J Optim Theory Appl (2008) 137: 435–451
DOI 10.1007/s10957-007-9331-9

Q-Learning Algorithms with Random Truncation Bounds and Applications to Effective Parallel Computing

G. Yin · C.Z. Xu · L.Y. Wang

Published online: 12 December 2007

Abstract Motivated by an important problem of load balancing in parallel computing, this paper examines a modified algorithm to enhance Q-learning methods, especially in asynchronous recursive procedures for self-adaptive load distribution at runtime. Unlike the existing projection method that utilizes a fixed region, our algorithm employs a sequence of growing truncation bounds to ensure the boundedness of the iterates. Convergence and rates of convergence of the proposed algorithm are established. This class of algorithms has broad applications in signal processing, learning, financial engineering, and other related fields.

Keywords Q-learning · Recursive algorithms · Convergence · Rate of convergence

1 Introduction

With primary motivation of issues arising from parallel computing in asynchronous computing environments, this paper examines a modified algorithm to enhance

Communicated by C.T. Leondes.

G. Yin's research was supported in part by the National Science Foundation under Grants DMS-0603287 and DMS-0624849 and in part by the National Security Agency under Grant MSPF-068-029.

C.Z. Xu's research was supported in part by the National Science Foundation under Grants CCF-0611750, DMS-0624849, CNS-0702488, and CRI-0708232.

L.Y. Wang's research was supported in part by the National Science Foundation under Grants ECS-0329597 and DMS-0624849 and by the Michigan Economic Development Council.

G. Yin (✉)
Department of Mathematics, Wayne State University, Detroit, MI, USA
e-mail: gyin@math.wayne.edu

C.Z. Xu · L.Y. Wang
Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI, USA

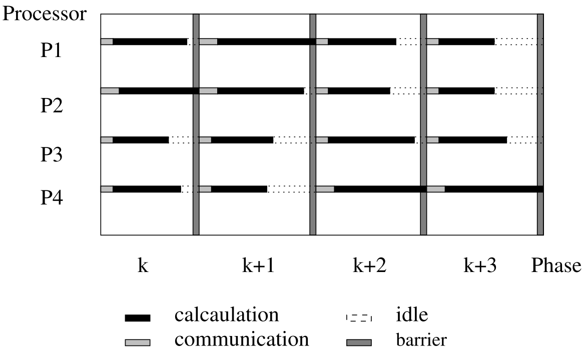

Fig. 1 Illustration of dynamic bulk synchronous computations

Q-learning methods. Unlike the existing projection method that utilizes a fixed projection region, our algorithm employs a sequence of growing truncation bounds to ensure the boundedness of the iterates. Under broad conditions, convergence and rates of convergence of the proposed algorithm are established.

One of our main motivations is assignment or mapping of the workload to processing nodes of a parallel computer. In such a computation environment, the primary performance goals of workload mapping are to balance the workloads among the processors, to minimize the interprocessor communication, and to reduce the run-time overhead of managing the assignment [1]. These can be accomplished statically or dynamically, depending upon the nature and predictability of the computation. Static assignment works for applications that have static and predictable computation and communication characteristics. For applications that have unpredictable characteristics, dynamic scheduling must be used at run-time to adapt to workload changes. A special form of dynamic scheduling is dynamic remapping, which redistributes the workload among processing nodes during the execution time. In general, achieving optimal dynamic remapping is computationally intractable, due to computational complexities. However, a bulk synchronous computation lends itself to a fairly structured approach to mapping. Its processes proceed in steps that are separated by global synchronization. During each step, they perform calculations independently and then communicate new results with their data-dependent peers at the end of the step. Due to the need of synchronization between steps, the duration or execution time of a step is determined by the most heavily loaded processor, which in turn determines the total execution time of the application. For example, consider a bulk synchronous computation that comprises of r processes labeled 1 through r. Assume the processes are running in parallel in a multiprogrammed distributed system with the same number of processing nodes. The computation proceeds in steps $k = 1, 2, \ldots,$ separated by barrier synchronization. The processes may exhibit time-varying stepwise computational requirements as the computation proceeds. Resources allocated to the computation can also vary with time as other jobs join and leave. Figure 1 shows the computation model with four processes running on different processors. The horizontal scale corresponds to computation steps; the vertical lines represent barrier synchronization operations which are due to begin.

In our recent paper [2], a workload assignment was treated by a stochastic control approach with the primary goals of balancing workloads among participating processors, minimizing interprocessor communication, and reducing run-time overheads of managing the assignment. The approach used an Markov decision process model. To use such models in optimal remapping strategies, it is crucial that the transition probability matrix is known. When such information is not available, one may use traditional estimation procedures to recover the unknown parameters. However, in a workload assignment setting, many parallel processors are involved. The dimensionality of the transition matrix is inevitably high. As a result, the usual methods of estimation that require rich data will become infeasible and impractical.

The Q-learning methods provide a promising alternative to resolve the problem. Compared with the traditional modeling and identification methodologies, the most appealing features of Q-learning include: (1) it directly uses system performance to evaluate the efficiency and fidelity of data acquisition and information processing; (2) it does not impose model structures. As a result, Q-learning is useful in developing control strategies for workload dynamic models. It should be emphasized that although our primary motivation stems from workload balancing, the learning methodologies developed in this paper will be of independent interest to other applications that can be formulated in the framework of adaptive Markovian optimization, including signal processing, learning, financial engineering, etc. Consequently, the problem is formulated generally and our results will be stated in generic forms.

The rest of the paper is arranged as follows. Section 2 develops the main algorithms of random regions of projection within the Q-learning methodologies. Convergence of the algorithms is first established in Sect. 3. Section 4 examines the issues of convergence rates. It is shown that the algorithms can be represented by a suitably scaled stochastic sequence that converges to a diffusion process. Finally, Sect. 5 concludes the paper with further remarks.

2 Q-Learning Algorithms

Roughly speaking, Q-learning is a technique of learning from delayed "costs," or a method for adaptive optimization of a controlled discrete-time Markov chain with a finite or countable state space. It belongs to the class of adaptive optimization algorithms of Markov decision processes. It is important to note that in the Q-learning method, the environment (the transition probability) is unknown, but the system can be simulated or observed under any choice of control actions. The Q-learning method was introduced by Watkins and Dayan in [3]. A comprehensive discussion of the Q-learning method and research progress in this area towards a general framework can be found in [4].

In what follows, we use the convention that i (in lieu of α_i) represents the state of the Markov chain. Let d be the decision variable taking values in a finite set $U(i)$ for each i, and $p_{ij}(d)$ the transition probability from state i to state j under decision d. By the (state, action) pair (i, d), we mean (α_i, d) throughout the rest of the paper. Denote by u_n the control action at time n, and let w_n be the state of the chain at time n. For example, in the simplest case of workload remapping problems, the control action takes only two possible values: either $d = 0$ or $d = 1$ depending on whether a remapping takes place. In this case, $U(i) = \{0, 1\}$. In the workload model, u_n represents the action of remapping or no remapping at step n, and w_n is the workload at n.

