SIAM J. CONTROL OPTIM.
Vol. 40, No. 3, pp. 681–698

LEARNING ALGORITHMS FOR MARKOV DECISION PROCESSES WITH AVERAGE COST*

J. ABOUNADI†, D. BERTSEKAS†, AND V. S. BORKAR‡

Abstract. This paper gives the first rigorous convergence analysis of analogues of Watkins's Q-learning algorithm, applied to average cost control of finite-state Markov chains. We discuss two algorithms which may be viewed as stochastic approximation counterparts of two existing algorithms for recursively computing the value function of the average cost problem—the traditional relative value iteration (RVI) algorithm and a recent algorithm of Bertsekas based on the stochastic shortest path (SSP) formulation of the problem. Both synchronous and asynchronous implementations are considered and analyzed using the ODE method. This involves establishing asymptotic stability of associated ODE limits. The SSP algorithm also uses ideas from two-time-scale stochastic approximation.

Key words. simulation-based algorithms, Q-learning, controlled Markov chains, average cost control, stochastic approximation, dynamic programming

AMS subject classification. 93E20

PII. S0363012999361974

1. Introduction.

Q-learning algorithms are simulation-based reinforcement learning algorithms for learning the value function arising in the dynamic programming approach to Markov decision processes. They were first introduced for the discounted cost problem by Watkins [27] and analyzed partially in Watkins [27] and then in Watkins and Dayan [28]. A more comprehensive analysis was given by Tsitsiklis [25] (also reproduced in Bertsekas and Tsitsiklis [7]), which made the connection between Q-learning and stochastic approximation. (See also Jaakola, Jordan, and Singh [15] for a parallel treatment, which made the connection between TD(λ) and stochastic approximation.) In particular, Q-learning algorithms for discounted cost problems or stochastic shortest path (SSP) problems were viewed as stochastic approximation variants of well-known value iteration algorithms in dynamic programming.

These techniques, however, do not extend automatically to the average cost problem, which is harder to analyze even when the model (i.e., controlled transition probabilities) is readily available. Not surprisingly, the corresponding developments for the average cost problem have been slower. One of the first was the "R-learning" algorithm proposed by Schwartz [22]. This is a two-time-scale algorithm that carries out a value iteration-type step to update values of state-action pairs and updates concurrently an estimate of the optimal average cost using the immediate reward along with an adjustment factor. The idea is to obtain a good estimate for the average cost while searching for the optimal policy using a value iteration-type update. Although Schwartz presents some intuitive arguments to justify his algorithm along with some

*Received by the editors September 29, 1999; accepted for publication (in revised form) March 14, 2001; published electronically September 7, 2001.
http://www.siam.org/journals/sicon/40-3/36197.html

†Laboratory for Information and Decision Systems, M.I.T., 77 Massachusetts Avenue, Cambridge, MA 02139 (jinane@mit.edu, dimitrib@mit.edu). The work of these authors was supported by NSF under grant 9600494-DMI and grant ACI-9873339.

‡School of Technology and Computer Science, Tata Institute of Fundamental Research, Homi Bhabha Road, Mumbai 400005, India (borkar@tifr.res.in). The work of this author was supported in part by the US Army grant PAAL 03-92-G-0115 at M.I.T., in part by the Homi Bhabha Fellowship, and by the Government of India, Department of Science and Technology grant No. III 5(12)/96-ET.

numerical results, he does not provide any convergence analysis. Singh [23] presents two Q-learning algorithms for the average cost problem: one is a slight modification of Schwartz's algorithm which updates the estimate of optimal cost at every step. The other one updates the estimate of average cost in a fashion similar to Jalali and Ferguson's deterministic asynchronous algorithm for average cost problems [16]. He provides simulation results for medium-sized problems but no convergence analysis. Finally, Mahadevan [20] discusses average cost problems and the need to consider the average cost criterion, with an emphasis on the difference between gain-optimal and bias-optimal policies. He presents extensive numerical experiments, highlighting the problems the algorithm can run into. He does not, however, provide any convergence analysis. It is also noteworthy that none of these algorithms use the relative value iteration (RVI) algorithm for average cost problems (see, e.g., [4], [21], [24]) as a basis for the learning algorithms because the latter may not converge asynchronously, as shown in [3]. Nevertheless, a diminishing stepsize does work around this problem, as we show in this paper.

We propose and give for the first time a complete convergence analysis of two Q-learning algorithms for average cost. The first is a stochastic approximation analogue of (RVI). The second is a stochastic approximation analogue of a recent value iteration algorithm of Bertsekas based on the SSP formulation of the average cost problem. We consider both synchronous and asynchronous implementations. The analysis relies on the ODE method, based on establishing first the boundedness of iterates and then the asymptotic stability of limiting ODEs. The rest then follows as in the Kushner–Clark approach [18] (see also Kushner and Yin [19]) in the synchronous case and by Borkar's theorem [10] in the asynchronous case.

The paper is organized as follows. The next section describes the two algorithms in both synchronous and asynchronous modes and states the assumptions required in each case. Section 3 provides the convergence analysis of the RVI-based Q-learning algorithm. Section 4 does likewise for the SSP Q-learning algorithm. Section 5 concludes with some general remarks. An Appendix collects some key facts from the literature that we used here.

2. Average cost Q-learning algorithms.

2.1. Preliminaries. We consider a controlled Markov chain $\{X_n\}$ on a finite state space $S = \{1, 2, \ldots, d\}$ with a finite action space $A = \{a_1, \ldots, a_r\}$ and transition probabilities $p(i, a, j) =$ the probability of transition from i to j under action a for $i, j \in S, a \in A$. Associated with this transition is a "cost" $g(i, a, j)$ and the aim is to choose actions $\{Z_n\}$ nonanticipatively (i.e., conditionally independent of the future state trajectory given past states and actions) so as to minimize the "average cost"

$$\limsup_{n\to\infty} \frac{1}{n} \sum_{m=0}^{n-1} E[g(X_m, Z_m, X_{m+1})]. \tag{2.1}$$

This problem is extensively treated in [4], [21], and [24] among others, to which we refer the reader for details. We recall here the minimal necessary background material required to motivate our algorithms.

We shall be interested in "stationary policies" wherein $Z_n = v(X_n)$ for a map $v : S \to A$. It is known that an optimal one exists under the following "unichain" condition which we assume throughout.

ASSUMPTION 2.1. *Under any stationary policy, the chain has a single communicating class and a common state (say, s) that is recurrent.*

In particular, this implies that "limsup" in (2.1) is a limit under any stationary policy. It is known that one can then associate a "value function" $V : S \to R$ with the problem, given as the solution to the dynamic programming equations

$$V(i) = \min_a \left[\sum_j p(i,a,j)(g(i,a,j) + V(j)) - \beta \right], \qquad i \in S, \tag{2.2}$$

where β is the optimal cost. $V(\cdot)$ is the unique solution to (2.2) modulo an additive constant. Let $Q(i,a)$ denote the expression inside the square brackets on the right-hand-side (r.h.s.) of (2.2). Equation (2.2) is useful because of the following fact: A stationary policy $v : S \to A$ is optimal if and only if $v(i) \in \text{Argmin } (Q(i,\cdot))$ for all i that are recurrent under v. $Q(\cdot,\cdot)$ is called the "Q-factor," also defined uniquely only up to an additive constant. Thus $V(i) = \min_a Q(i,a)$ for all i, and $Q(\cdot,\cdot)$ satisfies

$$Q(i,a) = \sum_j p(i,a,j) \left(g(i,a,j) + \min_b Q(j,b) \right) - \beta, \qquad i \in S, \quad a \in A. \tag{2.3}$$

The aim of any Q-learning algorithm is to "learn" the Q-factors when $p(\cdot,\cdot,\cdot)$ is not known, but one has access to a simulation device that can generate an independent S-valued random variable (i.e., independent of other random variables that might have been generated up to that point in time) ξ_{ia} whose probability law is $p(i,a,\cdot), i \in S, a \in A$. Let $\xi = [\xi_{ia}]$.

