Ann Oper Res (2013) 208:95–132
DOI 10.1007/s10479-012-1128-z

Q-learning and policy iteration algorithms for stochastic shortest path problems

Huizhen Yu · Dimitri P. Bertsekas

Published online: 18 April 2012

Abstract We consider the stochastic shortest path problem, a classical finite-state Markovian decision problem with a termination state, and we propose new convergent Q-learning algorithms that combine elements of policy iteration and classical Q-learning/value iteration. These algorithms are related to the ones introduced by the authors for discounted problems in Bertsekas and Yu (Math. Oper. Res. 37(1):66-94, 2012). The main difference from the standard policy iteration approach is in the policy evaluation phase: instead of solving a linear system of equations, our algorithm solves an optimal stopping problem inexactly with a finite number of value iterations. The main advantage over the standard Q-learning approach is lower overhead: most iterations do not require a minimization over all controls, in the spirit of modified policy iteration. We prove the convergence of asynchronous deterministic and stochastic lookup table implementations of our method for undiscounted, total cost stochastic shortest path problems. These implementations overcome some of the traditional convergence difficulties of asynchronous modified policy iteration, and provide policy iteration-like alternative Q-learning schemes with as reliable convergence as classical Q-learning. We also discuss methods that use basis function approximations of Q-factors and we give an associated error bound.

Keywords Markov decision processes · Q-learning · Approximate dynamic programming · Value iteration · Policy iteration · Stochastic shortest paths · Stochastic approximation

Work supported by the Air Force Grant FA9550-10-1-0412 and by NSF Grant ECCS-0801549.

H. Yu (✉) · D.P. Bertsekas
Lab. for Information and Decision Systems, MIT, Cambridge, MA 02139, USA
e-mail: janey_yu@mit.edu

D.P. Bertsekas
Dept. of Electr. Engineering and Comp. Science, MIT, Cambridge, MA 02139, USA
e-mail: dimitrib@mit.edu

1 Introduction

Stochastic shortest path (SSP) problems are a class of infinite horizon Markov decision processes (MDP) with the total cost criterion. They involve Markov chains with control-dependent transition probabilities and costs, and the objective is to reach a special destination state at minimum expected cost. In this paper we consider SSP with a finite state and control space, and the mathematical model is as follows. It involves a state space $S_o = \{0, 1, \dots, n\}$, with state 0 viewed as the destination state. Let $S = \{1, \dots, n\}$, and let $U(i)$ be the finite set of feasible controls at state $i \in S$. From state $i \in S$ under control $u \in U(i)$, a transition to state $j \in S_o$ occurs with probability $p_{ij}(u)$ and incurs a one-stage cost $\hat{g}(i, u, j)$. At state 0 the control set is $U(0) = \{0\}$, and we have $p_{00}(0) = 1$, $\hat{g}(0, 0, 0) = 0$, i.e., 0 is an absorbing and cost-free state. The goal is to reach state 0 with minimal total expected cost.

Policies of SSP are defined as in a standard MDP; however, a model assumption that ensures the suitability of the total cost criterion, will later be placed on certain policies of the SSP. More specifically, we denote a general history-dependent, randomized policy by π. A randomized Markov policy is a policy of the form $\pi = \{\nu_0, \nu_1, \dots\}$, where each function ν_t, $t = 0, 1, \dots$, maps each state $i \in S$ to a probability distribution $\nu_t(\cdot \mid i)$ over the set of feasible controls $U(i)$. If in addition, all ν_t are equal to some ν, π is said to be a stationary policy and is also denoted by ν. If all $\nu_t(\cdot \mid i)$ assign probability 1 to a single control $\mu_t(i)$, π is said to be deterministic, and if all μ_t are equal to some μ, π is said to be deterministic stationary and is also denoted by μ. The set of all deterministic stationary policies is denoted Π_{SD}. The notion of a proper policy will be used to classify policies of SSP. In particular, let us call a policy *proper*, if under that policy the destination state 0 is reached with probability 1 from every initial state, and *improper* otherwise.

We define the total cost of a policy π for initial state $i \in S$ to be

$$J_\pi(i) = \liminf_{k \to \infty} J_{\pi,k}(i)$$

with $J_{\pi,k}(i)$ being the expected k-stage costs of π for state i:

$$J_{\pi,k}(i) = \mathrm{E}\left[\sum_{t=0}^{k-1} \hat{g}(i_t, u_t, i_{t+1}) \,\middle|\, i_0 = i \right],$$

where i_t and u_t denote the state and control, respectively, at time t, and the expectation is with respect to the probability law of $\{i_0, u_0, \dots, i_k\}$ induced by π. The optimal cost at state $i \in S$, denoted $J^*(i)$, is the infimum of $J_\pi(i)$ over π, and the optimal cost vector J^* is the vector with components $J^*(i)$, $i = 1, \dots, n$.

SSP problems have a long history and have been studied by several authors, starting with Eaton and Zadeh (1962), who first introduced the problem and the notion of a proper policy. Veinott (1969) derived some of the underlying contraction properties, attributing them to A.J. Hoffman. Derman (1970) and Whittle (1983), among others, have streamlined the analysis (they referred to the problem as the "first passage problem," "transient programming"). Bertsekas and Tsitsiklis (1991) analyzed finite-state SSP problems with a compact control space, and introduced the following model assumption, which allows SSP to have both positive and negative one-stage costs. We will adopt their SSP model assumption in this paper.

Assumption 1.1

(i) There is at least one proper policy in Π_{SD}.
(ii) Any improper policy in Π_{SD} incurs infinite cost for at least one initial state.