Let $\mathcal{F}_n$ be the σ-algebra generated by $\{w_j, u_j : j < n, w_n\}$. The control action can be a random function of the past history. A control is admissible if it is selected in accordance with a probability law $P(u_n = d|\mathcal{F}_n)$ (i.e., in accordance with conditional probability on the data up to time n). Use π to denote an admissible control policy. Let c_n be a real-valued cost realized at time n and E_i^π be the expectation under initial state i and policy π. Given initial state i and admissible control π, the cost criterion is

$$J(i,\pi) = E_i^\pi \sum_{n=0}^{\infty} \beta^n c_n, \tag{1}$$

where $0 < \beta < 1$ is a discount factor. Suppose that the cost depends only on the current state and control action in that, for all c,

$$P\{c_n = c|w_j, u_j, j < n, w_n = i, u_n = d\}P\{c_n = c|w_n = i, u_n = d\},$$

where the right-hand side does not depend on n. Conditioning on the current state being i and the current control action being d, assume that the cost has a finite mean $\overline{c}_{id}$ and a uniformly bounded variance. Henceforth, for notational convenience, $c_{n,id}$ is used to denote the random cost realized at time n when the state and control pair is (i,d).

Define $J^*(i) = \inf_\pi J(i,\pi)$. Then, the dynamic programming equation for the minimal cost is

$$J^*(i) = \min_{d\in U(i)}\left[\overline{c}_{id} + \beta \sum_j p_{ij}(d) J^*(j)\right].$$

Since the environment (the transition probability) $p_{ij}(d)$ is unknown, the value function given above cannot be obtained directly. The Q-learning algorithm recursively estimates the so-called Q-functions $\overline{Q}_{id}$, which represents the cost given that one starts at state i, takes control action $d \in U(i)$, and then uses the optimal policy henceforth. The distinction between the value function and the Q function is: In the value function $J^*(i)$, starting from the state i, the optimal policy is used in the entire trajectory at *all times*. In fact, $\min_{d\in U(i)} \overline{Q}_{id} = J^*(i)$. An application of the dynamic programming principle leads to

$$\overline{Q}_{id} = \overline{c}_{id} + \beta \sum_j p_{ij}(d) \min_{v\in U(j)} \overline{Q}_{jv}. \tag{2}$$

Remark 2.1 Note that in the parallel computation models and many other applications, we are dealing with finite state space Markov chains. However, the results to be presented are also valid for countable state space Markov chains.

To estimate the desired $\overline{Q}_{id}$ values, one observes the current state i, and selects an action d. Then the value of the state at the next time is observed. One updates the Q-value for only that particular (state, action) pair (i,d). The stepsize depends on the number of times that the particular (state, action) pair has been observed before.

Suppose the pair (i,d) occurred at time n, and the next state w_{n+1} is observed. Denote the estimate of $\overline{Q}_{id}$ at n by $Q_{n,id}$. Then, $Q_{n,id}$ is recursively calculated by

$$Q_{n+1,id} = Q_{n,id} + \widetilde{\varepsilon}_{n,id}\Big[c_{n,id} + \beta \min_{v\in U(w_{n+1})} Q_{n,w_{n+1}v} - Q_{n,id}\Big], \tag{3}$$

where $\widetilde{\varepsilon}_{n,id}$ is the stepsize. With $Q=(Q_{id}:\ i,d,\ d\in U(i))$ denoting a vector, let

$$T_{id}(Q) = E\Big[\overline{c}_{id} + \beta \min_{v\in U(w_{n+1})} Q_{w_{n+1}v}\Big|\mathcal{F}_n, u_n = d, w_n = i\Big].$$

To implement the algorithm, we use (3). However, for the subsequent analysis, it is more convenient to separate the state and the noise. The Markov property yields

$$T_{id}(Q) = \overline{c}_{id} + \beta \sum_j p_{ij}(d) \min_{v\in U(j)} Q_{jv}.$$

Define further the "noise"

$$\widetilde{\delta}_{n,id} = \Big[c_{n,id} + \beta \min_{v\in U(w_{n+1})} Q_{n,w_{n+1}v}\Big] - T_{id}(Q_n).$$

It then follows that the algorithm becomes

$$Q_{n+1,id} = Q_{n,id} + \widetilde{\varepsilon}_{n,id}\left[T_{id}(Q_n) - Q_{n,id} + \widetilde{\delta}_{n,id}\right]. \tag{4}$$

In the above, $T(\cdot)$ is a continuous function, we have separated the noise from the state for analysis. Under quite general conditions, constrained Q-learning algorithms were considered in Chap. 12 of [5] and the convergence was proved via stochastic approximation methods for distributed and asynchronous recursive procedures ([6] and [7]). In particular, we demonstrated that how one may deal with constraints such as a hypercube $[-B,B]^r$, where r is the dimension of the state variable. Then the analysis was carried out by using the associated mean dynamics.

Note that (4) may be written synchronously; the analysis for this case is much simpler than its asynchronous counterpart. Nevertheless, asynchronous implementation has certain advantages but is mathematically more challenging to handle. The essence is that each (i,d) pair is updated in an asynchronous way. As illustrated in [5, p. 43] this is done by selecting the stepsize in a specific way and only updating one component $Q_{n,id}$ at a time. Let $\{\varepsilon_n\}$ be a sequence of positive real numbers such that

$$\varepsilon_n \to 0, \sum_{n=0}^{\infty} \varepsilon_n = \infty, \quad \text{and} \quad \sum_{n=0}^{\infty} \varepsilon_n^2 < \infty. \tag{5}$$

If the current (state, action) pair has been observed $(k-1)$ times up to the current time in simulation, then the stepsize $\widetilde{\varepsilon}_{n,id} = \varepsilon_{k,id}$. More generally, if (i,d) has been observed $k(i,d,n)$ times up to time n, then $\widetilde{\varepsilon}_{n,id} = \varepsilon_{(k,i,d),id}$. If (i,d) is the (state, action) pair at n, then for $(j,d_1)\neq(i,d)$ there is no update or $Q_{n+1,jd_1} = Q_{n,id}$. This idea can be equivalently explained by use of a random computation time model.