2.2. RVI Q-learning. The RVI algorithm is given by (see, e.g., [4], [21], [24])

$$V^{n+1}(i) = \min_a \left[\sum_j p(i,a,j)(g(i,a,j) + V^n(j)) - V^n(i_0) \right], \qquad i \in S, \tag{2.4}$$

where $i_0 \in S$ is an arbitrary but fixed state. This can be shown, under some additional aperiodicity conditions (see [4, Chap. 4]), to converge to the unique $V(\cdot)$ that satisfies (2.2) with $V(i_0) = \beta$. The purpose of subtracting the scalar "offset" $V^n(i_0)$ from each component on the r.h.s. of (2.4) is to keep the iterations stable—recall that $V(\cdot)$ is specified anyway only up to an additive constant. It turns out that $V^n(i_0) \to \beta$. However, $V^n(i_0)$ is not the unique choice of an offset term that makes the algorithm work. More generally, one can replace it by $f(V)$ for an $f : R^d \to R$ satisfying suitable hypotheses. (See below.)

Algorithm (2.4) suggests the "relative Q-factor iteration"

$$Q^{n+1}(i,a) = \sum_j p(i,\alpha,j) \left(g(i,a,j) + \min_b Q^n(j,b) \right) - Q^n(i_0,a_0), \qquad i \in S, \qquad a \in A,$$

with $i_0 \in S, a_0 \in A$ prescribed. The idea behind RVI Q-learning is to replace the conditional average with respect to the transition probability $p(i,a,\cdot)$ by an actual evaluation at a random variable ξ_{ia} with law $p(i,a,\cdot)$ and then "see" the conditional average by means of the well-known averaging effect of the stochastic approximation algorithm. Thus the synchronous RVI Q-learning algorithm is

$$Q^{n+1}(i,a) = Q^n(i,a) + \gamma(n)\Big(g(i,a,\xi^n_{ia}) + \min_b Q^n(\xi^n_{ia},b) - f(Q^n) - Q^n(i,a)\Big), \qquad i \in S, \qquad a \in A, \tag{2.5}$$

where ξ^n_{ia} are independent with the law of ξ^n_{ia} being $p(i,a,\cdot)$ for all n. $\{\gamma(k)\} \in (0,\infty)$ is the usual diminishing stepsize schedule of stochastic approximation satisfying

$$\sum_k \gamma(k) = \infty, \quad \sum_k \gamma^2(k) < \infty. \tag{2.6}$$

The function $f : R^{d\times r} \to R$ satisfies the following assumption.

ASSUMPTION 2.2. *f is Lipschitz, and, furthermore, for e equal to the constant vector of all 1's in $R^{d\times r}$, $f(e) = 1$ and $f(x + ce) = f(x) + c$ for $c \in R$.*

Examples are $f(Q) = Q(i_0,b_0)$ for prescribed i_0,b_0, $f(Q) = \min_u Q(i_0,u)$ for prescribed i_0, $f(Q) = \frac{1}{dr}\sum_{i,a} Q(i,a)$, and so on.

For the asynchronous algorithm, we hypothesize a set-valued process $\{Y^n\}$ taking values in the set of nonempty subsets of $S \times A$ with the interpretation: $Y^n = \{(i,a) :$ (i,a)th component of Q was updated at time $n\}$. (Thus $Y^n \equiv S\times A$ is the synchronous case.)

Remarks. As argued in [10], we may take $Y^n = \{\phi^n\}$ for some $\phi^n \in S \times A$, i.e., a singleton. This can be achieved by unfolding a single iteration that updates k components into k iterations that update one component each. While this introduces "delays" in the formulation of the algorithm below, that does not affect the results of [10] that we use here. Alternatively, we may use the results of [17, section 4], which work with the Y^n's directly. The only difference is that the resultant ODE is a time-scaled version of the one arising in the former approach with a nonautonomous time-scaling which, however, does not affect its qualitative behavior.

Define

$$\nu(n,i,a) = \sum_{k=0}^{n} I\{(i,a) \in Y^k\},$$

where $I\{\ldots\}$ is the indicator function. Thus $\nu(n,i,a) =$ the number of times $Q^m(i,a)$ was updated up to time n.

The asynchronous RVI Q-learning algorithm then is

$$Q^{n+1}(i,a) = Q^n(i,a) + \gamma(\nu(n,i,a))\Big(g(i,a,\xi^n_{ia}) + \min_u Q^n(\xi^n_{ia},u) - f(Q^n) - Q^n(i,a)\Big) I\{(i,a) \in Y^n\} \tag{2.7}$$

for $(i,a) \in S \times A$. For the asynchronous case, we need the following additional assumptions.

ASSUMPTION 2.3. *In addition to* (2.6), *$\{\gamma(n)\}$ satisfy the following: If $[\ldots]$ stands for "the integer part of ...," then for $x \in (0,1)$,*

$$\sup_k \gamma([xk])/\gamma(k) < \infty,$$

and

$$\frac{\sum_{m=0}^{[yk]} \gamma(m)}{\sum_{m=0}^{k} \gamma(m)} \to 1 \quad \textit{uniformly in } y \in [x,1].$$

Examples of stepsizes satisfying Assumption 2.3 are $\gamma(n) = \frac{1}{n}, \frac{1}{n \log n}, \frac{\log n}{n}$, etc., for $n \geq 2$.

ASSUMPTION 2.4. *There exists* $\Delta > 0$ *such that*

$$\liminf_{n \to \infty} \frac{\nu(n,i,a)}{n+1} \geq \Delta \text{ a.s.}, \qquad (i,a) \in S \times A.$$

Furthermore, for all $x > 0$ *and*

$$N(n,x) = \min \left\{ m \geq n : \sum_{k=n}^{m} \gamma(k) \geq x \right\},$$

the limit

$$\lim_{n \to \infty} \frac{\sum_{k=\nu(n,i,a)}^{\nu(N(n,x),i,a)} \gamma(k)}{\sum_{k=\nu(n,j,u)}^{\nu(N(n,x),j,u)} \gamma(k)}$$

exists a.s. for all i, j, a, u.

That is, all components are updated comparably often in an evenly distributed manner.

2.3. SSP Q-learning. SSP Q-learning is based on the observation that the average cost under any stationary policy is simply the ratio of expected total cost and expected time between two successive visits to the reference state s. This connection was exploited by Bertsekas in [5] to give a new algorithm for computing $V(\cdot)$, which we describe below.

Define a parametrized family of SSP problems parametrized by a scalar λ as follows.

(i) The state space is $S' = S \cup \{s'\}$, where s' is an artificially added terminal state (i.e., zero-cost and absorbing).

(ii) The action set is A for all states.

