Discrete Event Dyn Syst (2016) 26:477–509
DOI 10.1007/s10626-015-0216-z

Multiscale Q-learning with linear function approximation

Shalabh Bhatnagar[1] · **K. Lakshmanan**[2]

Received: 25 September 2012 / Accepted: 10 August 2015 / Published online: 30 August 2015

Abstract We present in this article a two-timescale variant of Q-learning with linear function approximation. Both Q-values and policies are assumed to be parameterized with the policy parameter updated on a faster timescale as compared to the Q-value parameter. This timescale separation is seen to result in significantly improved numerical performance of the proposed algorithm over Q-learning. We show that the proposed algorithm converges almost surely to a closed connected internally chain transitive invariant set of an associated differential inclusion.

Keywords Q-learning with linear function approximation · Reinforcement learning · Stochastic approximation · Ordinary differential equation · Differential inclusion · Multi-stage stochastic shortest path problem

1 Introduction

Markov decision process (MDP) (Bertsekas (2005, 2007); Puterman (1994)) is a general framework for studying problems of control under uncertainty. Classical approaches for solution of MDP such as *policy iteration* and *value iteration* aim at numerically solving the associated Bellman equation and work only when the system dynamics via the transition probabilities is known precisely. Such information is often not available in real-life systems. Moreover, computationally solving MDPs using traditional approaches when the state and

✉ Shalabh Bhatnagar
shalabh@csa.iisc.ernet.in

K. Lakshmanan
mpekl@nus.edu.sg

[1] Department of Computer Science and Automation, Indian Institute of Science, Bangalore, 560 012, India

[2] Department of Mechanical Engineering, National University of Singapore, Singapore, Singapore

action spaces are large is a challenging task. Reinforcement learning (RL) (Bertsekas and Tsitsiklis 1996; Sutton and Barto 1998) provides efficient alternatives to policy and value iteration methods. Many RL algorithms do not require any knowledge of the transition probabilities and work precisely with real or simulated data. Further, many of these algorithms incorporate parameterized classes of policies and value functions in order to improve the computational efficiency in the case of large scale systems. Value function approximation has been studied extensively using linear architectures because prediction algorithms such as temporal difference (TD) learning (Sutton 1988) have been shown convergent when linear architectures are used (Tsitsiklis and Van Roy 1997; Tsitsikis and Van Roy 1999). On the other hand, TD with nonlinear function approximation has been seen to diverge in some cases.

Q-learning (Watkins and Dayan 1992) is an RL algorithm that has been found to be efficient in various settings for the problem of control. This algorithm is based on Q-value iteration (a variant of value iteration) that updates values of feasible state-action tuples (also called Q-values) at each stage. Whereas Q-learning with full state representation is shown to be convergent (Abounadi et al. 2001; Borkar and Meyn 2000; Tsitsiklis 1994), this is not in general the case when function approximation (even linear) is used (see, however, Melo and Ribeiro 2007). It has been reported that Q-learning with function approximation can exhibit heavy oscillations and in some cases can even diverge (Baird 1995). This problem arises from "off-policy learning", i.e., learning the value function of one policy (the target policy) using data obtained from another (the behaviour policy). Under off-policy settings, TD with function approximation is known to suffer from this problem. When actions are picked according to a given policy, the Q-learning iterate resembles an on-policy TD update rule and is convergent. However, in the general case (of Q-learning), because of the explicit minimization in the update rule, the policy according to which the minimizing actions are chosen can be seen to change at each iterate resulting in the off-policy problem. More recently, however, in Sutton et al. (2009), Sutton et al. (2009), certain variants of the TD algorithm have been developed for the problem of prediction, that are shown convergent with off-policy learning when linear function approximators are used. In Maei et al. (2009), similar algorithms with smooth function approximators that could be nonlinear as well, have been proposed and shown convergent. Further, Maei et al. (2010) presents a convergent algorithm with linear function approximation for off-policy control that is derived from TD learning. Also, a variant of Q-learning where the function approximator satisfies certain interpolation properties is presented in Szepesvari and Smart (2004) and is shown convergent.

In this paper, we present a variant of the Q-learning algorithm with linear function approximation that is based on two-timescale stochastic approximation (Borkar 1997). The Q-value parameters for a given policy in our algorithm are updated on the slower timescale while the policy parameters themselves are updated on the faster scale. We perform a gradient search in the space of policy parameters. Since the objective function and hence the gradient are not analytically known, we employ the efficient one-simulation simultaneous perturbation stochastic approximation (SPSA) gradient estimates that employ Hadamard matrix based deterministic perturbations, see Bhatnagar et al. (2003). SPSA, orginally presented in Spall (1992), is an efficient gradient estimation technique that normally uses two simulations and incorporates random perturbations. It estimates the gradient of the objective at any parameter value by perturbing the parameter along each component direction using finite-valued, i.i.d., symmetric, zero-mean, random variates such as those having a symmetric Bernoulli distribution of ± 1 w.p.1/2. A one-simulation (random perturbation) variant of SPSA was presented in Spall (1997) that however does not exhibit good performance in

general. In Bhatnagar et al. (2003), certain deterministic perturbation variants of both one and two simulation SPSA were developed. In particular, the one-simulation variant with the perturbation vectors derived from a normalized Hadamard matrix construction was seen to show significant performance improvements over one-simulation random perturbation SPSA (Spall 1997). We therefore employ, in our algorithm, the deterministic perturbation, one-simulation, SPSA estimates based on normalized Hadamard matrices. For the case of full-state representations, a similar algorithm derived from Q-learning has been presented in (Bhatnagar and Babu 2008).

While our algorithm incorporates two timescales, it has a slightly different flavour from actor-critic (AC) algorithms, see for instance, (Konda and Borkar 1999; Konda and Tsitsiklis 2003; Bhatnagar and Kumar 2004; Abdulla and Bhatnagar 2007), that also incorporate two timescales. AC algorithms are in general based on the policy iteration (PI) technique, whereby the faster-scale recursion (the critic) is responsible for evaluating a given policy update (the *policy evaluation* step of PI) while the slower-scale recursion (the actor) updates the policy to find an improved policy (the *policy improvement* step). On the other hand, our starting point is the Q-learning algorithm that is based on the Q-value iteration procedure. Note that the minimization operation in this procedure is actually within the conditional expectation term in the corresponding Q-Bellman equation. Our scheme aims to solve the minimization problem on the faster timescale, and use the converged values of the faster iterates in the slower recursion. Thus, the timescales get reversed when one moves from AC algorithms to our scheme. Unlike AC algorithms, the faster scale now updates the policy paramater while the slower one updates the parameter of the Q-values. Even though AC type algorithms have been studied in the past, this particular combination has not been tried previously, see, however, the discussion on pp.317, Chapter 6, of (Bertsekas and Tsitsiklis 1996).

While our algorithm continuously updates both the value function and the policy, this is also the case with other schemes, for instance, with the algorithm in (Azar et al. 2011). The latter algorithm, however, incorporates a dynamic off-line policy programming technique and uses a combination of a numerical fixed point procedure together with Monte-Carlo simulation to estimate the conditional expectation of the cost-to-go in order to solve an 'approximate' Bellman equation. We do not require such a procedure as we resort to stochastic approximation. Moreover, our algorithm is an online scheme that works directly with real or simulated data.

We show that our algorithm converges to a closed connected internally chain transitive invariant set of an associated differential inclusion. As an application setting, we consider the problem of multi-stage stochastic shortest path routing (Walrand 1988) and consider various networks with multiple numbers of stages and nodes. A significant body of work in the literature on multi-stage queueing networks is dedicated towards obtaining results on the structure of the optimal policies (Ephremides et al. 1980; Weber 1978) such as Join-the-Shortest Queue (JSQ), Shortest-Remaining-Processing-Time (SRPT) etc., under certain conditions on the network and traffic settings. It is often assumed in these references that information on transition probabilities of the system is precisely known. We however consider a scenario where such information is not available in general. Moreover, it may be difficult to prescribe a general form for the optimal policies. We show performance comparisons of our algorithm with regular Q-learning as well as an actor-critic scheme and observe that our algorithm performs much better than both the other algorithms.

Finally, we would like to mention that in Prashanth et al. (2014), the average cost variant of our algorithm has recently been presented and the numerical performance of both average and discounted cost algorithms is studied on a problem of intrusion detection in wireless

sensor networks where the goal is to obtain optimal sleep-wake schedules for individual sensors in a network of battery operated sensor nodes while monitoring potential intruder movement. As with the experiments in our paper, it is observed in Prashanth et al. (2014) that two-timescale Q-learning outperforms Q-learning and other competing algorithms in the literature by requiring much less number of sensors to be awake at any given time while giving similar tracking accuracy.

The rest of the paper is organized as follows: The framework and our algorithm are presented in Section 2. The proof of convergence of our algorithm is presented in Section 3. The application setting and numerical results are given in Section 4. The concluding remarks are presented in Section 5. Finally, proofs of some of the more preliminary results have been given in an Appendix at the end of the paper.

2 The framework and algorithm

2.1 The framework

A stochastic process $\{X_n\}$ is referred to as a Markov decision process (MDP) if it is governed by a control sequence $\{Z_n\}$ and satisfies the controlled Markov property (below). We let S denote the state space i.e., the set in which $\{X_n\}$ takes values and $A(i)$ the set of feasible actions in state i. Further, let $A \stackrel{\triangle}{=} \cup_{i \in S} A(i)$ be the set of all actions or the action space. We assume that both S and A are finite sets. The controlled Markov property satisfied by $\{X_n\}$ is the following:

$$P(X_{n+1} = j \mid X_m, Z_m, m \leq n) = p(X_n, Z_n, j) \text{ a.s.},$$

where $p : S \times A \times S \to [0, 1]$ is a given function for which $\sum_{j \in S} p(i, a, j) = 1, \forall a \in A(i), i \in S$.

We define an admissible policy as a sequence $\psi \stackrel{\triangle}{=} \{\mu_0, \mu_1, \ldots\}$ of functions $\mu_n : S \to A$, with $\mu_n(i) \in A(i)\ \forall i \in S, n \geq 0$. At any instant n, the action chosen in state $k \in S$ when following policy ψ is $\mu_n(k)$. The policies that we consider are inherently Markovian since the resulting process under any admissible policy is Markov. Let Ψ denote the set of all admissible policies. When $\mu_n \equiv \mu, \forall n \geq 0$, i.e., the functions μ_n do not change with n, we call ψ or many times (by an abuse of notation) μ itself a stationary deterministic policy (SDP). Let $\mathcal{P}(A)$ (resp. $\mathcal{P}(A(i))$, $i \in S$) denote the set of all probability vectors on A (resp. $A(i)$). By a randomized policy (RP) χ, we mean a sequence $\chi = \{\pi_1, \pi_2, \ldots\}$ where each $\pi_n : S \to \mathcal{P}(A)$, $n \geq 0$, and such that for each $i \in S$, $n \geq 0$, $\pi_n(i) \in \mathcal{P}(A(i))$. Thus for each $i \in S$, $\pi_n(i)$ is a distribution on $A(i)$, $n \geq 1$. A stationary randomized policy (SRP) is a RP χ for which $\pi_n(i) = \pi(i), \forall n \geq 1, i \in S$, i.e., the distribution on $A(i)$ is stage-invariant. By an abuse of notation, we refer to π itself as the SRP. The ath component of $\pi(i)$, $a \in A(i)$, denoted $\pi(i, a)$ is the probability of choosing action a when in state i. Thus under an SRP π, the action $Z_n \in A(X_n)$ at instant n is picked according to the distribution $\pi(X_n)$, independent of all other states and actions realized till n.

In what follows, we shall formulate the problem in the infinite-horizon discounted cost setting. Let $g(X_n, Z_n)$ denote the real-valued single-stage cost when state is X_n and action chosen is Z_n. Further, let $\gamma \in (0, 1)$ denote the discount factor (a given constant). Note that

$\sup_n |g(X_n, Z_n)| < \infty$ almost surely since S and A are finite sets. Under a given admissible policy $\psi = \{\mu_0, \mu_1, \mu_2, \cdots\}$, define the infinite horizon discounted cost as

$$J_\psi(i) = \lim_{T\to\infty} E\left[\sum_{k=0}^{T} \gamma^k g(X_k, \mu_k(X_k))|X_0 = i\right],\ i \in S.$$

The aim is to find an admissible policy $\psi^* \in \Psi$ that minimizes $J_\psi(i)$ over all $\psi \in \Psi$ and $i \in S$. Let $J^*(i), i \in S$ denote the optimal cost or the value function. Thus,

$$J^*(i) \stackrel{\triangle}{=} J_{\psi^*}(i) = \min_{\bar{\psi}\in\Psi} J_{\bar{\psi}}(i), i \in S.$$

One can show that an optimal policy that is an SDP (and so trivially also an SRP) exists for this problem and that J^* satisfies the Bellman equation

$$J^*(i) = \min_{a\in A(i)} \left(g(i,a) + \gamma \sum_{j\in S} p(i,a,j)J^*(j)\right). \tag{2.1}$$

2.2 Q-learning with function approximation (QL-FA)

Under a given admissible policy ψ, let the Q-value function be defined as follows: $\forall i \in S$, $a \in A(i)$,

$$Q_\psi(i,a) = E\left[\sum_{k=0}^{\infty} \gamma^k g(X_k, \mu_k(X_k)) \mid X_0 = i, Z_0 = a\right]. \tag{2.2}$$

The process in this case starts from state i when action a is chosen. At the subsequent instants, i.e., $k = 1, 2, 3, \ldots$, actions are selected according to the admissible policy ψ. Let the optimal Q-values be defined according to

$$Q^*(i,a) = \min_{\psi\in\Psi} Q_\psi(i,a), i \in S, a \in A(i).$$

It can be shown that the following Q-Bellman equation gets satisfied:

$$Q^*(i,a) = g(i,a) + \gamma \sum_{j\in S} p(i,a,j) \min_{v\in A(j)} Q^*(j,v). \tag{2.3}$$

The Q-learning algorithm with full state representation aims to solve (2.3) using stochastic approximation and proceeds in the following manner: $\forall i \in S, a \in A(i)$,

$$Q_{n+1}(i,a) = Q_n(i,a) + c(n)\left(g(i,a) + \gamma \min_{v\in A(Y_n^{i,a})} Q_n(Y_n^{i,a}, v) - Q_n(i,a)\right). \tag{2.4}$$

Here, $Y_n^{i,a}$ is a simulated next state when the current state is i and action $a \in A(i)$ is chosen. The random variables $Y_n^{i,a}$, $n \geq 0$ are assumed to be independent and have the distribution $p(i,a,\cdot), i \in S, a \in A(i)$. Further, $c(n), n \geq 0$ are step-sizes that satisfy $c(n) > 0\ \forall n \geq 0$ and

$$\sum_n c(n) = \infty, \sum_n c(n)^2 < \infty. \tag{2.5}$$

One typically resorts to function approximation for the Q-values when the cardinality of the state-action space is so large that even storing a vector of the size of the state-action space is difficult. For $i \in S, a \in A(i)$, let $Q^*(i,a) \approx \theta^{*T}\phi_{i,a}$, where $\theta^* = (\theta^*(1), \ldots, \theta^*(d))^T$ is a d-dimensional parameter (for some $d \geq 1$) and $\phi_{i,a} = (\phi_{i,a}(1), \ldots, \phi_{i,a}(d))^T$ is the associated feature vector.