In light of the ideas of [7], consider a random computation time model as follows. For each $i \in \mathcal{M}$ and $d \in U(i)$, let $\{y_n^{id}\}$ be a sequence of positive integer-valued (assumed for notational simplicity) random variables such that y_n^{id} is the duration that $Q_{n,id}$ takes to complete the nth iteration. Define a cumulative process τ_n^{id} by $\tau_0^{id} = 0$, $\tau_n^{id} = \sum_{j=0}^{n-1} y_j^{id}$, which is the random computation time for $Q_{n,id}$. This sequence is similar to the conventional renewal process, although it is assumed that it is a sequence of data-dependent and non-i.i.d. random variables. For simplicity, we still refer to it as a renewal process. As for the noise process, we shall consider $\{\delta_n\}$ a random sequence satisfying $Q_{n,id} = Q_{\tau_j^{id},id}$, $\delta_{n,id} = \widetilde{\delta}_{n,id}$, for $n \in [\tau_j^{id}, \tau_{j+1}^{id})$. For further explanation of the description of $\delta_{n,id}$, we refer to Chap. 12 in [5].

To proceed, we also need the following definitions:

$$N_{id}(n) = \sup\{k; \tau_k^{id} \le n\}, \qquad \Delta_n^{id} = n - \tau_{N_{id}(n)}^{id}, \ I_n^{id} = I_{\{\Delta_n^{id}=0\}}. \tag{6}$$

Note that $N_{id}(n)$ is a counting process that counts the number of events or renewals (the number of iterations corresponding to the pair (i, d) up to time n). Similarly, Δ_n^{id} can be thought of as the "age" or "current life" in the conventional renewal language. It represents the time elapsed since last iteration. If $\Delta_n^{id} = 0$, then n is a random computation time. To keep track of all the state values used at each time n, we define $\widetilde{Q}_n \in \mathbb{R}^{r_0^2}$, a collection of r_0 r_0-vectors, as $\widetilde{Q}_n = (\widetilde{Q}_n^{id})' \in \mathbb{R}^{r_0^2 \times 1}$, where $\widetilde{Q}_n^{id} = Q_{\tau^{id}_{N_{id}(n)}} \in \mathbb{R}^{r_0}$. Note that $\widetilde{Q}_n^{id}$ is just the Q-value in use for the pair (i, d) at real time n. The notation is somewhat involved, but we have to keep track of all of the Q-values used at each time that may differ for distinct (state, action) pairs. Define

$$I_n = \operatorname{diag}(I_n^{id}), \ \delta_n = (\delta_{n,id} : \text{ all } (i, d)) \in \mathbb{R}^{r_0},$$
$$T(\widetilde{Q}_n) = (T_{id}(\widetilde{Q}_n^{id}) : \text{ all } (i, d))' \in \mathbb{R}^{r_0}$$

with a slight abuse of notation. Then, the Q-learning algorithm may be written as

$$Q_{n+1,id} = Q_{n,id} + \varepsilon_n [T_{id}(\widetilde{Q}_n^{id}) - Q_{n,id} + \delta_{n,id}] I_{n+1}^{id}, \tag{7}$$

or in a more compact form as

$$Q_{n+1} = Q_n + \varepsilon_n I_{n+1} [T(\widetilde{Q}_n) - Q_n + \delta_n]. \tag{8}$$

To keep track of the computation times, for each (i, d) pair, set

$$x_{n+1,id} = x_{n,id} + \varepsilon_n y_n^{id}. \tag{9}$$

Define also the interpolated sequences

$$t_n = \sum_{j=0}^{n-1} \varepsilon_j, \quad m(t) = \max\{n : t_n \le t\},$$
$$Q^0(t) = Q_n, \quad t \in [t_n, t_{n+1}), \ Q^n(t) = Q^0(t + t_n),$$
$$x_{id}^0(t) = x_{n,id}, \quad t \in [t_n, t_{n+1}), \ x_{id}^n(t) = x_{id}^0(t + t_n).$$

Note that $Q^n(\cdot)$ (resp. $x^n(\cdot)$) is a shifted sequence of $Q^0(\cdot)$ (resp. $x^0(\cdot)$). Its purpose is to bring the asymptotics to the foreground.

In this work, in lieu of an algorithm with a fixed constrained region, we use a sequence of varying truncation bounds to relax the *a prior* information and to ensure the boundedness of the iterates, which is especially suited from a learning point of view. The use of varying truncation bounds was first proposed and analyzed in Chen and Zhu [8] and subsequently developed in various applications; see [9–11], and references therein for motivation and related work. This method has been proved to be useful and easily implementable.

Let $\{M_n\}$ be a sequence of increasing positive real numbers, such that $M_n \to \infty$ as $n \to \infty$. Define a sequence of integer-valued random variables σ_n recursively by

$$\sigma_0 = 0, \quad \sigma_{n+1} = \sigma_n + 1_{\{|Q_n+\varepsilon_n I_{n+1}[T(\widetilde{Q}_n)-Q_n+\delta_n]|>M_{\sigma_n}\}},$$

where 1_A is the indicator function of the set A. Define

$$\chi_n = 1_{\{|Q_n+\varepsilon_n[T(\widetilde{Q}_n)-Q_n+\delta_n]|\le M_{\sigma_n}\}}, \qquad \chi_n^c = 1 - \chi_n. \tag{10}$$

With the random generated truncation bounds, rewrite the Q-learning algorithm as

$$Q_{n+1} = \big[Q_n + \varepsilon_n I_{n+1}(T(\widetilde{Q}_n) - Q_n + \delta_n)\big]\chi_n + Q^f \chi_n^c, \tag{11}$$

where Q_{id}^f is a fixed but otherwise arbitrary point.

Remark 2.2 For the remapping problem that we are interested in, a stationary policy is said to be proper if the probability of being at the absorbing state tends to 1 as time goes to ∞; otherwise, the policy is said to be improper (see [4]). It is well-known that since no discount factor is used, the operator T is in general not a contraction. However, as explained in the aforementioned reference, for the remapping problem, there is a unique fixed point $T(Q) = Q$ and the fixed point is nothing but the optimal value. We denote this set by Z.

3 Convergence Analysis

This section studies convergence of the proposed algorithm. Through the rest of the paper, we use the notation $\widetilde{E}_n^{id}$ and E_n^{id} to denote the conditional expectation with respect to the σ-algebras generated by $\{\widetilde{\delta}_{j,id} : j < n\}$ and $\{\delta_{j,id}, Q_{j+1} : j < n\}$, respectively. Choose a sequence of positive real numbers $\{\Delta_n\}$ such that $\Delta_n \to 0$ as $n \to \infty$. Consider an increasing sequence $n = \widetilde{m}(n,1) < \widetilde{m}(n,2) < \cdots$ such that $\sum_{j=\widetilde{m}(n,l)}^{\widetilde{m}(n,l+1)-1} \varepsilon_j/\Delta_n \to 1$. For notational simplicity, in what follows, we denote $\widetilde{m}(n,l)$ simply by m_l. To proceed, we assume the following conditions.