Under this assumption, the optimal cost vector J^* takes finite values and solves uniquely Bellman's equation (Bertsekas and Tsitsiklis 1991). Furthermore, the fundamental value iteration (VI) and policy iteration (PI) algorithms are valid: VI converges to the optimal cost vector, when started from an arbitrary initial condition, and PI terminates with an optimal policy, when started from a proper policy. Intermediate between VI and PI is another important algorithm for MDP, the modified PI method. It is similar to PI, but it performs policy evaluation approximately, with a finite number of value iteration for the associated policy. For discounted MDP and other DP problems for which Bellman's equation involves a sup-norm contraction, convergence of modified PI (with some form of synchronous or regular order of state updates, including Gauss-Seidel variants) has been shown by Rothblum (1979) (see also the more recent work by Canbolat and Rothblum 2012). For SSP problems under Assumption 1.1, no such result is presently available, to our knowledge, and convergence of modified PI is usually shown under certain restrictive assumptions, for example, by requiring that the initial policy is proper and that the initial cost estimates are such that the convergence to the optimal costs is monotonic from above (see e.g., Williams and Baird 1993; Puterman 1994; Bertsekas and Tsitsiklis 1996; Bertsekas 2007 for related accounts).

Simulation-based algorithms, commonly used for solving large MDP problems, often implement VI, PI, or modified PI on the space of the so-called Q-factors, which are more convenient to work with than costs in the context of sampling. These algorithms aim to find the optimal Q-factor $Q^*(i,u)$ of each state-control pair (i,u), i.e., the optimal cost starting from state i using u at the first step and using an optimal policy thereafter. From the optimal Q-factors Q^*, the optimal costs J^* and an optimal policy can be identified. Mathematically, Q^* and the deterministic versions of VI, PI, and modified PI for Q-factors are equivalent to their standard counterparts for costs; one can therefore deduce their properties in SSP problems, such as the optimality equation and convergence guarantees, from the results on SSP and from the theory of total cost MDP (e.g., Bertsekas and Tsitsiklis 1991; Feinberg 1992; Puterman 1994; Bertsekas 2007). These properties and deterministic algorithms form the basis for stochastic algorithms that involve simulation, which due to stochastic noise, asynchronous computation, and above all, the combination of both, have additional dimensions of complexity.

The classical Q-learning algorithm of Watkins (1989) is an asynchronous stochastic iterative version of VI for Q-factors (see also textbook accounts in e.g., Bertsekas and Tsitsiklis 1996; Sutton and Barto 1998). It is analyzed by Tsitsiklis (1994) as a special case of general asynchronous stochastic approximation algorithms involving contraction or monotone nonexpansive mappings, and it has strong convergence properties. Its convergence for discounted MDP was established in Tsitsiklis (1994), and for SSP under Assumption 1.1, it was established partially in Tsitsiklis (1994) and fully in the authors' paper (Yu and Bertsekas 2011). In contrast to the convergence guarantee of the VI-based classical Q-learning, the convergence of asynchronous stochastic modified PI schemes for Q-factors is subject to serious doubt because monotonic convergence is generally impossible to maintain in the stochastic setting. Even for discounted MDP, asynchronous implementations of modified PI may fail to converge, as shown through counterexamples by Williams and Baird (1993). We refer to Bertsekas and Tsitsiklis (1996, Sect. 5.4) for an extensive discussion of this convergence issue. As explained there, with a malicious ordering of the state-control pairs at which Q-factors and policies are updated, nonconvergence/cycling behavior is possible.

Despite the difficulties just mentioned, asynchronous stochastic Q-learning algorithms, relating to PI and modified PI, were proposed and proved to be convergent for discounted MDP by the authors (Bertsekas and Yu 2012). These algorithms differ notably from the classical Q-learning algorithm in that they iterate within the larger space of cost and Q-factor pairs (J, Q), and perform policy evaluation via an optimal stopping problem, where $J(i)$ plays the role of a stopping cost at state i. The algorithms do not involve a minimization over all controls at every iteration, and therefore require lower overhead per iteration than classical Q-learning, which is the generic advantage that modified PI has over VI. Yet as shown in Bertsekas and Yu (2012), for discounted problems, they retain the strong convergence guarantee that the classical Q-learning algorithm offers (Tsitsiklis 1994).

In this paper we adapt the algorithms of Bertsekas and Yu (2012) to solve SSP problems and we analyze their properties. The case of SSP problems deserves separate investigation because of a major difference from the discounted case: under the SSP model Assumption 1.1, the mappings underlying our algorithms, as well as the mapping underlying the classical Q-learning algorithm, are nonexpansive instead of contracting. A different line of analysis is thus needed, and it is not clear a priori to what extent convergence properties that hold in discounted problems also hold for SSP problems, partly because the conclusions one can obtain for monotone nonexpansive mappings are usually weaker than those for contraction mappings. Indeed, the convergence of the classical Q-learning algorithm under Assumption 1.1 has only recently been fully established in Yu and Bertsekas (2011) by the authors; the analysis there uses properties special to SSP to remove the boundedness condition on the Q-learning iterates that is required in the convergence theorem of Tsitsiklis (1994) for the case of SSP. Moreover in our context, each algorithm is associated with multiple nonexpansive mappings instead of the single mapping of the framework of Tsitsiklis (1994). Thus besides the need to study properties that are unique to the SSP context, the convergence issues of our algorithms of Bertsekas and Yu (2012) are largely unsettled for SSP problems prior to this work. Neither can their convergence for general SSP models be deduced from the analysis of classical Q-learning in Tsitsiklis (1994), nor can it be deduced from the authors' earlier results for discounted problems (Bertsekas and Yu 2012).

