2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

TD-Learning with Exploration

Sean P. Meyn and Amit Surana

Abstract— **We introduce exploration in the TD-learning algorithm to approximate the value function for a given policy. In this way we can modify the norm used for approximation, "zooming in" to a region of interest in the state space. We also provide extensions to SARSA to eliminate the need for numerical integration in policy improvement. Construction of the algorithm and its analysis build on recent general results concerning the spectral theory of Markov chains and positive operators.**

I. Introduction

This paper concerns a Markov Decision Process, or MDP, defined by a state space X, action space U, and controlled transition law P_u. Our goal is value function approximation: We focus on the discounted-cost optimal control problem with cost function $c\colon \mathsf{X}\times\mathsf{U}\to\mathbb{R}_+$, and discount factor $\beta\in(0,1)$. For a given control sequence $\boldsymbol{U}$, the resulting value function is given by

$$h(x)=\sum_{t=0}^{\infty}\beta^t\mathsf{E}_x\big[c(X(t),U(t))\big] \tag{1}$$

The subscript indicates that the initial condition is $X(0)=x$.

The state process $\boldsymbol{X}$ evolves on X, and the control (or *action*) process $\boldsymbol{U}$ evolves on U. The state space and action space are general, with associated sigma-algebras, denoted $\mathcal{B}(\mathsf{X})$ and $\mathcal{B}(\mathsf{U})$, respectively. There may be state constraints, in which case there is, for each $x\in\mathsf{X}$, a set $\mathsf{U}(x)\subset\mathsf{U}$ that consists of permissible values of $U(t)=u$ when $X(t)=x$. In this generality we cannot hope to compute an optimal policy, so we turn to approximation based on Monte-Carlo methods, or experiments on a physical system via reinforcement learning [2].

One approach is through approximate policy iteration. For this, we are given control sequence $\boldsymbol{U}$, and we require an approximation of the resulting value function h. Under general conditions, this approximation can be constructed using the Temporal-Difference (TD) learning algorithm. For a linear approximation $h^\theta(x)=\theta^{\mathrm{T}}\psi(x)$, where $\theta\in\mathbb{R}^n$ and $\psi\colon\mathsf{X}\to\mathbb{R}^n$, the LSTD algorithm can be applied. This coincides with stochastic Newton-Raphson, which is known to have minimal variance (the same optimal asymptotic variance as obtained in the two-time-scale stochastic approximation algorithm of Polyak and Juditsky [3]).

S. P. Meyn is with the Department of Electrical and Computer Engineering and the Coordinated Science Laboratory at UIUC meyn@illinois.edu

A. Surana is with United Technologies Research Center (UTRC) suranaa@utrc.utc.com

Financial support from UTRC, AFOSR grant FA9550-09-1-0190, ITMANET DARPA RK 2006-07284, and NSF grant CPS 0931416 is gratefully acknowledged.

However, TD-learning has a serious drawback: The norm under which the approximation is based depends on the steady-state distribution of the controlled Markov model. That is, the TD- or LSTD-learning algorithms compute the solution to the quadratic program,

$$\min_\theta \|h-h^\theta\|_\pi^2=\mathsf{E}_\pi[(h(X)-h^\theta(X))^2] \tag{2}$$

where in the expectation on the right we have $X\sim\pi$, where π is the steady-state distribution of $\boldsymbol{X}$ (it is assumed that this exists).

There is no reason why the norm defined in (2), or any of its weighted refinements, should provide an appropriate metric. In fact, for deterministic systems with a unique equilibrium, π will be a point-mass at the equilibrium, so any of these norms will only weight the equilibrium!

The Q-learning algorithm of Watkins addresses this difficulty [23], [24], [2], [21]. However, while the technique is provably convergent when the parameterization captures *all functions* on the joint state-action space, there is little theory to deal with approximations, and no Q-learning algorithm has been devised that approaches the efficiency of TD-learning.

In this paper we marry the benefits of these two approaches by introducing exploration in the TD-algorithm, and by extending the resulting algorithm to approximate the "Q-function" that appears in Q-learning. Construction of the algorithm and its analysis builds on recent general results concerning the spectral theory of Markov chains and positive operators.

The notion of exploration in machine learning and statistics has a very long history (see [7], [1] and the references therein). Note that in this paper the exploration is introduced to obtain an approximation of the value function *for a given policy*. The algorithms introduced here can be used to approximate the value function for a given policy, even for deterministic systems for which the state and action sequence converge to an equilibrium. Such approximations are valuable for performance evaluation, as well as policy improvement.

The remainder of this paper is organized as follows: Background on MDPs and Markov chains is surveyed in Sec. II. The Least Squares algorithms are developed in two sections: Sec. III contains a construction of LSTD with exploration, and in Sec. IV we show how this can be "lifted" to obtain the LS-SARSA algorithm, by viewing $(\boldsymbol{X},\boldsymbol{U})$ as a state process. Numerical examples are contained in Sec. V. Sec. VI contains conclusions, and suggests some topics for future research.

978-1-61284-799-3/11/$26.00 ©2011 IEEE

II. Markov and MDP background

A. MDP background

For any set $A \in \mathcal{B}(\mathsf{X})$, $x \in \mathsf{X}$, and $u \in \mathsf{U}(x)$, the transition law P_u introduced at the start of Sec. I defines the probability of transition in one step, $P_u(x, A) := \mathsf{P}\{X(t+1) \in A \mid X(t) = x,\ U(t) = u\}$. We view P_u as a mapping from functions on X to functions on $\mathsf{X} \times \mathsf{U}$: For any measurable $h\colon \mathsf{X} \to \mathbb{R}$ we denote,

$$P_u h\,(x) := \mathsf{E}[h(X(t+1)) \mid X(t) = x,\ U(t) = u]$$

We denote by $\phi\colon \mathsf{X} \to \mathsf{U}$ a stationary policy, giving $U(t) = \phi(X(t))$. The controlled process $\boldsymbol{X}$ is then a Markov chain with transition kernel denoted $P_\phi(x, dy) = P_{\phi(x)}(x, dy)$.

In this paper we focus on the discounted-cost optimal control problem: We wish to minimize the infinite-horizon discounted cost (1) over all policies, where c is a non-negative cost-function of states and actions. The minimal value function $h^*(x)$, if it exists, satisfies the discounted-cost optimality equation (DCOE),

$$\min_{u \in \mathsf{U}(x)} \{c(x,u) + \beta P_u\, h^*(x)\} = h^*(x), \qquad x \in \mathsf{X}. \quad (3)$$

The minimizer then defines an optimal policy in state feedback form.