(A1) The random computation times y_n^{id} are bounded and positive random variables. For each (i,d), there is a $\widetilde{\mu}^{id}(Q,\delta)$, continuous function of Q satisfying

$$E_{\tau_n^{id}} y_n^{id} = \widetilde{\mu}^{id}(Q_{\tau_n^{id}}, \widetilde{\delta}_{n,id});$$

there is a continuous function $\mu^{id}(\cdot)$ such that, for each Q and each l,

$$\frac{1}{\Delta_n}\sum_{j=m_l}^{m_{l+1}-1}\varepsilon_j \widetilde{E}^{id}_{m_l}\widetilde{\mu}^{id}(Q,\widetilde{\delta}_{j,id}) \to \mu^{id}(Q) \text{ in probability as } n\to\infty.$$

Denote $\mu^{-1}(x)=\operatorname{diag}(1/\mu^{id}(x):(i,d))$.

(A2) For each (i,d), $T_{id}(\cdot)$ is a continuous function; the sequence $\{\widetilde{\delta}_{n,id}\}$ is uniformly integrable; for each l, $\frac{1}{\Delta_n}\sum_{j=m_l}^{m_{l+1}-1}\varepsilon_j\widetilde{\delta}_{j,id}\to 0$ w.p.1 as $n\to\infty$.

(A3) There is at least one proper stationary policy. Each improper stationary policy yields an infinite cost for at least one initial state.

(A4) There are a twice continuously differentiable real-valued function $V(\cdot):\mathbb{R}^{r_0}\mapsto\mathbb{R}$ and a real number $M>0$ such that $|Q^0|<M$, that $V(Q)\neq V(y)$ for all $Q\in Z$ and $y\notin Z$, that $V'_Q(Q)(T(Q)-Q)<0$ for all $Q\notin Z$, that $V(Q^f)<\inf\{V(Q):|Q|=M\}=d_0$ and $[V(Q^f),d_0]\cap V(Z)\neq[V(Q^f),d_0]$, where $V_Q(\cdot)$ denotes the gradient of $V(\cdot)$ and $Z=\{Q:T(Q)=Q\}$. Moreover, the related differential equation

$$\frac{d}{dt}Q(t)=\mu^{-1}(Q)(T(Q(t))-Q(t)) \tag{12}$$

has a unique solution for each initial condition.

Remark 3.1 We comment on the conditions briefly. The averaging condition in (A1) is an ergodicity requirement for the random computation times. This condition mimics the "asymptotic rate of change" condition in [5]. But here we use convergence in the weak sense, and use the interface of the varying truncation and that of weak convergence to establish the w.p.1 convergence of the algorithm, which is a new twist. The new angle may shed more light in the study of general stochastic approximation algorithms.

A specification is that for each (i,d) pair, τ_n^{id} is a stationary sequence with mean $\mu^{id}>0$. Then the average condition reduces to the usual ergodicity condition $\tau_n^{id}/n\to\mu^{id}$ w.p.1. Note that our formulation is rather general. The random computation times are, in fact, data dependent. The condition with conditional expectation is weaker than that without. Note that (A2) is essentially a law of large numbers for the noise process.

In fact, for our analysis of the recursive algorithm, there are essentially two scales. The noise and the computation time evolve much faster than that of Q_n. Thus, for large enough m and in an interval as in the above average, the values of Q can be viewed as a constant value. Then we simply average out other random effects. This condition does assume that the computation time to be Q-value and noise dependent.

In addition, more general assumptions (such as state dependence) on the noise $\{\delta_n\}$ can be given; see the related work in stochastic approximation in [7]. However, for our consideration, the current setup appears to be sufficient. Condition (A4) requires the existence of a Liapunov function $V(\cdot)$. Its precise form needs not be known.

We will show that the truncation eventually terminates, and that the iterates become bounded. Once boundedness is established, the limiting ordinary differential equation for an appropriate interpolated process can be established. In contrast to the usual projection algorithms, a priori information on the truncation region need not be known. In addition, the growth conditions of functions involved can be much relaxed. These are particularly suited for the remapping problem that we are considering. Note that the significance of the above differential equation is that its stationary point is the parameter values of the system (the Q-values) we are searching for.

Theorem 3.1 *Under assumptions* (*A*1)–(*A*4), $Q^n(\cdot)$ *converges to* $Q(\cdot)$ *w.p.*1 *such that* $Q(\cdot)$ *satisfies the limit ordinary differential equation* (12).

To prove the theorem, we establish first a series of lemmas.

Lemma 3.1 *Assume that the conditions of Theorem* 3.1 *are satisfied and that* $\{\widetilde{Q}_n\}$ *is bounded w.p.*1 *whenever* $\{Q_n\}$ *is. Let* $m(n,\eta)=\max\{m; \sum_{j=n}^{m}\varepsilon_j\le\eta\}$.

(a) *Suppose that* $\{Q_{n_k}\}$ *is a convergent subsequence of* $\{Q_n\}$. *Then, there exists a* $\delta>0$ *such that, for any* $\eta>0$ *with* $0<\eta<\delta$, *for sufficiently large* k,
   (i) *for all* $m\in[n_k, m(n_k,\eta)]$,

$$\left|\sum_{j=n_k}^{m}\varepsilon_j I_{j+1}(T(\widetilde{Q}_j)-Q_j+\delta_j)\right|\le K_1, \quad \text{w.p.1}; \tag{13}$$

   (ii) *for all* $m\in[n_k, m(n_k,\eta)+1]$,

$$|Q_m-Q_{n_k}|\le K_2\eta, \quad \text{w.p.1}. \tag{14}$$

(b) $\lim_n\sigma_n=\sigma<\infty$, w.p.1.

Remark 3.2 Assertion (a) states that for sufficiently small $\eta>0$, (13) and (14) hold. Because $\{\sigma_n\}$ is increasing, either it has a finite limit or it grows without bound. Assertion (b) indicates that the second alternative is not possible.

Proof of Lemma 3.1 The proof of Assertion (a) follows along the same line of argument as that of Lemma 3.2 in p. 70 of [9]. We thus omit the verbatim details.