(iii) The transition probabilities are

$$p'(i,a,j) = \begin{cases} p(i,a,j) & \text{if} \quad j \neq s, s', \\ 0 & \text{if} \quad j = s, \\ p(i,a,s) & \text{if} \quad j = s'. \end{cases}$$

(iv) The costs are defined by

$$g'(i,a,j) = \begin{cases} g(i,a,j) - \lambda & \text{if} \quad j \neq s, s', \\ 0 & \text{if} \quad j = s, \\ g(i,a,s) - \lambda & \text{if} \quad j = s'. \end{cases}$$

By Assumption 2.1, s' is reached from every state with probability 1. Thus all policies are proper (as defined in [4]). Let $V_\lambda(\cdot)$ denote the value function given as the unique solution to the dynamic programming equations

$$V_\lambda(i) = \min_a \left[\sum_{j=1}^{d} p(i,a,j) \left(g(i,a,j) + \sum_{j \neq s} p(i,a,j) V_\lambda(j) \right) - \lambda \right], \quad 1 \leq i \leq d,$$

$$V_\lambda(s') = 0.$$

For each fixed policy, the cost is linear in λ with negative slope. Thus $V_\lambda(\cdot)$, which by standard dynamic programming arguments is the lower envelope thereof, is piecewise linear with finitely many linear pieces and concave decreasing in λ for each component. If $\lambda = \beta$, we "recover" (2.2), which can be shown to happen when $V_\lambda(s) = 0$. This suggests the coupled iterations

$$V^{k+1}(i) = \min_a \left[\sum_{j=1}^{d} p(i,a,j) \left(g(i,a,j) + \sum_{j \neq s} p(i,a,j) V^k(j) \right) - \lambda^k \right], \qquad i \in S,$$
$$\lambda^{k+1} = \lambda^k + b(k) V^k(s),$$

where $\{b(k)\} \subset (0,\infty)$ with $\sum_k b(k) = \infty$ and $\sum_k b^2(k) < \infty$. This is the algorithm of [5], wherein the first "fast" iteration sees λ^k as quasi-static ($b(k)$'s are "small") and thus tracks $V_{\lambda^k}(\cdot)$, while the second "slow" iteration gradually guides λ^k to the desired value.

This suggests the SSP Q-learning algorithm (synchronous) as

(2.8a)
$$Q^{n+1}(i,a) = Q^n(i,a) + \gamma(n) \left[(g(i,a,\xi^n_{ia}) + \min_u Q^n(\xi^n_{ia},u)) I\{\xi^n_{ia} \neq s\} - \lambda^n - Q^n(i,a) \right],$$

(2.8b)
$$\lambda^{n+1} = \lambda^n + b(n) \min_u Q^n(s,u),$$

where $b(n) = o(\gamma(n))$. Unfortunately, it appears hard to ensure boundedness of $\{\lambda^n\}$. So we propose replacing (2.8b) by

(2.8b′)
$$\lambda^{n+1} = \Gamma \left(\lambda^n + b(n) \min_u Q^n(s,u) \right),$$

where $\Gamma(\cdot)$ is the projection onto an interval $[-K,K]$ with K chosen so that $\beta \in (-K,K)$. (This assumes prior knowledge of a bound on β, but this can be obtained from a bound on $g(\cdot,\cdot,\cdot)$.)

As in the case of RVI Q-learning, we impose Assumptions 2.3 and 2.4 for the asynchronous SSP Q-learning, which is

$$Q^{n+1}(i,a) = Q^n(i,a) + \gamma(\nu(n,i,a)) \left[\left(g(i,a,\xi^n_{ia}) + \min_u Q^n(\xi^n_{ia},u) I\{\xi^n_{ia} \neq s\} \right) \right.$$

(2.9a)
$$\left. -\lambda^n - Q^n(i,a) \right] I\{(i,a) \in Y^n\},$$

(2.9b)
$$\lambda^{n+1} = \Gamma \left(\lambda^n + b(n) \min_u Q^n(s,u) \right).$$

3. Convergence of RVI Q-learning.

3.1. ODE analysis. We can rewrite the synchronous RVI Q-learning algorithm (2.5) as

(3.1)
$$Q^{n+1} = Q^n + \gamma(n)(T(Q^n) - f(Q^n)e - Q^n + M^{n+1}),$$

where Q^n stands for $Q^n(i,a)$, $T : R^{d \times r} \to R^{d \times r}$ is the map defined by

$$(TQ)(i,a) = \sum_j p(i,a,j) \left(g(i,a,j) + \min_u Q(j,u) \right),$$

and, for $n \geq 0$,

$$M^{n+1}(i,a) = g(i,a,\xi_{ia}^n) + \min_u Q^n(\xi_{ia}^n, u) - (TQ^n)(i,a).$$

Letting $\mathcal{F}_n = \sigma(Q^m, M^m, m \leq n), n \geq 0$, we note that, for all n,

$$E[M^{n+1} \mid \mathcal{F}_n] = 0, \tag{3.2}$$

$$E[||M^{n+1}||^2 \mid \mathcal{F}_n] \leq C_1(1 + ||Q^n||^2) \tag{3.3}$$

for a suitable constant $C_1 > 0$.

Define $\hat{T} : R^{d\times r} \to R^{d\times r}, T' : R^{d\times r} \to R^{d\times r}$ by

$$\begin{aligned} \hat{T}(Q) &= T(Q) - \beta e, \\ T'(Q) &= T(Q) - f(Q)e = \hat{T}(Q) + (\beta - f(Q))e, \end{aligned}$$

where, as before, $e \in R^{d\times r}$ is the constant vector of all 1's.

Let $||x||_\infty = \max_{i,a} |x_{ia}|, ||x||_s = \max_{i,a} x_{ia} - \min_{i,a} x_{ia}$ for $x \in R^{d\times r}$. These are, respectively, the max-norm and the span seminorm, the latter having the property that $||x||_s = 0$ if and only if x is a scalar multiple of e. The following "nonexpansivity" properties are then easily verified:

$$||T(Q) - T(Q')||_\infty \leq ||Q - Q'||_\infty,$$

and likewise for $\hat{T}(\cdot)$. Also,

$$||T(Q) - T(Q')||_s \leq ||Q - Q'||_s,$$

and likewise for $\hat{T}(\cdot), T'(\cdot)$. In fact, $||T(Q)||_s = ||T'(Q)||_s = ||\hat{T}(Q)||_s$ since $||e||_s = 0$.

Algorithm (3.1) is in the form of a standard stochastic approximation algorithm with the martingale difference sequence $\{M^{n+1}\}$ serving as the "noise." The ODE approach to analyzing the convergence of such algorithms (described in [2], [13], [18], and [19], among others) is based on the stability of the associated ODE

$$\dot{Q}(t) = T'(Q(t)) - Q(t). \tag{3.4}$$

This subsection is devoted to studying the stability properties of this ODE. We do so through a succession of lemmas. The analysis is inspired by a similar analysis in [9] in the context of value iteration. (See also [17].)

We shall also consider the related ODE

$$\dot{Q}(t) = \hat{T}(Q(t)) - Q(t). \tag{3.5}$$

Note that by the properties of $T(\cdot), f(\cdot)$, both (3.4) and (3.5) have Lipschitz r.h.s.'s and thus are well-posed.

The set G of equilibrium points of (3.5) is precisely the set of fixed points of $\hat{T}(\cdot)$, i.e., the solutions of (2.3) which are unique up to an additive constant. Thus $G = \{Q : Q = Q^* + ce, c \in R\}$, where Q^* is the solution to (2.3) satisfying $f(Q^*) = \beta$. (That there will indeed be one such solution follows from the fact that $f(x + ce) = f(x) + c$ for $c \in R$.)