Let Φ denote a matrix with rows $\phi_{i,a}^T,\ i \in S, a \in A(i)$. Assuming that the total number of states is n and the number of feasible actions in any state i (i.e., the cardinality of $A(i)$) is $m_i \geq 1$, the number of rows of the matrix Φ is $q = \sum_{j=1}^{n} m_j$. The number of columns of this matrix is d. One can also represent Φ as $\Phi = (\Phi(k), k = 1, \ldots, d)$, where $\Phi(k)$ is the column vector

$$\Phi(k) = (\phi_{i,a}(k),\ i \in S,\ a \in A(i))^T,\ k = 1, \ldots, d.$$

Now $Q^* = (Q^*(i,a), i \in S, a \in A(i))^T$ can be approximated as

$$Q^* \approx \sum_{i=1}^{d} \Phi(i)\theta^*(i), \text{ or alternatively, } Q^* \approx \Phi\theta^*.$$

The estimates $Q_n(i,a),\ n \geq 0$, of $Q^*(i,a),\ i \in S,\ a \in A(i)$ will be approximated as $Q_n(i,a) \approx \theta_n^T \phi_{i,a}$, where $\theta_n \stackrel{\triangle}{=} (\theta_n(1), \ldots, \theta_n(d))^T$ denotes the nth estimate of θ^*. This estimate is obtained from the algorithm below. The Q-learning algorithm with function approximation that we refer to as **QL-FA** updates θ_n according to

$$\theta_{n+1} = \theta_n + c(n)\phi_{X_n,Z_n}\left(g(X_n, Z_n) + \gamma \min_{v \in A(X_{n+1})} \theta_n^T \phi_{X_{n+1},v} - \theta_n^T \phi_{X_n,Z_n}\right), \tag{2.6}$$

where θ_0 is set arbitrarily and the step-sizes $c(n),\ n \geq 0$ satisfy (2.5). It is important to note that (2.6) is an on-line algorithm as it works with a single trajectory of (feasible) state-action tuples $(X_n, Z_n),\ n \geq 0$ and updates the parameter θ as new states are observed and actions chosen. Further, $\nabla_{\theta_n} Q_n(X_n, Z_n) \approx \nabla_{\theta_n} \theta_n^T \phi_{X_n,Z_n} = \phi_{X_n,Z_n}$. As discussed briefly in the earlier section, the algorithm (2.6) is known to suffer from the off-policy problem (Baird 1995; Sutton and Barto 1998) and may not converge in some cases.

2.3 Two-timescale Q-learning with function approximation (QW-FA)

We present here a new algorithm based on Q-learning with function approximation. Let $\pi_w = (\pi_w(i), i \in S)^T$ represent a class of SRP that are parameterized by w, where each $\pi_w(i)$ is the distribution $\pi_w(i) = (\pi_w(i,a), a \in A(i))^T$ over the set of feasible actions $A(i)$ in state i. We let $w \stackrel{\triangle}{=} (w_1, \ldots, w_N)^T \in C \subset \mathcal{R}^N$ (for some $N \geq 1$). We also let $\theta \in \mathcal{R}^d$ take values in the set $D \subset \mathcal{R}^d$. In what follows, we restrict our attention to SRP parameterized by w.

We make the following assumptions.

Assumption 1 *The Markov process $\{X_n\}$ under any SRP π_w is aperiodic and irreducible.*

Assumption 2 *The probabilities $\pi_w(i,a),\ i \in S,\ a \in A(i)$ are continuously differentiable in the parameter $w \in C$. Further, $\pi_w(i,a) > 0\ \forall i \in S,\ a \in A(i),\ w \in C$.*

A well-studied example of parameterized policies that satisfy Assumption 2 are the parameterized Boltzmann policies described by

$$\pi_w(i,a) = \frac{\exp(w^T \phi_{i,a})}{\sum_{b \in A(i)} \exp(w^T \phi_{i,b})}.$$

We also use these policies for our experiments.

Assumption 3 *The basis functions* $\{\Phi(k), k = 1, \ldots, d\}$ *are linearly independent. Further,* $d \leq |S|$.

Assumption 4 *The sets* C *and* D *are (both) compact and convex subsets of* $\mathcal{R}^N$ *and* $\mathcal{R}^d$*, respectively. In particular,* C *has the form*

$$C = \{x \mid q_i(x) \leq 0, i = 1, \ldots, x\},$$

where $q_i(\cdot)$, $i = 1, \ldots, s$ *are continuously differentiable. Further, at each* $x \in \partial C$ *(the boundary of* C*), the gradients of the active constraints are linearly independent.*

Since S is a finite set, it follows from Assumption 1 that $\{X_n\}$ is also positive recurrent under the SRP π_w. Assumption 1 holds for the multi-stage routing example that we study in Section 4. Assumption 2 is a standard requirement in policy gradient approaches, see for instance (Bhatnagar et al. 2009). Similarly, Assumption 3 is routinely used as well, see Tsitsiklis and Van Roy (1997), Bertsekas and Tsitsiklis (1996). The requirement (Assumption 4) that both w and θ take values in suitable compact sets ensures stability of both the w and the θ updates (below). Thus, using the recursions in our algorithm, $\sup_n \| w_n \|, \sup_n \| \theta_n \| < \infty$ with probability one. When C is an N-dimensional rectangle $\prod_{i=1}^{N}[c_{i,\min}, c_{i,\max}]$, with $c_{i,\min} < c_{i,\max}, \forall i$, it can be written in the form as in Assumption 4 with $q_i(x)$ being linear functions of the form $q_i(x) = c_i x_i + d_i$, which are continuously differentiable. The form of the set C in Assumption 4 allows us to use a key result from Kushner and Clark (1978) on convergence of projected stochastic iterates to associated ODEs for the analysis of the faster (w) recursion (see Theorem 2). On the other hand, the slower recursion will be seen to track an associated differential inclusion, for which we will use a key result from (Borkar 2008). The latter result does not require the set D to have a form similar to C.

Our algorithm uses two step-size sequences $\{a(n)\}$ and $\{b(n)\}$ that satisfy the following requirements:

Assumption 5 *For all* $n \geq 0$, $a(n), b(n) > 0$, *and*

$$\sum_n a(n) = \sum_n b(n) = \infty, \quad \sum_n (a(n)^2 + b(n)^2) < \infty, \quad b(n) = o(a(n)), \tag{2.7}$$

respectively. Further, for any integer $M > 0$,

$$\lim_{n\to\infty} \max_{1\leq l\leq M} \frac{a(n+l)}{a(n)} = \lim_{n\to\infty} \max_{1\leq l\leq M} \frac{b(n+l)}{b(n)} = 1.$$

While the first two requirements in (2.7) on $a(n), b(n)$, $n \geq 0$, are similar to (2.5), the last condition there implies that $b(n)$ tends to zero at a rate faster than $a(n)$. Thus, recursions governed by $a(n)$ (resp. $b(n)$) operate on a 'faster' (resp. 'slower') timescale. The requirements in Assumption 5 are seen to be satisfied by many step-size sequences such as $a(n) = 1/n^\alpha$, $\alpha \in (0.5, 1.0)$ and $b(n) = 1/n$, $n \geq 1$, with $a(0) = b(0) = 1$.

We require gradient estimates of the approximate Q-value function (with respect to the parameter w) for which we use a one-simulation simultaneous perturbation gradient approximation procedure based on a Hadamard matrix construction as in (Bhatnagar et al. 2003). We briefly describe these estimates here. Let $\Delta_n = (\Delta_n(1), \ldots, \Delta_n(N))^T, n \geq 0$ be certain column vectors of perturbation variables obtained from a normalized Hadamard matrix

(see below). These vectors will be used to perturb the parameter updates w_n. In particular, an estimate $D(n)$ of $\nabla f(w_n)$ for a given function f is given by (see Bhatnagar et al. 2003):

$$D(n) = \frac{f(w_n + \delta\Delta_n)}{\delta}(\Delta_n)^{-1},$$

for a given (small) $\delta > 0$, where $(\Delta_n)^{-1} \stackrel{\triangle}{=} (\Delta_n^{-1}(1), \ldots, \Delta_n^{-1}(N))^T$. It will be shown in the convergence analysis in Section 3 (see Theorem 2) that $D(n)$ serves as a close approximation to $\nabla f(w_n)$.

2.3.1 Hadamard matrix based deterministic perturbations

Let $H_{2^k}, k \geq 1$ be matrices of order $2^k \times 2^k$ that are recursively obtained as:

$$H_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \text{ and } H_{2^k} = \begin{pmatrix} H_{2^{k-1}} & H_{2^{k-1}} \\ H_{2^{k-1}} & -H_{2^{k-1}} \end{pmatrix}, k > 1.$$

Such matrices are called normalized Hadamard matrices. All elements in the first row and column of these matrices are 1. Let $P = 2^{\lceil \log_2(N+1) \rceil}$. It is easy to see that $P \geq N + 1$. Consider now the matrix H_P (with P chosen as above). Let $h(1), \ldots, h(N)$ be any N columns of H_P other than the first column. Form a new matrix H'_P of order $P \times N$ that has $h(1), \ldots, h(N)$ as its columns. Let $e(p)$, $p = 1, \ldots, P$ be the P rows of H'_P. Now set $\Delta_n^T = e(n \bmod P + 1), \forall n \geq 0$. In other words, the perturbations are generated by cycling through the rows of H'_P with $\Delta_0^T = e(1), \Delta_1^T = e(2), \ldots, \Delta_{P-1}^T = e(P), \Delta_P^T = e(1)$ etc. Some results related to the perturbation sequence obtained from a Hadamard matrix based construction are given in Bhatnagar et al. (2003) and are described in Section 3 for the sake of completeness.

Next, we present our two-timescale Q-learning algorithm with function approximation that we refer to as **QW-FA**. The letter 'W' in QW-FA signifies the additional weight parameter 'w' used in our algorithm.

2.3.2 The algorithm QW-FA

Let θ_n and w_n denote the nth updates of the parameters θ and w, respectively. Let $\pi_{(w_n+\delta\Delta_n)} \stackrel{\triangle}{=} (\pi_{(w_n+\delta\Delta_n)}(i,a), i \in S, a \in A(i))^T$, where $\delta > 0$ is a given small constant, be the randomized policy followed during the nth update of the algorithm. Note that this (randomized) policy is governed by the parameter $(w_n + \delta\Delta_n)$. Here $\Delta_n, n \geq 0$ are perturbations obtained from the Hadamard matrix construction described in Section 2.3.1. Let $\Gamma_1 : \mathcal{R}^d \to D$ denote the projection operator that projects any $z \in \mathcal{R}^d$ to the set D. Similarly, let $\Gamma_2 : \mathcal{R}^N \to C$ denote the projection operator that projects any $x \in \mathcal{R}^N$ to the set C. We let both projections be with respect to suitable Euclidean norms in the two spaces. We denote both norms (by abuse of notation) as $\| \cdot \|$. Thus, for any $z \in \mathcal{R}^d$ ($x \in \mathcal{R}^N$), $\Gamma_1(z) = \arg\min_{r \in D} \| r - z \|$ ($\Gamma_2(x) = \arg\min_{y \in C} \| y - x \|$).

The algorithm is as follows: For all $n \geq 0$,

$$\theta_{n+1} = \Gamma_1\left(\theta_n + b(n)\phi_{X_n,Z_n}\left(g(X_n, Z_n) + \gamma\theta_n^T\phi_{X_{n+1},Z_{n+1}} - \theta_n^T\phi_{X_n,Z_n}\right)\right), \tag{2.8}$$

$$w_{n+1} = \Gamma_2\left(w_n - a(n)\left(\frac{\theta_n^T\phi_{X_n,Z_n}}{\delta}\right)(\Delta_n)^{-1}\right). \tag{2.9}$$

Remark 1 1. Note that unlike Q-learning where only the Q-values are parameterized, our algorithm parameterizes both Q-values and policies (in fact, SRPs). In (2.8)–(2.9), both the actions Z_n and Z_{n+1} are sampled from the same policy $\pi_{(w_n+\delta\Delta_n)}$. We show in our analysis that given θ, the w_n update in (2.9) converges to the set of local minima of the stationary average Q-value function denoted $R(\theta, w)$, see (3.2) and the subsequent discussion. Thus, the 'minimization over actions' in Q-learning that is performed over instantaneous Q-value updates is replaced by a 'minimization over policy parameters' in our algorithm, that is in effect performed with the stationary average Q-value function as the objective. Thus, as is seen in our experiments, whereas Q-learning suffers from the problem of high variance/instabilities, the iterates of our algorithm stabilize rapidly because of the timescale difference and show significantly low variance in comparison.

2. It has been shown, see for instance (Baird 1995), that Q-learning with function approximation can diverge. A natural question to ask is whether convergence of Q-learning can be guaranteed if its iterates are projected to a compact set (the way we do in our algorithm). Indeed a divergent sequence of iterates when projected to a compact set will settle on the projection set boundary. It is not clear, however, whether a similar differential inclusions (DI) based analysis as ours can be worked through in the case of Q-learning. The most important step in that direction would be to obtain an analogous result as Proposition 2, which it is not clear would carry through for projected Q-learning. In the absence of such a result, even if one were to identify a suitable DI with an appropriate limit set, the stochastic iterates may still not converge to that set. In other words, Theorem 3 will not hold unless Proposition 2 does. Thus, if the projection region is large enough so that the limit set of DI is strictly contained in the interior of the projection region, the iterates may not converge to the limit set. This would indeed be true of situations where Q-learning diverges (Baird 1995), and as mentioned above, the projected Q-learning iterates would settle on the projection set boundary and not in the aforementioned limit set of DI. On the other hand, we show that our algorithm converges to the limiting set of the associated DI. We only use projection in our algorithm to ensure stability of iterates (a key requirement in the analysis of stochastic recursive schemes) and not to 'force' convergence to some undesirable point (in order to avoid divergence), and which projection based Q-learning might end up doing. Nonetheless, it would be an interesting future direction to prove the stability of our algorithm's iterates without resorting to projection, that would considerably strengthen the theoretical analysis of our algorithm.

3 Convergence analysis

We begin with an overview of how the proof proceeds.

3.1 An overview of the proof

The algorithm QW-FA incorporates two-timescale stochastic approximation, hence, the faster timescale update essentially sees the slower recursion as quasi-static while the slower update sees the faster one as equilibrated. We show in the analysis (cf. Theorem 2) that corresponding to any given value of the slower parameter (θ) update, the faster (i.e., policy) parameters (w_n) converge to a set of fixed points $w(\theta)$. For convergence of the faster timescale recursion, we rely on a key result from Kushner and Clark (1978) (cf. Theorem 5.3.1) for projected stochastic approximations.

