimanesh et al. [2009] study the problem of designing features for the state-action value function under the Gaussian actor policies. They also show the proto-value functions originally devised for discretized action spaces (see Mahadevan and Maggioni [2007]), generalized to the continuous action actor-critic domain.

All of these methods above have been developed for approximating the value function of a given policy for the discounted cost MDP. Further, extension of these methods in the context of control (or policy improvement) with adaptive bases for the discounted case has not been studied in the literature. The problem of control with adaptive bases for the average cost setting has been considered in Castro and Mannor [2010] and actor-critic algorithms developed. One of the natural actor-critic algorithms in Bhatnagar et al. [2009] is extended in Castro and Mannor [2010] by incorporating an adaptive basis selection scheme. However, the basis functions in Castro and Mannor [2010] are parameterized, and the parameters are updated using the method given in Menache et al. [2005].

The rest of the article is organized as follows: In Section 2, we discuss the problem setting of MDP under both the cost criteria and linear function approximation. In Section 3, we obtain the expression for the gradient on the Grassmann manifold of features using a result in Edelman et al. [1998] under both the cost objectives. In Section 4, we present our actor-critic control algorithms for the average and the discounted cost objectives, respectively. Proofs of convergence using the ordinary differential equation (ODE) technique for the two objectives are presented in Section 5. We also show here (see Section 5.1) that the feature recursion corresponding to any given policy in fact converges to a global optimum (in the feature space). This result has not been shown in previous literature including Bhatnagar et al. [2013a]. Results of numerical experiments using our algorithms and TD are presented in Section 6. Finally, we present our concluding remarks and discuss future work in Section 7.

2. THE MDP FRAMEWORK AND ACTOR-CRITIC PRELIMINARIES

We consider an MDP with finite state and action spaces denoted by S and A, respectively. We assume $S = \{1, 2, \ldots, N\}$. For simplicity, we assume that all actions in A are feasible in every state. The probabilistic transition mechanism governing the evolution of the MDP is described via the map $p : S \times S \times A \to \mathcal{R}$, where $p(i, j, a), i, j \in S, a \in A$ gives the probability of moving to a next state j from the current state i under the current action a. The cost function is a mapping $k : S \times A \to \mathcal{R}$, where $k(i, a), i \in S, a \in A$ denotes the single-stage cost when the state is i and the action chosen is a.

It is often convenient to view actions as being chosen from a given policy, that is, a rule or method for selecting actions. A deterministic policy $\bar{\pi}$ is a sequence of maps $\bar{\pi} \stackrel{\triangle}{=} \{\mu_0, \mu_1, \ldots\}$ with $\mu_j : S \to A, j \geq 0$. If the policy can be represented via a single map, that is, $\mu_j \equiv \mu, \forall j \geq 0$, where μ does not depend on j, we call $\bar{\pi}$ or, by abuse of notation, μ itself, a stationary deterministic policy (SDP). A stationary randomized policy (SRP) π is a mapping that assigns for each state $i \in S$ a probability distribution over A. In our work, we consider only SRPs and parameterize the policy π using a

parameter $\theta \in \mathcal{R}^L$. We will denote such a parameterized policy as π_θ. For each pair $(i, a) \in S \times A$, $\pi_\theta(i, a)$ denotes the probability of choosing action a when the current state is i. With a slight abuse of notation, we will interchangeably use SRP θ for SRP π_θ (corresponding to parameter θ). Note that under any SRP, the state sequence $\{X_n\}$ as well as the state-action sequence $\{(X_n, Z_n)\}$ of the MDP form Markov chains with state spaces S and $S \times A$, respectively. We make the following assumptions about the policy parameterization.

Assumption 1. *Under any SRP $\theta \in \mathcal{R}^L$, the Markov chains $\{X_n\}$ and $\{(X_n, Z_n)\}$ resulting from the MDP are both aperiodic and irreducible.*

Assumption 2. *For any state-action pair (i, a), $\pi_\theta(i, a)$ is continuously differentiable in the parameter θ.*

These assumptions are natural and essential for any actor-critic architecture (see Konda and Tsitsiklis [2003] and Bhatnagar et al. [2009]). A commonly used parameterization that satisfies Assumption 2 is the parameterized Gibbs distribution, that is, $\pi_\theta(i, a) = \frac{\exp(\theta^T \sigma(i,a))}{\sum_a \exp(\theta^T \sigma(i,a))}$, where $\sigma(i, a) \in \mathcal{R}^L$ corresponds to the policy feature for the state-action tuple (i, a).

2.1. The Long-Run Average Cost Criterion

In the average cost case, the goal is to find an SRP θ^* that minimizes over all SRP θ the long-term average cost $\rho(\theta)$ given by

$$\rho(\theta) = \lim_{T\to\infty} \frac{1}{T} \mathbb{E}\left[\sum_{n=0}^{T-1} k(X_n, Z_n) \Big| \theta\right] \tag{1}$$

$$= \sum_{i\in S} d^\theta(i) \sum_{a\in A} \pi_\theta(i, a) k(i, a), \tag{2}$$

where $d^\theta(i), i \in S$ corresponds to the stationary distribution of the Markov chain $\{X_n\}$.

Further, the differential state-action value function for a given SRP θ is defined by

$$Q^\theta(i, a) = \sum_{n=1}^{\infty} \mathbb{E}[k(X_n, Z_n) - \rho(\theta) | X_0 = i, Z_0 = a, \theta], \forall i \in S, a \in A. \tag{3}$$

Note that Q^θ gives the long-term expected sum of differences between the single-stage and the average costs incurred by choosing an action a in state i at time 0 and following the SRP θ thereafter. We also define the differential state value function $V^\theta(i), i \in S$ for an SRP θ as

$$V^\theta(i) = \sum_{a\in A} \pi_\theta(i, a) Q^\theta(i, a) \forall i \in S. \tag{4}$$

We accomplish the goal of minimizing the average cost $\rho(\theta)$ by performing a gradient search in the parameter θ. These (policy gradient) methods compute the gradient of $\rho(\theta)$ and update θ along the negative gradient direction of $\rho(\theta)$ as follows:

$$\theta_{n+1} = \theta_n - c_n \nabla \rho(\theta_n),$$

where c_n is the step-size parameter. Assuming stability of the previous iterates, under reasonable conditions, the updates θ_n converge almost surely to a parameter θ^* such that π_{θ^*} is a locally optimum policy. Normally, proving stability of the iterates is not straightforward and we use projection to a given compact subset of the parameter space to ensure the same.

The gradient of the average cost $\rho(\theta)$ for the class of parameterized policies satisfying Assumptions 1 and 2 can be computed using the policy gradient theorem (see Sutton et al. [2000] and Konda and Tsitsiklis [2003]) as

$$\nabla \rho(\theta) = \mathbb{E}_{i\sim d^{\theta}(.),a\sim\pi_{\theta}(i,.)}[(Q^{\theta}(i,a) - V^{\theta}(i))\nabla \ln \pi_{\theta}(i,a)]. \tag{5}$$

Note that to estimate the gradient, V^{θ} in Equation (5) is not needed as it is a function of the state alone. However, including this term minimizes the variance in the estimation of the state-action value function Q^{θ} (see Bhatnagar et al. [2009]). From Equation (5), one can see that to obtain the gradient of $\rho(\theta)$, we need to estimate the advantage function $A^{\theta} \stackrel{\Delta}{=} Q^{\theta} - V^{\theta}$.

In the actor-critic methods, the gradient of the objective at parameter θ is computed in two steps. The actor follows the SRP at the current policy parameter θ. For the SRP θ, the critic obtains the estimate of the advantage function A^{θ} through simulation. Using the critic's estimate, the actor computes the gradient and updates the policy parameter θ. The stochastic update rule (where this computation is said to be performed by the actor) is given by

$$\theta_{n+1} = \theta_n - c_n \delta_n \psi(X_n, Z_n). \tag{6}$$

Here, $\delta_n = k^{\theta_n}(X_n) - \rho_n + \hat{V}^{\theta_n}(X_{n+1}) - \hat{V}^{\theta_n}(X_n)$ corresponds to the temporal difference term, with $k^{\theta_n}(X_n)$ being the single-stage cost for visiting state X_n when following the SRP θ_n, $\hat{V}^{\theta_n}(.)$ being the estimate of the differential state value function, ρ_n being the current estimate of the average cost for the given policy π, and $\psi(X_n, Z_n) \stackrel{\Delta}{=} \nabla \ln \pi_{\theta_n}(X_n, Z_n)$ being the compatible feature associated with the tuple (X_n, Z_n) (see Bhatnagar et al. [2009] and Sutton et al. [2000]). Now to follow the update rule in Equation (6), the actor needs an estimate of the differential state value function. This will be obtained by the critic through a search in the Grassmannian.

The critic solves the problem of prediction by estimating the differential value function of each state under a given SRP θ. The differential state value function $V^{\theta}(i), i \in S$ satisfies the Poisson equation,

$$V^{\theta}(i) + \rho(\theta) = \sum_{a\in A} \pi_{\theta}(i,a) \left[k(i,a) + \sum_{j\in S} p(i,j,a) V^{\theta}(j) \right], \ i \in S. \tag{7}$$

Alternatively, in vector-matrix notation, the same can be written as

$$V^{\theta} = k^{\theta} - \rho(\theta) e + P^{\theta} V^{\theta}, \tag{8}$$

where P^{θ} is the transition probability matrix of $\{X_n\}$ under SRP θ with the (i,j)th component being $P^{\theta}(i,j) = \sum_{a\in A} p_{i,j}(a)\pi_{\theta}(i,a)$. Further, $k^{\theta} \stackrel{\Delta}{=} (\sum_{a\in A} \pi_{\theta}(i,a)k(i,a), i \in S)^T$ is the vector of single-stage costs, and $V^{\theta} \stackrel{\Delta}{=} (V^{\theta}(i), i \in S)^T$ is the differential value function or the vector of differential cost values of individual states under the SRP θ. Also, $e \in \mathcal{R}^N$ is a vector of all ones. Equation (8) does not have a unique solution since if V^{θ} is a solution to Equation (8), then $V^{\theta} + ce$ for any $c \in \mathcal{R}$ is also a solution to Equation (8). A possible workaround is to arbitrarily pick a state s, set its value to 0 (or any number), and solve for the resulting system of equations, which would then yield a unique solution. In effect, this amounts to replacing $V^{\theta}(i)$ by $V^{\theta}(i) - V^{\theta}(s), \forall i \in S$ in Equation (8).