The next lemma is a special case of Theorem 4.1 of [14] (see the appendix).

LEMMA 3.1. *Let $y(\cdot)$ and z be a solution and an equilibrium point of* (3.5)*, respectively. Then $||y(t)-z||_\infty$ is nonincreasing, and $y(t) \to y^*$ for some equilibrium point y^* of* (3.5) *that may depend on $y(0)$.*

We use this to analyze (3.4). But first note the following.

LEMMA 3.2. *Equation* (3.4) *has a unique equilibrium point at Q^*.*

Proof. Since $f(Q^*) = \beta$, it follows that $T'(Q^*) = \hat{T}(Q^*) = Q^*$; thus Q^* is an equilibrium point for (3.4). Conversely, if $T'(Q) = Q$, then $Q = \hat{T}(Q) + (\beta - f(Q))e$. But the Bellman equation

$$Q = \hat{T}(Q) + ce$$

has a solution if and only if $c = 0$. (This can be deduced from the corresponding statement for (2.2), which is well known, and the relation $V(i) = \min_u Q(i,u)$ modulo an additive constant.) Thus $f(Q) = \beta$, implying $Q = Q^*$. □

LEMMA 3.3. *Let $x(\cdot), y(\cdot)$ satisfy* (3.4) *and* (3.5)*, respectively, with $x(0) = y(0) = x_0$. Then $x(t) = y(t) + r(t)e$, where $r(\cdot)$ satisfies the ODE*

$$\dot{r}(t) = -r(t) + (\beta - f(y(t))).$$

Proof. By the variation of constants formula,

$$x(t) = x_0 e^{-t} + \int_0^t e^{-(t-s)} \hat{T}(x(s))ds + \left[\int_0^t e^{-(t-s)}(\beta - f(x(s)))ds\right] e,$$
$$y(t) = x_0 e^{-t} + \int_0^t e^{-(t-s)} \hat{T}(y(s)).$$

Therefore, with $\hat{T}_i(\cdot)$ denoting the ith component of $\hat{T}(\cdot)$,

$$\max_i (x_i(t) - y_i(t)) \le \int_0^t e^{-(t-s)} \max_i (\hat{T}_i(x(s)) - \hat{T}_i(y(s)))ds + \int_0^t e^{-(t-s)}(\beta - f(x(s)))ds,$$
$$\min_i (x_i(t) - y_i(t)) \ge \int_0^t e^{-(t-s)} \min_i (\hat{T}_i(x(s)) - \hat{T}_i(y(s)))ds + \int_0^t e^{-(t-s)}(\beta - f(x(s)))ds.$$

Subtracting, we have

$$\begin{aligned} ||x(t) - y(t)||_s &\le \int_0^t e^{-(t-s)} ||\hat{T}(x(s)) - \hat{T}(y(s))||_s ds \\ &\le \int_0^t e^{-(t-s)} ||x(s) - y(s)||_s ds. \end{aligned}$$

By Gronwall's inequality, $||x(t)-y(t)||_s = 0$ for all $t \ge 0$. Since $||x||_s = 0$ if and only if $x = ce$ for some $c \in R$, we have $x(t) = y(t) + r(t)e, t \ge 0$. Since $x(0) = y(0), r(0) = 0$. Since

$$\hat{T}(x + ce) = \hat{T}(x) + ce,$$
$$f(x + ce) = f(x) + c,$$

for $r \in R$ we have

$$\begin{aligned} \dot{r}(t)e &= \dot{x}(t) - \dot{y}(t) \\ &= (\hat{T}(x(t)) - x(t) + \beta - f(x(t))e) - (\hat{T}(y(t)) - y(t)) \\ &= (-r(t) + \beta - f(y(t)))e. \end{aligned}$$ □

THEOREM 3.4. *Q^* is the globally asymptotically stable equilibrium point for* (3.4).

Proof. By the variation of constants formula, in the foregoing,

$$r(t) = \int_0^t e^{-(t-s)}(\beta - f(y(s)))ds. \tag{3.6}$$

Let $y(t) \to y^* \in G$. Then $r(t) \to \beta - f(y^*)$ so that $x(t) \to y^* + (\beta - f(y^*))e$, which must coincide with Q^*, since that is the only equilibrium point for (3.4). To claim asymptotic stability, we also need to prove Liapunov stability. (That is, we need to show that given any $\epsilon > 0$, we can find a $\delta > 0$ such that $\|x(0) - Q^*\|_\infty < \delta$ implies $\|x(t) - Q^*\|_\infty < \epsilon$ for $t \geq 0$.) Now

$$\begin{aligned} ||x(t) - Q^*||_\infty &\leq ||y(t) - Q^*||_\infty + |r(t)| \\ &\leq ||y(0) - Q^*||_\infty + \int_0^t e^{-(t-s)}|\beta - f(y(s))|ds \\ &\leq ||x(0) - Q^*||_\infty + \int_0^t e^{-(t-s)}|f(Q^*) - f(y(s))|ds. \end{aligned} \tag{3.7}$$

Since $f(\cdot)$ is Lipschitz,

$$\begin{aligned} |f(Q^*) - f(y(s))| &\leq L||y(s) - Q^*||_\infty \\ &\leq L||y(0) - Q^*||_\infty \\ &= L||x(0) - Q^*||_\infty \end{aligned} \tag{3.8}$$

for a suitable $L > 0$. Thus

$$||x(t) - Q^*||_\infty \leq (1 + L)||x(0) - Q^*||_\infty.$$

Liapunov stability follows, completing the proof. □

3.2. Boundedness and convergence. The ODE method described variously in [2], [13], [18], [19], etc. immediately yields the following.

THEOREM 3.5. *In both the synchronous and the asynchronous Q-learning iterations (cf.* (2.5), (2.7)*), if $\{Q^n\}$ remain bounded a.s., then $Q^n \to Q^*$ a.s.*

Proof. The synchronous case follows from the standard ODE approach in view of Theorem 3.4. The asynchronous case follows likewise from the results of [10]. (In either case, given our prior assumptions, the only things left to verify are the a.s. boundedness of the iterates, which we simply *assumed* for the time being, and the global asymptotic stability of the associated ODE, which we just proved in the previous subsection.) □

The problem of proving a.s. boundedness remains. We shall indicate two proof approaches. The first, which works only for the synchronous case, is based on Lemma 2.2 of [1] (see Appendix). Note that by Theorem 3.4 and the converse Liapunov theorem [29], there exists a C^1 Liapunov function $V : R^{d\times r} \to R^+$ with $\lim_{||x||\to\infty} V(x) = \infty$ and

$$\langle \nabla V(x), T'(x) - x\rangle < 0, \quad x \neq Q^*.$$

Let B be an open neighborhood of Q^*, and let $C = \{x : V(x) \leq c\}$, where $c > 0$ is chosen sufficiently large so as to ensure that $B \subset$ interior (C). Note that C is compact. Define $\pi : R^{d\times r} \to C$ by

$$\pi(x) = x \qquad \text{if} \quad x \in C,$$

$$= Q^* + \eta(x)(x - Q^*) \quad \text{if} \quad x \notin C,$$

where $\eta(x) = \max\{a > 0 : Q^* + a(x - Q^*) \in B\}$. Consider the "scaled" version of (2.5) given by

$$\begin{aligned} \overline{Q}^{n+1}(i,a) = \tilde{Q}^n(i,a) + \gamma(n) \Big(& g(i,a,\xi^n_{ia}) + \min_u \tilde{Q}^n(\xi^n_{ia},u) \\ & -f(\tilde{Q}^n) - \tilde{Q}^n(i,a) \Big), \qquad i \in S, \quad a \in A, \end{aligned} \tag{3.9}$$

where $\tilde{Q}^n = \pi(\overline{Q}^n)$. The iterates (3.9) remain bounded a.s. by construction. To use Lemma 2.2 of [1], we need the following:

(i) The maps $x \to (1-\gamma(n))x + \gamma(n)T'(x)$ are nonexpansive with respect to $||\cdot||_s$ (where without any loss of generality we take $\gamma(n) < 1$). Note that they are so if $T'(\cdot)$ is, which it indeed is, as already observed.