We then proceed to analyze the slower recursion and observe that this is a stochastic recursive inclusion because of the set-valued nature of $w(\theta)$. We show that this recursion tracks a differential inclusion in the asymptotic limit and in particular converges almost surely to a closed connected internally chain transitive invariant set associated to this inclusion. In the process, we also show that the associated set-valued map satisfies certain nice properties required to prove convergence of stochastic recursive inclusions. We make use of the fact that because of the projection, both the slow and the fast iterates are stable. A detailed convergence analysis of stochastic recursive inclusions is given in (Benaim et al. 2005), see also Chapter 5 of (Borkar 2008), and applications of the same have been given in (Benaim et al. 2006). A convergence analysis of projected stochastic recursive inclusions for various settings is also available in Chapter 5 of (Borkar 2008). Our analysis of the slower recursion (that also incorporates projection in its update rule) is based on the latter reference.

The main result is in Theorem 3 that gives the convergence of the slower recursion and is derived from a sequence of other smaller related results. Specifically, we follow the sequence of steps (below) in order to prove Theorem 3.

(i) We begin with some preliminary results. Proposition 1–Lemma 1 show that the state-action Markov chain $\{(X_n, Z_n)\}$ is ergodic and its stationary distribution f_w is continuously differentiable in the policy parameter w. Next, in Lemma 2, we show that the partial derivatives of the stationary average Q-value $R(\theta, w)$ exist and are continuous. In fact, Theorem 2 shows that the partial derivative of $R(\theta, w)$ w.r.t. w for a given θ is tracked by the faster (w) recursion.

(ii) Next, we analyze the convergence of the faster (w) recursion. Towards this end, Lemma 3–Corollary 1 characterize the martingale difference component of the noise sequence and show that the aggregate sequence is a convergent martingale. Then Lemmas 4–6 are used in the proof of Theorem 2 to establish that the w-update direction as given by the one-simulation SPSA based faster recursion (2.9) indeed corresponds to a 'nearly steepest descent' gradient direction.

(iii) The main result pertaining to convergence of the faster (w) recursion is Theorem 2 that establishes that for any given θ and $\epsilon > 0$, there exists $\delta_0 > 0$ such that for all $\delta \leq \delta_0$, the parameter sequence $\{w_n\}$ converges to a set $w(\theta)^\epsilon$ of stable fixed points. The proof is based on an application of a convergence result from Kushner and Clark (1978) (cf. Theorem 5.3.1) that we also describe here as Theorem 1.

(iv) Next, we analyze the convergence of the slower (θ) recursion. We first observe that because the faster recursion converges to a set $w(\theta)^\epsilon$ of fixed points, the slower recursion in fact resembles a projected stochastic recursive inclusion. The projection operation ensures that the recursion remains stable. Our analysis here largely follows along the lines of Chapter 5 of (Borkar 2008). In fact, Chapter 5.4 of Borkar (2008) deals with the case of projected recursive inclusions. The idea here is to show that the stochastic recursive inclusion essentially tracks a closed connected internally chain transitive invariant set associated with a corresponding differential inclusion (DI). A detailed analysis of stochastic recursive inclusions by characterizing the associated limit points of DI is given in (Benaim et al. 2005).

(vi) We identify a suitable DI for the slower (θ) recursion and show in Proposition 2 that the DI satisfies certain nice regularity properties under which it is guaranteed to have at least one solution that is absolutely continuous (Aubin and Cellina 1984).

(vii) Under the aforementioned regularity properties of the DI given by Proposition 2, we finally show in Theorem 3 that the slower recursion converges asymptotically almost

surely to a closed connected internally chain transitive invariant set of the associated DI. This result follows from an application of Lemma 1 (pp.53), Theorem 2 (pp.53–54), Lemma 3 (pp.54–55) and Corollary 4 (pp.55), Chapter 5 of (Borkar 2008).

We provide the proofs of some of the preliminary results in an Appendix at the end of the paper.

3.2 The proof of convergence

For a given SRP π_w, define $\check{p}(i, j, \pi_w), i, j \in S$ as

$$\check{p}(i, j, \pi_w) = \sum_{a \in A(i)} \pi_w(i, a) p(i, a, j).$$

Under the SRP π_w, $\{X_n, n \geq 0\}$ is a Markov process with transition probabilities $\check{p}(i, j, \pi_w)$, $i, j \in S$. Let $d^{\pi_w} \stackrel{\triangle}{=} (d^{\pi_w}(i), i \in S)$ be the stationary distribution of this process. Consider now the joint process $\{(X_n, Z_n), n \geq 0\}$, i.e., that obtained from the state-action tuples at each instant. This process takes values in the set $S \times A(S) \stackrel{\triangle}{=} \{(i, a) \mid i \in S, a \in A(i)\}$. It is easy to see that $\{(X_n, Z_n)\}$ with Z_n, $n \geq 0$ obtained from π_w is also a Markov process with transition probabilities

$$p_w(i, a; j, b) \stackrel{\triangle}{=} P(X_{n+1} = j, Z_{n+1} = b \mid X_n = i, Z_n = a) = p(i, a, j)\pi_w(j, b), \quad (3.1)$$

$i, j \in S, a \in A(i), b \in A(j)$, respectively.

Recall from earlier discussion that $q = \sum_{i=1}^{n} m_i$ is the cardinality of $S \times A(S)$. Let P_w denote the $q \times q$ transition probability matrix of the joint Markov process $(X_n, Z_n), n \geq 0$ when the policy parameter w is fixed.

Proposition 1 *Under Assumptions 1 and 2, the process $(X_n, Z_n), n \geq 0$ with Z_n, $n \geq 0$ obtained from the SRP π_w, for any given $w \in C$, is an ergodic Markov process.*

Proof See Appendix for a proof. □

As a consequence of Proposition 1, $\{(X_n, Z_n)\}$ has a unique stationary distribution $f_w(i, a), i \in S, a \in A(i)$. We will denote by $\mho_w$ the row vector of stationary probabilities $\mho_w = (f_w(i, a), i \in S, a \in A(i))$. Note that

$$f_w(j, b) = \sum_{i \in S, a \in A(i)} f_w(i, a) p_w(i, a; j, b), \; j \in S, b \in A(j).$$

It is easy to verify that $f_w(i, a) = d^{\pi_w}(i)\pi_w(i, a), i \in S, a \in A(i)$.

Lemma 1 *Under Assumptions 1 and 2, $f_w(i, a)$, $i \in S$, $a \in A(i)$ are continuously differentiable in the parameter $w \in C$.*

Proof See Appendix for a proof. □

Let

$$R(\theta, w) \stackrel{\triangle}{=} \sum_{i \in S, a \in A(i)} f_w(i, a)\theta^T \phi_{i,a}, \quad (3.2)$$

denote the stationary average Q-value under the parameters θ and w, respectively.

Lemma 2 *The partial derivatives of $R(\theta, w)$ with respect to any $\theta \in D$ and $w \in C$ exist and are continuous.*

Proof See Appendix for a proof. □

Let $u(\cdot)$ (resp. $v(\cdot)$) denote a vector field on D (resp. C). Define now the vector fields $\hat{\Gamma}_1(u(\cdot))$ and $\hat{\Gamma}_2(v(\cdot))$ on D and C, respectively, as follows:

$$\hat{\Gamma}_1(u(\theta)) = \lim_{\eta \downarrow 0} \left(\frac{\Gamma_1(\theta + \eta u(\theta)) - \theta}{\eta} \right) \text{ and}$$

$$\hat{\Gamma}_2(v(w)) = \lim_{\alpha \downarrow 0} \left(\frac{\Gamma_2(w + \alpha v(w)) - w}{\alpha} \right),$$

respectively. Since D and C are compact and convex sets, $\hat{\Gamma}_1$ and $\hat{\Gamma}_2$ are both well defined. In particular, if $\theta \in D^o$ (resp. $w \in C^o$), $\hat{\Gamma}_1(u(\theta)) = u(\theta)$ (resp. $\hat{\Gamma}_2(v(w)) = v(w)$). Here D^o (resp. C^o) denotes the interior of D (resp. C). On the other hand, if $\theta \in \partial D$ (resp. $w \in \partial C$) is such that $\theta + \eta u(\theta) \notin D$ (resp. $w + \alpha v(w) \notin C$) for small η (resp. α), then $\hat{\Gamma}_1(u(\theta))$ (resp. $\hat{\Gamma}_2(v(w))$) is the projection of $u(\theta)$ (resp. $v(w)$) onto D (resp. C). Here ∂D (resp. ∂C) denotes the boundary of D (resp. C).

The following ODE is associated with (2.9):

$$\dot{w}(t) = \hat{\Gamma}_2 \left(-\nabla_w R(\theta(t), w(t)) \right). \tag{3.3}$$

Assuming for now that $\theta(t) \equiv \theta$ (a constant independent of t), (3.3) can be rewritten as

$$\dot{w}(t) = \hat{\Gamma}_2 \left(-\nabla_w R(\theta, w(t)) \right). \tag{3.4}$$

The (faster) recursion (2.9) will be seen to track (3.4). It will also be seen that along the faster timescale, one may indeed let $\theta(t) \equiv \theta$.

Let $w(\theta)$ denote the set of stationary points of the ODE (3.4). Given $\epsilon > 0$, let $w(\theta)^\epsilon$ denote the ϵ-neighborhood of $w(\theta)$, i.e.,

$$w(\theta)^\epsilon = \{w \in C \mid\mid w - w_0 \mid\mid < \epsilon, w_0 \in w(\theta)\}.$$

Now let $\mathcal{F}_n$, $n \geq 0$ denote a sequence of σ-fields defined according to

$$\mathcal{F}_n = \sigma(X_j, Z_j, j < n; \theta_j, w_j, j \leq n), n \geq 1.$$

Let $\{K_n^i, n \geq 0\}$, $i = 1, \ldots, N$ be defined as $K_0^i = 0$ and for $n \geq 1$,

$$K_n^i = \sum_{j=0}^{n-1} a(j) \left(\frac{\theta_j^T \phi_{X_j, Z_j} - E[\theta_j^T \phi_{X_j, Z_j} \mid \mathcal{F}_j]}{\delta \Delta_j^i} \right).$$

Let $M_{n+1}^i = \frac{\theta_n^T \phi_{X_n, Z_n} - E[\theta_n^T \phi_{X_n, Z_n} | \mathcal{F}_n]}{\delta \Delta_n^i}$. Then $K_n^i = \sum_{j=0}^{n-1} a(j) M_{j+1}^i$, $i = 1, \ldots, N$.

Lemma 3 *For all $i = 1, \ldots, N$, $(K_n^i, \mathcal{F}_n)$, $n \geq 0$ are almost surely convergent martingale sequences.*

Proof We show the proof for a given $i \in \{1, \ldots, N\}$ as the same proof holds for all other values of i as well. Note that K_n^i is measurable with respect to $\mathcal{F}_n, \forall n \geq 0$. Also, the random variables K_n^i are integrable as well. Further,

$$E\left[K_{n+1}^i \mid \mathcal{F}_n\right] = E\left[\sum_{j=0}^{n} a(j)\left(\frac{\theta_j^T \phi_{X_j,Z_j} - E[\theta_j^T \phi_{X_j,Z_j} \mid \mathcal{F}_j]}{\delta\Delta_j^i}\right) \mid \mathcal{F}_n\right]$$

$$= K_n^i + \frac{a(n)}{\delta\Delta_n^i}\left(E\left[\theta_n^T \phi_{X_n,Z_n} \mid \mathcal{F}_n\right] - E\left[\theta_n^T \phi_{X_n,Z_n} \mid \mathcal{F}_n\right]\right)$$

$$= K_n^i \text{ a.s.}$$

Now note that since $\{\theta_n\}$ take values in D, a compact set, we have $\sup_n \parallel \theta_n \parallel < \infty$ w.p. 1. Further, because the state-action spaces are finite, $\sup_{i,a} \phi_{i,a} < \infty$ as well. Also, $(\Delta_j^i)^2 = 1$ $\forall j \geq 0$ and $\sum_n a(n)^2 < \infty$. Thus, it follows that $\sup_n E[(K_n^i)^2] < \infty$. The claim follows from the martingale convergence theorem. □

Let $\hat{K}_n^i = \sum_{j=0}^{n-1} a(j)\hat{M}_{j+1}^i$, $i = 1, \ldots, N$, where $\hat{M}_{j+1}^i$ is the same as M_{j+1}^i, $\forall j, \forall i$, with the difference that $\theta_j \equiv \theta, \forall j$. Let $\hat{\mathcal{F}}_n$ be the same as $\mathcal{F}_n$ except with $\theta_n \equiv \theta$ $\forall n$ in the definition.

Corollary 1 *For all $i = 1, \ldots, N$, $(\hat{K}_n^i, \hat{\mathcal{F}}_n), n \geq 0$ is an almost surely convergent martingale.*

Proof Follows from Lemma 3 by setting $\theta_n \equiv \theta, \forall n$. □

Recall that $P = 2^{\lceil \log_2(N+1) \rceil}$ denotes the number of rows of the associated Hadamard matrix (cf. Section 2.3.1).

Lemma 4 *The vectors Δ_n, $n \geq 0$ satisfy the following properties:*

1. *For any $s \geq 0$ and all $k \in \{1, \ldots, N\}$, $\sum_{n=s+1}^{s+P} \frac{1}{\Delta_n^k} = 0$.*
2. *For any $s \geq 0$ and all $i, j \in \{1, \ldots, N\}$, $i \neq j$, $\sum_{n=s+1}^{s+P} \frac{\Delta_n^i}{\Delta_n^j} = 0$.*

Proof The claim is obvious from the construction, see Lemma 3.5 of (Bhatnagar et al. 2003). □

Lemma 5 *The iterates w_n, $n \geq 0$ governed according to (2.9) satisfy*

$$\parallel w_{n+k} - w_n \parallel \rightarrow 0 \text{ as } n \rightarrow \infty,$$

almost surely for all $k \in \{1, \ldots, P\}$.

Proof Follows from an application of Lemma 2.2 of Bhatnagar et al. (2003) using the facts that $\sup_n \parallel \theta_n \parallel < \infty$ almost surely and $\sup_{(i,a)} |\phi_{i,a}| < \infty$. □

We now have the following lemma whose proof we provide in Appendix as it has not been stated for the case of Hadamard matrix perturbations in (Bhatnagar et al. 2003).

Lemma 6 *The following results hold for any $k, l \in \{1, \ldots, N\}$, $k \neq l$: With probability one,*

$$\| \sum_{n=m}^{m+P-1} \frac{a(n)}{a(m)} \frac{\triangle_n^k}{\triangle_n^l} \nabla_{w,k} R(\theta, w_n) \| \to 0 \text{ as } m \to \infty, \tag{3.5}$$

$$\| \sum_{n=m}^{m+P-1} \frac{a(n)}{a(m)} \frac{1}{\triangle_n^l} R(\theta, w_n) \| \to 0 \text{ as } m \to \infty, \tag{3.6}$$

respectively.

Proof See Appendix for a proof. □

Before we proceed further, we describe an important result from Kushner and Clark (1978) (Theorem 5.3.1 on pp. 191-196 of Kushner and Clark (1978)) for general projected stochastic approximations. While the result, as given in Kushner and Clark (1978), is more generally applicable, we present its adaptation here that is relevant to the setting that we consider.