2.2. The Weighted Discounted Cost Criterion

Our goal here is to find an SRP θ^* that minimizes the weighted infinite horizon discounted cost criterion. The weighted discounted cost $\omega(\theta)$ of an SRP θ with given weights

$\beta(l), l \in \{1, 2, \ldots, N\}$ (where $\beta(l) \geq 0, \forall l$ and $\sum_{l=1}^{N} \beta(l) = 1$) is given by

$$\omega(\theta) = \sum_{l=1}^{N} \beta(l) J^{\theta}(l), \tag{9}$$

where J^{θ} corresponds to the (state) value function for a given SRP θ and is defined for all $i \in S$ by

$$J^{\theta}(i) = \sum_{n=0}^{\infty} \mathbb{E}[\gamma^n k(X_n, Z_n) | X_0 = i, \theta], \tag{10}$$

where $\gamma \in (0, 1)$ is a given discount factor of the MDP. The value function J^{θ} gives the expected long-term discounted single-stage cost incurred by following the SRP θ. The weight vector $(\beta(l), l = 1, \ldots, N)^T$ in Equation (9) corresponds to the initial distribution over states of the process $\{X_n\}$.

We achieve the goal of minimizing $\omega(\theta)$ by performing a gradient search for the actor parameter θ. In our algorithm, we update θ along the negative gradient direction of $\rho(\theta)$ using SPSA gradient estimates; see Equation (11). The kth component of the policy parameter θ for $k \in \{1, 2, \ldots, L\}$ gets updated as

$$\theta_{n+1}(k) = \theta_n(k) - c(n) \left[\frac{\omega(\theta_n + \epsilon \Delta_n) - \omega(\theta_n - \epsilon \Delta_n)}{2\epsilon \Delta_n(k)} \right], \tag{11}$$

where Δ_n is a vector of independent Bernoulli random variables $\Delta_n(l), l \in \{1, 2, \ldots, L\}$, taking values ± 1 with probability $\frac{1}{2}$; $\epsilon > 0$ is a given (fixed) parameter; and $c(n)$ is a suitable step-size parameter. We estimate the gradient at the policy parameter θ_n in the SPSA scheme by performing two independent simulations to determine the objectives $\omega(\theta_n + \epsilon \Delta_n)$ and $\omega(\theta_n - \epsilon \Delta_n)$, respectively. However, since our algorithm is online, we update θ_n after every $2M$ iterations. In the first M iterations, we set the policy parameter at $\theta_n + \epsilon \Delta_n$ and estimate the objective $\omega(\theta_n + \epsilon \Delta_n)$. In the subsequent M iterations, we set the policy parameter to $\theta_n - \epsilon \Delta_n$ and estimate $\omega(\theta_n - \epsilon \Delta_n)$. Using these two estimates, we update θ_n according to Equation (11), at the end of $2M$ iterations. Assuming stability of the iterates in Equation (11), under mild conditions, $\{\theta_n\}$ can be shown to converge to a parameter θ^* almost surely so that π_{θ^*} is a locally optimum policy. Again, as before, since showing the stability of iterates is usually not straightforward, we use projection to an a priori chosen compact subset of the parameter space.

In order to estimate the objective $\omega(\theta)$ (for a policy parameter θ) so as to follow the update rule in Equation (11), the actor needs an estimate of the value function J^{θ}. This will be obtained in our algorithm by the critic in a similar manner as for the average cost. The critic solves the problem of prediction by estimating the value function of each state under a given SRP θ. Owing to the same reasons as with the average cost case, one cannot use methods like the value iteration in Equation (8) to obtain J^{θ} and thus one needs to resort to value function approximation.

2.3. Linear Function Approximation

We use function approximation to approximate the differential value function $V^{\theta}(\cdot)$ in the case of average cost and the value function $J^{\theta}(\cdot)$ in the discounted cost setting, respectively. In the rest of the section, we will describe the approximation for the average cost case as the discounted cost value function approximation follows in a similar manner. We approximate the differential value function $V^{\theta}(i) \approx \phi_i^T r$ using a linear function approximator where $\phi_i = (\phi_i(1), \ldots, \phi_i(K))^T$ is a K-dimensional feature

associated with state i. Also, $r = (r_1, \ldots, r_K)^T$ is an associated approximation parameter that assigns weights to the various feature components. The linear approximation architecture is simple to use and provides convergence guarantees when combined with temporal difference learning algorithms (see Tsitsikilis and Van Roy [1997, 1999]). The nonlinear architectures, on the other hand, in some cases are seen to diverge [Baird 1995; Tsitsiklis and Van Roy 1997]. Let Φ denote the $N \times K$ feature matrix with ϕ_i^T, $i \in S$, as its rows. Thus, $\Phi = [[\phi_i(k)]]_{i \in S, k=1,\ldots,K}$. Let $\phi(k) \stackrel{\triangle}{=} (\phi_i(k), i \in S)^T$ denote the kth column of Φ, $k \in \{1, \ldots, K\}$ having dimension N. From the foregoing, the (j,k)th element of Φ corresponds to $\phi_j(k)$. We now make the following assumptions.

Assumption 3. *The K columns of the matrix Φ, that is, $\phi(1), \ldots, \phi(K)$, are linearly independent. Further, $K \leq N$.*

Assumption 4. *$\Phi r \neq ce$, where $e = (1, 1, \ldots, 1)^T$ is a vector of all ones of dimension N and $c \in \mathcal{R} \setminus \{0\}$.*

From Assumption 1, $\{X_n\}$ is positive recurrent since the state space of this Markov chain is finite. Assumption 2 is a technical requirement used for the analysis of most policy gradient algorithms (see Bhatnagar et al. [2009]). Also, from Assumption 3, Φ has full column rank and is a standard requirement as well. In most real-life applications, the value of K is typically much less than N. Assumption 4 is also a technical requirement needed for the analysis of the average cost algorithm (see, for instance, Tsitsiklis and Van Roy [1999] and Bhatnagar et al. [2009], where a similar assumption has been used). Assumption 4 is, however, not required in the discounted cost setting.

Let $d^\theta(i)$ be the stationary probability of $\{X_n\}$ (under SRP θ) being in state $i \in S$, and D^θ be the diagonal matrix with entries $d^\theta(i)$, along the diagonal. Let the weighted Euclidean norm $\|\cdot\|_{D^\theta}$ be defined according to $\| z \|_{D^\theta} = \sqrt{z^T D^\theta z}$, where $z \in \mathcal{R}^N$. The convergence of algorithms such as TD has been shown (see Tsitsikilis and Van Roy [1997, 1999]), in the norm $\|\cdot\|_{D^\theta}$.

We use here the mean square Bellman error (MSBE) objective to measure the approximation error resulting from function approximation. The average cost MSBE objective is defined as

$$G_\theta(\Phi, r) = \| \Phi r - (k^\theta - \rho(\theta)e + P^\theta \Phi r) \|^2_{D^\theta}, \tag{12}$$

with $r \in \mathcal{R}^K$, and the aim is to find a parameter $r^*_{\theta,\Phi} \in \mathcal{R}^K$ that minimizes $G_\theta(\Phi, r)$ over all r. In a similar manner, the discounted cost MSBE objective is defined as

$$\hat{G}_\theta(\Phi, \hat{r}) = \| \Phi \hat{r} - (k^\theta + \gamma P^\theta \Phi \hat{r}) \|^2_{D^\theta}, \tag{13}$$

with the goal of finding a parameter $\hat{r}^*_{\theta,\Phi} \in \mathcal{R}^K$ that minimizes $\hat{G}_\theta(\Phi, \hat{r})$ over all $\hat{r} \in \mathcal{R}^K$. Given Φ and θ, our algorithms incorporate the residual gradient scheme from Baird [1995] to minimize $G_\theta(\Phi, r)$ over r.

In the next section, we obtain the gradient on the Grassmannian of features for both the average cost and the discounted cost criteria. For ease of notation, when not needed, we drop the dependence on θ of the two objective functions $G_\theta(\Phi, r)$ and $\hat{G}_\theta(\Phi, r)$, respectively, as well as the single-stage cost k^θ, the transition probability P^θ, the stationary distribution matrix D^θ, and the average cost $\rho(\theta)$, and will simply denote these quantities as $G(\Phi, r)$, k, P, D, and ρ, respectively, for the underlying SRP θ. However, we will incorporate the aforementioned dependence when the same is needed.

3. GRADIENT ON THE GRASSMANNIAN OF FEATURES

3.1. Average Cost Criterion

As mentioned in the last section, the critic adapts the features so as to estimate the differential value function by minimizing the MSBE. In this section, we assume complete knowledge of transition probabilities p, k in deriving the gradient on the Grassmannian of features. In the next section, we provide stochastic approximation algorithms to compute the quantities involved in the gradient from online samples. Let

$$G(\Phi, r) = \| \Phi r - (k - \rho e + P\Phi r) \|_D^2 = \| (I - P)\Phi r - (k - \rho e) \|_D^2. \tag{14}$$

We perform the minimization of $G(\Phi, r)$, first over r for a given Φ, and then subsequently over Φ. Let

$$F(\Phi) = \min_r G(\Phi, r).$$

This function can be rewritten as

$$F(\Phi) = G(\Phi, r^*(\Phi)), \tag{15}$$

where $r^*(\Phi)$ is the minimizer over r of $G(\Phi, r)$ for a given Φ. We make the following assumption:

ASSUMPTION 5. *The feature matrices Φ are orthonormal, that is, $\Phi^T \Phi = I$ (the identity matrix).*

Our goal in Equation (15) is to minimize the function $F(\Phi)$ over the space of orthonormal matrices. It can be easily verified that F satisfies the homogeneity condition $F(\Phi) = F(\Phi Q)$, where Q is any $k \times k$ orthogonal matrix; that is, the objective function depends only on the subspace spanned by columns of Φ and is invariant to the choice of basis.

The Grassmann manifold (Grassmannian) ($\mathcal{M}$) is the set of all K-dimensional subspaces of $\mathcal{R}^N$. Equivalently, each point in the Grassmann manifold corresponds to a K-dimensional subspace of $\mathcal{R}^N$; that is, each point corresponds to subspace $S \stackrel{\triangle}{=} \{\Phi r \mid r \in \mathcal{R}^K\} \subset \mathcal{R}^N$ for which the (feature) matrices Φ satisfy Assumption 5. As the objective function F satisfies the homogeneity condition, we need to minimize $F(\Phi)$ over the Grassmannian $\mathcal{M}$. If we relax the homogeneity assumption, then one needs to optimize over the Stiefel manifold. The Stiefel manifold consists of all $N \times K$ orthogonal matrices. The Grassmannian can be obtained from the Stiefel manifold by defining the equivalence relation on the space of all $N \times K$ matrices where two matrices Φ_1 and Φ_2 are said to be equivalent, if the columns of Φ_1 and Φ_2 span the same subspace, that is, $\exists\ Q$ satisfying $Q^T Q = I$ and with $\Phi_1 = \Phi_2 Q$.