(ii) The iterates $\{\overline{Q}^n\}$ converge to Q^* a.s., which, in view of Theorem 3.4, is proved exactly as in [1, section 3].

We shall also need the following additional assumption on $f(\cdot)$.

ASSUMPTION 3.6. $|f(Q)| \le ||Q||_\infty$ for all $Q \in R^{d\times r}$.

Note that this is satisfied by the examples of $f(\cdot)$ that follow Assumption 2.2.

LEMMA 3.7. *Under the additional Assumption* 3.6, $\{Q^n\}$ *given by the synchronous Q-learning iteration* 2.5 *is bounded a.s.*

Proof. In view of above remarks and Lemma 2.2 of [1], $||\overline{Q}^n - Q^n||_s$ remains bounded a.s. Note that

$$\sup_n ||Q^n||_s \le \sup_n ||\overline{Q}^n||_s + \sup_n ||Q^n - \overline{Q}^n||_s \triangleq K < \infty.$$

Let $D = \max(||Q^0||, \max_{i,a,j} |g(i,a,j)| + K)$. Then by Assumptions 2.2 and 3.6,

$$\begin{aligned} \left|\min_u Q^n(\xi^n_{ia},u) - f(Q^n)\right| &= \left|f\left(Q^n - \left(\min_u Q^n(\xi^n_{ia},u)\right)e\right)\right| \\ &\le \left|\left|Q^n - \left(\min_u Q^n(\xi^n_{ia},u)\right)e\right|\right|_\infty \\ &\le ||Q^n||_s \le K. \end{aligned}$$

Then

$$\begin{aligned} |Q^{n+1}(i,a)| &\le (1-\gamma(n))||Q^n||_\infty + \gamma(n)\left(\max_{i,a,j} g(i,a,j) + K\right) \\ &\le (1-\gamma(n))||Q^n||_\infty + \gamma(n)D. \end{aligned}$$

A simple induction shows that $||Q^n||_\infty \le D$ for all n. □

The boundedness argument above does not work for the asynchronous iteration (2.7). The reason is as follows: The term $f(Q^n)e$ (resp., $f(\overline{Q}^n)e$) being subtracted from the r.h.s. of (2.5) (resp., (3.7)) implies that exactly the same "offset" is being subtracted from all the components. These terms, being scalar multiples of e, contribute nothing to the span seminorm, a fact that is crucial in the analysis of [1] used above. In the asynchronous case, there is no way of achieving this without artificial restrictions.

The second technique is that of [13], which applies to both synchronous and asynchronous cases. We need the following assumption.

ASSUMPTION 3.6′. $f(cQ) = cf(Q)$ for all $c \in R, Q \in R^{d\times r}$.

Once again, this is satisfied by all the examples of $f(\cdot)$ given in the preceding section. Define $T_0 : R^{d\times r} \to R^{d\times r}$ by

$$(T_0(x))_{ia} = \sum_j p(i,a,j) \min_b x_{jb}, \quad x = [[x_{ia}]] \in R^{d\times r}.$$

The technique of [13] requires that we look at

$$\begin{aligned} h(x) &\triangleq \lim_{c\to\infty} (T'(cx) - cx)/c \\ &= T_0(x) - x - f(x)e \end{aligned}$$

(in view of Assumption 3.6′) and requires that the origin be the globally asymptotically stable equilibrium point of the ODE $\dot{x}(t) = h(x(t))$. But this is merely a special case of Theorem 3.4, corresponding to $g(\cdot,\cdot,\cdot)$ being identically zero. Thus Theorem 2.2 of [13] applies, implying that $\{Q^n\}$ remains bounded a.s. for both the synchronous iteration (2.5), and its asynchronous version (2.7). (For the latter, see section 4 of [13].) We state this conclusion as a lemma.

LEMMA 3.8. *Under the additional Assumption* 3.6′, $\{Q^n\}$ *given by the synchronous Q-learning iteration* (2.5) *and its asynchronous version* (2.7) *is bounded a.s.*

4. Convergence of SSP Q-learning.

4.1. ODE analysis. Redefine T, T', f as follows. $T : R^{d\times r} \to R^{d\times r}, T' : R^{d\times r\times 1} \to R^{d\times r}, f : R^{d\times r} \to R$ are given by

$$\begin{aligned} (TQ)(i,a) &= \sum_{j=1}^{d} p(i,a,j) \left(g(i,a,j) + \sum_{j\neq s} p(i,a,j) \min_u Q(j,u) \right), \\ (T'(Q,\lambda))(i,a) &= (TQ)(i,a) - \lambda, \\ f(Q) &= \min_u Q(s,u). \end{aligned}$$

Then the synchronous iteration (2.8a)–(2.8b′) can be rewritten as

$$Q^{n+1} = Q^n + \gamma(n)[T'(Q^n,\lambda^n) - Q^n + M^{n+1}], \tag{4.1}$$

$$\lambda^{n+1} = \Gamma(\lambda^n + b(n) f(Q^n)), \tag{4.2}$$

where $M^{n+1} = [M^{n+1}(i,a)]$ with

$$\begin{aligned} M^{n+1}(i,a) = &\left[g(i,a,\xi_{ia}^n) + \min_u Q^n(\xi_{ia}^n, u) \right] I\{\xi_{ia}^n \neq s\} \\ &- \lambda^n - T'(Q^n,\lambda^n). \end{aligned}$$

In this new setup one verifies (3.2), (3.3) as before. Note that (4.2) can be rewritten as

$$\lambda^{n+1} = \lambda^n + e(n),$$

where $e(n) = O(b(n)) = o(\gamma(n))$. Thus the limiting ODE associated with (4.1)–(4.2) is

$$\dot{Q}(t) = T'(Q(t),\lambda(t)) - Q(t),$$

$$\dot{\lambda}(t) = 0.$$

Thus it suffices to consider

$$\dot{Q}(t) = T'(Q(t), \lambda) - Q(t) \tag{4.3}$$

for a fixed λ. As observed in [25], the map $T(\cdot)$, and therefore the map $T'(\cdot, \lambda)$ for fixed λ, is a contraction on $R^{d\times r}$ with respect to a certain weighted max-norm

$$||x||_w \stackrel{\Delta}{=} \max |w_i x_i|, \quad x \in R^{d\times r},$$

for an appropriate weight vector $w = [w_1, \ldots, w_{rd}], w_i > 0$ for all i. In particular, $T'(\cdot, \lambda)$ has a unique fixed point $Q(\lambda)$. A straightforward adaptation of the arguments of [14] then shows the following.

LEMMA 4.1. *$Q(\lambda)$ is the globally asymptotically stable equilibrium for* (4.1). *In fact, $||Q(t) - Q(\lambda)||_w$ decreases monotonically to zero.*

4.2. Boundedness and convergence. Once again we present two alternative schemes for proving the a.s. boundedness of $\{Q^n\}$. (Note that $\{\lambda^n\}$ are bounded anyway, as they are constrained to remain in $[-K, K]$.) The first approach is based on [25].