In this work we first prove the convergence of the asynchronous deterministic version of our Q-learning algorithm. We then address the convergence of our asynchronous stochastic Q-learning algorithm using the framework of Tsitsiklis (1994), to which we add proper extensions to address the use of multiple mappings in an asynchronous stochastic approximation algorithm. Building on the convergence results of Tsitsiklis (1994) and the boundedness results of Yu and Bertsekas (2011), we prove that the iterates of our stochastic Q-learning algorithm are bounded and convergent under Assumption 1.1. Discounted MDP and undiscounted SSP problems are the only two types of MDP for which classical Q-learning with a totally asynchronous implementation is proved to converge. Thus with this paper and Bertsekas and Yu (2012), we establish that our new PI-like Q-learning algorithms have convergence guarantees that fully match those of the classical Q-learning algorithm.

The paper is organized as follows. In Sect. 2, we introduce the deterministic synchronous and asynchronous forms of our Q-learning algorithm for the case of exact/lookup table representation of Q-factors, and we discuss their connection with PI and VI. We also analyze basic properties of the associated mappings and prove convergence of the algorithms. This section serves as the basis for the subsequent sections. In Sect. 3, we introduce the asynchronous stochastic Q-learning algorithm, and we establish its convergence. In Sect. 4, we discuss function approximation of costs and Q-factors, including a simulation-based approximation algorithm, and we give an associated error bound.

2 Deterministic forms of new Q-learning algorithms

In this section, we introduce deterministic versions of our PI-like Q-learning algorithms, presenting first a synchronous prototype algorithm and then its asynchronous implementations.

We analyze the mappings underlying the algorithms and provide a convergence proof for the general class of SSP models satisfying Assumption 1.1. These deterministic algorithms and basic properties of their associated mappings will be the basis for the development of stochastic algorithms in later sections.

We begin this section by introducing notation and the Bellman equation mappings associated with VI and PI for Q-factors in SSP problems.

2.1 Background and notation

We consider SSP problems whose models satisfy Assumption 1.1. In such an SSP, the optimal cost function J^* is finite-valued and it is the unique solution of the Bellman equation (Bertsekas and Tsitsiklis 1991):

$$J^*(i) = \min_{u\in U(i)} \left\{ g(i,u) + \sum_{j\in S} p_{ij}(u) J^*(j) \right\}, \quad i \in S, \tag{2.1}$$

where $g(i,u) = \sum_{j\in S_o} p_{ij}(u)\hat{g}(i,u,j)$ denotes the expected one-stage cost of applying control u at state i. Any policy in Π_{SD} that minimizes the right-hand side for every state is an optimal policy for the SSP. For a state i and feasible control $u \in U(i)$, the optimal Q-factor $Q^*(i,u)$ is the cost of starting from state i, using u at the first step and using an optimal policy thereafter. The optimal costs and optimal Q-factors are thus related by

$$J^*(i) = \min_{u\in U(i)} Q^*(i,u), \quad i \in S, \tag{2.2}$$

and

$$Q^*(i,u) = g(i,u) + \sum_{j\in S} p_{ij}(u) J^*(j), \quad i \in S,\ u \in U(i). \tag{2.3}$$

Once the optimal Q-factors $Q^*(i,u)$ have been computed, the optimal cost function J^* and an optimal policy in Π_{SD} can be identified by the minimization in Eq. (2.2).

Denote the set of all state and feasible control pairs by $R = \{(i,u) \mid i \in S,\ u \in U(i)\}$. Let $Q \in \Re^{|R|}$ denote a vector with components $Q(i,u)$: $Q = \{Q(i,u) \mid (i,u) \in R\}$, and let Q^* denote the vector of optimal Q-factors.

The relations (2.1)–(2.3) show that under Assumption 1.1, Q^* is real-valued and solves uniquely the equation,

$$Q = FQ, \tag{2.4}$$

where F is the mapping given by

$$(FQ)(i,u) = g(i,u) + \sum_{j\in S} p_{ij}(u) \min_{v\in U(j)} Q(j,v), \quad (i,u) \in R. \tag{2.5}$$

This is Bellman's equation for Q-factors.[1]

For other policies, Q-factors can be similarly defined. In particular, for any policy $\mu \in \Pi_{\mathrm{SD}}$ and state-control pair $(i,u) \in R$, the Q-factor for μ and (i,u), denoted $Q_\mu(i,u)$, is the cost of starting from state i, applying control u, and afterwards following policy μ. When μ

[1] That the Bellman equation (2.4) for Q-factors has a unique solution can alternatively be established by considering an equivalent SSP problem on the state space $R \cup \{(0,0)\}$ and applying the results of Bertsekas and Tsitsiklis (1991).

is a proper policy, these Q-factors are finite-valued and solve uniquely the Bellman equation corresponding to μ,

$$Q = F_\mu Q,$$

where F_μ is the mapping given by

$$(F_\mu Q)(i,u) = g(i,u) + \sum_{j \in S} p_{ij}(u) Q\bigl(j, \mu(j)\bigr), \quad (i,u) \in R, \tag{2.6}$$

and $\mu(j)$ denotes the control applied by the deterministic policy μ at state i.