In order to minimize F, one can perform gradient descent on the Grassmannian $\mathcal{M}$. The gradient of the function $F(\Phi)$ in the Grassmannian $\mathcal{M}$ can be computed as the following:

$$\nabla F(\Phi) = \left(I - \Phi\Phi^T\right)\frac{dF}{d\Phi} \tag{16}$$

(see Equation (2.70), pp. 321 of Edelman et al. [1998]), with Φ taking values in the set of orthornormal $N \times K$ matrices. The Euclidean gradient $\frac{dF}{d\Phi}$ is obtained by taking partial derivatives of the function F with respect to entries in Φ. This gradient need not lie in the tangent space of $\mathcal{M}$. In order to obtain the gradient along the tangent space (manifold gradient), we need to premultiply the Euclidean gradient by $(I - \Phi\Phi^T)$ (see Edelman et al. [1998] for the derivation of the manifold gradient).

The Euclidean gradient $\frac{dF}{d\Phi}$ can be obtained using the envelope theorem, that is,

$$\frac{dF(\Phi)}{d\Phi} = \frac{\partial(G(\Phi, r(\Phi)))}{\partial \Phi}\bigg|_{r=r^*(\Phi)}.$$

In Bhatnagar et al. [2013a], the gradient of a similar function $G(\Phi, r)$ is derived for the discounted cost MDP. The difference between our optimization in the average cost case and the discounted case is the presence of an extra average cost term instead of the discount factor $\gamma \in (0, 1)$. The presence of γ in the minimization problem of Bhatnagar et al. [2013a] helps in the invertibility of matrices for computing $r^*(\Phi)$. But this is not so in the average cost setting. Nevertheless, under Assumption 4, the invertibility issues get addressed (see Lemma 5.5 in Section 5). Set $\Delta = (I - P)$ for notational simplicity. Then, Equation (12) can be rewritten as

$$G(\Phi, r) = (\Delta\Phi r - (k - \rho e))^T D(\Delta\Phi r - (k - \rho e)).$$

By differentiating w.r.t. r the previous equation and equating to zero, one obtains

$$r^*(\Phi) = \arg\min_r G(\Phi, r) = (\Phi^T \Delta^T D \Delta\Phi)^{-1} \Phi^T \Delta^T D(k - \rho e). \tag{17}$$

The inverse on the RHS of Equation (17) can be seen to exist because of Assumptions 1 and 4. It is, in fact, shown in Lemma 5.5 that $\Delta\Phi$ is a full rank matrix. Since we use multitimescale stochastic approximation, for any given update of Φ, $r^*(\Phi)$ will be estimated in our scheme along the faster timescale.

The computation of $\frac{dF(\Phi)}{d\Phi}$ can be done along similar lines as Bhatnagar et al. [2013a] (see Absil et al. [2009]) or using matrix calculus. We obtain (see Equation (18))

$$\frac{dF}{d\Phi} = 2\Delta^T D(\Delta\Phi r^*(\Phi) - (k - \rho e))(r^*(\Phi))^T. \tag{18}$$

Under Assumption 5, from Equation (16) (cf. Equation (2.70) of Edelman et al. [1998]), one can write using Equation (18) that

$$\begin{aligned} \nabla F &= (I - \Phi\Phi^T)\frac{dF}{d\Phi} \\ &= -2(I - \Phi\Phi^T) y^*(\Phi)(r^*(\Phi))^T, \end{aligned} \tag{19}$$

where $y^*(\Phi) = (I - P)^T D(k - \rho e - (I - P)\Phi r^*(\Phi))$ is an N-dimensional column vector. Thus, to compute the gradient in Equation (19), we need to compute both $y^*(\Phi)$ and $r^*(\Phi)$. So far in the derivation of ∇F, we have assumed knowledge of the underlying transition probabilities and the cost function. In the RL setting, we do not have access to these quantities; however, we can simulate the system. Thus, we need to estimate the gradient in Equation (19) from simulation samples. Our algorithm runs a separate recursion to track y along the same faster timescale recursion of r. Φ is then updated along the direction given by Equation (19) on a timescale slower compared to the y and r updates (see Section 4.1 for the update rules of r, y, and Φ).

Remark 3.1. Edelman et al. [1998] provides a range of more sophisticated methods, such as Newton and conjugate gradient methods on manifolds to minimize a function on the Grassmannian. These methods have additional computational overhead compared to the simple gradient method that we consider in this article. In the next section, to compute ∇F, we present stochastic approximation algorithms that use simulation samples to estimate y and r^* in Equation (19). So, if one wants to utilize Newton or conjugate gradient methods for faster convergence, the stochastic approximation recursion would have to be designed to estimate the Hessian where the estimation procedure

would typically require additional simulation samples. However, it is an interesting future direction to compare the Newton or conjugate-gradient-based schemes with the simple gradient scheme in terms of the convergence rate and number of simulation samples required.

3.2. Weighted Discounted Cost Criterion

In a similar manner as the average cost case, one can define for the weighted discounted cost criterion the function to be minimized for estimating the value function. Let

$$\begin{aligned}\hat{G}(\Phi, \hat{r}) &= \| \Phi\hat{r} - (k + \gamma P\Phi\hat{r}) \|_D^2 \\ &= \| (I - \gamma P)\Phi\hat{r} - k \|_D^2, \end{aligned} \tag{20}$$

and let

$$\hat{F}(\Phi) = \min_{\hat{r}} \hat{G}(\Phi, \hat{r}).$$

In this case, one obtains the gradient of $\hat{F}(\Phi)$ as

$$\nabla\hat{F}(\Phi) = -2(I - \Phi\Phi^T)\hat{y}(\Phi)(\hat{r}^*(\Phi))^T, \tag{21}$$

where $\hat{y}(\Phi) = \hat{\Delta}^T D(k - \hat{\Delta}\Phi\hat{r}^*(\Phi))$ is an N-dimensional column vector with $\hat{\Delta} = (I - \gamma P)$.

4. THE ACTOR-CRITIC CONTROL ALGORITHMS

In this section, we describe the stochastic update rules corresponding to our control algorithm for the average cost objective in Section 4.1 and the weighted discounted cost objective in Section 4.2, respectively.

4.1. Average Cost Criterion

Our online control algorithm with feature adaptation is given as follows:

In Algorithm 1, we employ multitimescale stochastic approximation to compute the local optimal policy. Our algorithm updates the policy parameter along the negative gradient direction. In order to obtain an unbiased estimate of the gradient, the critic adaptively tunes the value function features. In step A,[1] we estimate quantities along the faster timescale $\{a(n)\}$. Specifically, in step A1 (Equation (22)), we estimate the average cost for a given policy. In step A2 (Equation (23)), we use the residual gradient scheme to estimate the weight vector r for the current policy parameter θ_n and the critic's feature Φ_n. Note that in Equation (23), $\tilde{X}_{n+1}$ is a sample generated with the distribution $p(\cdot \mid X_n, \theta_n)$ that is conditionally independent of X_{n+1} given X_n. This poses a problem in online situations as one would require, as with the original residual gradient algorithm (see Baird [1995]), two independent samples for the "next" state given the current state. One possible remedy is to use TD(0) with function approximation (see Tsitsiklis and Van Roy [1999]) instead of the residual gradient scheme as it requires only one sample. Even though the TD(0) scheme does not minimize the MSBE error objective, it shows comparable performance as the residual gradient scheme in numerical experiments (see Section 6). In step A3, we estimate $y^*(\Phi)$ to compute the gradient on the Grassmannian.

In step B, we update the feature matrix along the medium timescale $\{b(n)\}$ in the direction of the negative gradient of $F(\Phi)$ on the Grassmannian. In Equation (25), $\Gamma^1(\cdot)$ is the operator that performs the Gram-Schmidt orthonormalization step. This

[1]This step is similar to Bhatnagar et al. [2013a] except for the presence now of ρ_n in the definition of $\zeta(n)$ and $\tilde{\zeta}(n)$ and the absence of a discount factor multiplying the terms $\phi^T_{X_{n+1}}(n)r_n$ and $\phi^T_{\tilde{X}_{n+1}}(n)r_n$, respectively.

ALGORITHM 1: Actor-critic control algorithm for the long-run average cost criterion

Input: Policy features σ
Output: Policy parameter θ, value function parameter r, and feature Φ

(1) **Initialization**:
—Actor's parameter $\theta = \theta_0$,
—Critic's parameters $r = r_0$, $y = y_0$, $\Phi = \Phi_0$,
—Initial state $X_0 = x_0$

(2) **Execution**:
for $n \leftarrow 0, 1, 2, \ldots,$ **do**
—Choose $Z_n \sim \pi_\theta(X_n, \cdot)$
—Observe the next state $X_{n+1} \sim p(X_n, \cdot, Z_n)$
—Observe the cost $k(X_n, Z_n)$

(A) [**First (Fastest) Timescale Update**]:
(A1) *Average cost ρ update*:

$$\rho_{n+1} = \rho_n + a(n)(k(X_n, Z_n) - \rho_n). \tag{22}$$

(A2) *Residual gradient r update:*

$$\begin{aligned} r_{n+1} \;=\; & r_n + a(n)\big(k(X_n, Z_n) - \rho_n + \phi^T_{X_{n+1}}(n) r_n - \phi^T_{X_n}(n) r_n\big) \\ & \times \big(\phi_{X_n}(n) - \phi_{\tilde{X}_{n+1}}(n)\big). \end{aligned} \tag{23}$$

(A3) *Intermediate y update:*

$$\zeta(n) \stackrel{\triangle}{=} k(X_n, Z_n) - \rho_n + \phi^T_{X_{n+1}}(n) r_n - \phi^T_{X_n}(n) r_n,$$

$$\tilde{\zeta}(n) \stackrel{\triangle}{=} k(X_n, Z_n) - \rho_n + \phi^T_{\tilde{X}_{n+1}}(n) r_n - \phi^T_{X_n}(n) r_n$$

,

$$y_{n+1}(i) = y_n(i) + a(n) 1_{\{X_{n+1}=i\}} \big(\zeta(n+1) - \tilde{\zeta}(n) - y_n(i)\big). \tag{24}$$

(B) [**Second (Medium) Timescale Grassmannian Feature Φ Update**]:

$$\Phi(n+1) = \Gamma^1 \left(\Phi(n) + b(n) 2\big(I - \Phi(n)\Phi(n)^T\big) y_n (r_n)^T\right). \tag{25}$$

(C) [**Third (Slowest) Timescale Policy Parameter θ Update**]:

$$\theta_{n+1} = \Gamma^2 \left(\theta_n - c(n)\delta_n \psi(X_n, Z_n)\right). \tag{26}$$

end
return θ, Φ, r

recursion will converge to Φ^* such that $\Phi^* r^*$ will give the state value function V^{θ_n} of the policy with parameter θ_n.