We proceed to prove (b). Since $\{\sigma_n\}$ is an increasing sequence of positive random variables, either there is a finite limit $\sigma<\infty$ w.p.1, or $\sigma=\infty$ w.p.1. To prove (2), suppose the contrary, $\sigma=\infty$. Then starting from Q^f, Q_n would be across the sphere $\{Q; |Q|=M\}$ infinitely often. Then there exists an interval $[\Delta_1,\Delta_2]\subset[V(Q^f),d]$ with $\Delta_1\ne V(Q^f)$, $\Delta_1\notin V(Z)$. If $|Q_n|\le M$, then $|\widetilde{Q}_n|\le M$, $|\varepsilon_n I_{n+1}(T(\widetilde{Q}_n)-Q_n+\delta_n)|\to 0$ w.p.1, and hence $V(Q_n)$ would be across the interval $[\Delta_1,\Delta_2]$ infinitely often from the left. It then yields that there exist $\{Q_{n_k}\}$ and $\{Q_{m_k}\}$, subsequences of $\{Q_n\}$, satisfying $n_k\le m_k$, $|Q_{n_k}|\le M$,

$$\begin{aligned} &V(Q_{n_k-1})<\Delta_1, \qquad \Delta_1\le V(Q_j)\le\Delta_2, \qquad n_k\le j\le m_k-1,\\ &V(Q_{m_k})>\Delta_2. \end{aligned} \tag{15}$$

The continuity of $V_Q(\cdot)$, the gradient of $V(\cdot)$, implies that $|V(Q_{n_k}) - V(Q_{n_k-1})| \to 0$, by the mean value theorem. Thus, (15) leads to $V(Q_{n_k}) \to \Delta_1$. Define

$$F = \{Q : \Delta_1 \le V(Q) \le \Delta_2\} \cap \{Q; |Q| \le M\}.$$

Then, F is closed.

Choose a convergent subsequence of $\{Q_{n_k}\}$, denote it by $\{Q_k\}$ for notational simplicity, and denote the limit by $\overline{Q}$. Then, $\lim_k V(Q_k) = V(\overline{Q}) = \Delta_1$, $\overline{Q} \in F$, and $\overline{Q} \notin Z$. It can be verified that

$$\begin{aligned} V(Q_{m(k,\eta)+1}) - V(Q_k) &= V_Q'(\overline{Q})(Q_{m(k,\eta)+1} - Q_k) + O(\eta^2) \\ &= V_Q'(\overline{Q}) \sum_{j=k}^{m(k,\eta)} \varepsilon_j I_{j+1}(T(\widetilde{Q}_j) - Q_j) \\ &\quad + V_Q'(\overline{Q}) \sum_{j=k}^{m(k,\eta)} \varepsilon_j I_{j+1} \delta_j + O(\eta^2). \end{aligned}$$

Note that the next to the last term in the above equation converges to 0 as $k \to \infty$. Thus, it suffices to examine the first term on the second line of the above equation.

Since for any $m \in [k, m(k,\eta)]$, $|Q_m - \overline{Q}| \le |Q_m - Q_k| + |Q_k - \overline{Q}| \le K\eta$, for some $K > 0$, the continuity of $T_{id}(\cdot)$ yields

$$\max_{\tau_j^{id} \in [N_{id}(k), N_{id}(m(k,\eta))]} |T_{id}(Q_{\tau_j^{id}}) - T_{id}(\overline{Q})| \to 0, \quad \text{as } \eta \to 0.$$

Hence,

$$\sum_{j=N_{id}(k)}^{N_{id}(m(k,\eta))} \varepsilon_{\tau_j^{id}} (T_{id}(Q_{\tau_j^{id}}) - Q_{\tau_j^{id}}) = \sum_{j=N_{id}(k)}^{N_{id}(m(k,\eta))} \varepsilon_{\tau_j^{id}} (T_{id}(\overline{Q}) - \overline{Q}_{id}) + o(\eta).$$

By virtue of (A4), $V_Q'(\overline{Q})(T(\overline{Q}) < 0$. Moreover,

$$\sum_{j=N_{id}(k)}^{N_{id}(m(k,\eta))} \varepsilon_{\tau_j^{id}} = \left[\frac{\sum_{j=N_{id}(k)}^{N_{id}(m(k,\eta))} \varepsilon_{\tau_j^{id}}}{\sum_{j=k+1}^{m(k,\eta)} \varepsilon_j}\right]\left[\sum_{j=k}^{m(k,\eta)} \varepsilon_j - \varepsilon_k\right] \ge \widetilde{k}_0 \eta,$$

for some $\widetilde{k}_0 > 0$. It follows that

$$V_Q'(\overline{Q}) T(\overline{Q}) \sum_{j=N_{id}(k)}^{N_{id}(m(k,\eta))} \varepsilon_{\tau_j^{id}} \le -\widetilde{k}_1 \eta,$$

for some $k_1 > 0$. It leads to that there is a $\delta > 0$ and, for all $0 < \eta \le \delta$,

$$\limsup_{k \to \infty} V(Q_{m(k,\eta)+1}) \le \limsup_{k \to \infty} V(Q_k) - K\eta \le \Delta_1 - K\eta, \tag{16}$$

for some $K > 0$. On the other hand, since $Q_k \to \overline{Q}$ as $k \to \infty$,

$$\max_{k \le m \le m(k,\eta)} |V(Q_m) - V(Q_k)| \to 0, \qquad \text{as } \eta \to 0$$

uniformly in k for sufficiently large k. For η sufficiently small, and $m \in [k, m(k,\eta)+1]$, $V(Q_m) \le \Delta_2$. Thus, we obtain $\Delta_1 \le V(Q_{m(k,\eta)+1}) \le \Delta_2$, for sufficiently large k, which is a contradiction to (16). Thus, the lemma is proved. □

Remark 3.3 Lemma 3.1 indicates that $\sigma_n \to \sigma < \infty$ w.p.1 and that there is an n_0 such that for all $n \ge n_0$, $|Q_n| \le M_\sigma$. That is, after finitely many steps, the truncations will not be executed and the iterates $\{Q_n\}$ will be uniformly bounded w.p.1. As a consequence, (11) becomes

$$Q_{n+1} = Q_n + \varepsilon_n I_{n+1}(T(\widetilde{Q}_n) - Q_n + \delta_n), \quad \text{for all } n \ge n_0. \tag{17}$$

Fix $\omega \in \Omega$. Lemma 3.1 and (17) imply that for all $n \ge n_0$, $\{Q^n(\cdot)\}$ is uniformly bounded. In addition, for any $t, s > 0$,

$$Q^n(t+s) - Q^n(t) = \sum_{j=m(t_n+t)}^{m(t_n+t+s)-1} \varepsilon_j I_{j+1}(T(\widetilde{Q}_j) - Q_j + \delta_j). \tag{18}$$

To proceed, we recall the notion of *equicontinuity in the extended sense*, an extension of the equicontinuity to a class of not necessarily continuous functions; see p. 102 of [5]. For a sequence of vector-valued measurable functions $\{g_n(\cdot)\}$ on $[0,\infty)$, if $\{g_n(0)\}$ is bounded and if for each $T > 0$ and $\varepsilon > 0$, there is a $\Delta > 0$ such that $\limsup_n \sup_{0 \le s < \Delta, t \in [0,T]} |g_n(t+s) - g_n(t)| \le \varepsilon$, then $\{g_n(\cdot)\}$ is said to be equicontinuous in the extended sense.