When μ is a proper policy, F_μ is a weighted sup-norm contraction, with the norm and the modulus of contraction depending on μ. If all policies in Π_{SD} are proper, then the Bellman equation mapping F is also a weighted sup-norm contraction. Both facts follow from Bertsekas and Tsitsiklis (1996, Proposition 2.2, pp. 23–24). In general, under the SSP model Assumption 1.1, F is not necessarily a contraction with respect to any norm. Instead, it is only guaranteed to be nonexpansive with respect to the unweighted sup-norm, and this is a source of analytical and algorithmic complications in SSP problems.

We specify some notational conventions. Throughout the paper, we adopt the following convention for treating the termination state 0 in various equations and algorithms. We write the optimality and other equations for all states except the cost-free and absorbing state 0, since for that state the cost of any policy is 0. In our notation, for example, $Q^* = FQ^*$ is Bellman's equation after eliminating the terms involving state 0. (Note in particular that $\sum_{j \in S} p_{ij}(u) \le 1$ in these equations.) For some of our algorithms, however, it will be simpler to use notation such as $J^*(0)$, $J(0)$, or $Q^*(0,0)$, $Q(0,0)$, to refer to the cost and Q-factor at state 0 with control 0. Therefore, for simplicity as well as convenience, we regard the space $\Re^n$ of cost vectors J^* and J as the n-dimensional subspace $\{(J(0), J(1), \dots, J(n)) \mid J(0) = 0\}$, embedded within $\Re^{n+1}$, and we will use the two notions interchangeably depending on the context. Similarly for Q-factors, we denote $R_o = R \cup \{(0,0)\}$, and we regard the space $\Re^{|R|}$ of Q^* and Q as the $|R|$-dimensional subspace $\{\{Q(i,u) \mid (i,u) \in R_o\} \mid Q(0,0) = 0\}$, embedded within $\Re^{|R_o|}$. We will use these two views interchangeably depending on the context. Furthermore, in the definition of any particular policy, we will implicitly assume that the policy applies at state 0 the only available control, 0.

Generally, all vectors in our development are viewed as column vectors in some Euclidean space $\Re^d$ of appropriate finite dimension d. All vector inequalities are meant to be componentwise.

2.2 A prototype algorithm

In its basic form, our algorithm operates on the joint cost/Q-factor space of (J, Q) and computes iteratively a sequence (J_k, Q_k), $k \ge 0$, in a way reminiscent of PI, in order to find the optimal costs and Q-factors (J^*, Q^*). The distinctive feature of the algorithm is that at the policy evaluation phase of each iteration, Q-factors Q_k are updated not by solving a linear equation $Q = F_\mu Q$ for some policy μ [cf. Eq. (2.6)], but by solving exactly or inexactly a certain optimal stopping problem, which is defined based on the progress of the algorithm. This phase involves mappings that are characteristic to our algorithms and are defined as follows.

Let Π_{SR} denote the set of stationary randomized policies. For each $\nu \in \Pi_{\mathrm{SR}}$, let $\nu(u \mid i)$, where $(i,u) \in R$, denote the probability of using control u at state i under ν. For a given $J \in \Re^{|S|}$ and $\nu \in \Pi_{\mathrm{SR}}$, define a mapping $F_{J,\nu} : \Re^{|R|} \mapsto \Re^{|R|}$ by

$$(F_{J,\nu}Q)(i,u) = g(i,u) + \sum_{j\in S} p_{ij}(u) \sum_{v\in U(j)} \nu(v \mid j) \min\bigl\{J(j), Q(j,v)\bigr\}, \quad (i,u) \in R. \tag{2.7}$$

The form of $F_{J,\nu}$ reveals its connection with an optimal stopping problem,[2] which is defined on the state space R_o, with the Markov chain being the same as the one induced by the randomized policy ν in the SSP problem, and with the stopping costs specified by the vector J. In particular, at a state $(i,u) \in R_o$, the probability of transition to state $(j,v) \in R_o$ is $p_{ij}(u)\nu(v \mid j)$, the cost of stopping is $J(i)$, and the expected one-stage cost of continuation is $g(i,u)$. The mapping $F_{J,\nu}$ and the equation $Q = F_{J,\nu}Q$ are in fact the Bellman operator and the Bellman equation of this optimal stopping problem, respectively (they both are for the Q-factors associated with the continuation action; cf. Footnote 2). When the SSP model satisfies Assumption 1.1, the equation $Q = F_{J,\nu}Q$ has a unique solution $Q_{J,\nu}$, which is the vector of optimal Q-factors (with the continuation action) of the optimal stopping problem, and $F^m_{J,\nu}Q$ converges to $Q_{J,\nu}$ for any initial Q. (Here by $F^m_{J,\nu}$ we mean the m-fold composition of $F_{J,\nu}$ with itself.) This fact will be helpful in understanding our algorithm. We state it, together with another important property, in the following proposition.