In step C, we update the policy parameter along the slowest timescale $\{c(n)\}$. In Equation (26), $\Gamma^2 : \mathcal{R}^k \to \mathcal{R}^k$ is a projection operator that projects any $\theta \in \mathcal{R}^L$ to a compact set $\breve{G} = \{\theta \in \mathcal{R}^L | q_i(\theta) \leq 0, i = 1, 2, \ldots, s\}$, with $q_i(\theta), i = 1, 2, \ldots, s$ being continuously differentiable functions on $\mathcal{R}^L$ that represent the constraints specifying the compact region. The TD error δ_n is found from Φ_n and r_n as $\delta_n = k(X_n, Z_n) - \rho_n + V_n(X_{n+1}) - V_n(X_n)$, where V_n is the N-dimensional vector $V_n = \Phi_n r_n$.

In our algorithm, we assume that the step-size schedules $\{a(n)\}$, $\{b(n)\}$, and $\{c(n)\}$ satisfy the following requirements:

ASSUMPTION 6. *The step-sizes $a(n), b(n), c(n) > 0, \forall n$. Further,*

$$\sum_n a(n) = \sum_n b(n) = \sum_n c(n) \;=\; \infty, \tag{27}$$

$$\sum_n (a^2(n) + b^2(n) + c^2(n)) \; < \; \infty, \tag{28}$$

$$\lim_{n\to\infty} \frac{b(n)}{a(n)} = \lim_{n\to\infty} \frac{c(n)}{b(n)} \; = \; 0. \tag{29}$$

Note that Equations (27) and (28) are standard requirements on step-size sequences. As a consequence of Equation (29), the timescale corresponding to $a(n), n \geq 0$ is the fastest, while the one corresponding to $c(n), n \geq 0$ is the slowest. Also, the timescale corresponding to $b(n), n \geq 0$ falls in between the two aforementioned timescales.

Remark 4.1. A limitation of the update rule in step A2 of the algorithm is that it requires two simulation samples, as a result of which our algorithm may not work in situations involving real data from a physical system. The problem arises because of the use of the Bellman error objective that gets minimized by the residual gradient scheme [Baird 1995] that we have adopted. Nonetheless, in such scenarios, a possible alternative is to replace the residual gradient update by the TD(0) update. TD(0) does not minimize the MSBE, and to extend the theoretical analysis to include TD(0) in place of the residual gradient scheme is not straightforward. In our experiments, however, we found that TD(0) shows comparable performance to the residual gradient scheme.

Remark 4.2. Step B of the algorithm is a matrix update rule. This takes superlinear time in the number of states N of the MDP. Thus, our algorithm in its current form cannot be applied directly to high-dimensional state spaces as storing the entire feature matrix becomes computationally challenging. Thus, in the future, we would like to develop more efficient algorithms that work with a subset of the Grassmannian by suitably parameterizing the aforementioned subset. Another promising direction is to put a sparsity penalty in the Grassmannian gradient descent as, for example, in compressive sensing and exploit the sparse matrix computations. While this is on our agenda, it is not a minor tweak. Yet another possibility is to fold in aspects of "random basis search" as in Bhatnagar et al. [2012].

Remark 4.3. It may be noted that multiple timescales can be induced by performing the slower iterates along a subsequence. This has a twofold advantage: (1) it allows better choice of the step-size for the slower timescale and (2) it amortizes the computational effort. We follow this strategy here. We have observed empirically that a combination of smaller step-sizes and subsequential updates perform better than either scheme in isolation (see Bhatnagar et al. [2013a]). Thus, as we do in our experiments, we perform the updates in the steps (A1) through (A3) (Equations (22) through (24)) 100 times before step (B) is performed. Note that we fix other parameters to their current update values. Then, steps (B) and (C) (Equation (25) through (26)) are carried out once. This ensures faster convergence of the algorithm. Also, the computational overhead is reduced as the number of Φ iterations that need to be performed is reduced. Also, in step (B) of the algorithm, we do not apply the Gram-Schmidt orthonormalization procedure on every step but instead apply it only after several iterations as it leads to computational savings. The computational loss in increased dimensionality caused by the medium timescale iterates is mitigated by updating along a subsample, that is, making those computations only sparsely so that the overall computational budget is reduced.

4.2. Weighted Discounted Cost Criterion

This algorithm works with SPSA gradient estimates for the policy gradient. SPSA estimates typically require two independent parallel simulations of the system with

different perturbed parameters. We propose a workaround by using a common (single) simulation as follows: we divide the total number of iterations into equal subintervals of length M as $\{0, 1, \ldots, M-1\}, \{M, \ldots, 2M-1\}$, and so forth. The policy parameter θ gets updated according to Equation (33) only at the end of every $2M$ epochs. The update rules for r, y, and Φ (i.e., Equations (30) through (32)) are followed with the step-sizes $a(n), b(n)$ fixed during the aforementioned intervals of $2M$ epochs. The step-sizes $\{a(n), b(n)\}$ as well as the perturbation parameters $\{\Delta_n\}$ are fixed during the (previous) $2M$ epochs and changed only at the end of the epochs. Here, $\Delta_n = (\Delta_n(1), \Delta_n(2), \ldots, \Delta_n(L))^T$, with $\Delta_n(k), k \in \{1, 2, \ldots, L\}, n \geq 0$ being independent and identically distributed (i.i.d), Bernoulli random variables taking values $\{\pm 1\}$ with probability $\frac{1}{2}$, and the perturbation parameter $\epsilon > 0$ chosen to be a small enough constant. One may replace ϵ by a slowly diminishing sequence $\epsilon_n \to 0$ (see Spall [1992]). Now, during the odd M-step subintervals, the policy parameter θ is set to $\theta_n + \epsilon\Delta_n$, and during the even such subintervals, the same is set to $\theta_n - \epsilon\Delta_n$. At the end of the odd and even subintervals, the objectives $\omega(\theta_n + \epsilon\Delta_n)$ and $\omega(\theta_n - \epsilon\Delta_n)$ are respectively computed as $\Phi_n r_n$. At the end of every $2M$ epochs, θ_n is updated according to Equation (33).

Remark 4.4. With appropriate modifications, the observations in Remarks 4.1 through 4.3 continue to hold in the discounted cost case as well. In particular, for our experiments, we perform the updates in steps (A1) through (A2) (Equations (30) and (31)) 100 times before steps (B) and (C) (Equations (32) and (33)) are carried out once. Further, we apply the Gram-Schmidt procedure in step (B) once after several iterations in order to reduce the computational effort. As with the average cost setting, the algorithm in the discounted cost case continues to show good empirical performance under these modifications.

The online control algorithm with feature adaptation is given as follows:

5. CONVERGENCE ANALYSIS

We begin in Section 5.1 with a characterization of the zeros of the gradient of the objective function $F(\Phi)$. Specifically, we show that all the minima of F are global minima and likewise the maxima are all global maxima. This will be followed with a convergence analysis for the average cost case in Section 5.2. Due to lack of space, we only provide the outline of the convergence analysis and state all the necessary lemmas, propositions, and theorems without proofs. Readers are referred to the Online Appendix for the proofs. The convergence analysis for the discounted cost can be carried along similar lines as the average cost. As mentioned previously, this analysis is new and has not been carried out for the policy evaluation scheme in Bhatnagar et al. [2013a].

5.1. Characterization of the Minima

We characterize next the minima of the objective function $F(\Phi)$ for the average cost case. We assume here that the minimization problem is an unconstrained minimization without the constraint on Φ. An unconstrained minimum Φ^* can be converted to a constrained minimum $\bar{\Phi}$ in our setting by orthonormalizing Φ^* to obtain $\bar{\Phi}$ satisfying $\bar{\Phi}^T\bar{\Phi} = I$. An alternative is to consider including Lagrangian for performing constrained minimization. However, in our case the constraint will be met nonetheless.

THEOREM 5.1. *Any local minimum of $F(\Phi)$ is a global minimum. Further, $\Phi^* r^*(\Phi^*)$ will correspond to the differential value function of the policy, where Φ^* is the minimizer of $F(\Phi)$.*

ALGORITHM 2: Actor-critic control algorithm for the weighted discounted cost criterion

Input: Policy features σ
Output: Policy parameter θ, value function parameter r and feature Φ

(1) **Initialization**:
—Actor's parameter $\theta = \theta_0$,
—Critic's parameters $r = r_0$, $y = y_0$, $\Phi = \Phi_0$,
—Initial state $X_0 = x_0$

(2) **Execution**:
for $n \leftarrow 0, 1, 2, \ldots,$ **do**
—Choose $Z_n \sim \pi_\theta(X_n, \cdot)$
—Observe the next state $X_{n+1} \sim p(X_n, \cdot, Z_n)$
—Observe the cost $k(X_n, Z_n)$

(A) [**First (Fastest) Timescale Update**]:

(A1) *Residual gradient r update:*

$$\begin{aligned}\hat{r}_{n+1} \;=\; & \hat{r}_n + a\left(\left\lfloor \frac{n}{2M} \right\rfloor\right)\left(k(X_n, Z_n) + \gamma\phi^T_{X_{n+1}}(n)\hat{r}_n - \phi^T_{X_n}(n)\hat{r}_n\right) \\ & \times \left(\phi_{X_n}(n) - \gamma\phi_{\tilde{X}_{n+1}}(n)\right), \end{aligned} \tag{30}$$

(A2) *Intermediate y update:*

$$\eta(n) \stackrel{\triangle}{=} k(X_n, Z_n) + \gamma\phi^T_{X_{n+1}}(n)\hat{r}_n - \phi^T_{X_n}(n)\hat{r}_n,$$

$$\tilde{\eta}(n) \stackrel{\triangle}{=} k(X_n, Z_n) + \gamma\phi^T_{\tilde{X}_{n+1}}(n)\hat{r}_n - \phi^T_{X_n}(n)\hat{r}_n.$$