Let $\Gamma : \mathcal{R}^L \to E \subset \mathcal{R}^L$ denote a projection operator mapping any $r \in \mathcal{R}^L$ to the set E. Consider the following L-dimensional stochastic recursion

$$r_{n+1} = \Gamma\left(r_n + c(n)(h(r_n) + \xi_n + \gamma_n)\right), \tag{3.7}$$

under the assumptions (A1)–(A5) listed below. Also, consider the following ODE associated with (3.7):

$$\dot{r}(t) = \bar{\Gamma}(h(r(t))), \tag{3.8}$$

where for any vector field $y(\cdot)$ on E,

$$\bar{\Gamma}(y(r)) = \lim_{\beta \to 0} \left(\frac{\Gamma(r + \beta y(r)) - r}{\beta} \right).$$

Let S denote the set of all stationary points of the ODE (3.8). Let $t(n), n \geq 0$ be a sequence of positive real numbers defined according to $t(0) = 0$ and for $n \geq 1$, $t(n) = \sum_{j=0}^{n-1} c(j)$. Let $m(t) = \max\{n \mid t(n) \leq t\}$. Let the following assumptions hold:

(A1) The function $h : \mathcal{R}^L \to \mathcal{R}^L$ is continuous.

(A2) The step-sizes $c(n), n \geq 0$ satisfy

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

(A3) The sequence $\gamma_n, n \geq 0$ is a bounded random sequence with $\gamma_n \to 0$ almost surely as $n \to \infty$.

(A4) There exists $T > 0$ such that $\forall \epsilon > 0$,

$$\lim_{n \to \infty} P\left(\sup_{j \geq n} \max_{t \leq T} \left| \sum_{i=m(jT)}^{m(jT+t)-1} c(i)\xi_i \right| \geq \epsilon \right) = 0.$$

(A5) The set $E \subset \mathcal{R}^L$ is compact. Also, $E = \{x \mid g_i(x) \leq 0, i = 1, \ldots, s_1\}$. The functions $g_i(\cdot), i = 1, \ldots, s_1$ are continuously differentiable. Further, at each $x \in \partial E$, the gradients of the active constraints of $g_i(x)$ are linearly independent.

As a consequence of (A2), $t(n) \to \infty$ as $n \to \infty$. Hence, $m(t) \to \infty$ as $t \to \infty$.

Theorem 1 [Kushner and Clark (1978), Theorem 5.3.1 (pp. 191–196)] *Let $h(\cdot) = -f_r(\cdot)$, where $f(\cdot)$ is a continuously differentiable function. Then under (A1)–(A5), $\{r_n\}$ governed by (3.7) satisfy $r_n \to S$ as $n \to \infty$ with probability one.*

The set S above is the set of Kuhn-Tucker points of the constrained optimization problem. Now note that one may rewrite (2.8) as

$$\theta_{n+1} = \Gamma_1\left(\theta_n + a(n)\hat{X}(n)\right), \tag{3.9}$$

where $\hat{X}(n) = \frac{b(n)}{a(n)}\check{X}(n)$ and with $\check{X}(n) = \phi_{X_n,Z_n}(g(X_n, Z_n) + \gamma\theta_n^T\phi_{X_{n+1},Z_{n+1}} - \theta_n^T\phi_{X_n,Z_n})$. Since $\sup_n \| \theta_n \| < \infty$ a.s., and $S \times A(S)$ is a finite set, $\sup_n \| \check{X}(n) \| < \infty$ with probability one. Now since $b(n)/a(n) \to 0$ as $n \to \infty$ (cf. Assumption 5), it follows that $\hat{X}(n) \to 0$ as $n \to \infty$ almost surely. Thus (3.9) is seen to track the ODE

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

Hence, when analyzing the faster recursion $\{w_n\}$, one may let $\theta_n \equiv \theta$ (a constant).

Theorem 2 *Let $\theta_n \equiv \theta, \forall n$, for some $\theta \in D \subset \mathcal{R}^d$. Then, given $\epsilon > 0$, there exists $\delta_0 > 0$ such that for all $\delta \in (0, \delta_0]$, $\{w_n\}$ governed by (2.9) satisfy $w_n \to w(\theta)^\epsilon$ as $n \to \infty$ with probability one.*

Proof The proof will be shown by verifying the requirements of Theorem 1. In our case, r_n in Theorem 1 corresponds to w_n that is N-dimensional, i.e., $L = N$. Further, $c(n) = a(n), n \geq 0$ here. From Lemma 2, for any $\theta \in D$ (fixed), $\nabla_w R(\theta, w)$ exists and is continuous in w. Thus, by a Taylor's expansion of $R(\theta, w_n + \delta\Delta_n)$ around the point (θ, w_n), one obtains

$$R(\theta, w_n + \delta\Delta_n) = R(\theta, w_n) + \delta\sum_{j=1}^{N}\Delta_n^j\nabla_{w,j}R(\theta, w_n) + o(\delta).$$

Hence,

$$\frac{R(\theta, w_n + \delta\Delta_n)}{\delta\Delta_n^i} = \frac{R(\theta, w_n)}{\delta\Delta_n^i} + \nabla_{w,i}R(\theta, w_n) + \sum_{j=1, j\neq i}^{N}\frac{\Delta_n^j}{\Delta_n^i}\nabla_{w,j}R(\theta, w_n) + O(\delta). \tag{3.10}$$

Let w_n^i, $i = 1, \ldots, N$ be the N components of w_n, i.e., $w_n = (w_n^1, \ldots, w_n^N)^T$. Recall that $\Gamma_2(x)$ is the closest point in C to any $x \in \mathcal{R}^N$. In particular, for $x \stackrel{\triangle}{=} (x_1, \ldots, x_N)^T \in \mathcal{R}^N$, we denote $\Gamma_2(x) = (\Gamma_{2,1}(x_1), \ldots, \Gamma_{2,N}(x_N))^T \in C$, where $\Gamma_{2,i}(x_i)$ is the ith component of $\Gamma_2(x)$. We use this identification of $\Gamma_2(x)$ as a vector of $\Gamma_{2,i}(x_i)$, $i = 1, \ldots, N$, only for a technical reason (below). Note that (2.9) can be rewritten as follows: For $i = 1, \ldots, N$,

$$\begin{aligned} w_{n+1}^i &= \Gamma_{2,i}\left(w_n^i - a(n)E\left[\frac{\theta^T\phi_{X_n,Z_n}}{\delta\Delta_n^i} \mid \mathcal{F}_n\right] - a(n)\hat{M}_{n+1}^i\right) \\ &= \Gamma_{2,i}\left(w_n^i - a(n)\frac{\theta^T E\left[\phi_{X_n,Z_n} \mid \mathcal{F}_n\right]}{\delta\Delta_n^i} - a(n)\hat{M}_{n+1}^i\right) \\ &= \Gamma_{2,i}\left(w_n^i - a(n)\frac{R(\theta, w_n + \delta\Delta_n)}{\delta\Delta_n^i} - a(n)\hat{M}_{n+1}^i - a(n)\xi_2^i(n)\right), \end{aligned} \tag{3.11}$$

where $\xi_2^i(n) = \frac{(\theta^T E[\phi_{X_n,Z_n}|\mathcal{F}_n] - R(\theta, w_n+\delta\Delta_n))}{\delta\Delta_n^i}$. Note that $\xi_2^i(n), n \geq 0$ is a bounded random sequence. From the results in Theorem 7 – Corollary 8 of (Borkar 2008, Chapter 6, pp.74), it follows that one may let $w_n + \delta\Delta_n$, a constant while analyzing the convergence of $\xi_2^i(n)$ (since this convergence is along the 'natural' timescale). Now as a consequence of Proposition 1, $\xi_2^i(n) \to 0$ almost surely as $n \to \infty$.

One can now rewrite (3.11) as

$$w_{n+1}^i = w_n^i - a(n)\frac{R(\theta, w_n + \delta\Delta_n)}{\delta\Delta_n^i} - a(n)\hat{M}_{n+1}^i - a(n)\xi_2^i(n) + a(n)Z_i(n), \tag{3.12}$$

where $Z_i(n)$ is the error because of the projection, see Chapter 5.1, page 89 of Kushner and Yin (1997) for a general description of such error terms.

From (3.10), one can rewrite (3.12) as

$$w_{n+1}^i = w_n^i - a(n)\frac{R(\theta, w_n)}{\delta\Delta_n^i} - a(n)\nabla_{w,i}R(\theta, w_n) - a(n)\sum_{j=1, j\neq i}^{N}\frac{\Delta_n^j}{\Delta_n^i}\nabla_{w,j}R(\theta, w_n)$$

$$-a(n)\hat{M}_{n+1}^i - a(n)\xi_2^i(n) - a(n)O(\delta) + a(n)Z_i(n).$$

Now,

$$w_{n+P}^i = w_n^i - a(n)\sum_{k=n}^{n+P-1}\frac{a(k)}{a(n)}\frac{R(\theta, w_k)}{\delta\Delta_k^i} - \sum_{k=n}^{n+P-1}a(k)\nabla_{w,i}R(\theta, w_k)$$

$$-a(n)\sum_{k=n}^{n+P-1}\sum_{j=1, j\neq i}^{N}\frac{a(k)}{a(n)}\frac{\Delta_k^j}{\Delta_k^i}\nabla_{w,j}R(\theta, w_n) - \sum_{k=n}^{n+P-1}a(k)\hat{M}_{k+1}^i - \sum_{k=n}^{n+P-1}a(k)\xi_2^i(k)$$

$$-\sum_{k=n}^{n+P-1}a(k)O(\delta) + \sum_{k=n}^{n+P-1}a(k)Z_i(k).$$

Thus, one obtains

$$w_{n+P}^i = w_n^i - a(n)\chi_1^i(n) - \sum_{k=n}^{n+P-1}a(k)\nabla_{w,i}R(\theta, w_k) - a(n)\chi_2^i(n)$$

$$-\sum_{k=n}^{n+P-1}a(k)\hat{M}_{k+1}^i - a(n)\chi_3^i(n) - \sum_{k=n}^{n+P-1}a(k)O(\delta) + \sum_{k=n}^{n+P-1}a(k)Z_i(k),$$

where from Lemma 6, it follows that as $n \to \infty$, with probability one,

$$\chi_1^i(n) \stackrel{\triangle}{=} \sum_{k=n}^{n+P-1}\frac{a(k)}{a(n)}\frac{1}{\delta\Delta_k^i}R(\theta, w_k) \to 0,$$

$$\chi_2^i(n) \stackrel{\triangle}{=} \sum_{k=n}^{n+P-1}\sum_{j=1, j\neq i}^{N}\frac{a(k)}{a(n)}\frac{\Delta_k^j}{\Delta_k^i}\nabla_{w,k}R(\theta, w_k) \to 0,$$

respectively. Further, it is easy to see that as $n \to \infty$,

$$\chi_3^i(n) \stackrel{\triangle}{=} \sum_{k=n}^{n+P-1}\frac{a(k)}{a(n)}\xi_2^i(k) \to 0 \text{ a.s.},$$

since $\xi_2^i(n) \to 0$ as $n \to \infty$ almost surely. Now

$$\chi^i(n) \stackrel{\triangle}{=} \chi_1^i(n) + \chi_2^i(n) + \chi_3^i(n) = o(1) \text{ a.s.}, \tag{3.13}$$

by the foregoing. Thus, the recursion (2.9) is analogous to the following: For $i = 1, \ldots, N$,

$$w_{n+1}^i = \Gamma_{2,i}\left(w_n^i - a(n)\nabla_{w,i}R(\theta, w_n) - a(n)\hat{M}_{n+1}^i - a(n)\chi^i(n) - a(n)O(\delta)\right).$$

In vector form, (2.9) is thus analogous to

$$w_{n+1} = \Gamma_2\left(w_n - a(n)\nabla_w R(\theta, w_n) - a(n)\hat{M}_{n+1} - a(n)\chi(n) - a(n)O(\delta)e\right), \tag{3.14}$$

where $\hat{M}_{n+1} = (\hat{M}_{n+1}^i, i = 1, \ldots, N)^T$, $\chi(n) = (\chi^i(n), i = 1, \ldots, N)^T$ and $e = (1, 1, \ldots, 1)^T$ is the N-dimensional unit vector.

As a consequence of Corollary 1, (A4) holds with ξ_n replaced with $\hat{M}_{n+1}$, $n \geq 0$. It is easy to see from Lemma 1 that $R(\theta, w)$ is a continuous function over $D \times C$ and hence is uniformly bounded (since $D \times C$ is a compact set). Further, $\sup_{(i,a)} |\phi_{i,a}| < \infty$. Now from Lemma 2 and the fact that C is compact, we have that $\sup_{w \in C} \| \nabla_w R(\theta, w) \| < \infty$ for any $\theta \in D$. By individually considering the terms $\chi_1^i(n)$, $\chi_2^i(n)$, $\chi_3^i(n)$, $n \geq 0$, it is easy to see that $\sup_n \| \chi(n) \| < \infty$ almost surely. Since $\chi(n) \to 0$ as $n \to \infty$ almost surely (see (3.13)), (A3) holds with $\gamma_n \equiv \chi(n)$. Also, (A5) holds for the set C ($E \equiv C$ here) as a consequence of Assumption 4. Further, (A2) holds from Assumption 5. Finally, (A1) holds as a consequence of Lemma 2 with $h(r_n)$ replaced by $-\nabla_w R(\theta, w_n)$ and $f(r_n)$ replaced by $R(\theta, w_n)$, respectively.

The ODE associated with (3.14) is

$$\dot{w}(t) = \hat{\Gamma}_2\left(-\nabla_w R(\theta, w(t)) - O(\delta)e\right). \tag{3.15}$$

Let $\hat{w}(\theta)$ denote the set of asymptotically stable fixed points of (3.15). By Theorem 1, (3.14) will almost surely converge to $\hat{w}(\theta)$. The claim now follows from the fact that the trajectories of the ODE (3.15) converge to those of (3.4) as $\delta \to 0$ uniformly over compacts for the same initial condition in both (cf. Bhatnagar and Borkar (1997)[Theorem 3.2], (Bhatnagar et al. 2009)[Theorem 1]). □

Remark 2 Recall that $w(\theta)$ is the set of stationary points of the ODE (3.4) that correspond to all Kuhn-Tucker points and not just stable equilibria. Under certain 'richness' conditions on the noise, it can be shown that the stochastic approximation recursion can asymptotically avoid the unstable invariant sets, see for instance, Borkar (2008)[Chapter 4.3], Brandiere (1998), Pemantle (1990), and converge almost surely to the set of stable equilibria of the ODE. In practice, however, due to the inherent randomness of the scheme, stochastic approximation recursions are seen to converge to the set of stable attractors (a subset of $w(\theta)$) even without any additional noise conditions. This set will be the set of local or global minima of $R(\theta, w)$ for given θ.

Note also that Theorem 2 only gives the existence of a $\delta_0 > 0$ for given $\epsilon > 0$ such that $\forall\, \delta \leq \delta_0$, convergence to $w(\theta)^\epsilon$ is assured. From the above, this would imply convergence to a neighborhood of the local or global minima of $R(\theta, w)$ (for given θ). A small δ has the same effect as that of a large step-size and which results in a large variance during the initial runs. On the other hand, however, it helps speed up convergence. We empirically study in our experiments, the variation in performance with δ (see Table 4), and observe that the performance is sensitive to the choice of δ. In particular, $\delta = 0.005$ seems to give the best results over the settings we considered. Moreover, performance seems to deteriorate as δ is either diminished or increased beyond this value.