LEMMA 4.2. *For both synchronous and asynchronous SSP Q-learning algorithms, $\{Q^n\}$ remain bounded a.s.*

Proof. Since $T(\cdot)$ is a contraction with respect to $||\cdot||_w$, we have

$$||T(Q)||_w \leq \alpha||Q||_w + D$$

for some $\alpha \in (0, 1), D > 0$. Thus

$$||T'(Q, \lambda)||_w \leq \alpha||Q||_w + D'$$

with $D' = D + K$. Since the r.h.s. does not involve λ, one can mimick the arguments of [25] to conclude. □

An alternative proof of Lemma 4.2 is to directly quote the results of [13]. For this, consider $T^0 : R^{d\times r} \to R^{d\times r}$ defined by

$$(T^0 Q)(i, a) = \sum_{j\neq s} p(i, a, j) \min_u Q(j, u).$$

Then

$$\lim_{a\to\infty} \frac{T'(cQ, \lambda)}{c} = T^0(Q),$$

and the ODE

$$\dot{Q}(t) = T^0(Q) - Q$$

has the origin as the globally asymptotically stable equilibrium. (This is just a special case of Lemma 4.1 with $g(\cdot) \equiv 0$.) Thus the results of [13] apply, allowing us to conclude Lemma 4.2.

Given the a.s. boundedness of iterates, one proves a.s. convergence for the synchronous case as follows.

LEMMA 4.3. *$||Q^n - Q(\lambda^n)|| \to 0$ a.s.*

Proof. Note that $Q(\lambda)$ is simply the Q-factor associated with the SSP problem described in section 2.3 with the prescribed λ, that is,

$$(Q(\lambda))(i,a) = \sum_{j=1}^{d} p(i,a,j)\left(g(i,a,j) + \sum_{j\neq s} p(i,a,j)V_\lambda(j) - \lambda\right), \qquad i \in S, \qquad a \in A.$$

Since the map $\lambda \to V_\lambda$ is concave, it is continuous, and therefore so is the map $\lambda \to Q(\lambda)$. In view of Lemmas 4.1 and 4.2, the claim now follows as in Corollary 2.1 of [8]. □

We shall also need the following lemma.

LEMMA 4.4.

$$\Pi_{i=0}^{n}(1-b(i)) \to 0, \quad \limsup_{n\to\infty} \sum_{i=0}^{n} \Pi_{j=i+1}^{n}(1-b(j))b(i) < \infty,$$

and for any sequence $\{a_n\}$ *with* $a_n \to 0$,

$$\sum_{i=0}^{n}\left(\prod_{j=i+1}^{n}(1-b(j))\right)b(i)a_i \to 0.$$

Proof. Since $\sum_i b(i) = \infty$ and $1 - x \leq e^{-x}$ for all x,

$$\prod_{i=0}^{n}(1-b(i)) \leq e^{-\sum_{i=0}^{n} b(i)} \to 0$$

as $n \to \infty$. Let $t(0) = 0, t(n) = \sum_{i=0}^{n-1} b(i), n \geq 1$. Then

$$\begin{aligned}\sum_{i=0}^{n}\prod_{j=i+1}^{n}(1-b(j))b(i) &\leq \sum_{i=0}^{n} b(i)\exp\left(-\sum_{j=i+1}^{n} b(j)\right)\\ &= \sum_{i=0}^{n} e^{-(t(n+1)-t(i))}(t(i+1)-t(i))\\ &\leq \int_0^{t(n+1)} e^{-(t(n+1)-s)}ds \to 1\end{aligned}$$

as $n \to \infty$. Define $h(t), t \geq 0$, by $h(t) = a_n$ for $t \in [t(n), t(n+1)), n \geq 0$. Then a similar argument shows that

$$\sum_{i=0}^{n}\prod_{j=i+1}^{n}(1-b(j))b(i)a_i \leq \int_0^{t(n+1)} e^{-(t(n+1)-s)}h(s)ds.$$

Since $h(t) \to 0$ as $t \to \infty$, the r.h.s. $\to 0$ as $n \to \infty$. □

THEOREM 4.5. *For the synchronous SSP Q-learning algorithm* (2.8a)–(2.8b′), $(Q^n, \lambda^n) \to (Q^*, \beta)$ *a.s.*

Proof. Define $\Delta^n = \lambda^n - \beta, r_n = f(Q^n) - f(Q(\lambda^n)), n \geq 0$. By Lemma 4.3, $r_n \to 0$. Also,

$$\begin{aligned}\Delta^{n+1} &= \Gamma(\Delta^n + \beta + b(n)f(Q^n)) - \beta \\ &= \Gamma(\Delta^n + \beta + b(n)f(Q(\lambda^n)) + b(n)r_n) - \beta. \end{aligned} \tag{4.4}$$

Since the map $\lambda \to V_\lambda$, and therefore also the map $\lambda \to f(Q(\lambda)) = \min_u(Q(\lambda))(s,u)$, is concave and piecewise linear with finitely many linear pieces, it follows that there exist $0 < L_1 \le L_2$ such that

$$-L_2(\lambda_1 - \lambda_2) \le f(Q(\lambda_1)) - f(Q(\lambda_2)) \le -L_1(\lambda_1 - \lambda_2)$$

for all $\lambda_1, \lambda_2 \in R$. (Basically, these are upper and lower bounds on the slope of the map $\lambda \to f(Q(\lambda))$.) With $\lambda_1 = \lambda^n, \lambda_2 = \beta$, and using the fact that $f(Q(\beta)) = f(Q^*) = 0$, this reduces to

$$-L_2\Delta^n \le f(Q(\lambda^n)) \le -L_1\Delta^n.$$

Using this, (4.4), and the fact that $\Gamma(\cdot)$ is nondecreasing, we have

$$\begin{aligned}&\Gamma((1 - L_2 b(n))\Delta^n + b(n)r_n + \beta) - \beta \le \Delta^{n+1} \\ &\quad \le \Gamma((1 - L_1 b(n))\Delta^n + b(n)r_n + \beta) - \beta.\end{aligned}$$

Note that for $i = 1, 2$,

$$\begin{aligned}(1 - L_i b(n))\Delta^n + b(n)r_n + \beta &= \lambda^n + b(n)(r_n - L_i\Delta^n) \\ &= \lambda^n + O(b(n)).\end{aligned}$$

Since $\lambda^n \in [-K, K]$ for all n and $b(n) \to 0$, it follows from the definition of $\Gamma(\cdot)$ that for any $\epsilon > 0$, there is an $N \ge 1$ sufficiently large so that for $n \ge N$,

$$\begin{aligned}&(1 - L_2 b(n))\Delta^n + (b(n)r_n - \epsilon) + \beta \\ &\quad \le \Gamma((1 - L_2 b(n))\Delta^n + b(n)r_n + \beta) \\ &\quad \le \Gamma((1 - L_1 b(n))\Delta^n + b(n)r_n + \beta) \\ &\quad \le (1 - L_i b(n))\Delta^n + (b(n)r_n + \epsilon) + \beta.\end{aligned}$$

Therefore,

$$(1 - L_2 b(n))\Delta^n + b(n)r_n - \epsilon \le \Delta^{n+1} \le (1 - L_1 b(n))\Delta^n + b(n)r_n + \epsilon.$$