Now for $i = 1, \ldots, N, n \geq 0$,

$$\hat{y}_{n+1}(i) = \hat{y}_n(i) + a\left(\left\lfloor \frac{n}{2M} \right\rfloor\right)\left(I^i_{n+1}(\eta(n+1) - \gamma\tilde{\eta}(n) - \hat{y}_n(i))\right), \tag{31}$$

(B) [**Second (Medium) Timescale Grassmannian feature** Φ **update**]:

$$\Phi(n+1) = \Gamma^1(\Phi(n) + b(\lfloor \frac{n}{2M} \rfloor)2\left(I - \Phi(n)\Phi(n)^T\right)\hat{y}_n(\hat{r}_n)^T), \tag{32}$$

(C) [**Third (Slowest) Timescale Policy parameter** θ **update**]: **for** $k \leftarrow 1$ **to** L **do**

$$\theta_{n+1}(k) = \Gamma^2\left(\theta_n(k) - c(n) \times \left[\frac{\omega(\theta_n + \epsilon\Delta_n) - \omega(\theta_n - \epsilon\Delta_n)}{2\epsilon\Delta_n(k)}\right]\right). \tag{33}$$

end
end
return θ, Φ, r

PROOF. By considering Equation (19) and setting the derivative $\frac{dF}{d\Phi}$ equal to zero, we get

$$\Delta^T D(\Delta\Phi r^*(\Phi) - (k - \rho e))(r^*(\Phi))^T = 0, \tag{34}$$

with $r^*(\Phi)$ as in Equation (17) and $\Delta = (I - P)$. For ease of notation, we will simply denote $r^*(\Phi)$ as r^*. It is easy to see that $\Delta^T D\rho e = 0$ and thus Equation (34) is an outerproduct ab^T of $a \stackrel{\triangle}{=} \Delta^T D(\Delta\Phi r^* - k) \in \mathcal{R}^N$ and $b \stackrel{\triangle}{=} r^* \in \mathcal{R}^K$. Now for Equation (34) to be zero, we need either of a and b to be the zero vector; otherwise, the outerproduct ab^T will not be zero.

Let us first assume $a = 0$, which implies $\Delta^T D(\Delta\Phi r^* - k) = 0$. For the aforementioned to be true, since $\Delta^T D$ is not full rank, the vector $(\Delta\Phi r^* - k)$ belongs to the null space of the matrix $\Delta^T D$ denoted by $\mathcal{N}(\Delta^T D)$. Let $z \in \mathcal{N}(\Delta^T D)$, and then

$$\begin{aligned}\Delta^T Dz \;&=\; 0, \text{ or} \\ P^T Dz \;&=\; Dz. \end{aligned} \tag{35}$$

From Equation (35), it follows that $z = ce$ for some $c \in \mathcal{R}$, since Dz corresponds to c times the stationary distribution of P. From this we can conclude that

$$\Delta\Phi r^* - k = ce. \tag{36}$$

Thus,

$$\Phi r^* = k + P\Phi r^* - ce. \tag{37}$$

Thus, if Φ satisfies Equation (36), then from Equation (37), it follows that Φr^* corresponds to the differential value function. Note that if Φ is such that $k - \rho e$ lies in the subspace spanned by the column vectors of $\Delta\Phi$ (denoted by S), then $\frac{dF}{d\Phi} = 0$ and $F(\Phi) = 0$, thus implying such a Φ is a local minimum. In this case, the Φ will correspond to a global minimum since $F(\Phi) = 0$.

Let us now consider the next case $b = 0$, that is, $r^* = 0$. From the expression for r^* in Equation (17), we have

$$\begin{aligned}((k - \rho e)^T D\Delta\Phi)(\Phi^T \Delta^T D\Delta\Phi)^{-1} &= 0, \text{ or} \\ ((k - \rho e)^T D\Delta\Phi) &= 0.\end{aligned} \tag{38}$$

From Equation (38), it follows that $k - \rho e$ is D-orthogonal to the column vectors of $\Delta\Phi$. For any Φ, $F(\Phi)$ is the projection error resulting from the projection of the cost vector $k - \rho e$ into the subspace S. This will be maximum if $k - \rho e$ lies in the orthogonal complement of S. In our present case, Equation (38) holds. Thus, Φ satisfying Equation (38) will correspond to the global maximum and the corresponding objective function value is $F(\Phi) = \| k - \rho e \|_D^2$.

If $k - \rho e$ lies neither in the span of $\Delta\Phi$ nor in its orthogonal complement, then at least one entry in the vector a and at least one entry in the vector b would each be nonzero. Hence, their outer product would be nonzero, indicating that the derivative is nonzero. Thus, we have shown that only if Φ is such that $k - \rho e$ either lies in the range space of $\Delta\Phi$ or lies in its orthogonal complement (with respect to the D-norm), the derivative of $F(\Phi)$ will be zero. In summary, the Φ's corresponding to $\frac{dF}{d\Phi} = 0$ will either be global minima or global maxima of the function $F(\Phi)$. In particular, Φ corresponding to global minima gives an exact representation of the value function, while the Φ corresponding to global maxima results in unstable equilibria. □

5.2. Convergence Analysis for the Average Cost Objective

We begin with the analysis of the faster timescale recursion (Equation (22)) (step (A1) of the algorithm). From Assumption 6, $c(n) = o(a(n))$. Hence, we may let θ_n be a constant θ while analyzing Equation (22) (see Borkar [1997] or Chapter 6 of Borkar [2008]). We show in Proposition 2 that the iterates in Equation (22) remain uniformly bounded almost surely. Thus, analyzing the previous recursion is equivalent to analyzing a limiting ODE that the recursion asymptotically tracks. The said limiting ODE here corresponds to

$$\dot{\rho}(t) = -\rho(t) + \rho_\theta, \tag{39}$$

where $\rho_\theta = E_{X_n \sim d^\theta, Z_n \sim \pi_\theta(X_n,.)}[k(X_n, Z_n)] = \sum_{i \in S} d^\theta(i) \sum_{a \in A} \pi_\theta(i, a) k(i, a)$ is the average cost incurred by following the policy π_θ. Note that for a fixed θ, ρ_θ is a constant term in the RHS of the ODE (Equation (39)). Let $h^1(\rho)$ denote the RHS of Equation (39) that is clearly Lipschitz continuous in ρ.

LEMMA 5.2. *The ODE (Equation (39)) has ρ_θ as its unique globally asymptotically stable equilibrium.*

PROPOSITION 5.3. *With $\theta_n \equiv \theta$, $\forall n$, ρ_n, and $n \geq 0$ governed according to Equation (39) are uniformly bounded almost surely. Further, $\rho_n \to \rho_\theta$ as $n \to \infty$ almost surely.*

Next we analyze the r recursion (step (A2) of the algorithm). From Assumption 6, $c(n) = o(a(n))$ and $b(n) = o(a(n))$. Hence, we may let θ_n to be a constant θ and $\Phi(n)$ be a constant Φ while analyzing Equation (40) (see Borkar [1997] or Chapter 6 of Borkar [2008]). In view of Proposition 5.3, the update (Equation (40)) can be rewritten as

$$r_{n+1} = r_n + a(n)\big(k(X_n, Z_n) - \rho_\theta + \phi_{X_{n+1}}^T r_n - \phi_{X_n}^T r_n\big)\big(\phi_{X_n} - \phi_{\tilde{X}_{n+1}}\big). \tag{40}$$

Let $\mathcal{F}_n = \sigma(X_m, Z_m, r_m, m \le n)$, $n \ge 0$ denote a sequence of sigma fields generated by the mentioned quantities and let

$$\begin{aligned} M_{n+1}^1 \;=\; & \big(k(X_n, Z_n) - \rho_\theta + \phi_{X_{n+1}}^T r_n - \phi_{X_n}^T r_n\big)\big(\phi_{X_n} - \phi_{\tilde{X}_{n+1}}\big) \\ & - E\big[\big(k(X_n, Z_n) - \rho_\theta + \phi_{X_{n+1}}^T r_n - \phi_{X_n}^T r_n\big)\big(\phi_{X_n} - \phi_{\tilde{X}_{n+1}}\big) \mid \mathcal{F}_n\big]. \end{aligned}$$

Lemma 5.4. *$(M_n^1, \mathcal{F}_n)$, $n \ge 0$ forms a martingale difference sequence with*

$$E[\| M_{n+1}^1 \|^2 | \mathcal{F}_n] \le \hat{K}(1 + \| r_n \|^2),$$

for some constant $\hat{K} > 0$.

A simple calculation shows that

$$\begin{aligned} & E\big[\big(k(X_n, Z_n) - \rho_\theta + \phi_{X_{n+1}}^T r_n - \phi_{X_n}^T r_n\big)\big(\phi_{X_n} - \phi_{\tilde{X}_{n+1}}\big) \mid \mathcal{F}_n\big] \\ & \quad = \left(k(X_n, Z_n) - \rho_\theta + \sum_j p_{X_n,j}(Z_n)\phi_j^T r_n - \phi_{X_n}^T r_n\right) \times \left(\phi_{X_n} - \sum_j p_{X_n,j}(Z_n)\phi_j\right). \end{aligned}$$

Now Equation (40) can be rewritten as

$$\begin{aligned} r_{n+1} \;=\; & r_n + a(n)\left(\left(k(X_n, Z_n) - \rho_\theta + \sum_j p_{X_n,j}(Z_n)\phi_j^T r_n - \phi_{X_n}^T r_n\right)\right. \\ & \left. \times \left(\phi_{X_n} - \sum_j p_{X_n,j}(Z_n)\phi_j\right) + M_{n+1}^1\right). \end{aligned} \tag{41}$$

The ODE associated with Equation (41) is the following:

$$\begin{aligned} \dot{r}(t) \;=\; & \sum_{i \in S} d^\theta(i) \sum_{a \in A} \pi_\theta(i, a)\left[\left(k(i, a) - \rho_\theta + \sum_{j \in S} p_{i,j}(a)\phi_j^T r(t) - \phi_i^T r(t)\right)\right. \\ & \left. \times \left(\phi_i - \sum_{j \in S} p_{i,j}(a)\phi_j\right)\right] \end{aligned} \tag{42}$$

$$= \Phi^T (I - P^\theta)^T D(k^\theta - \rho e - (I - P)\Phi r(t)) \stackrel{\triangle}{=} h^2(\theta, \Phi, r(t)), \tag{43}$$

where k^θ is a vector of dimension N whose ith component is $k^\theta(i) = \sum_{a \in A} \pi_\theta(i, a)k(i, a)$ and P^θ is a matrix of dimension $N \times N$ whose ij^{th} component is $P^\theta(i, j) = \sum_{a \in A} p_{i,j}(a)\pi_\theta(i, a)$.