Define $T_w : \mathcal{R}^{|S\times A(S)|} \to \mathcal{R}^{|S\times A(S)|}$ as the operator

$$T_w(J)(i,a) = g(i,a) + \gamma \sum_{j\in S, b\in A(j)} p_w(i,a;j,b)J(j,b). \tag{3.16}$$

Let G denote the column vector $G = (g(i,a), i \in S, a \in A(i))^T$. Further, let $J = (J(i,a), i \in S, a \in A(i))^T$ and $T_w(J) = (T_w(J)(i,a), i \in S, a \in A(i))^T$. In vector-matrix notation, (3.16) is analogous to

$$T_w(J) = G + \gamma P_w J.$$

Let $\mathbb{F}_w$ denote the diagonal matrix whose elements along the diagonal are $f_w(i,a)$, $i \in S$, $a \in A(i)$. As a consequence of Proposition 1, the matrix $\mathbb{F}_w$ is positive definite.

We shall now analyze the slower recursion (2.8). Define $Y(n+1)$, $n \geq 0$ according to

$$Y(n+1) = \left(g(X_n, Z_n) + \gamma\theta_n^T\phi_{X_{n+1},Z_{n+1}} - \theta_n^T\phi_{X_n,Z_n}\right)\phi_{X_n,Z_n}$$

$$-E\left[\left(g(X_n, Z_n) + \gamma\theta_n^T\phi_{X_{n+1},Z_{n+1}} - \theta_n^T\phi_{X_n,Z_n}\right)\phi_{X_n,Z_n} \mid \mathcal{G}(n)\right],$$

where $\mathcal{G}(n) = \sigma(\theta_r, w_r, X_r, Z_r, r \leq n)$, $n \geq 0$ is a sequence of associated sigma fields. Let

$$\check{Z}(n) = \sum_{m=0}^{n-1} b(m)Y(m+1), n \geq 1.$$

It is easy to see that $(\check{Z}(n), \mathcal{G}(n))$, $n \geq 0$ is a martingale sequence. Now

$$\sum_n E[(\check{Z}(n+1) - \check{Z}(n))^2 \mid \mathcal{G}(n)]$$

$$= \sum_n b(n)^2 E\left[\left(\left(g(X_n, Z_n) + \gamma\theta_n^T\phi_{X_{n+1},Z_{n+1}} - \theta_n^T\phi_{X_n,Z_n}\right)\phi_{X_n,Z_n}\right.\right.$$

$$\left.\left.-E\left[\left(g(X_n, Z_n) + \gamma\theta_n^T\phi_{X_{n+1},Z_{n+1}} - \theta_n^T\phi_{X_n,Z_n}\right)\phi_{X_n,Z_n} \mid \mathcal{G}(n)\right]\right)^2 \mid \mathcal{G}(n)\right]$$

$$< \infty \text{ w.p. } 1,$$

because (a) $\sup_{(i,a)\in S\times A(S)} \| \phi_{i,a} \|$ and $\sup_{(i,a)\in S\times A(S)} |g(i,a)|$ are both finite since $S \times A(S)$ is a finite set and $g(i,a)$ are real valued; (b) $\sup_n \| \theta_n \| < \infty$ almost surely and (c) $\sum_n b(n)^2 < \infty$ from Assumption 5. Hence, by the martingale convergence theorem, $\{\check{Z}(n)\}$ converges almost surely.

As a consequence of Theorem 2, one may consider the following recursion (that is a stochastic recursive inclusion) in place of (2.8):

$$\theta_{n+1} = \Gamma_1(\theta_n + b(n)(y_n + Y(n+1) + \kappa(n+1))), \tag{3.17}$$

where

$$y_n = \sum_{(i,a)} f_{w_n}(i,a)\left(g(i,a) + \gamma{\theta_n}^T \sum_{(j,b)} p_{w_n}(i,a;j,b)\phi_{j,b} - {\theta_n}^T\phi_{i,a}\right)\phi_{i,a},$$

$$\kappa(n+1) = E\left[\left(g(X_n, Z_n) + \gamma\theta_n^T\phi_{X_{n+1},Z_{n+1}} - \theta_n^T\phi_{X_n,Z_n}\right)\phi_{X_n,Z_n} \mid \mathcal{G}(n)\right] - y_n,$$

respectively, with $w_n \in w(\theta_n)^\epsilon$, $\forall n$. Note again that from the results in Theorem 7 – Corollary 8 on pp. 74 and Theorem 9 on pp. 75, Chapter 6 of Borkar (2008), one may ignore the

$\kappa(n+1)$ term as it converges almost surely to zero along the natural timescale. We thus consider the recursion

$$\theta_{n+1} = \Gamma_1(\theta_n + b(n)(y_n + Y(n+1))). \tag{3.18}$$

Now proceeding as in Chapter 5.4 of (Borkar 2008), one can rewrite (3.18) as follows:

$$\begin{aligned}\theta_{n+1} &= \theta_n + b(n)\left(\frac{\Gamma_1(\theta_n + b(n)(y_n + Y(n+1))) - \theta_n}{b(n)}\right)\\ &= \theta_n + b(n)\left(\gamma_1(\theta_n; y_n + Y(n+1)) + o(b(n))\right),\end{aligned} \tag{3.19}$$

where

$$\gamma_1(\theta; y) = \lim_{\eta \downarrow 0}\left(\frac{\Gamma_1(\theta + \eta y) - \theta}{\eta}\right)$$

is the directional derivative of Γ_1 at θ in the direction y. Let $z_n \stackrel{\triangle}{=} E[\gamma_1(\theta_n; y_n + Y(n+1)) \mid \mathcal{G}(n)]$ and $\check{Y}(n+1) \stackrel{\triangle}{=} \gamma_1(\theta_n; y_n + Y(n+1)) - z_n$, respectively. Thus, (3.19) can be rewritten as

$$\theta_{n+1} = \theta_n + b(n)(z_n + \check{Y}(n+1) + o(b(n))). \tag{3.20}$$

Let $h(\theta)$ denote the set-valued map

$$h(\theta) \stackrel{\triangle}{=} \left\{ \sum_{(i,a)\in S\times A(S)} f_w(i,a)(g(i,a) + \gamma\theta^T \sum_{(j,b)\in S\times A(S)} p_w(i,a;j,b)\phi_{j,b} - \theta^T\phi_{i,a})\phi_{i,a} \mid w \in \overline{w(\theta)^\epsilon} \right\},$$

where $\overline{w(\theta)^\epsilon}$ denotes the closure of the set $w(\theta)^\epsilon$. In vector-matrix notation, $h(\theta)$ can be written as

$$h(\theta) = \{\Phi^T \mathbb{F}_w(T_w(\Phi\theta) - \Phi\theta) \mid w \in \overline{w(\theta)^\epsilon}\}.$$

Note that $h(\theta)$ is a compact set since $f_w(i,a)$ and $p_w(i,a;j,b)$ are both continuous functions of w (cf. Assumption 2 and Lemma 1) and $\overline{w(\theta)^\epsilon}$ is a compact set for any θ (since $w(\theta)^\epsilon$ is a bounded set). Note also that y_n in (3.18) lies in the set $h(\theta_n)$ for each n.

Now recall that

$$\begin{aligned}Y(n+1) = &\left(g(X_n, Z_n) + \gamma\theta_n^T\phi_{X_{n+1},Z_{n+1}} - \theta_n^T\phi_{X_n,Z_n}\right)\phi_{X_n,Z_n}\\ &- E\left[\left(g(X_n, Z_n) + \gamma\theta_n^T\phi_{X_{n+1},Z_{n+1}} - \theta_n^T\phi_{X_n,Z_n}\right)\phi_{X_n,Z_n} \mid \mathcal{G}(n)\right].\end{aligned}$$

It is easy to see that

$$Y(n+1) \in \left\{ \gamma\theta_n^T\left(\phi_{X_{n+1},Z_{n+1}} - \sum_{(j,b)} p_w(X_n, Z_n; j, b)\phi_{j,b}\right)\phi_{X_n,Z_n} \mid w \in \overline{w(\theta_n)^\epsilon} \right\}.$$

Again since $\{p_w(i,a;j,b) \mid w \in \overline{w(\theta)^\epsilon}\}$ is a compact set for every θ, and X_n, Z_n take values from given finitely-valued sets, for each n, it follows that the conditional distribution of $Y(n+1)$ given $\mathcal{G}(n)$ has a compact support (say) $\mathcal{A}(\theta_n)$.

Let

$$\hat{\Gamma}_\theta(h(\theta)) \stackrel{\triangle}{=} \bigcap_{\epsilon>0} \bar{co}\left(\bigcup_{\|\beta-\theta\|<\epsilon} \{\gamma_1(\beta; y + Y) \mid y \in h(\beta), Y \in \mathcal{A}(\beta)\}\right),$$

where $\bar{co}(\cdot)$ denotes the closed and convex hull of '$\cdot$'. Then z_n in (3.20) satisfies $z_n \in \hat{\Gamma}_{\theta_n}(h(\theta_n))$ almost surely. We now have the following result:

Proposition 2 *We have*

(i) $\hat{\Gamma}_\theta(h(\theta))$ *is a convex and compact set for any* $\theta \in D$.
(ii) *For all* $\theta \in D$,

$$\sup_{\beta \in \hat{\Gamma}_\theta(h(\theta))} \| \beta \| < K(1 + \| \theta \|)$$

for some $K > 0$.
(iii) $\hat{\Gamma}_\theta(h(\theta))$ *is upper-semicontinuous, i.e., if* $\theta_n \to \theta$ *and* $\beta_n \to \beta$ *with* $\beta_n \in \hat{\Gamma}_{\theta_n}(h(\theta_n))$ $\forall n$, *then* $\beta \in \hat{\Gamma}_\theta(h(\theta))$.

Proof We start by showing (i). It is easy to see from its definition that $\hat{\Gamma}_\theta(h(\theta))$ is a convex and closed set for any θ. Further, it is easy to see that $\hat{\Gamma}_\theta(h(\theta))$ is bounded as well, implying that the same is a compact set.

For showing (ii), note that since $\hat{\Gamma}_\theta(h(\theta))$ is compact for each θ, we have

$$R \stackrel{\triangle}{=} \sup_{\theta \in D} \sup_{\beta \in \hat{\Gamma}_\theta(h(\theta))} \| \beta \| < \infty,$$

since D is a compact set. Thus, in particular, the claim holds with $K = R$.

Now consider (iii). Let $g(\beta) \stackrel{\triangle}{=} \bigcup\{\gamma_1(\beta; y + Y) \mid y \in h(\beta), Y \in \mathcal{A}(\beta)\}$. Then

$$\hat{\Gamma}_\theta(h(\theta)) = \bigcap_{\epsilon > 0} \bar{co}(\{g(\beta) \mid\mid \beta - \theta \mid\mid < \epsilon\}).$$

Let $H(\theta, \epsilon) \stackrel{\triangle}{=} \bar{co}(\{g(\beta) \mid\mid \beta - \theta \mid\mid < \epsilon\})$. Then $H(\theta, \epsilon)$ is a family of diminishing sets, i.e., $H(\theta, \epsilon_2) \subset H(\theta, \epsilon_1)$ if $\epsilon_2 < \epsilon_1$ and $H(\theta, \epsilon) \downarrow \hat{\Gamma}_\theta(h(\theta))$ as $\epsilon \downarrow 0$. Now let $\beta_n \in \hat{\Gamma}_{\theta_n}(h(\theta_n))$ for some n large. Then $\beta_n \in H(\theta_n, \frac{\epsilon}{3}) = \bar{co}(\{g(\eta) \mid\mid \eta - \theta_n \mid\mid < \frac{\epsilon}{3}\})$ for some $\epsilon > 0$. Now for large n, $\| \theta - \theta_n \| < \frac{\epsilon}{3}$. Further,

$$\| \eta - \theta \| \leq \| \eta - \theta_n \| + \| \theta_n - \theta \| < \frac{2\epsilon}{3}.$$

Thus, it is easy to see that $H(\theta_n, \frac{\epsilon}{3}) \subset H(\theta, \frac{2\epsilon}{3})$ for n large. Now since $\beta_n \in H(\theta, \frac{2\epsilon}{3})$ for all $n > N_0$ for some $N_0 > 0$ and $\beta_n \to \beta$, it follows that $\beta \in H(\theta, \frac{2\epsilon}{3})$, because $H(\theta, \frac{2\epsilon}{3})$ is a closed set. Finally, since $\epsilon > 0$ is arbitrary and $H(\theta, \frac{2\epsilon}{3}) \downarrow \hat{\Gamma}_\theta(h(\theta))$ as $\epsilon \downarrow 0$, it follows that $\beta \in \hat{\Gamma}_\theta(h(\theta))$. The claim follows. □

Before proceeding further, we recall a few definitions (see Benaim et al. 2005, Borkar 2008). Consider the following DI:

$$\dot{x}(t) \in g(x(t)), \tag{3.21}$$

where $g(\cdot)$ satisfies (i)–(iii) of Proposition 2 (in place of $\hat{\Gamma}_\theta(h(\theta))$).

Definition 1 1. A set B is invariant for the DI (3.21) if for $x \in B$, there is a trajectory $x(t), t \in (-\infty, \infty)$ with $x(0) = x$, that lies entirely in B.
2. A set B is said to be internally chain transitive for the DI (3.21) if it is compact and for any $x, y \in B$ and every $\epsilon, T > 0$, there exists an integer $n \geq 1$, solutions $x_1, \ldots, x_n$ to the DI (3.21), and real numbers $t_1, \ldots, t_n$ (all) greater than T such that

(a) $x_i(s) \in B$ for all $0 \leq s \leq t_i$, and all $i = 1, \ldots, n$,
(b) $\| x_i(t_i) - x_{i+1}(0) \| \leq \epsilon$ for all $i = 1, \ldots, n-1$,
(c) $\| x_1(0) - x \| \leq \epsilon$ and $\| x_n(t_n) - y \| \leq \epsilon$.

The reader is referred to Benaim et al. (2005) for various notions of invariant sets. It is also shown in Lemma 3.5 of Benaim et al. (2005) that an internally chain transitive set of a differential inclusion is invariant.

Let $t(n), n \geq 0$ be a sequence of nonnegative real numbers defined according to $t(0) = 0$ and $t(n) = \sum_{m=0}^{n-1} b(m), n \geq 1$. Consider now the following differential inclusion (DI) associated with (2.8):

$$\dot{\theta}(t) \in \hat{\Gamma}_\theta(h(\theta(t))). \tag{3.22}$$

Define $\bar{\theta}(\cdot)$ according to $\bar{\theta}(t(n)) = \theta_n, n \geq 0$, with linear interpolation on each interval $[t(n), t(n+1)]$. Let $G = \bigcap_{t \geq 0} \overline{\{\bar{\theta}(t+s) : s \geq 0\}}$. Under Proposition 2, the DI (3.22) is guaranteed to have at least one solution that is absolutely continuous, see Aubin and Cellina (1984) for details.