B. Reinforcement learning background

Two approaches to approximation of an optimal policy are TD-learning and Q-learning [2], [21]. In Q-learning we attempt to approximate the "Q-function" given by the term within the brackets in (3): $Q^*(x,u) := c(x,u) + \beta P_u\, h^*(x)$. Letting $\underline{Q}^*(x) = \min_{u \in \mathsf{U}(x)} Q^*(x,u)$, we deduce from (3) that Q^* solves a similar fixed point equation:

$$c(x,u) + \beta P_u\, \underline{Q}^*(x) = Q^*(x,u)\,.$$

In Watkins' Q-learning algorithm the goal is to compute Q^* exactly based on observations of $(\boldsymbol{X}, \boldsymbol{U})$ using a randomized, non-optimal policy [23], [24]. In the general state space case considered here we can only consider approximations.

For approximation consider an affine parameterization of the form

$$Q^\theta(x,u) = c(x,u) + \theta^{\mathrm{T}} \psi\,(x,u). \quad (4)$$

with $\theta \in \mathbb{R}^n$ and $\psi\colon \mathsf{X} \times \mathsf{U} \to \mathbb{R}^n$. With $\underline{Q}^\theta(x) := \min Q^\theta(x,u)$, the resulting *Bellman error* is denoted,

$$\mathcal{E}^\theta(x,u) := Q^\theta(x,u) - \big(c(x,u) + \beta P_u\, \underline{Q}^\theta(x)\big) \quad (5)$$

Minimization of the Bellman error in an appropriate metric can be transformed to the solution of a linear program, but the theory of on-line learning algorithms for optimization is not well developed. To the best of our knowledge, the only globally convergent algorithms are obtained under very special assumptions: [25] applies only to the optimal stopping problem, and [12] (with convergence justified in [19]) is applicable only for deterministic control problems.

Approximation theory is more complete for approximation of the value function for a fixed policy. Suppose that $P = P_\phi$ is a transition law obtained for a feedback law ϕ. The fixed-policy discounted-cost dynamic programming equation is (3) without the minimization,

$$c + \beta P h = h \quad (6)$$

Assuming that the Markov chain with transition law P has an invariant measure π, in TD-learning we assume that a parameterized family of approximations is given, denoted $\{h^\theta : \theta \in \mathbb{R}^n\}$, and we seek a solution to the minimization (2).

The main motivation for TD-learning is policy improvement: Given a solution h to (6), we obtain an improved policy via,

$$\phi^+(x) = \arg\min_{u \in \mathsf{U}(x)} \{c(x,u) + \beta P_u h\,(x)\}$$

This may be impossible to compute due to complexity of the integration required in computation of $P_u h$, or because P_u is not known.

SARSA is an alternative to TD-learning, designed to address this difficulty. The idea is to formulate a Q-function for the fixed-policy problem as follows:

$$Q(x,u) := c(x,u) + \beta P_u\, h(x) \quad (7)$$

where h is the solution to (6), so that $Q(x, \phi(x)) = h(x)$. If Q is known, then the updated policy is obtained without integration:

$$\phi^+(x) = \arg\min_{u \in \mathsf{U}(x)} Q(x,u) \quad (8)$$

SARSA is an acronym for State-Action-Reward-State-Action – it was introduced in the 1994 technical report of Rummery & Niranjan [18], and developed in several later papers and texts [21], [20]. It is in fact equivalent to Watkins' Q-learning algorithm, with the restriction to a trivial action space (equivalently, a fixed policy).

Just as in Q-learning, the theory for approximation in SARSA is currently weak. The main goal of this paper is to address these weaknesses, and thereby combine the advantages of TD- and Q-learning.

Basis selection In any of these approximation techniques, the question always then comes to this: *How do we choose the basis*? General approaches to basis selection are given in [22], [11]. A recent approach is to approximate the dynamic programming equations using a simpler model, such as a fluid or diffusion model that approximates the discrete-time MDP model, or an approximation via a limiting model, as constructed in mean-field games [8]. Some successful applications of this approach are presented in the final chapter of [13], and in [16], [12], [4], [15], [7].

A general procedure can be described as follows. Suppose that the evolution of the controlled Markov chain is described by the nonlinear state space model,

$$X(t+1) = X(t) + f(X(t), U(t), W(t+1)), \qquad t \ge 0, \quad (9)$$

where $\boldsymbol{W}$ is i.i.d., and the state and control evolve on Euclidean space. On denoting $\overline{f}(x,u) = \mathsf{E}[f(x,u,W(t))]$, the fluid model is defined by the controlled ODE,

$$\tfrac{d}{dt} x(t) = \overline{f}(x(t), u(t)) \quad (10)$$

A diffusion model that takes into account variability is defined similarly. In many cases, the fluid model gives enough insight to obtain a good basis using the procedure we describe.

Consider first the approximation of h appearing in (6), for a Markov model without control (or with fixed policy). In this case the models (9,10) are the same, except that the control terms are dropped:

$$X(t+1) = X(t)+f(X(t),W(t+1)), \qquad \tfrac{d}{dt}x(t) = \overline{f}(x(t))$$

Let $D = P-I$ denote the "generator" for the Markov model, and write (6) in the form,

$$c + \beta\mathcal{D}h = (1-\beta)h$$

Now, note that $\mathcal{D}h\,(x) = \mathsf{E}[h(X(t+1))-h(X(t)) \mid X(t) = x]$. We approximate this by the *differential generator* for the fluid model, $\mathcal{D}^{\text{F}}h\,(x) = \overline{f}(x)\cdot\nabla h(x)$. Hence our goal is to solve,

$$c + \beta\mathcal{D}^{\text{F}}h = (1-\beta)h$$

This has the solution $h = \beta^{-1}J_\gamma(x)$, with $\gamma = (1-\beta)/\beta$, and

$$J_\gamma(x) = \int_0^\infty e^{-\gamma t}c(x(t))\,dt, \qquad x(0) = x. \tag{11}$$

If this discounted-cost value function is computable, or if an approximation exists, then it is often a good starting point for a basis in the Markov model.

This approach is easily extended to SARSA. Returning to the model with control, we define the controlled generator for any function h by $\mathcal{D}_u h\,(x) = \mathsf{E}[h(X(t+1))-h(X(t)) \mid X(t) = x,\ U(t) = u]$. The fixed point equation (7) is expressed in terms of the generator by,

$$Q(x,u) = c(x,u) + \beta\big(h(x) + \mathcal{D}_u h\,(x)\big)$$

If $\{h^\theta = \theta^{\mathsf{T}}\psi\}$ is an approximation family for h, then a natural choice for Q is,

$$Q^\theta(x,u) := c(x,u) + \beta\big(h^\theta(x) + \mathcal{D}^{\text{F}}_u\, h^\theta(x)\big) \tag{12}$$

where $\mathcal{D}^{\text{F}}_u h(x) := \overline{f}(x,u)\cdot\nabla h(x)$.