By (A2) and (11), $\limsup_n |\sum_{j=m(t_n+t)}^{m(t_n+t+s)-1} \varepsilon_j I_{j+1} \delta_j| = 0$ w.p.1. Furthermore,

$$\limsup_n \sup_{0 \le s < \Delta, t \in [0,T]} \left| \sum_{j=m(t_n+t)}^{m(t_n+t+s)-1} \varepsilon_j I_{j+1}[T(\widetilde{Q}_j) - Q_j] \right| \le K\Delta,$$

for some $K > 0$. Thus, $\{Q^n(\cdot)\}$ is equicontinuous in the extended sense w.p.1. Lemma 3.1 also implies that it is uniformly bounded for $n \ge n_0$. Likewise, it is readily seen that

$$x_{id}^n(t+s) - x_{id}^n(t) = \sum_{j=m(t_n+t)}^{m(t_n+t+s)-1} \varepsilon_j y_j^{id}$$

and that $\{x_{id}^n(\cdot)\}$ is equicontinuous in the extended sense w.p.1. It then follows from Theorem 4.2.2 in [5], we can extract convergent subsequences of $\{Q^n(\cdot)\}$ and $\{x_{id}^n(\cdot)\}$, respectively. Denote the corresponding limits by $Q(\cdot)$ and $x_{id}(\cdot)$, respectively. Without loss of generality, we may choose n_k such that both $(Q^{n_k}(\cdot), x_{id}^{n_k}(\cdot)) \to (Q(\cdot), x_{id}(\cdot))$ w.p.1. We proceed to characterize the limit processes.

First note that the w.p.1 convergence of $(Q^{n_k}(\cdot), x^{n_k}_{id}(\cdot))$ to $(Q(\cdot), x_{id}(\cdot))$ implies the pair of processes $(Q^{n_k}(\cdot), x^{n_k}_{id}(\cdot))$ converges weakly to the same limit. To obtain the desired limit result, we need only figure out the weak limit of the pair given above owing to the uniqueness of the limit process. We first prove the following lemma.

Lemma 3.2 *Under the conditions of Theorem 3.1, the limit $x_{id}(\cdot)$ satisfies*

$$\frac{d}{dt}x_{id}(t) = \mu^{id}(Q(x_{id}(t))).$$

Proof of Lemma 3.2 As indicated above, we focus on obtaining the weak limit process. To this end, we need only show that $x_{id}(\cdot)$ is a solution of a degenerate martingale problem; see [5]. To this end, for any integer k_1, and any bounded and continuous functions $h_\iota(\cdot,\cdot)$ with $\iota \le k_1$,

$$\prod_{\iota=1}^{k_1} Eh_\iota(Q^{n_k}(t_i), x_{id}(t_i))\left[x_{id}(t+s) - x_{id}(t) - \sum_{j=m(t_{n_k}+t)}^{m(t_{n_k}+t+s)-1} y^{id}_j\right] \to 0.$$

Subdivide $[m(t_{n_k}+t), m(t_{n_k}+t+s)-1]$ to intervals $m_0 = m(t_{n_k}+t) \le m_1 \le m_2 \le m(t_{n_k}+t+s)-1$ such that $\sum_{j=m_l}^{m_{l+1}-1}\varepsilon_j/\Delta_{n_k} \to 1$. Note that for any $j \in [m_l, m_{l+1})$, $Q_j - Q_{m_l} \to 0$ and $x_{id,j} - x_{id,m_l} \to 0$ as $k \to \infty$. Thus, we obtain

$$\begin{aligned}
&\prod_{\iota=1}^{k_1} Eh_\iota(Q^{n_k}(t_i), x^{n_k}_{id}(t_i))\left[x^{n_k}_{id}(t+s) - x^{n_k}_{id}(t) - \sum_{j=m(t_{n_k}+t)}^{m(t_{n_k}+t+s)-1} y^{id}_j\right] \\
&\quad= \prod_{\iota=1}^{k_1} Eh_\iota(Q^{n_k}(t_i), x^{n_k}_{id}(t_i))\left[x^{n_k}_{id}(t+s) - x^{n_k}_{id}(t) - \sum_{j=m(t_{n_k}+t)}^{m(t_{n_k}+t+s)-1} E_{\tau^{id}_j} y^{id}_j\right] \\
&\quad= \prod_{\iota=1}^{k_1} Eh_\iota(Q^{n_k}(t_i), x^{n_k}_{id}(t_i))\Bigg[x^{n_k}_{id}(t+s) - x^{n_k}_{id}(t) \\
&\qquad - \sum_{j=m(t_{n_k}+t)}^{m(t_{n_k}+t+s)-1} \widetilde{E}^{id}_{m(t_{n_k}+t)}\widetilde{\mu}^{id}(Q_{\tau^{id}_j}, \widetilde{\delta}^{id}_j)\Bigg] \\
&\quad= \prod_{\iota=1}^{k_1} Eh_\iota(Q^{n_k}(t_i), x^{n_k}_{id}(t_i))\Bigg[x^{n_k}_{id}(t+s) - x^{n_k}_{id}(t) \\
&\qquad - \sum_{l}\Delta_{n_k}\frac{1}{\Delta_{n_k}}\sum_{j=m_l}^{m_{l+1}-1} \widetilde{E}^{id}_{m_l}\widetilde{\mu}^{id}(Q^{n_k}(x^{n_k}_{id}(t_{m_l}), \widetilde{\delta}^{id}_j)\Bigg] + o(1),
\end{aligned}$$

where $o(1) \to 0$ as $k \to \infty$ in probability. From the above estimates, it can be shown that, for the limit process, we have

$$\prod_{\iota=1}^{k_1} Eh_\iota(Q(t_i), x_{id}(t_i))\left[x_{id}(t+s) - x_{id}(t) - \int_t^{t+s} \mu^{id}(Q(x_{id}(r)))dr\right] = 0.$$

Hence, the limit ODE is obtained. A few details similar to those of pp. 224–226 in [7] are omitted for brevity. □

Now, we come back to the process $Q^{n_k}(\cdot)$. For each (i,d),

$$Q_{id}^{n_k}(t+s) - Q_{id}^{n_k}(t) = \sum_{j=N_{id}(m(t_{n_k}+t))}^{N_{id}(m(t_{n_k}+t+s))-1} \varepsilon_{\tau_j^{id}}(T_{id}(Q(x_{id}(t_j)) - Q(x_{id}(t_j) + \widetilde{\delta}_{j,id}).$$

Similar to pp. 226–227 in [7], we can show that the limit is precisely

$$Q_{id}(t) = Q_{id}(0) + \int_0^{\frac{1}{x_{id}(t)}} T(Q(x^{id}(r)) - Q_{id}(x^{id}(r)))dr.$$

Differentiating the above with respect to t leads to

$$\frac{d}{dt} Q_{id}(t) = \frac{1}{\mu^{id}(Q(t))}(T_{id}(Q(t) - Q_{id}(t)),$$

as desired. The proof of the theorem is concluded. □

Theorem 3.2 *The sequence* $Q_n \to Q^* = \overline{c}_{id} + \beta \sum_j p_{ij}(d) J^*(j)$ w.p.1.