Theorem 3 *The iterates $\theta_n, n \geq 0$ of the QW-FA algorithm converge to G almost surely. Further, the set G is a closed connected internally chain transitive invariant set of* (3.22).

Proof Note again that as a consequence of the projection Γ_1, $\sup_n \| \theta_n \| < \infty$ almost surely. Now as a consequence of Proposition 2, the results in Chapter 5 of Borkar (2008) (specifically Lemma 1-Theorem 2, pp. 53; Lemma 3, pp. 54; and Corollary 4, pp. 55) apply. The claim follows from Corollary 4, pp.55 of Borkar (2008)[Chapter 5]. □

Remark 3 Theorem 3 shows that the sequence $\theta_n, n \geq 0$ of iterates converges to the set G of limit points of these and that G is also a closed connected internally chain transitive invariant set of the associated DI (3.22). Again note that the set G depends on the samples θ_n (of these iterates) and in general may contain limit cycles of (3.22).

Also note that if instead of (3.22), the iterates $\theta_n, n \geq 0$ were to track a well-posed ODE, the result in Theorem 3 would be considerably stronger because notions of invariance and internal chain transitivity in such a scenario would be defined with respect to the underlying trajectory of the ODE, that would also be unique for any given initial condition.

While we have shown the convergence of QW-FA to the set G, we did not provide performance bounds on the obtained policies in relation to the optimal policy for the underlying MDP. Similarly, it would be of interest to obtain error bounds between the obtained and the optimal Q-values. For a fixed policy, such bounds have been provided in (Tsitsiklis and Van Roy 1997; Tsitsikis and Van Roy 1999), for the case of TD algorithms.

4 Numerical results – application to multistage routing

We consider the problem of routing in a (multi-stage) queueing network. The network has multiple stages with each queue in a given stage connected to all queues in the next stage, see Fig. 1. We assume that each queue has a unique server. Packets arrive at the source node **s** according to a Poisson process with rate λ.

After a packet completes service at a node, it is routed to a node in the next stage and this process is repeated until the destination node **d** is reached. The objective is to find a route that minimizes the expected delay from the source to the destination nodes. We assume for simplicity that service times at each node are exponentially distributed. However, in general, the parameter of the exponential distribution for the service times can be different at different nodes.

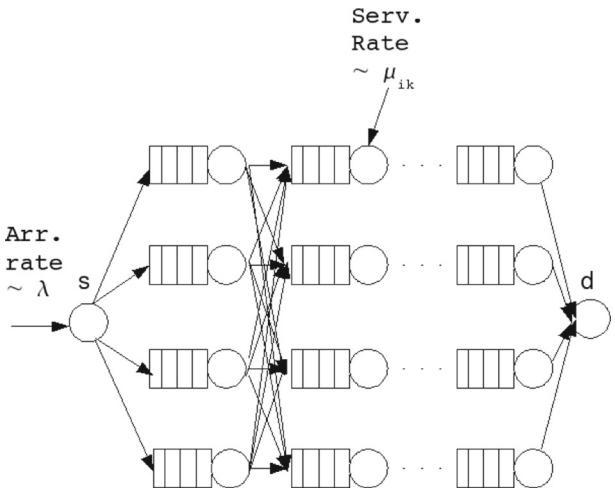

Fig. 1 Multi-stage routing problem

Let the total number of nodes be $\bar{N}$ with each stage having a given number $M = \bar{N}/L$ of nodes, where L is the number of stages in the network (we assume for simplicity here that $\bar{N}$ is a multiple of L). As explained previously, nodes at each stage except the last are connected to all the M nodes in the next stage. At the last stage all packets are routed to the destination node. Hence routing decisions are made for the first $\bar{N} - M + 1 = M(L-1)+1$ nodes that include the source node and exclude nodes in the last stage. We assume that all decisions are made by a central controller that has access to precise state information at each instant. The controller does its own computation of the routes that packets should take after service at each node and transmits this information back to the nodes. We assume no propagation and feed back delays in this process. We formulate the routing problem as a MDP and apply both the regular Q-learning algorithm with function approximation (QL-FA) as well as our proposed algorithm QW-FA. We also study performance comparisons with an actor-critic (AC) scheme, that is a similar algorithm as QW-FA except with the faster and slower timescales reversed.

The state $X_n \in S$ of the system at time n is the vector of queue lengths $\left(q_i(n),\, i = 1, \ldots, \bar{N}\right)^T$ at each of the nodes. The action $Z_n = \left(Z_n(i),\, i = 0, \ldots, (\bar{N} - M)\right)^T \in A(X_n)$ at time n is the vector of choices for the next nodes to route the packets after service at each of the first $(\bar{N} - M + 1)$ nodes. Thus, $Z_n(i) = j$ if after service at node $i \in \{0, 1, \ldots, (\bar{N} - M)\}$, the choice of the next node to route the packet to is j. We define the routing probabilities $p_{ij}(n),\ i \in \{0, 1, \ldots, (\bar{N} - M)\},\ j \in \{M \cdot \lceil i/M \rceil + 1, \ldots, M \cdot \lceil i/M \rceil + (M + 1)\},\ n \geq 0$ according to

$$p_{ij}(n) = \begin{cases} 1 & \text{if } Z_n(i) = j \\ 0 & o.w. \end{cases}$$

The single stage cost in state X_n when action Z_n is chosen is the expected sum of queue lengths along the path from source to destination,

$$g(X_n, Z_n) = \sum_{l=0}^{(\bar{N}-2M)/M} \sum_{i=lM+1}^{(l+1)M} \sum_{j=(l+1)M+1}^{(l+2)M} \left(p_{ij}(n) \cdot q_j(n)\right) + \sum_{j=1}^{M} p_{0j}(n) \cdot q_j(n).$$

Table 1 QL-FA: Expected total delay for different step-sizes $c(n) = 1/n^{\alpha}$, for various values of λ in a 2×4 network

α	$\lambda = 10$	$\lambda = 20$	$\lambda = 30$
0.95	2.18 ± 0.67	4.16 ± 0.35	5.94 ± 0.718
0.85	2.14 ± 0.571	4.2 ± 0.261	6.08 ± 0.5
0.75	2.2 ± 0.599	4.24 ± 0.275	5.94 ± 0.797
0.65	2.03 ± 0.312	4.28 ± 0.238	5.98 ± 0.698
0.55	2.13 ± 0.32	4.1 ± 0.532	5.19 ± 1.19

We selected the features ϕ_{X_n,Z_n} in both algorithms QL-FA and QW-FA as follows:

$$\phi_{X_n,Z_n} = \left(\phi_{q_i(n)} . \phi_{a_j(n)},\ i = 1, \ldots, \bar{N},\ j = 0, \ldots, (\bar{N} - M)\right)^T,$$

where

$$\phi_{q_i(n)} = \begin{cases} 0 & \text{if } q_i(n) < L_1 \\ 0.5 & \text{if } L_1 \le q_i(n) < L_2 \\ 1 & \text{o.w.,} \end{cases}$$

$$\phi_{a_i(n)} = \left(p_{ij}(n),\ j = M \cdot \lceil i/M \rceil + 1, \ldots, M \cdot \lceil i/M \rceil + (M+1)\right)^T.$$

Here L_1 and L_2 are two threshold levels for the queue lengths. We let $L_1 = 1$ and $L_2 = 2$ in the experiments. Further, we let $M = 4$ (the number of nodes in each stage) for simplicity. The servers at each stage are non-identical with the first server in each of these stages having a ten times lower service rate than the other three that serve with a common rate. We let the compact set in which w takes values be $C = [-100, 100]^N$, where $N = \bar{N} \cdot (\bar{N} - M) \cdot M$ in the experiments. Also, we did not project the θ-updates in our experiments as they were seen to remain bounded as such. We run both QL-FA and QW-FA algorithms with 100 different initial seeds and for 50,000 iterations in each simulation run. In the tables below, we show both the mean and the standard error from these simulation runs upon termination.

Note that there are no queues at the source and destination nodes. As soon as a packet arrives at the source node, it is forwarded to one of the nodes in the next stage where it gets queued if there are packets waiting for service at that node. Further, packets are served according to the first come first serve (FCFS) schedule at each node. The service rate is set

Table 2 QW-FA: Expected total delay for different step-sizes $a(n) = 1/n^{\alpha}$, $b(n) = 1/n^{\beta}$ with $\alpha < \beta$, and for different arrival rates λ, on a 2×4 network

β	α	$\lambda = 10$	$\lambda = 20$	$\lambda = 30$
0.95	0.55	1.47 ± 0.141	2.17 ± 0.171	2.86 ± 0.237
0.95	0.65	1.69 ± 0.239	2.48 ± 0.279	3.26 ± 0.422
0.95	0.75	1.86 ± 0.335	2.81 ± 0.607	3.7 ± 0.748
0.95	0.85	2.06 ± 0.595	3.07 ± 0.973	3.89 ± 1.214
0.85	0.55	1.44 ± 0.11	2.12 ± 0.169	2.8 ± 0.191
0.85	0.65	1.68 ± 0.245	2.55 ± 0.348	3.32 ± 0.467
0.85	0.75	1.89 ± 0.379	2.95 ± 0.548	3.77 ± 0.584
0.75	0.55	1.43 ± 0.142	2.14 ± 0.173	2.76 ± 0.21
0.75	0.65	1.69 ± 0.262	2.56 ± 0.336	3.23 ± 0.389
0.65	0.55	1.43 ± 0.127	2.1 ± 0.177	2.77 ± 0.237

Table 3 AC : Total delay (mean ± standard error) for different step-sizes $a(n) = 1/n^{\alpha}$, $b(n) = 1/n^{\beta}$ with $\beta < \alpha$, and for different arrival rates λ, on a 2×4 network

β	α	$\lambda = 10$	$\lambda = 20$	$\lambda = 30$
0.55	0.95	2.28 ± 0.697	3.18 ± 1.145	3.97 ± 0.843
0.65	0.95	2.33 ± 0.70	3.49 ± 1.566	4.18 ± 1.48
0.75	0.95	2.16 ± 0.645	3.44 ± 1.384	4.17 ± 1.629
0.85	0.95	2.23 ± 0.689	3.32 ± 1.226	4.31 ± 1.601
0.55	0.85	2.12 ± 0.532	3.09 ± 1.078	3.78 ± 0.80
0.65	0.75	2.11 ± 0.545	3.26 ± 1.215	3.95 ± 0.974
0.75	0.65	2.06 ± 0.518	3.23 ± 1.015	4.01 ± 0.902
0.55	0.75	1.96 ± 0.376	2.83 ± 0.604	3.55 ± 0.432
0.65	0.75	1.96 ± 0.407	2.87 ± 0.422	3.62 ± 0.456
0.55	0.65	1.75 ± 0.234	2.53 ± 0.393	3.27 ± 0.447

at $\mu = 5$ for the first node in each stage and at $\mu = 50$ for every other node. Thus the first node in each stage serves at a rate that is ten times less as compared to the other nodes. In all our experiments, except those whose results are shown in Tables 5 and 6, we set the discount factor γ at 0.9.

In our first set of experiments, we consider a 2×4-network, i.e., one with two stages and four nodes in each stage, and study the performance of both algorithms QL-FA and QW-FA over a range of step-size parameters. We consider step-sizes within the following classes for the two algorithms: For QL-FA, we let $c(n) = 1/n^{\alpha}$ and vary α in the range $(0.5, 1.0)$. Similarly for QW-FA, we let $a(n) = 1/n^{\alpha}$ and $b(n) = 1/n^{\beta}$, respectively, with $\alpha, \beta \in (0.5, 1.0)$. Table 1 shows the expected total delay (i.e., along the entire path from source to destination) obtained when QL-FA is used and the step-size parameter α is varied as described above. Similarly, Table 2 shows the expected total delay when QW-FA is used and the step-size parameters α and β in this case are varied with $\alpha < \beta$. The last condition ensures that $b(n) = o(a(n))$, see (2.7). We consider cases when the arrival rate is chosen to be $\lambda = 10, 20$ and 30, respectively. We observe from Table 1 that a step-size of $c(n) = 1/n^{0.55}$ shows, on the whole, better results than the other choices in the case of QL-FA. From Table 2, it is seen that $a(n) = 1/n^{0.55}$ shows the best results for any given choice

Table 4 QW-FA : Total delay (mean ± standard error) for different values of δ and for different arrival rates λ, on a 2×4 network

δ	$\lambda = 10$	$\lambda = 20$	$\lambda = 30$
0.001	2.33 ± 0.128	3.28 ± 0.188	4.11 ± 0.263
0.002	1.69 ± 0.098	2.37 ± 0.154	3.1 ± 0.211
0.003	1.5 ± 0.105	2.2 ± 0.18	2.83 ± 0.209
0.004	1.47 ± 0.11	2.12 ± 0.158	2.79 ± 0.199
0.005	1.45 ± 0.119	2.09 ± 0.169	2.79 ± 0.224
0.007	1.51 ± 0.161	2.24 ± 0.25	2.97 ± 0.269
0.01	1.6 ± 0.198	2.45 ± 0.403	3.15 ± 0.345
0.03	1.91 ± 0.372	2.92 ± 0.554	4.0 ± 0.596
0.05	2.05 ± 0.438	3.16 ± 0.647	4.26 ± 1.047

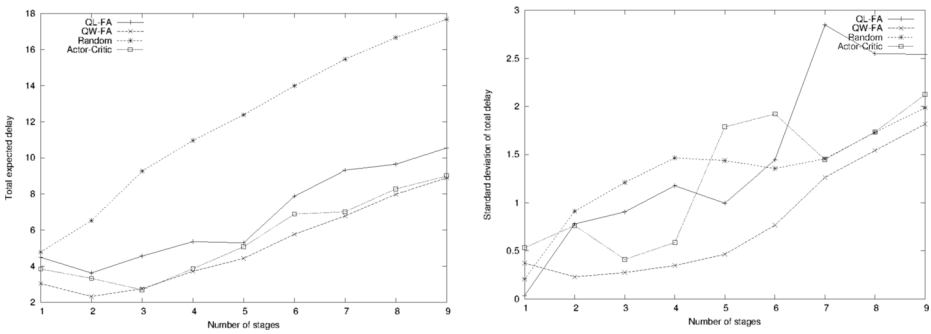

Fig. 2 Total expected delay and standard error obtained from 100 simulation runs with different initial seeds for different network sizes using the algorithms

of $b(n)$ in the case of QW-FA. In Table 3, we consider the AC algorithm that is the same as QW-FA except with reversed time scales, i.e., $\beta < \alpha$. Thus, the update of the weight parameter w is now on the slower time scale. It can be seen from the results in this table that the performance of the algorithm deteriorates in both the mean and the standard error in comparison to QW-FA, i.e., when $\alpha < \beta$ (Table 2). We also observed that for networks with more number of stages, $\alpha = 0.55$ for QL-FA and $\alpha = 0.55$, $\beta = 0.85$ for QW-FA, respectively, show the best results overall. Hence, in our subsequent experiments, we set the aforementioned values for the step-size parameters.