C. Markov background

The Markov chains considered in this paper are assumed to be geometrically ergodic. This is essentially equivalent to a solution $V\colon \mathsf{X}\to[1,\infty)$ to the drift condition (V4) of [14]:

$$PV\,(x) \le \lambda V(x) + b\mathbb{I}_S(x), \qquad x\in\mathsf{X}, \tag{13}$$

where $\lambda\in(0,1)$, $b<\infty$, and the set S is "small". We refer the reader to [14] for further background.

Following the notation of [14], we denote by L^V_∞ the Banach space of functions bounded by V, with norm,

$$\|f\|_V = \sup_{x\in\mathsf{X}} \frac{|f(x)|}{V(x)}\,.$$

We let $\|\!|\cdot|\!\|_V$ denote the induced operator norm: For two transition laws P and P' we denote,

$$\|\!|P-P'|\!\|_V = \sup_{\|f\|_V=1} \|Pf-P'f\|_V$$

Under mild conditions, geometric ergodicity is equivalent to a solution to (V4), and this in turn is equivalent to the V-*uniform ergodicity*: The existence of an invariant measure π such that P^t converges to π in the induced operator norm, geometrically fast: For some $R_0<\infty$, $r_0>1$,

$$\|\!|P^t - 1\otimes\pi|\!\|_V \le R_0 r_0^{-t}, \quad t\ge 0.$$

This implies that $\mathsf{E}_x[f(X(t))]\to\pi(f)$ at rate r_0^{-t} for any function $f\in L^V_\infty$.

The *exploration* introduced in this paper is analogous to *importance sampling*: We will perform a transformation to obtain a new Markov chain, and then transform the invariance equation (such as (6)) so that the solutions of the two equations match.

The transformed equation will involve a scaling of the transition law, of the following form: For a bounded function $G\colon\mathsf{X}\to\mathbb{R}$, denote $g = e^G$, and $\widehat{P}(x,A) = g(x)P(x,A)$, $x\in\mathsf{X}$, $A\in\mathcal{B}(\mathsf{X})$. The new invariance equation has the form,

$$\widehat{P}h = h - c \tag{14}$$

where $c\colon\mathsf{X}\to\mathbb{R}$ is a simple function of c. Solutions to identities of this form can be characterized by appealing to the Perron-Frobenius theory of positive matrices [17], [9], [10].

Under (V4), we view $\widehat{P}$ as a bounded linear operator on L^V_∞. Its log-spectral radius is denoted,

$$\Lambda(G) := \lim_{n\to\infty}\frac{1}{n}\log(\|\!|\widehat{P}^n|\!\|_V) \tag{15}$$

If $\Lambda(G)<0$, this means that the spectral radius of $\widehat{P}$ is less than one. If moreover the function $c\colon\mathsf{X}\to\mathbb{R}$ satisfies $c\in L^V_\infty$, it then follows that $\widehat{P}^n c\to 0$ geometrically fast in L^V_∞. We conclude that the unique solution to (14) is given by the infinite sum,

$$h = \sum_{t=0}^\infty \widehat{P}^t c \tag{16}$$

where $\widehat{P}^0 = I$. This representation will be used to construct learning-algorithms in the next two sections.

III. LSTD WITH EXPLORATION

We begin in the simpler setting without control. We have transition matrix P and a cost function $c\colon\mathsf{X}\to\mathbb{R}_+$. For a given discount factor $\beta\in(0,1)$, we seek an approximation to the fixed-policy discounted-cost dynamic programming equation (6). The starting point of the construction of the algorithm is the construction of a regeneration time, and a regeneration distribution μ on $\mathcal{B}(\mathsf{X})$.

The following assumptions are taken for granted throughout this section:

Assumptions for Sec. III: The chain is V-uniformly ergodic, so that the drift condition (13) holds for some $V\colon\mathsf{X}\to[1,\infty)$. The cost function satisfies $c^2\in L^V_\infty$, and the n-dimensional basis satisfies $\psi_i^2\in L^V_\infty$ for each $1\le i\le n$. The regeneration distribution μ satisfies $\mu(V)<\infty$. ■

Under these ergodicity assumptions, there is a unique steady-state distribution π for $\boldsymbol{X}$, and the steady state cost $\pi(c) := \int c(x)\,\pi(dx)$ as well as its variance are finite.

A. Regeneration

We construct a new transition matrix $\widetilde{P}$ by restarting the chain: Let μ denote a probability measure on $\mathcal{B}(\mathsf{X})$, and suppose that the process is restarted according to μ at a randomized stopping time, denoted $\mathcal{T}$. Assume there is a function $\delta\colon \mathsf{X} \to [0,1]$, such that for any state x and set $A \in \mathcal{B}(\mathsf{X})$,

$$\widetilde{P}(x,A) = (1-\delta(x))P(x,A) + \delta(x)\mu(A) \qquad (17)$$

In the notation of [17], [9] this is expressed $\widetilde{P} = I_{1-\delta}P + \delta \otimes \mu$.

The distribution of the randomized stopping time $\mathcal{T}$, taking values in $\{0,1,2,\dots\}$, is specified by $\mathsf{P}\{\mathcal{T}=0 \mid \tilde{X}(0)\} = \delta(\tilde{X}(0))$, and for $n \ge 1$,

$$\mathsf{P}\{\mathcal{T} > n \mid \mathcal{T} > n-1,\, \tilde{X}(0),\dots,\tilde{X}(n)\} = 1-\delta(\tilde{X}(n))\,.$$

When $\delta(x) \equiv \delta$ is constant then $\mathcal{T}$ is a geometric random time. The process evolves according to P until the time $\mathcal{T}$, and then regenerates with distribution μ. The chain with transition law $\widetilde{P}$ is denoted $\widetilde{\boldsymbol{X}}$.

We assume throughout that the chain with transition law $\widetilde{P}$ is V-uniformly ergodic. This is justified by the following simple lemma:

Lemma 3.1: Suppose there is $\bar{\delta} < 1$ such that $\delta(x) \le \bar{\delta}$ for all x. Then the chain with transition law $\widetilde{P}$ is V-uniformly ergodic.