Proposition 2.1 *Under Assumption 1.1, the mapping $F_{J,\nu}$ given by Eq. (2.7) has the following properties:*

(i) *For any J and $\nu \in \Pi_{\mathrm{SR}}$, $F_{J,\nu}$ has a unique fixed point $Q_{J,\nu}$, and $\lim_{m\to\infty} F^m_{J,\nu}Q = Q_{J,\nu}$ for any $Q \in \Re^{|R|}$.*

(ii) *For any $\nu \in \Pi_{\mathrm{SR}}$, Q^* is the unique fixed point of $F_{J^*,\nu}$, i.e., $Q^* = F_{J^*,\nu}Q^*$.*

Proof Due to its length, we give the proof of (i) in Appendix A. The uniqueness part in (ii) follows from (i). We show here $F_{J^*,\nu}Q^* = Q^*$. By Eq. (2.2), $J^*(j) \le Q^*(j,v)$ for all $v \in U(j)$, $j \in S$, so by Eqs. (2.7), (2.3),

$$\begin{aligned}\bigl(F_{J^*,\nu}Q^*\bigr)(i,u) &= g(i,u) + \sum_{j\in S} p_{ij}(u) \sum_{v\in U(j)} \nu(v \mid j) J^*(j)\\ &= g(i,u) + \sum_{j\in S} p_{ij}(u) J^*(j) = Q^*.\end{aligned}$$

□

Our basic algorithm is as follows. At iteration k, given (J_k, Q_k), the algorithm selects a randomized policy ν_k and computes Q_{k+1} by

$$Q_{k+1} = F^{m_k}_{J_k,\nu_k} Q_k \tag{2.8}$$

[2] The type of optimal stopping problem we encounter here is the standard optimal stopping problem in MDP (see e.g., Puterman 1994). It involves an uncontrolled finite-state Markov chain and the option to stop the process at any state. Described in the MDP framework, it has two controls at each state: to continue the process and to stop the process. Suppose the state space is S and the transition probabilities of the Markov chain are p_{ij}, $i, j \in S$. Then the Bellman equation is

$$J(i) = \min\Bigl\{c(i),\ g(i) + \sum_{j\in S} p_{ij} J(j)\Bigr\}, \quad i \in S,$$

where $c(i)$ is the stopping cost and $g(i)$ the expected one-stage cost for continuation at state i. Correspondingly, the Bellman equation for Q-factors, with u_s standing for "to stop" and u_c "to continue," is given by

$$Q(i,u_s) = c(i), \qquad Q(i,u_c) = g(i) + \sum_{j\in S} p_{ij} \min\bigl\{c(j), Q(j,u_c)\bigr\}, \quad i \in S,$$

and it is a system of equations in the Q-factors $Q(i,u_c)$ associated with the control "to continue."

for some chosen integer $m_k \geq 1$. This is the policy evaluation phase. Subsequently the algorithm computes J_{k+1} by

$$J_{k+1}(i) = \min_{u \in U(i)} Q_{k+1}(i,u), \quad i \in S. \tag{2.9}$$

This cost minimization step is analogous to the policy improvement phase.

To convey some intuition about the algorithm and its variants to be introduced shortly, we note a correspondence between the Q-factors of the original SSP problem and the optimal Q-factors $Q_{J,\nu}$ of the optimal stopping problem associated with $F_{J,\nu}$ for a vector J and a randomized policy ν. If J is the cost vector J_π of some policy π (possibly randomized and history-dependent) in the SSP, then $Q_{J,\nu}$ is the Q-factor vector of a policy that switches optimally from following the policy ν to following the policy π. This means that if ν is some trial policy and $J(i)$ are costs known to be achievable from each state i, by solving the corresponding optimal stopping problem $Q = F_{J,\nu}Q$ for policy evaluation, the "improving part" of ν can be taken into account, while the "non-improving part" of ν can be avoided, when estimating the least costs that are attainable at each state. [In particular, if $J = J^*$, then $Q_{J,\nu} = Q^*$ regardless of ν by Proposition 2.1(ii); similarly, if $J \approx J^*$, we still have $Q_{J,\nu} \approx Q^*$ regardless of the choice of ν.] The property of $F_{J,\nu}$ just described is desirable in model-free learning. There, due to the stochastic nature of the environment, one generally does not obtain policies with successively improving performance over the entire state space. Moreover, in addition to assessing the most promising policies, one also needs to try out other policies in order to explore the environment. The ability of an algorithm to improve cost estimates incrementally, on only parts of the state space at a time if necessary, with flexibility in the choice of policies, helps to address the multiple objectives present in learning. Adding to this discussion the fact that fixed point iterations with $F_{J,\nu}$ converge to $Q_{J,\nu}$ under our SSP model assumption (Proposition 2.1(i)), our motivation for using the mapping $F_{J,\nu}$ in the policy evaluation phase of the algorithm (2.8)–(2.9) can be seen.

We note, however, that the character of the algorithm (2.8)–(2.9) is strongly affected by the choice the randomized policies ν_k and the number m_k of fixed point iterations in the policy evaluation phases. As the preceding discussion already suggested, for the algorithm to have a strong PI character, m_k should not be too small, and ν_k should be chosen based on policy improvement: either ν_{k+1} is equal to the policy μ_{k+1} attaining the minimum in Eq. (2.9), or it is an "exploration-enhanced" version of μ_{k+1}, which combines μ_{k+1} with some policy, for example, the one that randomly samples the control space.

Generally, depending on the choices of ν_k and m_k, the behavior of the algorithm (2.8)–(2.9) is intermediate between PI and VI. In particular, the algorithm (2.8)–(2.9) reduces to VI, if $m_k = 1$, or if ν_k has rather poor performance so that it is always better to take the stopping action in the optimal stopping problem corresponding to F_{J_k,ν_k}. This is similar to the qualitative behavior of the algorithm for discounted MDP; more detailed discussion can be found in Bertsekas and Yu (2012, Sect. 2), where the algorithm was first proposed.

Under Assumption 1.1, the basic algorithm (2.8)–(2.9) converges to (J^*, Q^*) regardless of the choices of ν_k and m_k, and in particular, regardless of whether ν_k is proper. This is in contrast to PI and modified PI, which need additional care to handle improper policies in SSP. The convergence of our basic algorithm will be proved as a special case of the convergence of asynchronous algorithms which we present next.