Next, in Table 4, we show the effect of varying the parameter δ in the recursion (2.9), on the performance of the QW-FA algorithm over a 2×4-network. We vary the value of δ between 0.001 and 0.05. From the table, it can be seen that for all arrival rates, the best performance is obtained around 0.005. This was also the case with other network sizes. Hence, we fix the value of δ to be 0.005 in all our subsequent experiments.

In our next set of experiments, we study performance comparisons between QL-FA, QW-FA, and the AC algorithms in terms of the total delay using both the mean and the standard error metrics on networks of various sizes. For the AC algorithm, we let the tuple (β, α) in Table 3 to be (0.55, 0.65) as it is seen to be the best setting for the AC algorithm across all settings considered in Table 3. Figure 2 shows the plots of the mean and standard error of the total expected delay as functions of the number of stages in the network that is in turn varied between 1 and 9. Thus the network sizes here are varied between 1×4 and 9×4. The total arrival rate at the source node **s** is set at $\lambda = 20$ in all cases. Further, we let the discount factor to be $\gamma = 0.9$ for all the network sizes.

Table 5 Expected total delay for different arrival rates λ and discount factors γ in a 2×4 network

γ	$\lambda = 10$		$\lambda = 20$		$\lambda = 30$	
	QL-FA	QW-FA	QL-FA	QW-FA	QL-FA	QW-FA
0.9	2.19 ± 0.339	1.43 ± 0.107	3.63 ± 0.862	2.3 ± 0.228	5.05 ± 1.13	3.33 ± 0.453
0.7	2.01 ± 0.389	1.45 ± 0.138	4.09 ± 0.584	2.35 ± 0.277	5.47 ± 1.027	3.36 ± 0.408
0.5	2.06 ± 0.33	1.47 ± 0.132	4.25 ± 0.351	2.48 ± 0.54	5.56 ± 0.754	3.25 ± 0.652
0.3	2.13 ± 0.32	1.45 ± 0.119	4.1 ± 0.532	2.12 ± 0.15	5.09 ± 1.291	2.82 ± 0.227
0.1	2.18 ± 0.411	1.46 ± 0.144	4.17 ± 0.468	2.12 ± 0.162	4.81 ± 1.092	2.82 ± 0.216

Table 6 Expected total delay for different arrival rates λ and discount factors γ in a 3×4 network

γ	$\lambda = 10$		$\lambda = 20$		$\lambda = 30$	
	QL-FA	QW-FA	QL-FA	QW-FA	QL-FA	QW-FA
0.9	2.56 ± 0.357	2.34 ± 0.182	4.56 ± 0.905	2.75 ± 0.274	5.72 ± 1.85	2.98 ± 0.302
0.8	2.6 ± 0.615	2.38 ± 0.207	4.38 ± 1.134	2.69 ± 0.246	5.69 ± 1.622	2.93 ± 0.30
0.7	2.67 ± 0.596	2.38 ± 0.302	4.16 ± 1.045	2.77 ± 0.283	5.65 ± 1.698	2.94 ± 0.301
0.6	2.69 ± 0.616	2.43 ± 0.222	4.15 ± 1.018	2.77 ± 0.236	5.47 ± 1.407	2.96 ± 0.29
0.5	2.64 ± 0.518	2.52 ± 0.554	4.33 ± 1.183	2.79 ± 0.301	5.25 ± 1.549	2.94 ± 0.326

We also make performance comparisons in this case with the algorithm termed "Random" that uses fixed and equal probabilities for selecting each of the queues in the 'next' stage. It can be seen that QW-FA almost always exhibits the lowest mean delay when compared with the other algorithms. It is also interesting to observe that the AC algorithm outperforms QL-FA as well in most cases. Further, as expected, all three algorithms show better results as compared to "Random". The standard error performance of QW-FA is again the best overall. In fact, on networks of sizes 7×4 to 9×4, the standard error obtained using QL-FA is significantly higher than all the other algorithms including "Random". The standard error obtained from AC is low for smaller-sized networks. However, for networks of sizes 5×4 and 6×4, the AC algorithm shows poor performance as well. Thus, we observe that QW-FA is more robust and exhibits superior numerical performance when compared with QL-FA as well as the AC algorithms.

Next, in Tables 5 and 6, we show the results of experiments on 2×4 and 3×4 networks, respectively, using QL-FA and QW-FA for different values of the discount factor γ and arrival rate λ. We observe that, QW-FA shows significantly better results in both the mean

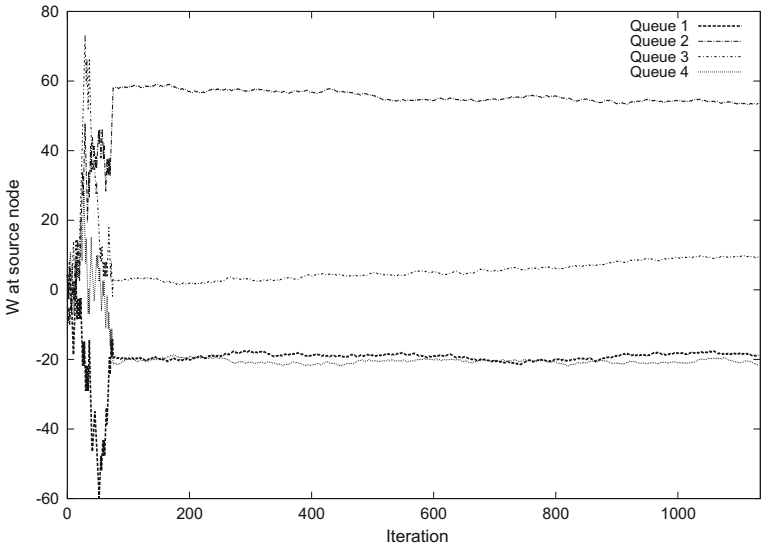

Fig. 3 Plots of convergence of four w-components in QW-FA for the source node in a 2×4 network with $\lambda = 20$ for states with queue lengths above L_2 at each node in the first stage

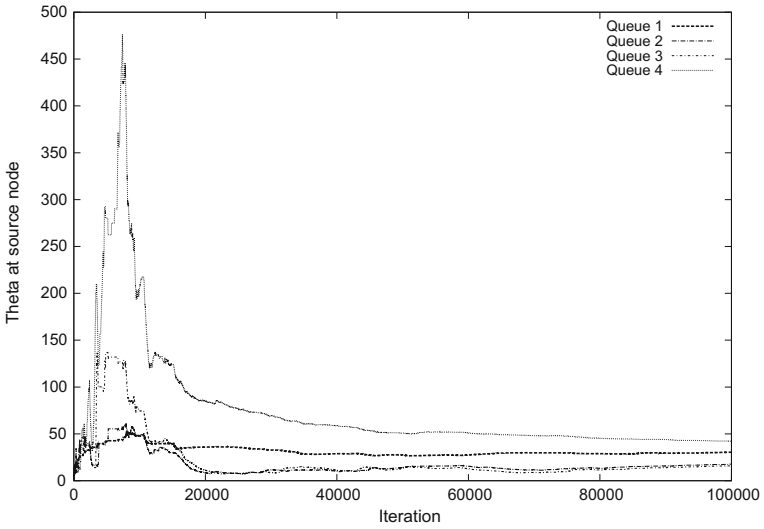

Fig. 4 Plots of convergence of four θ-components in QW-FA for the source node in a 2×4 network with $\lambda = 20$ for states with queue lengths above L_2 at each node in the first stage

and the standard error (of the total delay) performances, in almost all cases, over QL-FA. For higher values of λ, the standard error performance when QL-FA is used deteriorates significantly in comparison to the same when QW-FA is used.

Next, in Figs. 3 and 4, we show plots of convergence of some of the components of the w and θ parameters, respectively, of QW-FA, for a 2×4-network, as functions of the iteration number for a single simulation run. In this case, both w and θ are vectors of dimension $\bar{N} \cdot (\bar{N} - M) \cdot M = 9 \cdot 5 \cdot 4 = 180$ each. The graphs show the convergence of four of

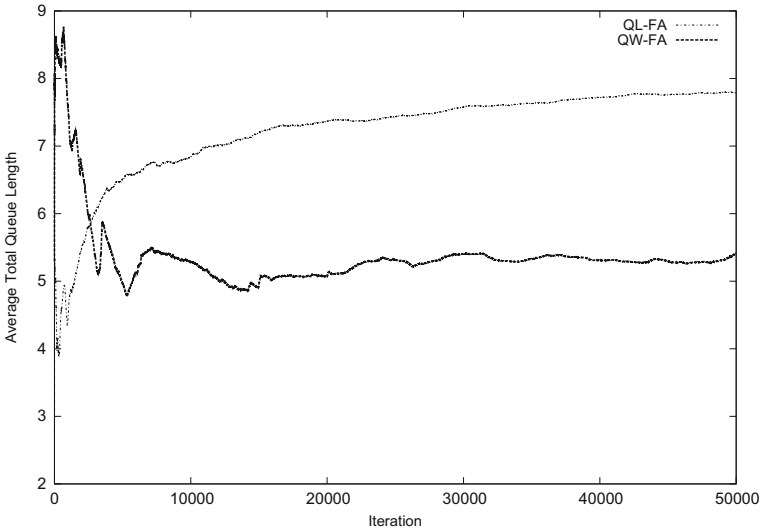

Fig. 5 Average total queue lengths for QL-FA and QW-FA on a 2×4 network with $\lambda = 20$

the parameter components that correspond to the state with queue lengths above L_1 at all the nodes in the first stage out of the total of 180 parameter components. Finally, in Fig. 5, we show plots for both QL-FA and QW-FA, respectively, of the average total queue lengths on a 2×4 network with $\lambda = 20$, as functions of the iteration count (for these algorithms). From the figure, it is seen that the average total queue length for QW-FA is higher initially, however, as the iterations in the two algorithms progress, the average total queue length using QW-FA becomes lower. The algorithm QW-FA consistently gives a lower average total queue length as compared to QL-FA.

5 Conclusions

We presented in this paper a multi-scale version of Q-learning with function approximation. Our algorithm has the advantage that, unlike Q-learning, it does not suffer from high oscillations. This is because our algorithm incorporates two-timescale stochastic recursions, whereby the explicit minimization over instantaneous Q-value updates in Q-learning is replaced with a gradient search in the (parameterized) policy space along a faster timescale. This is seen to help in the convergence of the algorithm as the faster (w) iterates, for any given θ, track the stationary average Q-value, and thus unlike Q-learning, the minimization does not result in high variance/instabilities in the scheme. The gradient estimate that we use for the local search procedure (on the faster scale) is a one-simulation estimate incorporating Hadamard matrix based deterministic perturbations. Our procedure is seen to be computationally efficient in our experiments. A global search scheme such as simulated annealing may be used for the faster timescale update, however, global search procedures, in general, are known to be computationally expensive. In Gelfand and Mitter (1991), addition of a slowly diminishing Gaussian noise sequence within a local search procedure is proposed, in the spirit of simulated annealing. Such a recursion may be used for our faster timescale update.

We showed the almost sure convergence of our algorithm to a closed connected internally chain transitive invariant set of an associated differential inclusion. As an application setting, we considered a problem of routing in multi-stage queueing networks. We compared the performance of our algorithm with Q-learning as well as an actor-critic scheme (that is similar to QW-FA but with timescales reversed) on various network configurations. We observed that our algorithm is more robust as compared to both Q-learning and the actor-critic scheme as it gives significantly lower standard error. Further, our algorithm also exhibits better mean performance as compared to both Q-learning and actor-critic. The large variation in performance across different seeds in the case of Q-learning could be the result of problems due to off-policy learning. Similar observations have also been made in Prashanth et al. (2014) on the numerical performance of our algorithm (in both discounted and average cost cases) in relation to corresponding Q-learning algorithms on a problem of finding optimal sleep-wake schedules for individual sensor nodes in a wireless sensor network, while tracking potential intruder movement.

A possible future direction would be to develop Hessian based algorithms, see for instance, Bhatnagar (2005, 2007); Bhatnagar et al. (2013) as well as algorithms that incorporate functional (inequality) constraints (Bhatnagar and Lakshmanan 2012). Finally, the analysis that we carried out in this paper is only a first step and more detailed convergence analyses of this and similar algorithms must be carried out in the future. In particular, it would be interesting to extend the analysis to show stability of the iterates when projection

is not used. Analyses of performance and error bounds between the obtained and optimal policies as well as Q-values must also be carried out.

Acknowledgments The authors thank the Editor Prof. C. G. Cassandras, the Associate Editor, and all the anonymous reviewers for their detailed comments and criticisms on the various drafts of this paper, that led to several corrections in the proof and presentation. In particular, the authors gratefully thank the reviewer who suggested that they follow a differential inclusions based approach for the slower scale dynamics. The authors thank Prof. V. S. Borkar for helpful discussions. This work was partially supported through projects from the Department of Science and Technology (Government of India), Xerox Corporation (USA), and the Robert Bosch Centre (Indian Institute of Science).

Appendix

In this section, we present detailed proofs of some of the results given in Section 3.

Proof of Proposition 1 Note that the n-step ($n > 1$) transition probability of going from state (i, a) to (j, b) is

$$p_w^n(i, a; j, b) = P(X_n = j, Z_n = b \mid X_0 = i, Z_0 = a, \pi_w)$$

$$= q_w^n(i, a, j)\pi_w(j, b),$$

where $q_w^n(i, a, j)$ is the n-step probability of going to state j when the initial state is i and action a is chosen (in state i), while actions in other stages (from 1 to $n-1$) are chosen according to the SRP π_w. It is easy to see that $\sum_{j \in S} q_w^n(i, a, j) = 1, \forall i \in S, a \in A(i)$.

Let $l \in S$ be such that $p(i, a, l) > 0$. Now from Assumption 1, X_n, $n \geq 0$, under any SRP π_w is irreducible. Thus, given SRP π_w and states l, j, there exists an integer $n_1 > 0$ such that

$$p^{n_1}(l, j, \pi_w) \stackrel{\triangle}{=} P(X_{n_1} = j \mid X_0 = l, \pi_w) > 0.$$

Note that in estimating $p^n(l, j, \pi_w)$, it is assumed that the actions at each of the n stages are picked according to the policy π_w. This is unlike estimating $q_w^n(l, a, j)$ where the first action to be picked is a in state l while the actions in the remaining $n-1$ stages are picked according to π_w. Now observe that

$$p_w^n(i, a; j, b) \geq p(i, a, l)p^{n-1}(l, j, \pi_w)\pi_w(j, b).$$

Thus, $p_w^{n_1+1}(i, a; j, b) > 0$. Similarly, it can be shown that there exists an integer $n_2 > 0$ such that $p_w^{n_2+1}(j, b; i, a) > 0$. Thus, $\{(X_n, Z_n)\}$ is an irreducible Markov chain when $Z_n, n \geq 0$ are obtained according to π_w.