Proof: Condition (V4) (the bound (13)) holds for this chain. Moreover, under the assumed bound, a set S is *small* for $\widetilde{P}$ (see [14]), whenever it is small for P. ∎

The dynamic programming equation for P results in a similar invariance equation for $\widetilde{P}$. However, it will be helpful to first refine the construction of the Markov chain $\widetilde{\boldsymbol{X}}$. Let $\boldsymbol{I}$ denote the randomized function of $\widetilde{\boldsymbol{X}}$, taking values in $\{0,1\}$, defined via

$$\mathsf{P}\{I(t)=1 \mid \tilde{X}_0^t,\ I_0^{t-1}\} = \delta(\tilde{X}(t))$$

That is, the process $\boldsymbol{I}$ explicitly models the 'coin flip' used in the construction of the regeneration epochs for $\widetilde{\boldsymbol{X}}$. We denote the joint process by $\tilde{Y}(t) = (\tilde{X}(t), I(t))$. Since $I(t)$ is a randomized function of $\tilde{X}(t)$, the transition kernel for $\widetilde{\boldsymbol{Y}}$ is specified by,

$$\begin{aligned}\mathsf{P}\{\tilde{X}(t+1)\in A \mid \tilde{Y}(t) = (\tilde{X}(t), I(t)) = (x,a)\} \\ = \begin{cases} P(x,A) & \text{If } a=0; \\ \mu(A) & \text{If } a=1. \end{cases}\end{aligned}$$

We can then interpret the difference $\widetilde{P} - \delta\otimes\mu$ as follows:

$$[\widetilde{P} - \delta\otimes\mu](x,A) = \mathsf{P}\{\tilde{X}(t+1)\in A \textit{ and } I(t)=0 \mid \tilde{X}(t)=x\} \qquad (18)$$

Returning to the dynamic programming equation (6), we have

$$\beta[\widetilde{P} - \delta\otimes\mu]h = (1-\delta)\beta P h = (1-\delta)(h-c)\,. \qquad (19)$$

We denote, $g(x) = \beta/(1-\delta(x))$ and $G = \log(g) = \log(\beta) - \log(1-\delta)$. Denote $\widehat{P} = I_g[\widetilde{P} - \delta\otimes\mu]$, or equivalently

$$\widehat{P}(x,A) = g(x)[\widetilde{P}(x,A) - \delta(x)\mu(A)], \quad x\in\mathsf{X},\ A\in\mathcal{B}(\mathsf{X}).$$

Then, the expression (19) gives,

$$\widehat{P}h = h - c \qquad (20)$$

The identity (20) is not surprising since, combining all of the definitions above, $\widehat{P}$ is nothing but βP! It is the interpretation of (20) in terms of the new Markov chain that will lead to a more flexible family of algorithms for approximating h.

The solution to (20) is obtained as the infinite sum (16):

$$h = \sum_0^\infty \widehat{P}^i c\,. \qquad (21)$$

To understand the solution we must provide an interpretation of the products of $\widehat{P}$. For this we generalize the interpretation given in (18), which we write in the form

$$\begin{aligned}\widehat{P}(x,A) &= g(x)\mathsf{P}\{\tilde{X}(t+1)\in A \textit{ and } I(t)\ne 1 \mid \tilde{X}(t)=x\} \\ &= \mathsf{E}[e^{G(\tilde{X}(0))}\mathbb{I}\{\tilde{X}(1)\in A \textit{ and } I(0)\ne 1\} \mid \tilde{X}(0)=x]\end{aligned}$$

Using this as a starting point, we can show by induction that for any $i \ge 1$, and any function $c \in L_\infty^V$,

$$\widehat{P}^i c\,(x) = \mathsf{E}_x\Big[\exp\Big(\sum_{t=0}^{i-1} G(\tilde{X}(t))\Big)c(\tilde{X}(i))\mathbb{I}\{\mathcal{T}\ge i\}\Big]$$

Recall that the subscript represents the conditioning on $X(0)=x$. In the special case $i=0$, the sum is interpreted as *zero*, giving $\widehat{P}^0 c\,(x) = c(x)$. For $i \ge 1$, the notation $\mathbb{I}\{\mathcal{T}\ge i\}$ is equivalent to the restriction that $I(t)=0$ for $0 \le t < i$.

Given this interpretation for $\widehat{P}^i$, the function h given in (21) has the representation,

$$h(x) = \mathsf{E}_x\Big[\sum_{i=0}^{\mathcal{T}} \exp\Big(\sum_{t=0}^{i-1} G(\tilde{X}(t))\Big)c(\tilde{X}(i))\Big] \qquad (22)$$

B. Approximation

Now we assume we have a linearly parameterized family $h^\theta = \theta^{\mathsf{T}}\psi$, where $\theta \in \mathbb{R}^n$ and $\psi\colon \mathsf{X} \to \mathbb{R}^n$. Let $\tilde{\pi}$ denote the invariant measure for $\widetilde{P}$. We seek an approximation in $L_2(\tilde{\pi})$:

$$\min_\theta \|h - h^\theta\|_{\tilde{\pi}}^2 = \min_\theta \mathsf{E}_{\tilde{\pi}}[(h(\tilde{X}) - h^\theta(\tilde{X}))^2] \qquad (23)$$

To characterize a minimum we simply compute the derivative of $\|h-h^\theta\|_{\tilde{\pi}}^2$ with respect to each θ_i, and set it equal to zero. On denoting $\langle f,g\rangle = \mathsf{E}_{\tilde{\pi}}[f(\tilde{X})g(\tilde{X})]$ for any functions $f,g \in L_2(\tilde{\pi})$, this gives

$$0 = \frac{\partial}{\partial\theta_i}\|h-h^\theta\|_{\tilde{\pi}}^2 = 2\langle h^\theta - h, \psi_i\rangle\,, \quad 1\le i\le n.$$

On denoting $\Xi_{i,j} = \langle\psi_j,\psi_i\rangle$ and $b_i = \langle h,\psi_i\rangle$, we conclude that the optimizer θ^* is any solution to the linear equation, $\Xi\theta^* = b$. We assume that Ξ is full-rank, so that the the optimizer $\theta^* = \Xi^{-1}b$ is unique.

The apparent difficulty in TD learning, as well as the approximation technique described here, is computation of b. This is resolved by further consideration of the representation of h given in (22).

Let $\widehat{R}$ denote the *potential kernel*,

$$\widehat{R} = \sum_{t=0}^{\infty} \widehat{P}^t \tag{24}$$

so that $h = \widehat{R}c$. Letting $\widehat{R}^\dagger$ denote its adjoint, we have

$$b_i = \langle \widehat{R}c, \psi_i \rangle = \langle c, \widehat{R}^\dagger \psi_i \rangle \tag{25}$$

The adjoint has an elegant interpretation that we now describe.

Subject to growth conditions on any two functions f and c, we obtain from (22) that $\langle f, \widehat{R}c \rangle =$

$$\sum_{i=0}^{\infty} \mathsf{E}_{\tilde{\pi}} \Big[f(\tilde{X}(0)) \exp\Big(\sum_{t=0}^{i-1} G(\tilde{X}(t)) \Big) c(\tilde{X}(i)) \mathbb{I}\{\mathcal{T} \geq i\} \Big] \tag{26}$$