2.3 Deterministic asynchronous algorithms

The basic algorithm (2.8)–(2.9) may be viewed as synchronous in the sense that the Q-factors of all state-control pairs are simultaneously updated at each iteration. In corresponding asynchronous versions, J is updated selectively, for only some of the states, and Q is also updated at some iterations and for some of the state-control pairs. Asynchronous algorithmic models are necessary for analysis of PI algorithms where the states and state-control pairs at which J and Q are updated, respectively, are generated by simulating the policy that is currently evaluated. Asynchronous models are also needed for analysis of distributed algorithms involving a network of processors, each assigned a subset of components of J and Q, and updating asynchronously these components.

An asynchronous implementation of the basic algorithm (2.8)–(2.9) is as follows. We generate a sequence of pairs (J_k, Q_k), starting from an arbitrary pair (J_0, Q_0). Given (J_k, Q_k), to obtain the next pair (J_{k+1}, Q_{k+1}), we first select a randomized policy ν_k, a subset R_k of state-control pairs, and a subset of states S_k such that $R_k \cup S_k \neq \emptyset$. We then generate Q_{k+1} according to

$$Q_{k+1}(i,u) = \begin{cases} (F_{J_k,\nu_k} Q_k)(i,u) & \text{if } (i,u) \in R_k, \\ Q_k(i,u) & \text{if } (i,u) \notin R_k, \end{cases} \tag{2.10}$$

and J_{k+1} according to

$$J_{k+1}(i) = \begin{cases} \min_{u \in U(i)} Q_k(i,u) & \text{if } i \in S_k, \\ J_k(i) & \text{if } i \notin S_k. \end{cases} \tag{2.11}$$

The basic synchronous algorithm (2.8)–(2.9) is obtained from the above algorithm if we either update all the costs but none of the Q-factors, or update none of the costs but all the Q-factors (possibly multiple times with the same policy ν_k).

In an efficient implementation of the above method, the sets S_k are empty for many iterations and only the update (2.10) is performed on Q-factors. Moreover, ν_k may be selected in special ways to give the algorithm a PI character. For example, assume that a deterministic policy μ_k is also maintained and $\nu_k = \mu_k$. The algorithm updates Q according to

$$Q_{k+1}(i,u) = \begin{cases} (F_{J_k,\mu_k} Q_k)(i,u) & \text{if } (i,u) \in R_k, \\ Q_k(i,u) & \text{if } (i,u) \notin R_k, \end{cases} \tag{2.12}$$

and it updates J and μ according to

$$J_{k+1}(i) = \min_{u \in U(i)} Q_k(i,u), \qquad \mu_{k+1}(i) \in \arg\min_{u \in U(i)} Q_k(i,u), \qquad \text{if } i \in S_k; \tag{2.13}$$

$$J_{k+1}(i) = J_k(i), \qquad \mu_{k+1}(i) = \mu_k(i), \qquad \text{if } i \notin S_k. \tag{2.14}$$

We may view Eq. (2.12) as a policy evaluation iteration only for the state-control pairs in R_k, and Eqs. (2.13)–(2.14) as a policy improvement iteration only for the states in S_k.

The algorithmic variant (2.12)–(2.14) resembles an asynchronous version of modified PI, but differs from the latter in a major way by employing the mappings F_{J_k,μ_k} instead of F_{μ_k} [cf. Eq. (2.6)]. Asynchronous modified PI in general does not have convergence guarantees without additional restrictions on policy types and the initial conditions, as demonstrated by the counterexamples of Williams and Baird (1993) for discounted MDP (which can be viewed as special cases of SSP problems). By contrast our algorithm (2.12)–(2.14) will be shown to be convergent for any policies and initial conditions. This advantage over modified PI is not confined to Q-factor computation: an algorithm similar to (2.12)–(2.14) that operates on costs instead of costs/Q-factors, was proposed and proved convergent by the authors in Bertsekas and Yu (2010).

Our convergence result is stated in the proposition below for the general asynchronous algorithm (2.10)–(2.11). The proof will be given in the next subsection.

Theorem 2.1 *Under Assumption 1.1, any sequence $\{(J_k, Q_k)\}$ generated by iteration (2.10)–(2.11) converges to (J^*, Q^*), if every state or state-control pair is included in the subset S_k or R_k, respectively, for infinitely many k.*

We note that there are more general versions of the algorithm (2.10)–(2.11), which can use cost/Q-factor values generated at earlier times when updating (J_k, Q_k) to (J_{k+1}, Q_{k+1}). These variants are natural for a parallel distributed implementation with a network of processors, where communication delays between processors need to be taken into account. As algorithmic models, these variants also entail an extension of the algorithm (2.10)–(2.11), which employs a different policy ν_k for each selected state-control pair in the set R_k. We do not introduce these algorithms here, for the extra notation needed can obscure the main ideas and the basic properties of our algorithms, which are the focus of this section. However, we note that with slight modification, our proof of Theorem 2.1 can be extended to show the convergence of these asynchronous algorithms when communication delays are present. (The framework of asynchronous distributed computation with delays will be considered in Sect. 3, where the stochastic version of our algorithm is analyzed.)

2.4 Basic properties of mappings and convergence analysis

We analyze some basic properties of the mappings underlying the algorithms (2.8)–(2.9) and (2.10)–(2.11), which ensure their convergence. Whereas before we were focusing primarily on the PI-type properties of the algorithms, in this section we shift the emphasis to VI-type properties. In particular, we will analyze properties relating to monotonicity, nonexpansiveness, fixed points, and in special cases, contraction. This analysis shows that the convergence guarantee does not depend critically on the choices of policies ν_k and subsets S_k, R_k, so in this sense our analysis has a worst-case character.