Iterating the inequalities, we have for $n > N$

$$\begin{aligned}&\prod_{i=N}^{n+1}(1 - L_2 b(i))\Delta^N + \sum_{i=N}^{n}\prod_{j=i+1}^{n}(1 - L_2 b(j))(b(i)r_i - \epsilon) \le \Delta^{n+1} \\ &\qquad \le \prod_{i=N}^{n+1}(1 - L_1 b(i))\Delta^N + \sum_{i=N}^{n}\prod_{j=i+1}^{n}(1 - L_1 b(j))b(i)(r_i - \epsilon).\end{aligned}$$

Letting $n \to \infty$ and using Lemma 4.4, we have $\Delta^n \to [-C\epsilon, C\epsilon]$ for a suitable constant $C > 0$. Since $\epsilon > 0$ was arbitrary, $\Delta^n \to 0$, i.e., $\lambda^n \to \beta$. Since $\lambda \to Q(\lambda)$ is continuous, $Q(\lambda^n) \to Q(\beta) = Q^*$, and, by Lemma 4.3, $Q^n \to Q^*$. ∎

Remark. If we consider instead the SSP Q-learning algorithm (2.8a)–(2.8b) that does not use the projection $\Gamma(\cdot)$, it is possible to argue as above to conclude that *if* the iterates $\{\lambda^n\}$ remain bounded a.s., then Theorem 4.5 holds.

Finally, we have the following theorem.

THEOREM 4.6. *For the asynchronous SSP Q-learning algorithm* (2.9a)–(2.9b), $(Q^n, \lambda^n) \to (Q^*, \lambda^*)$ *a.s.*

Proof. The analysis of [10], [17] applies, implying, in particular, that Lemma 4.3 holds exactly as before. The only difference is that now the interpolated algorithm would track a time-scaled version of (4.1). The rest is as before because the iteration scheme for $\{\lambda^n\}$ is unchanged. ∎

5. Conclusions. We have presented two Q-learning algorithms for average cost control of finite Markov chains—one based on RVI and another on an SSP formulation of the average cost problem. We have rigorously established their stability and convergence to the desired limits with probability one. As already remarked in the introduction, this is the first rigorous analysis of any Q-learning algorithms for average cost problems. Nevertheless, this is only a first step toward a better understanding of these algorithms. In conclusion, we mention three important directions for future work in this area:

(i) Typically, the state space can be very large. This calls for approximations, such as state aggregation or considering a parametrized family of candidate Q-factor functions with a low dimensional parameter space. (See, e.g., [7], [26].) The algorithms presented need to be interlaced with such approximation architectures and analyzed as such. A popular architecture is a linear combination of suitable basis functions, the weights in question being the parameters that are tuned [7], [26]. A good choice of basis functions is crucial, and it has been suggested that they be based upon sample simulation runs [6]. Yet another technique for reducing computation is to update not at every sample but at an appropriately chosen subsequence of samples. This can be combined with "kernel" methods, where one updates in a neighborhood of the sample in a weighted fashion. While there is an enormous amount of empirical work on such ideas in recent years, the theory has been lacking. Finally, it is worthwhile exploring the use of acceleration techniques in traditional Monte Carlo methods (such as importance sampling) to reinforcement learning.

(ii) Simulation-based algorithms are slow. An analysis of rate of convergence and good speed-up procedures are needed. To some extent, the rate of convergence statements for general stochastic approximation algorithms, based on associated limit theorems or asymptotics for moments, will also bear upon these algorithms. However, they have enough special structure that one should be able to say more. A recent work [12] takes a step in this direction by estimating the number of steps required by Q-learning for discounted cost problems to attain a prescribed level of accuracy with a prescribed probability.

(iii) Extension to the case where the state space is not finite is an open issue. See, however, [11] for the discounted cost problem.

Appendix. We briefly recall the key results from the literature that have been used here in a crucial manner. To begin with, let $F(\cdot,\cdot) = [F_1(\cdot,\cdot),\ldots,F_d(\cdot,\cdot)]^T : R^d \times R^m \to R^d$ be Lipschitz in the first argument uniformly with respect to the second, i.e., for some scalar K, we have

$$\|F(x,y) - F(y,u)\| \le K\|x-y\| \qquad \forall\ x,y,u.$$

Consider the stochastic approximation algorithm of the form

$$x^{k=1} = x^k + \gamma(k)F(x^k,\xi^k), \qquad k \ge 0,$$

for $x^k = [x_1^k,\ldots,x_d^k]$, with $\{\xi^k\}$ i.i.d., R^n-valued random variables. Let $h(x) = E[F(x,\xi)]$. The ODE approach views the above recursion as

$$x^{k=1} = x^k + \gamma(k)[h(x^k) + M^{k+1}]$$

with

$$M^{k+1} = F(x^k,\xi^k) - h(x^k), \qquad k \ge 0,$$

a "martingale difference" sequence. The term in square brackets is viewed as a noisy measurement of $h(x^k)$ with M^{k+1} as the "noise." The iteration can then be viewed as a noisy discretization of the ODE $\dot{x}(t) = h(x(t))$ with diminishing time steps. We assume that this ODE has a globally asymptotically stable equilibrium x^*. If $\{x^k\}$ remains bounded and the martingale $\sum \gamma(k) M^{k+1}$ converges with probability one, both the discretization error and the error due to noise in the above "approximation" of the ODE become asymptotically negligible, and therefore the iterates track the ODE. In particular, $x^k \to x^*$ a.s.

The asynchronous version of this algorithm is

$$x_i^{k+1} = x_i^k + \nu(k,i) I(i \in Y^k) F_i(x_1^{k-\tau_{1i}}(k), \ldots, x_d^{k-\tau_{di}}(k), \xi^k), \qquad 1 \le i \le d,$$

for $k \ge 0$, where (1) $\{Y^k\}$ is a set-valued random process taking values in the subsets of $\{1, \ldots, d\}$, representing the components that do get updated at time k, (2) $\{\tau_{ij}(k)\}$, $1 \le i, j \le d$, $k \ge 0\}$, are bounded random delays, and (3) $\nu(k,i) = \sum_{m=0}^{k} I(i \in Y^m)$ denotes the number of times component i gets updated up to time k. Under the kind of assumptions on $\{\gamma(k)\}$ and $\{\nu(k,i)\}$ we have used here, Borkar [10] shows that the asynchronous iterations track the ODE $\dot{x}(t) = \frac{1}{d} h(x(t))$, which is a time-scaled version of $\dot{x}(t) = h(x(t))$ and has the same trajectories.

The two-time-scale stochastic approximation algorithm of Borkar [8] considers the following iteration:

$$x^{k+1} = x^k + \gamma(k) F(x^k, y^k, \xi^k),$$

$$y^{k=1} = y^k + \beta(k) G(x^k, y^k, \zeta^k),$$

where $\{\xi^k\}$, $\{\zeta^k\}$ are i.i.d., and $\{\beta(k)\}$ satisfy

$$\sum_{k=0}^{\infty} \beta(k) = \infty, \qquad \sum_{k=0}^{\infty} \beta(k)^2 < \infty, \qquad \beta(k) = o(\gamma(k)).$$

Thus $\{y^k\}$ (resp., $\{x^k\}$) is the slow (resp., fast) component of the iteration. One can analyze $\{x^k\}$ viewing $\{y^k\}$ as quasi-static, and then analyze $\{y^k\}$, viewing $\{x^k\}$ as essentially equilibrated. In other words, consider the ODE $\dot{x}(t) = h(x(t), \overline{y})$, where $h(x,y) = E[F(x,y,\xi)]$ and $\overline{y}$ is treated as a fixed parameter. Suppose it has a globally asymptotically stable equilibrium $\lambda(\overline{y})$, where $\lambda(\cdot)$ is a Lipschitz function. Then $\{x^k\}$ tracks $\{\lambda(y^k)\}$. In turn, $\{y^k\}$ tracks the ODE $\dot{y}(t) = g(\lambda(y(t)), y(t))$, where $g(x,y) = E[G(x,y,\zeta)]$. If the latter ODE has a globally asymptotically stable equilibrium y^*, one can show that $(x^k, y^k) \to (\lambda(y^*), y^*)$ a.s.