To obtain the adjoint, we consider a stationary version of $\widetilde{\boldsymbol{Y}} = (\widetilde{\boldsymbol{X}}, \boldsymbol{I})$, defined on the two-sided time axis. We have by stationarity,

$$\begin{aligned} &\mathsf{E}_{\tilde{\pi}} \Big[f(\tilde{X}(0)) \exp\Big(\sum_{t=0}^{i-1} G(\tilde{X}(t)) \Big) c(\tilde{X}(i)) \mathbb{I}\{\mathcal{T} \geq i\} \Big] \\ &\quad = \mathsf{E}_{\tilde{\pi}} \Big[f(\tilde{X}(-i)) \exp\Big(\sum_{t=-i}^{-1} G(\tilde{X}(t)) \Big) c(\tilde{X}(0)) \mathbb{I}\{\mathcal{T}^- > i\} \Big] \end{aligned}$$

where $\mathbb{I}\{\mathcal{T}^- > i\} = \mathbb{I}\{I(t) \neq 1,\ -i \leq t < 0\}$. That is, $\mathcal{T}^-$ is a regeneration time for the time-reversed process:

$$\mathcal{T}^- = \min\{t \geq 1 : I(-t) = 1\}$$

Thus, we arrive at a representation for the adjoint appearing in the right hand side of (25):

Lemma 3.2: For any function f satisfying $f^2 \in L_\infty^V$, the adjoint can be expressed,

$$\widehat{R}^\dagger f\,(x) = \mathsf{E}_x \Big[\sum_{i=0}^{\mathcal{T}^- - 1} f(\tilde{X}(-i)) \exp\Big(\sum_{t=-i}^{-1} G(\tilde{X}(t)) \Big) \Big] \tag{27}$$

where the sum within the exponential is defined to be zero when $i = 0$. ∎

The form (27) is useful because it involves the *past* of the process, rather than the expression for h in (22), which depends on the future.

Following the standard development of TD-learning, define the sequence of *eligibility vectors*,

$$\psi_g(t+1) = \psi(\tilde{X}(t+1)) + \mathbb{I}\{I(t) = 0\} g(\tilde{X}(t)) \psi_g(t), \tag{28}$$

with $\psi_g(0) \in \mathbb{R}^n$ given as initial condition. For any t denote $\mathcal{T}^-(t) = \min\{T \geq 1 : I(t - T) = 1\}$. On iterating (28), we obtain for any $t > \mathcal{T}$ (so that at least one regeneration has occurred),

$$\psi_g(t) = \sum_{i=t-\mathcal{T}^-(t)+1}^{t} \exp\Big(\sum_{k=i}^{t-1} G(\tilde{X}(k)) \Big) \psi(X(i)) \tag{29}$$

Under general conditions, we can combine this representation with (25) and (27) to conclude that b is given by the ergodic limit,

$$b = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \psi_g(t) c(\tilde{X}(t)) \tag{30}$$

Much simpler is the Law of Large Numbers for Ξ:

$$\Xi = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \psi(\tilde{X}(t)) \psi(\tilde{X}(t))^{\intercal} \tag{31}$$

The *LSTD algorithm with exploration* is then defined as the sequence of approximations, $\hat{\theta}_T = \Xi_T^{-1} b_T$, where

- b_T appears on the right hand side of (30):

$$b_T := \frac{1}{T} \sum_{t=1}^{T} \psi_g(t) c(\tilde{X}(t)) \tag{32}$$

- $\{\Xi_t : t \geq 0\}$ approximates the average appearing in (31), defined recursively by,

$$\Xi_{t+1} = \Xi_t + \frac{1}{t+1} \big(\psi(\tilde{X}(t+1)) \psi(\tilde{X}(t+1))^{\intercal} - \Xi_t \big)\,,\ \ t \geq 0\,,$$

with $\Xi_0 > 0$ arbitrary.

The inverses of $\{\Xi_t : t \geq 0\}$ can be obtained recursively using the Matrix Inversion Lemma.

We believe that convergence of this algorithm will hold under the conditions we have imposed. All that is required for convergence is the justification of the Law of Large Numbers to establish convergence of $\{b_t,\ \Xi_t\}$. This will follow from the fact that the triple $\{\tilde{X}(t), I(t), \psi_g(t)\}$ is Markov, but we may require additional assumptions to establish sufficient regularity of this chain to apply Theorem 17.3.2 of [14].

IV. Extension to SARSA

We now introduce control: The state process $\boldsymbol{X}$ evolves on X as before, but we return to the MDP setting outlined in Sec. II-A: The model is defined by a controlled transition law P_u, cost function $c(x, u)$, and discount factor β. We fix a policy ϕ, and let P_ϕ denote the resulting transition law, $P_\phi(x, dy) = P_{\phi(x)}(x, dy)$, and where c_ϕ denotes the resulting cost function on X: $c_\phi(x) = c(x, \phi(x))$, $x \in \mathsf{X}$. The value function h solves the DCOE (6):

$$c_\phi + \beta P_\phi h = h\,. \tag{33}$$

Our goal is to approximate the Q-function defined in (7). Given the form $Q(x, u) = c(x, u) + \beta P_u h\,(x)$, we take the affine parameterization (4), giving $Q(x, u) - Q^\theta(x, u) = \beta P_u h\,(x) - \theta^{\intercal} \psi\,(x, u)$. We proceed by embedding this approximation problem in the TD-learning framework of the previous section. This embedding is based on the simple observation that the joint process $\boldsymbol{\Phi} := (\boldsymbol{X}, \boldsymbol{U})$ is a Markov chain.

For the joint-process $\boldsymbol{\Phi}$, the Q-function defined in (7) is entirely analogous to the function h that solves (6). Hence we can proceed as in the previous section to solve

$$\min_\theta \|Q - Q^\theta\|_{\tilde{\pi}}^2 = \min_\theta \mathsf{E}_{\tilde{\pi}} [(Q(\widetilde{\Phi}) - Q^\theta(\widetilde{\Phi}))^2] \tag{34}$$

where here $\tilde{\pi}$ is the steady-state distribution for $\tilde{\boldsymbol{\Phi}}$. The main difference here is a slightly more general construction of the regenerated process $\tilde{\boldsymbol{\Phi}}$, since we regenerate the joint-process $\boldsymbol{\Phi}$. We impose the same assumptions used in Sec. III for the joint process:

Assumptions for Sec. IV: The chain $\mathbf{\Phi}$ is V-uniformly ergodic, so that the drift condition (13) holds for some $V\colon \mathsf{X}\times\mathsf{U}\to[1,\infty)$. The cost function satisfies $c^2\in L_\infty^V$, and the n-dimensional basis satisfies $\psi_i^2\in L_\infty^V$ for each $1\le i\le n$. The regeneration distribution μ on $\mathcal{B}(\mathsf{X}\times\mathsf{U})$ satisfies $\mu(V)<\infty$. ∎