In what follows, we shall show the convergence of Equation (41) using Theorem 7–Corollary 8 (p. 74) and Theorem 9 (p. 75) of Borkar [2008]. It is easy to see that $\frac{h^2(\theta, \Phi, rc)}{c}$

converges as $c \to \infty$ to $h^2_\infty(\theta, \Phi, r)$ uniformly on compacts, where

$$h^2_\infty(\theta, \Phi, r) \stackrel{\triangle}{=} \lim_{c\uparrow\infty} \frac{h^2(\theta, \Phi, cr)}{c} = -\Phi^T (I - P^\theta)^T D(I - P^\theta)\Phi r. \tag{44}$$

LEMMA 5.5. *For $\Delta = (I - P^\theta)$, under Assumption 4, the matrix $\Delta\Phi$ is full rank.*

COROLLARY 5.6. *Under Assumption 4, the matrix $E \stackrel{\triangle}{=} (\Delta\Phi)^T D(\Delta\Phi)$ is positive definite and symmetric.*

LEMMA 5.7. *The ODE (43) has $r^*_{\theta,\Phi} \stackrel{\triangle}{=} ((\Delta\Phi)^T D(\Delta\Phi))^{-1}(\Delta\Phi)^T Dk^\theta$ as its unique globally asymptotically stable equilibrium.*

PROPOSITION 5.8. *With $\theta_n \equiv \theta$ and $\Phi(n) \equiv \Phi$, $\forall n$, $r(n)$, and $n \geq 0$ governed according to Equation (40) are uniformly bounded almost surely. Further, $r(n) \to r^*_{\theta,\Phi}$ as $n \to \infty$ almost surely.*

We now analyze the y recursion in step (A3) of the algorithm. Since $c(n) = o(a(n))$, one may again let $\theta_n \equiv \theta$ and $\Phi(n) \equiv \Phi$ (constant) while analyzing the update (A3). Now let

$$\zeta^*(n) \stackrel{\triangle}{=} k(X_n, Z_n) - \rho_\theta + \phi^T_{X_{n+1}} r^*_{\theta,\Phi} - \phi^T_{X_n} r^*_{\theta,\Phi},$$

$$\tilde{\zeta}^*(n) \stackrel{\triangle}{=} k(X_n, Z_n) - \rho_\theta + \phi^T_{\tilde{X}_{n+1}} r^*_{\theta,\Phi} - \phi^T_{X_n} r^*_{\theta,\Phi},$$

respectively. In view of Propositions 5.3 and 5.8, one may analyze the following recursion in place of Equation (24):

$$y_{n+1}(i) = y_n(i) + b(n) I^i_{n+1}\big(\zeta^*(n+1) - \tilde{\zeta}^*(n) - y_n(i)\big).$$

In a similar manner to the r recursion, one can obtain the ODE associated with the recursion in Equation (24) to be (see Online Appendix for more details)

$$\dot{y}(t) \;=\; (I - P^\theta)^T D z^*_{\theta,\Phi} - y(t), \tag{45}$$

where $z^*_{\theta,\Phi} = (k^\theta - (I - P^\theta)\Phi r^*_{\theta,\Phi})$.

Let $\mathcal{G}(n) = \sigma(X_m, Z_m, y_m, m \leq n)$, $n \geq 0$ denote an increasing sequence of sigma fields and let M^2_n, $n \geq 0$ be defined as follows:

$$M^2_{n+1} = \big(\zeta^*(n+1) - \tilde{\zeta}^*(n)\big) - E\big[\big(\zeta^*(n+1) - \tilde{\zeta}^*(n)\big) \mid \mathcal{G}(n)\big].$$

It is easy to see that $(M^2_n, \mathcal{G}(n))$, $n \geq 0$ forms a martingale difference sequence.

PROPOSITION 5.9. *Given $\theta_n \equiv \theta$ and $\Phi(n) \equiv \Phi$, $\forall n$, the updates y_n, and$n \geq 0$ governed by Equation (24) are uniformly bounded almost surely and converge to $y^*_{\theta,\Phi} \stackrel{\triangle}{=} (I - P^\theta)^T Dz^*_{\theta,\Phi}$ as $n \to \infty$.*

Consider now the medium timescale recursion in step (B) of the algorithm. From Assumption 6, $c(n) = o(b(n))$. Hence, we may let θ_n be a constant θ while analyzing the update in Equation (25). Note that for an $N \times K$–matrix $\Phi = (\phi^T_i, i = 1, \ldots, N)^T$, $\Gamma^1(\Phi)$ is the operator that performs the Gram-Schmidt orthonormalization step. As a consequence of the Γ^1-operator, the iterates in Equation (25) remain almost surely uniformly bounded. One can rewrite the recursion in Equation (25) in the following manner:

$$\Phi(n+1) = \Gamma^1\Big(\Phi(n) + c(n)2\big(I - \Phi(n)\Phi(n)^T\big)y^*_{\theta,\Phi(n)}(r^*_{\theta,\Phi(n)})^T + \epsilon(n)\Big), \tag{46}$$

where $\epsilon(n) = 2(I - \Phi(n)\Phi(n)^T)(y_n(r_n)^T - y^*_{\theta,\Phi(n)}(r^*_{\theta,\Phi(n)})^T)$. From Propositions 5.8 and 5.9, it follows that $\epsilon(n) \to 0$ almost surely as $n \to \infty$.

Consider now the following ODE corresponding to the recursion in Equation (46):

$$\dot{\Phi}(t) = \hat{\Gamma}^1(-\nabla F(\Phi(t))), \tag{47}$$

where for any continuous function $v : \mathcal{R}^{N\times K} \to \mathcal{R}^{N\times K}$,

$$\hat{\Gamma}^1(v(y)) = \lim_{0<\eta\to 0}\left(\frac{\Gamma^1(y+\eta v(y)) - y}{\eta}\right).$$

In case the aforementioned limit is not unique, we may let $\hat{\Gamma}^1(v(y))$ be the set of all possible limit points (see p. 191 of Kushner and Clark [1978]). Also, if y is an interior point, $\hat{\Gamma}^1(v(y)) = v(y)$. Let

$$\mathcal{K}^1 \stackrel{\triangle}{=} \{\Phi \in \mathcal{M} \mid \hat{\Gamma}^1(\nabla F(\Phi)) = 0\}$$

denote the set of all fixed points of Equation (47). Note that $\check{V}(\Phi) = F(\Phi)$ itself serves as an associated strict Lyapunov function for the ODE (Equation (47)).

Now, we recall a crucial result from Kushner and Clark (cf. Theorem 5.3.1, pp. 191–196 of Kushner and Clark [1978]). While the result stated in Kushner and Clark [1978] is more generally applicable, we present its adaptation here that is relevant to the setting that we consider (see also Appendix E of Bhatnagar et al. [2013b]). Consider the following $\hat{L}$-dimensional stochastic recursion:

$$\tau(n+1) = \Gamma\big(\tau(n) + \alpha(n)(h(\tau(n)) + \xi(n) + \beta(n))\big), \tag{48}$$

under the conditions (C1) through (C5) later. Here, $\Gamma : \mathcal{R}^{\hat{L}} \to \check{C} \subset \mathcal{R}^{\hat{L}}$ is a projection map and $\check{C}$ is a subset of $\mathcal{R}^{\hat{L}}$. Consider also the following ODE associated with Equation (48):

$$\dot{\tau}(t) = \hat{\Gamma}(h(\tau(t))), \tag{49}$$

where for any continuous function $w : \mathcal{R}^{\hat{L}} \to \mathcal{R}^{\hat{L}}$,

$$\hat{\Gamma}(w(\tau)) = \lim_{0<\eta\to 0}\left(\frac{\Gamma(\tau+\eta w(\tau)) - \tau}{\eta}\right).$$

Let

$$\mathcal{B} \stackrel{\triangle}{=} \{\tau \in \mathcal{R}^{\hat{L}} \mid \hat{\Gamma}(h(\tau)) = 0\}$$

denote the set of all fixed points of Equation (49). We now state the following conditions in relation to Equation (48):

(C1) The function $h : \mathcal{R}^{\hat{L}} \to \mathcal{R}^{\hat{L}}$ is continuous.

(C2) The step-sizes $\alpha(n), n \geq 0$ satisfy

$$\alpha(n) > 0 \ \forall n, \ \sum_n \alpha(n) = \infty, \ \alpha(n) \to 0 \text{ as } n \to \infty.$$

(C3) The sequence $\beta(n), n \geq 0$ is a bounded random sequence with $\beta(n) \to 0$ almost surely as $n \to \infty$.

(C4) $\forall \delta > 0$,

$$\lim_{n\to\infty} P\left(\sup_{m\geq n}\left|\sum_{i=n}^{m}\alpha(i)\xi(i)\right| \geq \delta\right) = 0.$$

(C5) The set $\check{C}$ is compact.

Theorem 5.3.1 (pp. 191–196 of Kushner and Clark [1978]) in this setting says the following:

THEOREM 5.10. *Under (C1) through (C5),* $\tau(n) \to \mathcal{B}$ *as* $n \to \infty$ *almost surely.*

Using Theorem (5.10), we can analyze Equation (25) and we have:

THEOREM 5.11. *Given* $\theta_n \equiv \theta$*, as* $n \to \infty$*,* $\Phi(n) \to \mathcal{K}^1$ *almost surely.*

Remark 5.12. Note that $\mathcal{K}^1$ is composed of both minima and unstable equilibria. From the discussion in Section 5.1, all the minima in $\mathcal{K}^1$ correspond to global minima. In principle, a stochastic search algorithm may get trapped in unstable equilibria. Under additional assumptions on the noise (see Chapter 4 of Borkar [2008]), one can ensure nonconvergence to unstable points. Most often in practice, the scheme even without additional noise conditions will converge to stable equilibria, that is, minima. Thus, $\mathcal{K}^1$ will in fact correspond to the set of global minima.

Now consider the slowest recursion, that is, the θ-update corresponding to Equation (26), that is,

$$\theta_{n+1} = \Gamma^2\left(\theta_n - c(n)\delta_n\psi(X_n, Z_n)\right), \tag{50}$$

where the scalar $\delta_n = k(X_n, Z_n) - \rho_n + V_n(X_{n+1}) - V_n(X_n)$ corresponds to the TD error and the vector $V_n = \Phi_n r_n$ is the estimate of the differential value function.