For the convergence analysis, it is more convenient to consider the joint space of (J, Q) and introduce a set of mappings $\{L^\nu \mid \nu \in \Pi_{\mathrm{SR}}\}$ on this space, which combine the policy evaluation and policy improvement/cost minimization phases of the basic algorithm. More specifically, let $M : \Re^{|R|} \mapsto \Re^{|S|}$ denote the operator of minimization of Q-factors over the controls feasible at each state: for all $Q \in \Re^{|R|}$,

$$(MQ)(i) = \min_{u\in U(i)} Q(i,u), \quad i \in S. \tag{2.15}$$

For each $\nu \in \Pi_{\mathrm{SR}}$, define a mapping $L^\nu : \Re^{|S\cup R|} \mapsto \Re^{|S\cup R|}$ by

$$L^\nu(J, Q) = (MQ, F_{J,\nu} Q), \tag{2.16}$$

i.e., L^ν maps (J, Q) to (J', Q') given by

$$\begin{aligned} J'(i) &= (MQ)(i) = \min_{u\in U(i)} Q(i,u), \quad i \in S; \\ Q'(i,u) &= (F_{J,\nu} Q)(i,u), \quad (i,u) \in R. \end{aligned} \tag{2.17}$$

The algorithms introduced earlier (as well as their more general variants) can be viewed as asynchronous fixed point iterations with mappings L^ν, whereby for each selected state or state-control pair $\ell \in S \cup R$, we update the ℓth component of (J, Q) to be the ℓth component of $L^\nu(J, Q)$ for some mapping L^ν, $\nu \in \Pi_{\mathrm{SR}}$, which may be chosen arbitrarily. Thus properties common to this set $\{L^\nu \mid \nu \in \Pi_{\mathrm{SR}}\}$ of mappings are important for the algorithmic analysis, and will be our focus below.

Let us introduce in addition two mappings associated with the set $\{L^\nu \mid \nu \in \Pi_{\mathrm{SR}}\}$, which will play a key role in analyzing convergence in the presence of asynchrony. Define the mappings $\overline{L}$ and $\underline{L}$ by taking componentwise supremum and infimum, respectively, of $L^\nu(J, Q)$ over $\nu \in \Pi_{\mathrm{SR}}$, i.e., for all $x = (J, Q) \in \Re^{|S\cup R|}$,

$$(\overline{L}x)(\ell) = \sup_{\nu\in\Pi_{\mathrm{SR}}} \big(L^\nu x\big)(\ell), \qquad (\underline{L}x)(\ell) = \inf_{\nu\in\Pi_{\mathrm{SR}}} \big(L^\nu x\big)(\ell), \quad \forall \ell \in S \cup R. \tag{2.18}$$

Stated more explicitly,

$$\overline{L}(J,Q) = (MQ, \overline{F}_J Q), \qquad \underline{L}(J,Q) = (MQ, \underline{F}_J Q),$$

where $\overline{F}_J$, $\underline{F}_J$ are mappings given by for all $(i,u) \in R$,

$$(\overline{F}_J Q)(i,u) = \sup_{\nu \in \Pi_{\mathrm{SR}}} (F_{J,\nu} Q)(i,u) = g(i,u) + \sum_{j \in S} p_{ij}(u) \min\Big\{ J(j), \max_{v \in U(j)} Q(j,v) \Big\},$$

$$(\underline{F}_J Q)(i,u) = \inf_{\nu \in \Pi_{\mathrm{SR}}} (F_{J,\nu} Q)(i,u) = g(i,u) + \sum_{j \in S} p_{ij}(u) \min\Big\{ J(j), \min_{v \in U(j)} Q(j,v) \Big\}.$$

Note that the componentwise supremum or infimum of $L^\nu(J,Q)$ in the definition of $\overline{L}$ or $\underline{L}$ is attained simultaneously for all components by some $\nu \in \Pi_{\mathrm{SR}}$. In other words, for any given (J,Q), there exist some $\overline{\nu}$, $\underline{\nu} \in \Pi_{\mathrm{SR}}$ such that

$$\overline{L}(J,Q) = L^{\overline{\nu}}(J,Q), \qquad \underline{L}(J,Q) = L^{\underline{\nu}}(J,Q). \tag{2.19}$$

Some basic properties of the mappings L^ν, $\nu \in \Pi_{\mathrm{SR}}$, and $\overline{L}$ and $\underline{L}$ are given in the following proposition.

Proposition 2.2 *The mappings L^ν, $\nu \in \Pi_{\mathrm{SR}}$ given by Eq. (2.16), and their associated $\overline{L}$, $\underline{L}$ mappings given by Eq. (2.18) are monotone and nonexpansive with respect to the sup-norm. Under Assumption 1.1, they all have (J^*, Q^*) as their unique fixed point.*