The transition matrix $\widetilde{P}$ for $\tilde{\mathbf{\Phi}}$ is defined using the probability μ, and function $\delta\colon \mathsf{X}\times\mathsf{U}\to[0,1]$. For any $\zeta=(x,u)$ and set $A\in\mathcal{B}(\mathsf{X}\times\mathsf{U})$, the definition of $\widetilde{P}$ coincides with (17) on this larger state space:

$$\widetilde{P}(\zeta,A)=(1-\delta(\zeta))P(\zeta,A)+\delta(\zeta)\mu(A)$$

As in previous section, we let $\boldsymbol{I}$ denote the randomized function of $\tilde{\mathbf{\Phi}}$, taking values in $\{0,1\}$, defined via

$$\mathsf{P}\{I(n)=1\mid\widetilde{\Phi}_0^n,\ I_0^{n-1}\}=\delta(\widetilde{\Phi}(n)),\ n\ge0,$$

and the distribution of the randomized stopping time $\mathcal{T}$, taking values in $\{0,1,2,\dots\}$, is specified by $\mathsf{P}\{\mathcal{T}=0\mid\widetilde{\Phi}(0)\}=\delta(\widetilde{\Phi}(0))$, and for $n\ge1$,

$$\mathsf{P}\{\mathcal{T}=n\mid\widetilde{\Phi}_0^n,\ \mathcal{T}\ge n\}=\delta(\widetilde{\Phi}(n)).$$

Next, we recall that Q solves a fixed point equation identical to (3) for the joint-process:

$$c(x,u)+\beta\mathsf{E}[Q(\Phi(t+1))\mid\Phi(t)=(x,u)]=Q(x,u) \tag{35}$$

On letting PQ denote the conditional distribution for the joint-process,

$$PQ\,(x,u):=\mathsf{E}[Q(\Phi(t+1))\mid\Phi(t)=(X(t),U(t))=(x,u)]\,,$$

the fixed point equation (35) can be expressed $c+\beta PQ=Q$. Consequently, the Q function satisfies a fixed point equation for $\tilde{\mathbf{\Phi}}$ that is identical to (19):

$$\beta[\widetilde{P}-\delta\otimes\mu]Q=(1-\delta)\beta PQ=(1-\delta)(Q-c)\,. \tag{36}$$

With the state process lifted in this way, the *LS-SARSA algorithm with exploration* is essentially the algorithm constructed in the previous section, with $\boldsymbol{X}$ replaced by $\mathbf{\Phi}$. The only difference is the different parameterization (4).

We arrive at the sequence of approximations, $\hat{\theta}_T=\Xi_T^{-1}b_T$, where the definition is a variation on (30),

$$b_T:=\frac{1}{T}\sum_{t=1}^{T}\big(\psi_g(t)-\psi(\widetilde{\Phi}(t))\big)c(\widetilde{\Phi}(t)) \tag{37}$$

and the positive matrices $\{\Xi_t:t\ge0\}$ are generated as before by the recursion,

$$\Xi_{t+1}=\Xi_t+\frac{1}{t+1}\big(\psi(\widetilde{\Phi}(t+1))\psi(\widetilde{\Phi}(t+1))^{\intercal}-\Xi_t\big)\,,\quad t\ge0\,.$$

V. Numerical Examples

We now present a examples to illustrate these methods. For purposes of evaluating an approximation we consider two types of Bellman error:

$$\mathcal{E}^\theta(x) = h^\theta(x)-\big(c_\phi(x)+\beta P_\phi\,h^\theta(x)\big) \tag{38}$$

$$\mathcal{E}_\circ^\theta(x) = \min_u\big[h^\theta(x)-\big(c(x,u)+\beta P_u\,h^\theta(x)\big)\big] \tag{39}$$

If θ optimizes (23), then we expect that the first error will be small on the support of $\tilde{\pi}$, in a mean-square sense. The error $\mathcal{E}_\circ^\theta$ will be small only if h^θ approximates h_β^*.

A. LQR with exploration

Consider first a deterministic linear model on $\mathbb{R}^2$, without control,

$$X(t+1)=AX(t),\qquad t\ge0$$

We take a purely quadratic cost function $c\colon\mathbb{R}^2\to\mathbb{R}_+$. To impose stability we assume that the eigenvalues of A are strictly within the unit circle in $\mathbb{C}$. In this case, the assumptions for Sec. III are not quite satisfied, but the *transformed* chain will be V-uniformly ergodic, with $V=1+\|x\|^2$, provided μ admits a second moment (i.e., $\mu(V)<\infty$).

The value function for the deterministic model is of the form $h_\beta(x)=c_\beta+x^{\intercal}Qx$, with $Q\ge0$. There is not theory to support the application of standard TD learning to approximate h_β because $\pi=\delta_0$. The value of exploration in this case is clear: Shown in Figure 1 is a comparison of a sample path of $\boldsymbol{X}$ for a particular example in which A has complex eigenvalues within the unit circle. The trajectory spirals to the origin exponentially quickly. The dashed-line trajectory shows a sample path of $\widetilde{\boldsymbol{X}}$, where μ is uniformly distributed on $[-4,4]\times[-4,4]$, and $\delta(x)=\delta=0.1$.

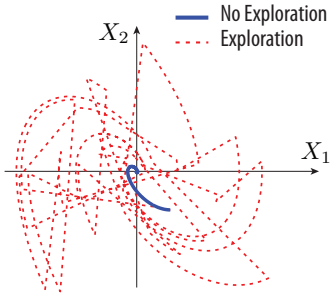

Fig. 1: *Sample paths of $\boldsymbol{X}$ and $\widetilde{\boldsymbol{X}}$ under LSTD with exploration.*

Next, we consider an example to illustrate the application of LS-SARSA with exploration. Consider the scalar, deterministic model with control, $X(t+1)=0.8X(t)+U(t)$. The discounted total-cost criterion (1) is considered, with $c(x,u)=\frac{1}{2}(x^2+u^2)$, and discount factor $\beta=0.9$. The optimal policy is static gain feedback; for any such policy, there is a linear relationship between $X(t)$ and $U(t)$, for all t, which rules out application of standard TD learning algorithms.

The exploration procedure was defined as follows. Whenever $(X(t),U(t))$ lies in the region $[-0.2,0.2]\times[-0.2,0.2]$, the process regenerates at time $t+1$, with probability $\delta_0>0$; the regeneration distribution μ is uniform on the square annulus, $[-1,1]\times[-1,1]\setminus[-0.2,0.2]\times[-0.2,0.2]$.

The LS-SARSA algorithm with exploration was run using an ideal basis, $\{\psi_i\}=\{x^2,xu,u^2\}$. An initial policy was chosen as linear state feedback, with $K=A/18$. After each run, a new policy was obtained using approximate policy iteration, via (8).

Figure 2 shows results from two experiments. The density plots and Bellman error plots shown were obtained in the third iteration of the algorithm; the policy was updated three times. The density for $\tilde{\pi}$ is the steady-state distribution for $\tilde{\mathbf{\Phi}}$, and the Bellman error is defined in (5), using θ obtained at the conclusion of the third run.