Proof The proof is similar to that of [9]. Since $V(\cdot)$ is a Liapunov function for (12), the well-known stability theory yields that $\lim_{t\to\infty} d(Q(t), Z) = 0$, where $d(\cdot,\cdot)$ is the usual distance function,

$$d(x, Z) = \inf_{y \in Z} |x - y|.$$

The uniform convergence of $Q^n(\cdot) \to Q(\cdot)$ on bounded intervals enables us to extract a subsequence $\{Q_{n_k}\}$ of $\{Q_n\}$ with $\lim_k d(Q_{n_k}, Z) = 0$. If $\lim_n d(Q_n, Z) \neq 0$, we would have another subsequence $Q_{m_k} \to \overline{Q} \notin Z$ and $a_0 = \inf\{|V(Q) - V(\overline{Q})|, Q \in Z\} > 0$. Since Z is closed and $\{Q_n\}$ is bounded, for any $Q \in Z$, either $V(\overline{Q}) + a_0 \le V(Q)$ or $V(Q) \le V(\overline{Q}) - a_0$. Suppose $V(\overline{Q}) + a_0 \le V(Q)$ for any $Q \in Z$. The continuity of $V(\cdot)$ and the boundedness of $\{Q_{n_k}\}$ imply $V(\overline{Q}) + \frac{2}{3}a_0 \le V(Q_{n_k})$. Thus $\{V(Q_n)\}$ would be across the interval $[V(\overline{Q}) + \frac{1}{3}a_0, V(\overline{Q}) + \frac{2}{3}a_0]$ infinitely often. Then for all $Q \in E$, $V(Q) \notin [V(\overline{Q}) + \frac{1}{3}a_0, V(\overline{Q}) + \frac{2}{3}a_0]$, which leads to a contradiction. The case $V(Q) \le V(\overline{Q}) - a_0$ can be handled analogously. The uniqueness of the fixed point of T in Remark 2.2 implies the desired result. □

Remark 3.4 Note that in the remapping application, the Markov state 0 (or α_0) is an absorbing state. We assume $p_{00}(u) = 1$ and $c_{0d} = 0$ for all $u \in U(0)$. That is, the balanced state 0 is a cost free state. As noted in [4], (A3) implies that the set $\{V(i) : J(0) = J(i = 0) = 0\}$ contains a fixed point J^* ($J^* = T(J^*)$). Corresponding to the above theorem, we assume that in addition to the assumptions of Theorem 3.2, $Q_{0,(0,d)} = 0$, which implies that for the absorbing state $i = 0$, $Q_{n+1,(0,d)} = Q_{n,(0,d)} = 0$ for all n.

4 Rate of Convergence

This section is devoted to the rate of convergence study of the asynchronous Q-learning algorithm. The pertinent notion here is in the sense of weak convergence; see for instance, [5, 12], and references therein.

Let us first recall the definition of weak convergence. Consider $\mathbb{R}^r$-valued random variables X_n and X. The sequence X_n is said to converge weakly to X if and only if for any bounded and continuous function $g(\cdot)$, $Eg(X_n) \to Eg(X)$; $\{X_n\}$ is said to be tight if and only if for each $\eta > 0$, there is a compact set K_η such that $P(X_n \in K_\eta) \geq 1 - \eta$ for all n. The definitions of weak convergence and tightness extend to random variables in a metric space. Weak convergence is a substantial generalization of convergence in distribution. On a complete separable metric space, the notion of tightness is equivalent to sequential compactness, which is known as the Prohorov's Theorem. Due to this theorem, we are able to extract convergent subsequences once tightness is verified. Let $D^r[0, \infty)$ denote the space of $\mathbb{R}^r$-valued functions that are right continuous and have left-hand limits, endowed with the Skorohod topology. For various notations and terms in weak convergence theory such as Skorohod topology, Skorohod representation etc. and many others, we refer to [5] and the references therein.

Under the conditions of Theorem 3.2, $Q_n \to Q^*$ w.p.1. The rate of convergence study refers to the asymptotic properties of normalized estimation errors about the point Q^*. In fact, as in [9] (see also [11]), by use of the truncation, we have that there is an n_0 such that for all $n \geq n_0$, no truncations will be executed and the iterates become bounded. Thus, for the rate of convergence study, we need examine only an algorithm of the form

$$Q_{n+1} = Q_n + \varepsilon_n I_{n+1}(T(\widetilde{Q}_n) - Q_n + \delta_n)$$

without the truncations and assume that $\{Q_n\}$ is bounded w.p.1.

To proceed, we note that $(\varepsilon_n/\varepsilon_{n+1}) = 1 + \eta_n$, where

$$\eta_n = \begin{cases} 1/(2n) + o(\varepsilon_n), & \text{if } \varepsilon_n = 1/n, \\ o(\varepsilon_n), & \text{if } (1/n)/\varepsilon_n \to 0. \end{cases}$$

Define $U_n = (Q_n - Q^*)/\sqrt{\varepsilon_n}$. In view of (11), we have

$$U_{n+1} = U_n + I_{n+1}\left[\varepsilon_n \frac{T(\widetilde{Q}_n)}{\sqrt{\varepsilon_n}} - U_n + \sqrt{\varepsilon_n}\delta_n\right] + o(1),$$

where $o(1) \to 0$ in probability as $n \to \infty$. This term is due to the stepsize ε_n and the expansion of $(\varepsilon_n/\varepsilon_{n+1})$. Define a continuous-time interpolation by

$$U^n(t) = U_{n+i} \quad \text{for } t \in [t_{n+i} - t_n, t_{n+i+1} - t_n).$$

The analysis involve much technical details, but the main idea can be explained as follows. We first linearize $T(Q)$ about its fixed point Q^*, next prove that the iterates $\{U_n\}$ is bounded in the mean squares sense (i.e., $E|U_n|^2 = O(\varepsilon_n)$), and then construct interpolations of $U_n(\cdot)$ similar as what has been done in the last section. Then using averaging ideas, we show that the interpolated sequence converges in the weak sense to a system of stochastic differential equations.