Next, we show that $\{(X_n, Z_n)\}$ is aperiodic. Again let $l \in S$ be such that $p(i, a, l) > 0$. Since the process $\{X_n\}$ is aperiodic under π_w, from Assumption 1, there exists an integer

$M > 0$ such that $p^n(l, l, \pi_w) > 0 \ \forall n \geq M$, see for instance, Lemma 5.3.2, pp.99, of (Borkar 1995). By irreducibility of $\{X_n\}$ under π_w, there exists $n_3 > 0$ (integer) such that $p^{n_3}(l, i, \pi_w) > 0$. Now note that

$$p_w^{1+n+n_3}(i, a; i, a) \geq p(i, a, l)p^n(l, l, \pi_w)p^{n_3}(l, i, \pi_w)\pi_w(i, a)$$
$$> 0, \forall n \geq M.$$

Thus, $p_w^n(i, a; i, a) > 0 \ \forall n \geq (1 + M + n_3)$. Hence, $\{(X_n, Z_n)\}$ is aperiodic under π_w as well. Finally, since $S \times A(S)$ is a finite set, $\{(X_n, Z_n)\}$ is also positive recurrent. The claim follows. □

Proof of Lemma 1 We shall use a key result from Schweitzer (1968) for the proof. Let $P_w^\infty = \lim_{m\to\infty} \frac{1}{m} \sum_{n=1}^{m} P_w^n$ and $Z_w \stackrel{\triangle}{=} (I - P_w - P_w^\infty)^{-1}$, respectively, where I denotes the $(|S \times A(S)| \times |S \times A(S)|)$-identity matrix and P_w^m is the matrix of m-step transition probabilities $p_w^m(i, a; j, b)$, $i, j \in S$, $a \in A(i)$, $b \in A(j)$. From Theorem 2, pp.402-403 of (Schweitzer 1968), one can write

$$\mho_{w+\xi e_i} = \mho_w(I + (P_{w+\xi e_i} - P_w)Z_w + o(\xi)),$$

where $\xi > 0$ is a small quantity and $e_i, i \in \{1, \ldots, N\}$ is a unit vector with 1 as its ith entry and 0s elsewhere. Hence, we get

$$\nabla_{w,i}\mho_w = \mho_w \nabla_{w,i} P_w Z_w, i = 1, \ldots, N.$$

Thus, $\nabla_w \mho_w = \mho_w \nabla P_w Z_w$. Now since $p_w(i, a; j, b) = p(i, a, j)\pi_w(j, b)$, it follows from Assumption 2, it follows that $p_w(i, a; j, b)$ are continuously differentiable, i.e., $\nabla_w P_w$ exists and is continuous. Hence, $\nabla_w \mho_w$ exists.

Next we verify that $\nabla_w \mho_w$ is continuous as well. Note that $\mho_w$ is continuous since it is differentiable. Further, $\nabla_w P_w$ is continuous as noted above. Also, from Cramer's rule, it follows that Z_w is continuously differentiable and hence also continuous over $w \in C$. Since the set C is a compact subset of $\mathcal{R}^N$, it is easy to see that $\nabla_w \mho_w$ is continuous as well. The claim follows. □

Proof of Lemma 2 It is easy to see from the definition of $R(\theta, w)$ and Lemma 1 that the partial derivatives of $R(\theta, w)$ with respect to any $\theta \in \mathcal{R}^d$ and $w \in C$ exist. Note that from definition, for a given $w \in C$,

$$\nabla_\theta R(\theta, w) = \sum_{(i,a)\in S\times A(S)} f_w(i, a)\phi_{i,a},$$

which is a constant function of θ, hence continuous. Now consider

$$\nabla_w R(\theta, w) = (\nabla_{w,1} R(\theta, w), \ldots, \nabla_{w,N} R(\theta, w))^T,$$

where $\nabla_{w,i} R(\theta, w)$ is the partial derivative of $R(\theta, w)$ with respect to w_i, given $\theta \in D$. Note that $\sup_{\theta\in D} \| \theta \| < \infty$, since D is bounded. Now, given $\theta \in D$,

$$\nabla_w R(\theta, w) = \sum_{(i,a)\in S\times A(S)} \nabla_w f_w(i, a)\theta^T \phi_{i,a},$$

since $S \times A(S)$ is a finite set. Let w^1 and w^2 be two points in C. Then,

$$\| \nabla_w R(\theta, w^1) - \nabla_w R(\theta, w^2) \|$$
$$\leq \sum_{(i,a)\in S\times A(S)} \| \nabla_w f_{w^1}(i, a)\theta^T \phi_{i,a} - \nabla_w f_{w^2}(i, a)\theta^T \phi_{i,a} \|$$

$$\leq \sum_{(i,a)\in S\times A(S)} \parallel \nabla_w f_{w^1}(i,a) - \nabla_w f_{w^2}(i,a) \parallel |\theta^T \phi_{i,a}|.$$

Now since D is a compact set, note that

$$L_2 \stackrel{\triangle}{=} \max_{(i,a)\in S\times A(S)} \max_{\theta\in D} |\theta^T \phi_{i,a}| < \infty.$$

The claim now follows since $\nabla_w f_w(i,a)$ is a continuous function from Lemma 1 (in fact also uniformly continuous since $w \in C$, a compact set). □

Proof of Lemma 6 We first show the claim in (3.5). Recall from Lemma 5 that

$$\parallel w_{n+s} - w_n \parallel \to 0 \text{ as } n \to \infty,$$

almost surely, for all $s \in \{1,\ldots,P\}$. From Lemma 2 and the above, it follows that

$$\parallel \nabla_{w,k} R(\theta, w_{n+s}) - \nabla_{w,k} R(\theta, w_n) \parallel \to 0 \text{ as } n \to \infty,$$

$\forall s \in \{1,\ldots,P\}$, $k \in \{1,\ldots,N\}$. By letting $M = P$ in Assumption 5, it follows that $a(j)/a(m) \to 1$ as $m \to \infty$ for any $j \in \{m,\ldots,m+P-1\}$. Note also that P is an even integer. As a consequence of Lemma 4, one can split any set of the type $A_m \stackrel{\triangle}{=} \{m, m+1,\ldots,m+P-1\}$ into two disjoint subsets $A^+_{m,k,l}$ and $A^-_{m,k,l}$ each having the same number of elements, with $A^+_{m,k,l} \cup A^-_{m,k,l} = A_m$ and such that $\frac{\triangle^k_n}{\triangle^l_n}$ takes value $+1$ $\forall n \in A^+_{m,k,l}$ and -1 $\forall n \in A^-_{m,k,l}$, respectively. Thus,

$$\parallel \sum_{n=m}^{m+P-1} \frac{a(n)}{a(m)} \frac{\triangle^k_n}{\triangle^l_n} \nabla_{w,k} R(\theta, w_n) \parallel = \parallel \sum_{n\in A^+_{m,k,l}} \frac{a(n)}{a(m)} \nabla_{w,k} R(\theta, w_n) - \sum_{n\in A^-_{m,k,l}} \frac{a(n)}{a(m)} \nabla_{w,k} R(\theta, w_n) \parallel .$$

It now follows as a consequence of the above that

$$\parallel \sum_{n=m}^{m+P-1} \frac{a(n)}{a(m)} \frac{\triangle^k_n}{\triangle^l_n} \nabla_{w,k} R(\theta, w_n) \parallel \to 0,$$

almost surely as $m \to \infty$. Finally, the claim in (3.6) follows from Lemma 5, Lemma 2 and Assumption 5, in a similar manner as (3.5). □

References

Abdulla MS, Bhatnagar S (2007) Reinforcement learning based algorithms for average cost Markov decision processes. Discrete Event Dyn Syst Theory Appl 17(1):23–52

Abounadi J, Bertsekas D, Borkar VS (2001) Learning algorithms for Markov decision processes. SIAM J Control Optim 40:681–698

Aubin J, Cellina A (1984) Differential inclusions: set-valued maps and viability theory. Springer, New York

Azar MG, Gomez V, Kappen HJ (2011) Dynamic policy programming with function approximation. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics (AISTATS), Fort Lauderdale

Baird LC (1995) Residual algorithms: reinforcement learning with function approximation. In: Proceedings of ICML. Morgan Kaufmann, pp 30–37

Benaim M, Hofbauer J, Sorin S (2005) Stochastic approximations and differential inclusions. SIAM J Control Optim 44(1):328–348

Benaim M, Hofbauer J, Sorin S (2006) Stochastic approximations and differential inclusions, Part II: applications. Math Oper Res 31(4):673–695

Bertsekas DP (2005) Dynamic programming and optimal control, 3rd ed. Athena Scientific, Belmont

Bertsekas DP (2007) Dynamic programming and optimal control, vol II, 3rd ed. Athena Scientific, Belmont

Bertsekas DP, Tsitsiklis JN (1996) Neuro-dynamic programming. Athena Scientific, Belmont
Bhatnagar S, Babu KM (2008) New algorithms of the Q-learning type. Automatica 44(4):1111–1119
Bhatnagar S, Borkar VS (1997) Multiscale stochastic approximation for parametric optimization of hidden Markov models. Probab Eng Inf Sci 11:509–522
Bhatnagar S, Fu MC, Marcus SI, Wang I-J (2003) Two-timescale simultaneous perturbation stochastic approximation using deterministic perturbation sequences. ACM Transactions on Modelling and Computer Simulation 13(2):180–209
Bhatnagar S, Kumar S (2004) A simultaneous perturbation stochastic approximation based actor–critic algorithm for Markov decision processes. IEEE Trans Autom Control 49(4):592–598
Bhatnagar S (2005) Adaptive multivariate three-timescale stochastic approximation algorithms for simulation based optimization. ACM Transactions on Modeling and Computer Simulation 15(1):74–107
Bhatnagar S (2007) Adaptive Newton-based multivariate smoothed functional algorithms for simulation optimization. ACM Transactions on Modeling and Computer Simulation 18(1):2:1–2:35
Bhatnagar S, Prasad HL, Prashanth LA (2013) Stochastic recursive algorithms for optimization: simultaneous perturbation methods, lecture notes in control and information sciences. Springer, London
Bhatnagar S, Sutton RS, Ghavamzadeh M, Lee M (2009) Natural actor-critic algorithms. Automatica 45:2471–2482
Bhatnagar S, Lakshmanan K (2012) An online actor-critic algorithm with function approximation for constrained Markov decision processes. J Optim Theory Appl 153(3):688–708
Borkar VS (1995) Probability theory: an advanced course. Springer, New York
Borkar VS (1997) Stochastic approximation with two timescales. Syst Control Lett 29:291–294
Borkar VS (2008) Stochastic approximation: a dynamical systems viewpoint. Cambridge University Press and Hindustan Book Agency
Borkar VS, Meyn SP (2000) The O.D.E. method for convergence of stochastic approximation and reinforcement learning. SIAM J Control Optim 38(2):447–469
Brandiere O (1998) Some pathological traps for stochastic approximation. SIAM J Contr Optim 36:1293–1314
Ephremides A, Varaiya P, Walrand J (1980) A simple dynamic routing problem. IEEE Trans Autom Control 25(4):690–693
Gelfand SB, Mitter SK (1991) Recursive stochastic algorithms for global optimization in $\mathcal{R}^{d_*}$. SIAM J Control Optim 29(5):999–1018
Konda VR, Borkar VS (1999) Actor–critic like learning algorithms for Markov decision processes. SIAM J Control Optim 38(1):94–123
Konda VR, Tsitsiklis JN (2003) On actor–critic algorithms. SIAM J Control Optim 42(4):1143–1166
Kushner HJ, Clark DS (1978) Stochastic approximation methods for constrained and unconstrained systems. Springer, New York
Kushner HJ, Yin GG (1997) Stochastic approximation algorithms and applications. Springer, New York
Maei HR, Szepesvari C, Bhatnagar S, Precup D, Silver D, Sutton RS (2009) Convergent temporal-difference learning with arbitrary smooth function approximation. Proceedings of NIPS
Maei HR, Szepesvari Cs, Bhatnagar S, Sutton RS (2010) Toward off-policy learning control with function approximation. Proceedings of ICML, Haifa
Melo F, Ribeiro M (2007) Q-learning with linear function approximation. Learning Theory, Springer, pp 308–322
Pemantle R (1990) Nonconvergence to unstable points in urn models and stochastic approximations. Annals Prob 18:698–712
Prashanth LA, Chatterjee A, Bhatnagar S (2014) Two timescale convergent Q-learning for sleep scheduling in wireless sensor networks. Wirel Netw 20:2589–2604
Puterman ML (1994) Markov decision processes: discrete stochastic dynamic programming. Wiley, New York
Schweitzer PJ (1968) Perturbation theory and finite Markov chains. J Appl Probab 5:401–413
Sutton RS (1988) Learning to predict by the method of temporal differences. Mach Learn 3:9–44
Sutton RS, Barto A (1998) Reinforcement learning: an introduction. MIT Press, Cambridge
Sutton RS, Szepesvari Cs, Maei HR (2009) A convergent $O(n)$ temporal-difference algorithm for off-policy learning with linear function approximation. In: Proceedings of NIPS. MIT Press, pp 1609–1616
Sutton RS, Maei HR, Precup D, Bhatnagar S, Silver D, Szepesvari Cs, Wiewiora E (2009) Fast gradient-descent methods for temporal-difference learning with linear function approximation. In: Proceedings of ICML. ACM, pp 993–1000
Spall JC (1992) Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. IEEE Trans Autom Control 37(3):332–341

Spall JC (1997) A one-measurement form of simultaneous perturbation stochastic approximation. Automatica 33:109–112
Szepesvari C, Smart WD (2004) Interpolation-based Q-learning. In: Proceedings of ICML. Banff, Canada
Tsitsiklis JN (1994) Asynchronous stochastic approximation and Q-learning. Mach Learn 16:185–202
Tsitsiklis JN, Van Roy B (1997) An analysis of temporal-difference learning with function approximation. IEEE Trans Autom Control 42(5):674–690
Tsitsikis J, Van Roy B (1999) Average cost temporal-difference learning. Automatica 35:1799–1808
Walrand J (1988) An introduction to queueing networks. Prentice Hall, New Jersey
Watkins C, Dayan P (1992) Q-learning. Mach Learn 8:279–292
Weber RW (1978) On the optimal assignment of customers to parallel servers. J Appl Probab 15:406–413

Shalabh Bhatnagar received his Ph.D degree from the Indian Institute of Science, Bangalore in 1998. He was a Postdoctoral Fellow at the Institute for Systems Research, University of Maryland, as well as the Free University Amsterdam, before moving back to IISc where he works as a Professor. He has held visiting positions at University of Alberta, Canada, as well as at the Indian Institute of Technology, Delhi.

His interests are in stochastic control and optimization with special emphasis on simulation based methods. He is also interested in applications of these techniques in engineering applications in the domains of communication and wireless networks, as well as vehicular traffic control. He is a Senior Associate of the Abdus Salam International Centre for Theoretical Physics, Trieste, Italy, and is a Fellow of the Indian National Academy of Engineering as well as the Institution of Electronics and Telecommunication Engineers.

K. Lakshmanan received his Ph.D degree from the Indian Institute of Science, Bangalore in 2013. He has held Research Associate positions at IISc, IIT Bombay, University of Leoben, Austria, and is currently working as a Postdoctoral Fellow at the National University of Singapore, Singapore.

His interests are in reinforcement learning and simulation optimization.