In dynamic programming applications, an important special class of ODEs arises, wherein $h(x) = f(x) - x$ for an $f : R^d \to R^d$ satisfying the "nonexpansivity property"

$$\|f(x) - f(y)\|_\infty \le \|x - y\|_\infty.$$

The set $B = \{x : f(x) = x\}$ of fixed points of $f(\cdot)$, assumed to be nonempty, is precisely the set of equilibria for the ODE $\dot{x}(t) = f(x(t)) - x(t)$. It is shown in Borkar and Soumyanath [14] that $x(\cdot)$ converges to a point in B and, furthermore, for any $x^* \in B$, $\|x(t) - x^*\|_\infty$ is nonincreasing in t.

Finally, we recall Lemma 2.2 of [1], which has been used here. Let $D \subset R^d$ be an open bounded set, and let $C \subset R^d$ be a set containing D. Define $\Pi_{D,C} : R^d \to \overline{D}$ by

$$\prod_{D,C}(x) = \alpha_{D,C}(x) \cdot x,$$

where

$$\alpha_{D,C}(x) = \begin{cases} 1 & \text{if } x \in C, \\ \max\{\beta > 0 \,:\, \beta x \in \overline{D}\} & \text{if } x \notin C. \end{cases}$$

Let $\|x\|_s = \max_i x_i - \min_i x_i$ define the span seminorm on R^d. Consider an iteration

$$x^{k+1} = G^k(x^k, \xi^k),$$

where $\{\xi^k\}$ is a random process and $\{G^k\}$ satisfy

$$\|G^k(x,\xi) - G^k(y,\xi)\|_s \le \|x - y\|_s \qquad \forall\, x, y, \xi.$$

Suppose the sequence $\{\tilde{x}^k\}$ generated by the "scaled" iteration

$$\tilde{x}^{k+1} = G^k\left(\prod_{D,C}(\tilde{x}^k), \xi^k\right)$$

converges a.s. Lemma 2.1 of [1] then says that $\{\|x^k\|_s\}$ remains bounded with probability one.

REFERENCES

[1] J. Abounadi, D. P. Bertsekas, and V. S. Borkar, *Stochastic Approximation for Nonexpansive Maps: Application to Q-Learning*, Report LIDS-P-2433, Laboratory for Information and Decision systems, MIT, Cambridge, MA, 1998.

[2] A. Benveniste, M. Metivier, and P. Priouret, *Adaptive Algorithms and Stochastic Approximations*, Springer-Verlag, Berlin, Heidelberg, 1990.

[3] D. P. Bertsekas, *Distributed dynamic programming*, IEEE Trans. Automat. Control, 27 (1982), pp. 610–616.

[4] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Vol.* 2, Athena Scientific, Belmont, MA, 1995.

[5] D. P. Bertsekas, *A new value iteration method for the average cost dynamic programming problem*, SIAM J. Control Optim., 36 (1998), pp. 742–759.

[6] D. P. Bertsekas and D. A. Castanon, *Rollout algorithms for stochastic scheduling problems*, J. Heuristics, 5 (1999), pp. 89–108.

[7] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*, Athena Scientific, Belmont, MA, 1996.

[8] V. S. Borkar, *Stochastic approximation with two time scales*, Systems Control Lett., 29 (1996), pp. 291–294.

[9] V. S. Borkar, *Recursive self-tuning control of finite Markov chains*, Appl. Math. (Warsaw), 24 (1996), pp. 169–188.

[10] V. S. Borkar, *Asynchronous stochastic approximations*, SIAM J. Control Optim., 36 (1998), pp. 840–851.

[11] V. S. Borkar, *A learning algorithm for discrete time stochastic control*, Probab. Engrg. Inform. Sci., 14 (2000), pp. 243–248.

[12] V. S. Borkar, *On the number of samples required for Q-learning*, in Proceedings of the 38th Allerton Conference, University of Illinois at Urbana-Champaign, Urbana-Champaign, IL, 2000.

[13] V. S. Borkar and S. P. Meyn, *The ODE method for convergence of stochastic approximation and reinforcement learning*, SIAM J. Control Optim., 38 (2000), pp. 447–469.

[14] V. S. Borkar and K. Soumyanath, *A new analog parallel scheme for fixed point computation, part* I: *Theory*, IEEE Trans. Circuits Systems I Fund. Theory Appl., 44 (1997), pp. 351–355.

[15] T. Jaakola, M. I. Jordan, and S. P. Singh, *On the convergence of stochastic iterative dynamic programming algorithms*, Neural Computation, 6 (1994), pp. 1185–1201.

[16] A. Jalali and M. Ferguson, *Adaptive control of Markov chains with local updates*, Systems Control Lett., 14 (1990), pp. 209–218.

[17] V. R. Konda and V. S. Borkar, *Actor-critic–type learning algorithms for Markov decision processes*, SIAM J. Control Optim., 38 (2000), pp. 94–123.
[18] H. J. Kushner and D. Clark, *Stochastic Approximation Methods for Constrained and Unconstrained Systems*, Springer-Verlag, New York, 1978.
[19] H. J. Kushner and G. Yin, *Stochastic Approximation Algorithms and Applications*, Springer-Verlag, New York, 2000.
[20] S. Mahadevan, *Average reward reinforcement learning: Foundations, algorithms and empirical results*, Machine Learning, 22 (1996), pp. 1–38.
[21] M. L. Puterman, *Markov Decision Processes*, John Wiley and Sons, New York, 1994.
[22] A. Schwartz, *A reinforcement learning method for maximizing undiscounted rewards*, in Proceedings of the 10th International Conference on Machine Learning, Morgan Kaufmann, San Mateo, 1993, pp. 298–305.
[23] S. P. Singh, *Reinforcement learning algorithms for average payoff Markovian decision processes*, in Proceedings of the 12th National Conference on Artificial Intelligence, MIT Press, Cambridge, MA,1994, pp. 202–207.
[24] H. C. Tijms, *Stochastic Modeling and Analysis: A Computational Approach*, John Wiley and Sons, New York, 1986.
[25] J. N. Tsitsiklis, *Asynchronous stochastic approximation and Q-learning*, Machine Learning, 16 (1994), pp. 185–202.
[26] J. N. Tsitsiklis and B. van Roy, *Feature-based methods for large scale dynamic programming*, Machine Learning, 22 (1996), pp. 59–94.
[27] C. Watkins, *Learning from Delayed Rewards*, Ph.D. thesis, Cambridge University, Cambridge, U.K., 1989.
[28] C. Watkins and P. Dayan, *Q-learning*, Machine Learning, 8 (1992), pp. 279–292.
[29] F. W. Wilson, *Smoothing derivatives of functions and applications*, Trans. Amer. Math. Soc., 139 (1967), pp. 413–428.