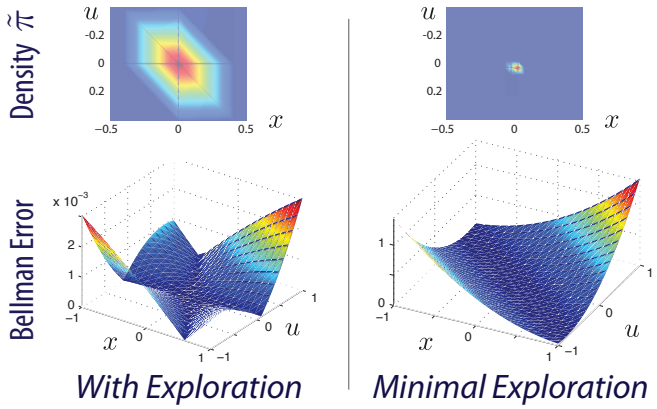

Fig. 2: *LS-SARSA for the scalar LQR model in two experiments, differentiated by the level of exploration.*

The column denoted "With Exploration" is based on $\delta_0 = 0.1$, and "Minimal Exploration" is $\delta_0 = 10^{-4}$. When using exploration, the Bellman error was less that 3×10^{-3} over the range shown. The algorithm failed to converge in the case of minimal exploration.

B. Dynamic speed scaling

We consider here the example of [4], based on the controlled-queue models considered in [5]. In the latter reference, the controlled queue is meant to model *dynamic speed scaling*, which is an approach to power management in computer system design. The goal is to control processing speed so as to optimally balance energy and delay costs – reducing (increasing) the speed in times when the workload is small (large). However, the general model has many applications. In particular, speed scaling approaches can be applied in wireless applications (see the aforementioned references, and the recent work [6] for an investigation of the analysis of the computation of "cost" in these applications).

The model is a simple controlled random walk (the CRW model of [13]),

$$X(t+1) = X(t) - U(t) + A(t+1), \qquad t \ge 0, \tag{40}$$

in which $\boldsymbol{X}$, $\boldsymbol{U}$, $\boldsymbol{A}$ each evolve on $\mathbb{R}_+$, and $\boldsymbol{A}$ is i.i.d.: Its marginal is supported on $\mathbb{R}_+$, with finite mean α, and variance σ_A^2.

The state $X(t)$ represents 'workload' in a processor, or packets in a communication buffer, and $U(t)$ is the processing rate at time t. Both evolve on $\mathbb{R}_+$: That is, $\mathsf{X} = \mathsf{U} = \mathbb{R}_+$. In addition, we have $\mathsf{U}(x) = [0, x]$ for each x. In computer system applications the following cost function is well motivated,

$$c(x, u) = x + \nu u^2\,, \tag{41}$$

where $\nu > 0$. This is meant to balance the cost of unfinished work, with the cost in terms of power for high processing rates.

The associated fluid model is defined by $\dot{x} = u - \alpha$, where α denotes the mean of $A(t)$. The value function (11), minimized over all $\boldsymbol{u}$, is derived in [4], in the special case $\alpha = 0$ and $\gamma = 0$. It has the simple form,

$$J_0^*(x) = kx^{3/2}\,, \qquad k = 4/(3\sqrt{\nu})\,. \tag{42}$$

This solves the DP equation $\min_u\{c(x,u) + \mathcal{D}_u^{\mathsf{F}} J_0^*(x)\} = 0$, $x \ge 0$, where $\mathcal{D}_u^{\mathsf{F}} h(x) = -u\frac{d}{dx}h(x)$, for any $h\colon \mathbb{R} \to \mathbb{R}$. It is shown in [4] that $J_0^*(x)$ is a tight approximation to the ACOE for the stochastic model, under general assumptions on $\boldsymbol{A}$.

When $\gamma > 0$, we do not have a closed form expression for the optimal value function. However, we can show that the function below is a tight approximation,

$$\begin{aligned} J_\gamma^*(x) &:= \min \int_0^\infty e^{-\gamma t} c(x(t), u(t)) dt \\ &\approx J_\gamma(x) := \frac{x}{\gamma}(1 - \exp(-\gamma k\sqrt{x})) \end{aligned} \tag{43}$$

where k is defined in (42). Observe that $J_\gamma(x) \to J_0^*(x)$ as $\gamma \downarrow 0$, uniformly on any bounded interval.

In view of these results we choose a three dimensional basis for approximation: $h_\theta = \theta_1\psi_1 + \theta_2\psi_2 + \theta_3\psi_3$ for $\theta \in \mathbb{R}^3$, with

$$\psi_1 = x, \quad \psi_2(x) = x\exp(-\gamma k\sqrt{x}), \quad \psi_3 = 1. \tag{44}$$

We take $\gamma = (1-\beta)/\beta$, as explained above (11). The value function approximation given in (43) can be expressed as a linear combination of $\{\psi_1, \psi_2\}$. The constant is chosen because the value function h_β grows with increasing β. If the controlled chain satisfies the assumptions for Sec. III, then $\lim_{\beta\to 1}(1-\beta)h_\beta(x) = \pi(c)$ for each x.

The marginal of $\boldsymbol{A}$ was chosen to be the scaled geometric distribution of the form used in [4], with the scaling factor denoted by Δ_A. For exploration we consider μ to be uniformly distributed on $[5, 15]$ and $\delta(x) \equiv \delta = 0.2$ (we did not consider state-dependent regeneration rates). The control $U(t)$ is taken to be integral multiple of Δ_A and is saturated to lie in $\mathcal{U} = \{0, \Delta_A, \cdots, 30\}$.

For any function h, and any stationary policy ϕ, the function Ph is given as follows

$$Ph\,(x) = \sum_{i=0}^{\infty} h([x - \phi(x) + \Delta_A i])p_A^i(1 - p_A), \tag{45}$$

where, $[\cdot]$ denotes projection on $\mathbb{R}_+$. For the purposes of computation, the above sum will be truncated to $i \le N_{\text{approx}}$. In all of our experiments we took $N_{\text{approx}} = 100$, $\Delta_A = 1/24$, and $p_A = 0.96$.

To illustrate the importance of exploration, we choose for the initial policy $U^0(t) = X(t)$. That is, $\phi_0(x) = x$. This results in a minimal state process, $X(t) = A(t)$ for $t \ge 1$, but is very costly for non-zero ν: The resulting value function (1) is quadratic,

$$h^0(x) = x + \nu x^2 + \frac{\beta}{1-\beta}(\alpha + \nu(v_A^2 + \alpha^2))$$

The optimal value function h_β^* for the discounted-optimal control problem has linear growth. It is easily shown that $x^{-1}h_\beta^*(x) \to (1-\beta)^{-1}$, as $x \to \infty$.