Let $\mathcal{I}_n^1 = \sigma\{\theta_m, \rho_m, r_m, y_m, \Phi_m, m \le n\}$. Then, the recursion in Equation (26) can be rewritten as

$$\begin{aligned}\theta_{n+1} \;=\; \Gamma^2\big(\theta_n - c(n)\big(&E[\delta_n^{\theta_n}\psi(X_n, Z_n)|\mathcal{I}_n^1] + (\delta_n\psi(X_n, Z_n) - E[\delta_n\psi(X_n, Z_n)|\mathcal{I}_n^1]) \\ &+ E[(\delta_n - \delta_n^{\theta_n})\psi(X_n, Z_n)|\mathcal{I}_n^1]\big)\big),\end{aligned} \tag{51}$$

where the scalar $\delta_n^{\theta_n} = k(X_n, Z_n) - \rho_n + V^{\theta_n}(X_{n+1}) - V^{\theta_n}(X_n)$ corresponds to the temporal difference and the vector $V^{\theta_n} = \Phi^*_{\theta_n} r^*_{\theta_n, \Phi^*_{\theta_n}}$ is the converged differential value function for the policy fixed at π_{θ_n}. The latter is obtained as the product of the equilibrated values of the Φ and r recursions, respectively. The converged differential value function will in practice correspond to the true differential value function (see Remark 5.12).

The term $\chi_n^1 \equiv E[(\delta_n - \delta_n^{\theta_n})\psi(X_n, Z_n)|\mathcal{I}_n^1]$ in Equation (51) is $o(1)$ almost surely. This is because the Φ and r recursions on the faster timescale converge to values such that their product will (in practice) converge to the true differential state value function.

Let $M_{n+1}^3 = (\delta_n\psi(X_n, Z_n) - E[\delta_n\psi(X_n, Z_n)|\mathcal{I}_n^1])$. It is easy to see that

$$E[\parallel M_{n+1}^3 \parallel^2 | \mathcal{I}_n^1] \le C(1 + \parallel \theta_n \parallel^2 + \parallel \rho_n \parallel^2 + \parallel r_n \parallel^2) \tag{52}$$

for some $C < \infty$. From Equation (52), one can see that $E[\parallel M_{n+1}^3 \parallel^2] < \bar{C}, \forall n$ for some $\bar{C} < \infty$, since the ρ and θ iterates are uniformly bounded and the r-iterates are stable.

Now, Equation (51) becomes

$$\theta_{n+1} = \Gamma^2\left(\theta_n - c(n)\big(h^4(\theta_n) + (E[\delta_n^{\theta_n}\psi(X_n, Z_n)|\mathcal{I}_n^1] - h^4(\theta_n)) + M_{n+1}^3 + \chi_n^1\big)\right), \tag{53}$$

where

$$h^4(\theta) = \sum_{i \in S} d^\theta(i) \sum_{a \in A} \nabla\pi_\theta(i, a)[k(i, a) - \rho(\theta) + \sum_{j \in S} P_{ij}(a)V^\theta(j) - V^\theta(i)] \tag{54}$$

$$= E_{X_n \sim d^\theta, a \sim \pi(X_n, .)}[(Q^\pi(X_n, Z_n) - V^\pi(X_n))\nabla \ln \pi(X_n, Z_n)]. \tag{55}$$

From the policy gradient theorem (see Sutton et al. [2000]), it can be seen that the term on the RHS of Equation (55) is the gradient of the average cost, that is, $h^4(\theta) = \nabla\rho(\theta)$. Using Theorem 7–Corollary 8 (p. 74) and Theorem 9 (p. 75) of Borkar [2008] one can see that in Equation (53), because of the "natural" timescale averaging, $\chi_n^2 \equiv (E[\delta_n^{\theta_n}\psi(X_n, Z_n)|\mathcal{I}_n^1] - h^4(\theta_n)) \to 0$ almost surely as $n \to \infty$.

Consider now the following ODE corresponding to the recursion in Equation (53):

$$\dot{\theta}(t) = \hat{\Gamma}^2(-\nabla\rho(\theta(t))), \tag{56}$$

where for any continuous function $v : \mathcal{R}^L \to \mathcal{R}^L$,

$$\hat{\Gamma}^2(v(y)) = \lim_{0<\eta\to 0}\left(\frac{\Gamma^2(y+\eta v(y)) - y}{\eta}\right).$$

In case the aforementioned limit is not unique, we let $\hat{\Gamma}^2(v(y))$ be the set of all possible limit points (see p. 191 of Kushner and Clark [1978]). Also, if y is an interior point, $\hat{\Gamma}^2(v(y)) = v(y)$. Let

$$\mathcal{K}^2 \stackrel{\triangle}{=} \{\theta \in \mathcal{R}^L \mid \hat{\Gamma}^2(\nabla\rho(\theta)) = 0\}$$

denote the set of all fixed points of Equation (56).

We now have the following main result:

THEOREM 5.13. *As $n \to \infty$, $\theta(n) \to \mathcal{K}^2$ almost surely.*

Remark 5.14. As with $\mathcal{K}^1$, $\mathcal{K}^2$ also corresponds to the set of all Kuhn-Tucker points of the ODE (Equation (56)). These include both local minima and unstable equilibria. As explained in Remark 5.12, the θ-recursion in practice will converge to stable equilibrium points, that is, local minima. From the foregoing and the discussion in Section 5.1, the algorithm will converge almost surely to local minima of the true long-run average cost function.

Remark 5.15. In the actor-critic algorithms of Konda and Tsitsiklis [2003], at certain parameter values of the policy parameter θ, the feature vectors of the critic can be close to zero or linearly dependent. This will cause the projection operator projecting the state-action value function into the critic's feature space to be ill-conditioned, rendering the algorithm unstable. Our algorithms do not suffer from such instability issues as our critic directly estimates the temporal difference by implicitly learning the state value function as with algorithms in Bhatnagar et al. [2009]. On the other hand, the critic in the actor-critic algorithms of Konda and Tsitsiklis [2003] learn the projection of the state-action value function onto the critic's feature space. An important difference between algorithms in Bhatnagar et al. [2009] and our algorithms lies in the fact that the estimates of the value function in the algorithms of Bhatnagar et al. [2009] are biased, whereas our estimates are unbiased. This is because for any given policy update, the approximate value function that we obtain from function approximation converges to the true value function (under that policy) as a result of the Φ update converging to the global minimum.

Remark 5.16. Note that we did not analyze the rate of convergence of our algorithms. In Konda and Tsitsiklis [2004], the rate of convergence of two timescale stochastic algorithms with linear objectives has been analyzed. Carrying out a similar analysis in the case of nonlinear objectives appears hard, though it would be an interesting research direction.

Table I. Details of the MDP Settings Used in the Experiments

S.No.	N	A	K	L
1. MDP1	10	10	3	20
2. MDP2	100	100	10	100
3. MDP3	1,000	1000	20	200

6. EXPERIMENTS

In this section, we show the performance of Algorithm 1 on three random MDP settings (MDP1, MDP2, and MDP3, respectively) whose parameters are given in Table I. We will refer to our algorithm as BR in the plots. We compare the results of BR with another algorithm where the residual gradient scheme is replaced with TD(0). Note that the other update rules for feature adaptation (Equations (22), (24), and (25)) and policy improvement (Equation (26)) remain unchanged. We will refer to this algorithm as TD(0) in the plots. We compare our algorithms BR and TD(0) with the actor-critic algorithms ABBE and ABTD in Castro and Mannor [2010]. In the critic part of the ABBE and ABTD algorithms, we used the basis functions ϕ (feature vectors for state) parameterized by s as $\phi(i, s) = \cos(\frac{i}{d}s + \rho_{i,d})$, where $1 \leq i \leq N, 1 \leq d \leq K$, and $\rho_{i,d}$ are uniform phases. Note that this choice of the parameterization and basis functions is the same as in Castro and Mannor [2010].

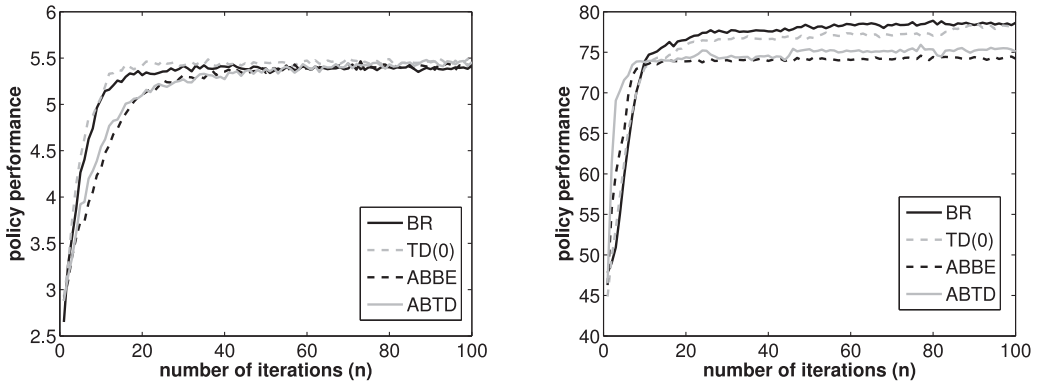
In all the MDP settings, we simulate the transition probability to the next state given the current state i and action a as $j \sim \min(1 + \mathit{Binomial}(N, i * a/(S+1) * (A+1)), N)$, where $\mathit{Binomial}(N, \tilde{p})$ denotes the binomial random variable with success probability $\tilde{p}$, and set the cost function g as $g(i, u, j) = -\mid i - u + j \mid$.

We also consider another setting involving a machine replacement problem and compare all algorithms. Please refer to Bertsekas [2011] for a description of the transition dynamics and cost structure for this problem.

The actor-critic feature adaptation algorithm was run on each individual setting where the critic first estimates the value function and the actor subsequently uses the estimate to improve the policy. We let the step-sizes be $a(n) = 1/n^{0.51}$, $b(n) = 1/n^{0.6}$, and $c(n) = 1/n$, respectively. In the plots, the y-axis corresponds to the policy performance $-\rho(\theta)$ (negative of the average cost) and the x-axis corresponds to the number of slower timescale iterations. From Remarks 4.3 and 4.4, note that we update the slower timescale recursion after 100 updates of faster timescale recursion and a single update of the medium timescale recursion. The policy features $\sigma(s, a)$ were randomly generated and fixed during the experiments.