(A5) The following conditions hold:

- $T(Q) = A(Q - Q^*) + O(|Q - Q^*|^2)$, where A is a stable matrix (in the sense that all of its eigenvalues are on the left-hand of the complex plane). If $(1/n)/\varepsilon_n \to 0$, $H = A - I$ is a stable matrix and, if $\varepsilon_n = O(1/n)$, $H + (I/2)$ is a stable matrix.
- The sequences $\{\delta_{n,id}\}$, for all (i, d), are mutually independent stationary ϕ-mixing processes with rates $\psi_{n,id}$ such that $\sum_j \psi_{j,id}^{1/2} < \infty$ and $E\delta_{n,id} = 0$. For any $q_n \to \infty$,

$$\lim_n \frac{1}{\Delta_n} \sum_{j=N_{id}(q_n)}^{N_{id}(q_n)+m_l} \varepsilon_j \delta_{j,id} \to 0, \quad \text{in probability.}$$

  The averaging conditions in (A1) holds with m_l replaced by $N_{id}(q_n)$.
- Define

$$w_{n,id} = \sum_{j=q_n}^{q_n+n-1} \sqrt{\varepsilon}_j \delta_{j,id} \quad w_{id}^n(t) = w_{n+i,id}, \quad t \in [t_{n+i} - t_n, t_{n+i+1} - t_n).$$

  $w_{id}^n(\cdot)$ converges weakly to the mutually independent Brownian motions $w_{id}(\cdot)$ with covariance $\sigma_{id}^2 t$ as $n \to \infty$.

Remark 4.1 The proof of $E|U_n| = O(\varepsilon_n)$ can be accomplished by using a perturbed Liapunov function methods (see [5]). The weak convergence of the scaled sequence $U^n(\cdot)$ is similar to that of the convergence proof in the last section. The independence of the $\widetilde{\delta}_n^{id}$ implies that the limit of the interpolated sequences are independent.

Theorem 4.1 *In addition to the conditions of Theorem 3.2, let* (A5) *holds. Then, $U^n(\cdot)$ converges weakly to $U(\cdot)$ such that there exist standard mutually independent Brownian motions $\widetilde{w}_{id}(\cdot)$ and $U_{id}^n(\cdot)$ satisfies*

$$dU_{id} = \begin{cases} \frac{(H+(I/2))U}{\mu^{id}(Q^*)} dt + \frac{\sigma_{id}}{\sqrt{\mu^{id}(Q^*)}} d\widetilde{w}_{id}, & \textit{if } \varepsilon_n = 1/n, \\ \frac{HU}{\mu^{id}(Q^*)} dt + \frac{\sigma_{id}}{\sqrt{\mu^{id}(Q^*)}} d\widetilde{w}_{id}, & \textit{if } (1/n)/\varepsilon_n \to 0. \end{cases}$$

5 Further Remarks

This paper has been focused on asynchronous Q-learning algorithms. The main idea here is that each of the components or the component corresponding to the (i,d) pair can be implemented completely asynchronously. Thus each of them may have its own local clock. The updated estimates or Q-values need not be communicated to all other components at each iteration. Thus it has distinct advantage of occasional information transmission resulting less overhead. As can be seen from the development, one of the difficulties is that there is no iteration number that is common to all processors.

The algorithms presented can be easily adapted for synchronous algorithms if it is so desired. Here, the "time" is the iteration number and is common for all the processors. In fact, we can define χ_n and χ_n^c as in (10) and replace (11) by

$$Q_{n+1} = [Q_n + \varepsilon_n(T(Q_n) - Q_n + \delta_n)]\,\chi_n + Q^f\chi_n^c, \tag{19}$$

where $\delta_n \equiv \widetilde{\delta}_n$ and no random computation times are involved. Note also that in (19), $\widetilde{Q}_n$ is not needed. The proof and analysis, in fact, will be much simpler, since there is no need to keep track of all the processors' contents at any given instance. In addition, the update time is just deterministic time $n = 1, 2, \ldots$. The limit ODE takes the form

$$\frac{d}{dt}Q(t) = T(Q(t)) - Q(t). \tag{20}$$

Theorem 3.1, Theorem 3.2, and Theorem 4.1 can be readily adapted to the synchronous case. In the development, all the conditions related to the random computation time, renewal time etc. can all be removed.

Another modification is in lien of sequences of decreasing step sizes, use a constant step size. The algorithm (with random truncations) to be analyzed can be written as

$$Q_{n+1} = \left[Q_n + \varepsilon I_{n+1}(T(\widetilde{Q}_n) - Q_n + \delta_n)\right]\chi_n + Q^f\chi_n^c,$$

where $\varepsilon > 0$ is a small parameter, and other quantities are as defined in the previous sections. The analysis can then be carried out using a weak convergence approach in Chaps. 8 and 10 of [5]. An advantage of using a constant step size is its simplicity for implementation.

References

1. Xu, C., Lau, F.: Load Balancing in Parallel Computers: Theory and Practice. Kluwer Academic, Boston (1997)
2. Yin, G., Xu, C., Wang, L.Y.: Optimal remapping in dynamic bulk synchronous computations via a stochastic control approach. IEEE Trans. Parallel Distrib. Syst. **14**, 51–62 (2003)
3. Watkins, C.I.C.H., Dayan, P.: Q-learning. Mach. Learn. **8**, 279–292 (1992)
4. Tsitsiklis, J.N.: Asynchronous stochastic approximation and Q-learning. Mach. Learn. **16**, 185–202 (1994)
5. Kushner, H.J., Yin, G.: Stochastic Approximation and Recursive Algorithms and Applications, 2nd edn. Springer, New York (2003)
6. Kushner, H.J., Yin, G.: Asymptotic properties of distributed and communicating stochastic approximation algorithms. SIAM J. Control Optim. **25**, 1266–1290 (1987)

7. Kushner, H.J., Yin, G.: Stochastic approximation algorithms for parallel and distributed processing. Stochastics **22**, 219–250 (1987)
8. Chen, H.F., Zhu, Y.M.: Stochastic approximation procedure with randomly varying truncations. Sci. Sin. (Ser. A) **29**, 914–926 (1986)
9. Yin, G., Zhu, Y.M.: On w.p.1 convergence of a parallel stochastic approximation algorithm. Probab. Eng. Inf. Sci. **3**, 55–75 (1989)
10. Chen, H.-F.: Stochastic Approximation and Its Applications. Kluwer Academic, Dordrecht (2002)
11. Chen, H.-F., Yin, G.: Asymptotic properties of sign algorithms for adaptive filtering. IEEE Trans. Autom. Control **48**, 1545–1556 (2003)
12. Yin, G.: Stochastic approximation: Theory and applications. In: Kannan, D., Lakshmikantham, V. (eds.) Handbook of Stochastic Analysis and Applications, pp. 577–624. Marcel Dekker, New York (2002)