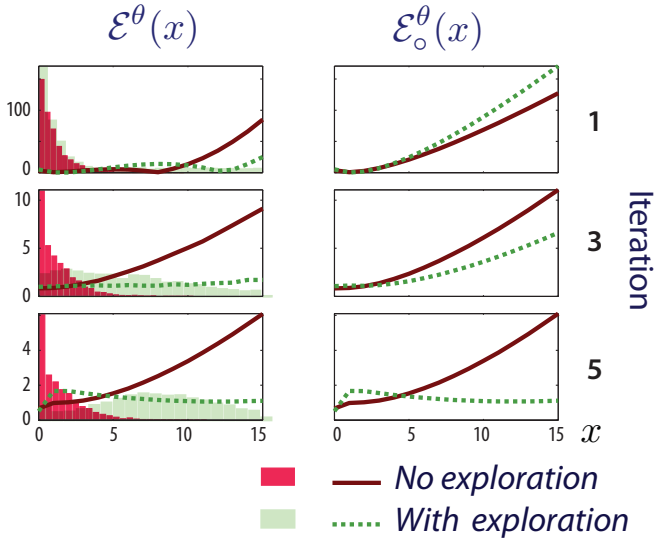

Fig. 3: *LSTD, with and without exploration, in the dynamic speed scaling model.*

The results of some of our experiments are illustrated in Figure 3. The three rows correspond to iterations 1, 3, and 5 of approximate policy iteration. With or without exploration, in this example we see rapid convergence, in the sense that the Bellman error (39) becomes small after just four or five iterations. The column on the left shows the pair of histographs for $\widetilde{\boldsymbol{X}}$ and $\boldsymbol{X}$, obtained respectively with and without exploration. Exploration led to a distribution with broader support. The column on the left also shows the policy-specific Bellman error (38): Exploration reduces this error significantly for larger values of x.

VI. Conclusions

We have shown how a simple restart mechanism leads to a new version of TD-learning that allows for exploration. In particular, value function approximation for a given policy is possible even for deterministic systems for which the state and action sequence converge to an equilibrium.

There are many open questions. We are interested in performance bounds for approximate policy iteration for TD-learning or SARSA with exploration. In the case of SARSA with exploration, we believe that the regenerative structure of the chain $\{\tilde{\boldsymbol{\Phi}}(t), I(t), \psi_g(t)\}$ will lead to simple characterization of ergodicity, but this requires further study. Also, we believe that this regenerative structure will simplify extension to the average-cost value function approximation, perhaps leading to algorithms with reduced variance as compared to the TD-learning algorithm introduced in Ch. 11 of [13].

We are also considering various applications of these techniques, to control and to anomaly detection.

References

[1] Dimitri P. Bertsekas and Huizhen Yu. Q-learning and enhanced policy iteration in discounted dynamic programming. In *49th IEEE Conference on Decision and Control (CDC)*, pages 1409 –1416, 2010.

[2] D.P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming*. Atena Scientific, Cambridge, Mass, 1996.

[3] V. S. Borkar. *Stochastic Approximation: A Dynamical Systems Viewpoint*. Hindustan Book Agency and Cambridge University Press (jointly), Delhi, India and Cambridge, UK, 2008.

[4] Wei Chen, Dayu Huang, Ankur A. Kulkarni, Jayakrishnan Unnikrishnan, Quanyan Zhu, Prashant Mehta, Sean Meyn, and Adam Wierman. Approximate dynamic programming using fluid and diffusion approximations with applications to power management. In *Proc. of the 48th IEEE Conf. on Dec. and Control*, pages 3575–3580, 2009.

[5] Jennifer M. George and J. Michael Harrison. Dynamic control of a queue with adjustable service rate. *Operations Research*, 49(5):720–731, September 2001.

[6] P. Grover, K.A. Woyach, and A. Sahai. Towards a communication-theoretic understanding of system-level power consumption. Arxiv preprint arXiv:1010.4855. Submitted to IEEE J. on Selected Areas in Communication, 2010.

[7] D. Huang, W. Chen, S. Mehta, P. Meyn, and A. Surana. Feature selection for neuro-dynamic programming. In F. Lewis, editor, *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*. Wiley, 2011.

[8] M. Huang, P. E. Caines, and R. P. Malhame. Large-population cost-coupled LQG problems with nonuniform agents: Individual-mass behavior and decentralized ε-Nash equilibria. *IEEE Trans. Automat. Control*, 52(9):1560–1571, 2007.

[9] I. Kontoyiannis and S. P. Meyn. Spectral theory and limit theorems for geometrically ergodic Markov processes. *Ann. Appl. Probab.*, 13:304–362, 2003. Presented at the INFORMS Applied Probability Conference, NYC, July, 2001.

[10] I. Kontoyiannis and S. P. Meyn. Large deviations asymptotics and the spectral theory of multiplicatively regular Markov processes. *Electron. J. Probab.*, 10(3):61–123 (electronic), 2005.

[11] S. Mannor, I. Menache, and N. Shimkin. Basis function adaptation in temporal difference reinforcement learning. *Annals of Oper. Res.*, 134(2):215–238, 2005.

[12] P. G. Mehta and S. P. Meyn. Q-learning and Pontryagin's minimum principle. In *Proc. of the 48th IEEE Conf. on Dec. and Control*, pages 3598–3605, Dec. 2009.

[13] S. P. Meyn. *Control Techniques for Complex Networks*. Cambridge University Press, Cambridge, 2007. Pre-publication edition available online.

[14] S. P. Meyn and R. L. Tweedie. *Markov chains and stochastic stability*. Cambridge University Press, Cambridge, second edition, 2009. Published in the Cambridge Mathematical Library. 1993 edition online.

[15] Sean Meyn, Wei Chen, and Daniel O'Neill. Optimal cross-layer wireless control policies using td learning. In *Proc. of the 49th IEEE Conf. on Dec. and Control*, pages 1951 –1956, 2010.

[16] C.C. Moallemi, S. Kumar, and B. Van Roy. Approximate and data-driven dynamic programming for queueing networks. Submitted for publication., 2006.

[17] E. Nummelin. *General Irreducible Markov Chains and Nonnegative Operators*. Cambridge University Press, Cambridge, 1984.

[18] G. A. Rummery and M. Niranjan. On-line Q-learning using connectionist systems. Technical report 166, Cambridge Univ., Dept. Eng., Cambridge, U.K. CUED/F-INENG/, 1994.

[19] S. Shirodkar and S. Meyn. Quasi stochastic approximation. In *Proc. of the 2011 American Control Conference (ACC)*, pages 2429–2435, July 2011.

[20] S. Singh, T. Jaakkola, M. L. Littman, and C. Szepesvári. Convergence results for single-step on-policy reinforcement-learning algorithms. *Mach. Learn.*, 38:287–308, 2000.

[21] C. Szepesvári. *Algorithms for Reinforcement Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2010.

[22] J. N. Tsitsiklis and B. Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE Trans. Automat. Control*, 42(5):674–690, 1997.

[23] C. J. C. H. Watkins. *Learning from Delayed Rewards*. PhD thesis, King's College, Cambridge, UK, 1989.

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

[25] H. Yu and D. P. Bertsekas. Q-learning algorithms for optimal stopping based on least squares. In *Proc. European Control Conference (ECC)*, July 2007.