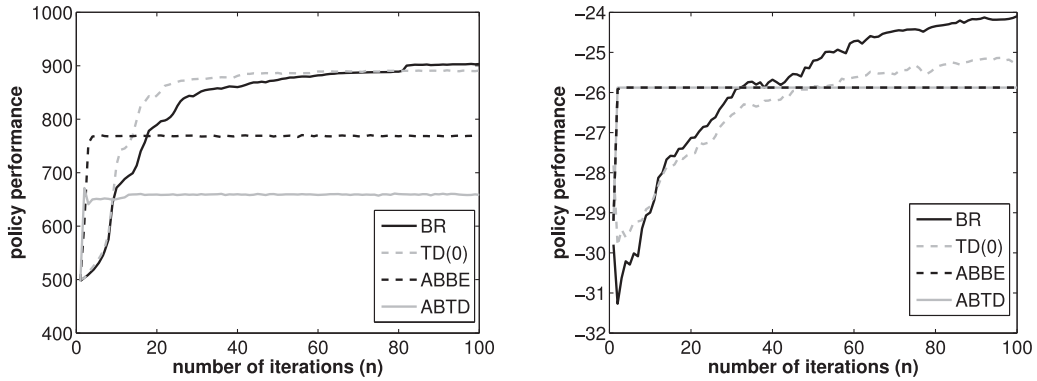
Figures 2(a), 2(b), and 3(a) compare all the algorithms based on the average cost criterion performance for the MDP1, MDP2, and MDP3 settings, respectively. Figure 3(b) compares the performance of all the algorithms for the machine replacement problem. It can be seen that our algorithms BR and TD(0) perform better than the ABBE and ABTD algorithms (see Figure 3(a) for significant improvement). This is due to the non-parameterized feature representation in our algorithms compared to parameterized features for the ABBE and ABTD algorithms. The performance improvement can be seen to be noisy as is the case for any stochastic update rule. Nonetheless, the average behavior of $-\rho(\theta)$ can be seen to increase with the number of iterations.

It can be seen from the plots that *TD*(0), despite not minimizing the MSBE error objective that we considered, shows good performance as it provides a good approximation to the differential value function. Although the theoretical convergence of the resulting scheme (using the MSBE objective) under TD(0) has not been proved, this scheme is computationally advantageous, as it uses only one simulation sample (for the "next" state generation, instead of two such samples) at each iterate.



(a) MDP1 setting (b) MDP2 setting

Fig. 2. Performance comparison plot $-\rho(\theta)$ versus n.



(a) MDP3 setting (b) Machine Replacement setting

Fig. 3. Performance comparison plot $-\rho(\theta)$ versus n.

We obtain similar results for the other control algorithm described in Section 4.2. For lack of space, we have provided these results in the Online Appendix.

7. CONCLUDING REMARKS

We presented online actor-critic schemes with feature adaptation on the Grassmannian for both the average and the discounted cost objectives. Our algorithms incorporate feature adaptation in the critic by using a simple gradient search on the Grassmannian of features. The average cost critic estimates the differential value function and provides this estimate to the actor to obtain the policy gradient in order to improve the policy performance. In the case of the discounted cost critic, the value function is estimated and provided to the actor to obtain the SPSA gradient estimate in order to improve the policy performance.

It would be interesting to investigate more sophisticated gradient methods like Newton's method and conjugate gradient methods on manifolds as discussed in Edelman et al. [1998] for adapting features as these methods have faster convergence guarantees, even though they would require more simulation samples. A modified version of our algorithms with TD(0) on the faster timescale is also seen to exhibit good performance even though TD(0) is not designed as such for finding an optimum for the

MSBE error objective. It would be interesting to extend the theoretical analysis to include convergence of our scheme with TD(0) in place of the residual gradient scheme on the faster timescale updates. As future work, we shall include a sparsity penalty, thereby modifying the scheme to prefer sparser basis vectors to improve scalability. Further, it would be interesting to apply similar techniques to the case of parameterized function approximation architectures that could then be used in the case of high-dimensional state-action spaces where storing the entire feature matrix may become computationally infeasible.

The natural actor-critic algorithms (see Bhatnagar et al. [2009]) provide faster convergence by utilizing the manifold nature of the policy parameterization. It is thus important to understand if the natural gradient schemes can be combined with our algorithms to obtain more efficient algorithms. Also, it has been shown in Thomas et al. [2013] that the natural actor-critic algorithms are equivalent to mirror descent methods. It would be interesting to study if one could do feature adaptation in the context of mirror-descent RL methods as well. Incorporating random search in our scheme as with Bhatnagar et al. [2013a] would likely result in more efficient algorithms as well.

ELECTRONIC APPENDIX

The electronic appendix for this article can be accessed in the ACM Digital Library.

REFERENCES

P. A. Absil, R. Mahony, and R. Sepulchre. 2009. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press.

L. C. Baird. 1995. Residual algorithms: Reinforcement learning with function approximation. In *Proceedings of the 12th International Conference on Machine Learning*. 30–37.

J. S. Baras and V. S. Borkar. 2000. A learning algorithm for Markov decision processes with adaptive state aggregation. In *Proceedings of the 39th IEEE Conference on Decision and Control*, Vol. 4. 3351–3356.

A. G. Barto. 1998. *Reinforcement Learning: An Introduction*. MIT Press.

A. G. Barto, R. S. Sutton, and C. W. Anderson. 1983. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man and Cybernetics,* 5 (1983), 834–846.

D. P. Bertsekas. 2011. *Dynamic Programming and Optimal Control*. Vol. 2, 4th ed. Athena Scientific, Belmont, MA.

S. Bhatnagar, V. S. Borkar, and K. J. Prabuchandran. 2013a. Feature search in the Grassmanian in online reinforcement learning. *IEEE Journal of Selected Topics in Signal Processing* 7, 5 (2013a), 746–758.

S. Bhatnagar, V. S. Borkar, and L. A. Prashanth. 2012. Adaptive feature pursuit: Online adaptation of features in reinforcement learning. *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control.* IEEE Press Computational Intelligence Science, IEEE Press and Wiley, 517–534.

S. Bhatnagar, H. L. Prasad, and L. A. Prashanth. 2013b. *Stochastic Recursive Algorithms for Optimization: Simultaneous Perturbation Methods*. Springer.

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

V. S. Borkar. 1997. Stochastic approximation with two time scales. *Systems & Control Letters* 29, 5 (1997), 291–294.

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

D. D. Castro and S. Mannor. 2010. Adaptive bases for reinforcement learning. *Machine Learning and Knowledge Discovery in Databases* (2010), 312–327.

A. Edelman, T. A. Arias, and S. T. Smith. 1998. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications* 20, 2 (1998), 303–353.

J. Hamm and D. D. Lee. 2008. Grassmann discriminant analysis: A unifying view on subspace-based learning. In *Proceedings of the 25th International Conference on Machine Learning*. ACM, 376–383.

P. W. Keller, S. Mannor, and D. Precup. 2006. Automatic basis function construction for approximate dynamic programming and reinforcement learning. In *Proceedings of the 23rd International Conference on Machine Learning*. ACM, 449–456.

V. R. Konda and J. N. Tsitsiklis. 2003. Onactor-critic algorithms. *SIAM Journal on Control and Optimization* 42, 4 (2003), 1143–1166.

V. R. Konda and J. N. Tsitsiklis. 2004. Convergence rate of linear two-time-scale stochastic approximation. *Annals of Applied Probability* 14, 2 (2004), 796–819.

H. J. Kushner and D. S. Clark. 1978. *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. Vol. 6. Springer-Verlag, New York.

M. G. Lagoudakis and R. Parr. 2003. Least-squares policy iteration. *Journal of Machine Learning Research* 4 (2003), 1107–1149.

S. Mahadevan and B. Liu. 2010. Basis construction from power series expansions of value functions. In *Advances in Neural Information Processing Systems*. 1540–1548.

S. Mahadevan and M. Maggioni. 2007. Proto-value functions: A Laplacian framework for learning representation and control in Markov decision processes. *Journal of Machine Learning Research* 8, 16 (2007), 2169–2231.

P. Marbach and J. N Tsitsiklis. 2001. Simulation-based optimization of Markov reward processes. *IEEE Transactions on Automatic Control,* 46, 2 (2001), 191–209.

I. Menache, S. Mannor, and N. Shimkin. 2005. Basis function adaptation in temporal difference reinforcement learning. *Annals of Operations Research* 134, 1 (2005), 215–238.

G. Meyer, S. Bonnabel, and R. Sepulchre. 2011. Regression on fixed-rank positive semidefinite matrices: A Riemannian approach. *Journal of Machine Learning Research* 12 (2011), 593–625.

R. Parr, C. Painter-Wakefield, L. Li, and M. Littman. 2007. Analyzing feature generation for value-function approximation. In *Proceedings of the 24th International Conference on Machine Learning*. 737–744.

K. J. Prabuchandran, S. Bhatnagar, and V. S. Borkar. 2014. An actor critic algorithm based on Grassmanian search. In *Proceedings of the 53rd IEEE Conference on Decision and Control*. 3597–3602.

K. Rohanimanesh, N. Roy, and R. Tedrake. 2009. Towards feature selection in actor-critic algorithms. In *Workshop on Abstraction in Reinforcement Learning*. 42–48.

S. T. Smith. 1993. *Geometric Optimization Methods for Adaptive Filtering*. Harvard University, Cambridge, MA.

J. C. Spall. 1992. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control,* 37, 3 (1992), 332–341.

Y. Sun, M. Ring, J. Schmidhuber, and F. J. Gomez. 2011. Incremental basis construction from temporal difference error. In *Proceedings of the 28th International Conference on Machine Learning*. 481–488.

R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. 2000. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, Vol. 12. 1057–1063.

P. S. Thomas, W. C. Dabney, S. Giguere, and S. Mahadevan. 2013. Projected natural actor-critic. In *Advances in Neural Information Processing Systems*. 2337–2345.

J. N. Tsitsiklis and B. Van Roy. 1997. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control,* 42, 5 (1997), 674–690.

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

L. Wolf and A. Shashua. 2003. Learning over sets using kernel principal angles. *Journal of Machine Learning Research* 4 (2003), 913–931.

H. Yu and D. P. Bertsekas. 2009. Basis function adaptation methods for cost approximation in MDP. In *Adaptive Dynamic Programming and Reinforcement Learning*. IEEE, 74–81.

Received June 2014; revised December 2015; accepted December 2015