Proof Consider any $\nu \in \Pi_{\mathrm{SR}}$ and any two cost/Q-factor pairs (J_1, Q_1), (J_2, Q_2). If $J_1 \le J_2$ and $Q_1 \le Q_2$, then by a direct calculation, $MQ_1 \le MQ_2$ and $F_{J_1,\nu} Q_1 \le F_{J_2,\nu} Q_2$, so $L^\nu(J_1, Q_1) \le L^\nu(J_2, Q_2)$ and L^ν is monotone. Observe that for any two sets of numbers $\{a_i\}_{i \in I}$, $\{b_i\}_{i \in I}$, where I is an index set, $|\min_{i \in I} a_i - \min_{i \in I} b_i| \le \max_{i \in I} |a_i - b_i|$.[3] Using this fact, a direct calculation gives

$$\|MQ_1 - MQ_2\|_\infty \le \|Q_1 - Q_2\|_\infty,$$
$$\|F_{J_1,\nu} Q_1 - F_{J_2,\nu} Q_2\|_\infty \le \max\big\{ \|J_1 - J_2\|_\infty, \|Q_1 - Q_2\|_\infty \big\},$$

and therefore,

$$\begin{aligned} \big\| L^\nu(J_1,Q_1) - L^\nu(J_2,Q_2) \big\|_\infty &\le \max\big\{ \|J_1 - J_2\|_\infty, \|Q_1 - Q_2\|_\infty \big\} \\ &= \big\| (J_1,Q_1) - (J_2,Q_2) \big\|_\infty. \end{aligned}$$

This shows that L^ν is nonexpansive with respect to the sup-norm $\|\cdot\|_\infty$.

We now show that L^ν has (J^*, Q^*) as its unique fixed point. Under Assumption 1.1, by Eq. (2.2) and Proposition 2.1(ii), we have $MQ^* = J^*$ and $F_{J^*,\nu} Q^* = Q^*$, so (J^*, Q^*) is a fixed point of L^ν. Now let $(\bar{J}, \bar{Q})$ be any fixed point of L^ν, i.e., $\bar{J} = M\bar{Q}$ and $\bar{Q} = F_{\bar{J},\nu} \bar{Q}$. Using the fact $\bar{J} = M\bar{Q}$ and the definition of the Bellman mapping F [cf. Eq. (2.4)], we obtain $F_{\bar{J},\nu} \bar{Q} = F\bar{Q}$. So

$$\bar{Q} = F_{\bar{J},\nu} \bar{Q} = F\bar{Q},$$

[3] The fact $|\min_{i \in I} a_i - \min_{i \in I} b_i| \le \max_{i \in I} |a_i - b_i|$ can be seen as follows. For every $i \in I$, since

$$a_i \le b_i + |a_i - b_i| \le b_i + \max_{i' \in I} |a_{i'} - b_{i'}|,$$

we have $\min_{i \in I} a_i \le \min_{i \in I} b_i + \max_{i \in I} |a_i - b_i|$. By the same argument, $\min_{i \in I} b_i \le \min_{i \in I} a_i + \max_{i \in I} |a_i - b_i|$. Hence, the desired inequality holds.

implying that $\bar{Q}$ is a fixed point of F. Since under Assumption 1.1 F has Q^* as its unique fixed point, $\bar{Q} = Q^*$. This in turn implies $\bar{J} = MQ^* = J^*$ [cf. Eq. (2.2)]. Thus (J^*, Q^*) is the unique fixed point of L^ν.

Regarding the mappings $\overline{L}$ and $\underline{L}$, as componentwise supremum and infimum of $L^\nu(J, Q)$ over ν, they inherit the following properties that are common to all mappings L^ν: monotonicity and nonexpansiveness with respect to $\|\cdot\|_\infty$, as well as having (J^*, Q^*) as their fixed point. It then follows from Eq. (2.19) that $\overline{L}$ and $\underline{L}$ cannot have a fixed point other than (J^*, Q^*) [otherwise, some mapping L^ν would have a fixed point other than (J^*, Q^*)]. □

We now recall a simple convergence result about fixed point iterations involving a nonexpansive mapping. The proof argument is standard and will also be used in the proof of Theorem 2.1 shortly.

Lemma 2.1 *Let H be a monotone, nonexpansive* (*with respect to* $\|\cdot\|_\infty$) *operator on $\Re^d$ with a unique fixed point x^*. Then $\lim_{k\to\infty} H^k x = x^*$ for all $x \in \Re^d$.*

Proof Let $c > 0$ and let $e \in \Re^d$ denote the vector of all ones. Consider $y_k = H^k(x^* + ce)$. Equivalently, $y_0 = x^* + ce$ and $y_k = Hy_{k-1}$ for $k \geq 1$. We show that $\{y_k\}$ converges monotonically to x^*. Since H is monotone and nonexpansive,

$$x^* = Hx^* \leq H\big(x^* + ce\big) \leq Hx^* + \big\|\big(x^* + ce\big) - x^*\big\|_\infty e = x^* + ce,$$

i.e., $x^* \leq y_1 \leq y_0$. From this and the monotonicity of H^k (implied by the monotonicity of H), we have $x^* = H^k x^* \leq y_{k+1} \leq y_k$. It follows that $\{y_k\}$ is monotonically nonincreasing and converges to some $\bar{y} \geq x^*$. Since H is continuous (as implied by the nonexpansiveness of H), the relation $y_k = Hy_{k-1}$ for $k \geq 1$ implies that $\bar{y} = H\bar{y}$, so $\bar{y} = x^*$ by the uniqueness of the fixed point of H. Thus $\{y_k\}$ converges monotonically to x^* from above.

Similarly, define $z_k = H^k(x^* - ce)$, and by an argument symmetric to the above, $\{z_k\}$ converges monotonically to x^* from below. Now let $c = \|x - x^*\|_\infty$ in the definition of y_k and z_k, and let $x_k = H^k x$. Then $x^* - ce \leq x_0 = x \leq x^* + ce$, so by the monotonicity of H^k, $z_k \leq x_k \leq y_k$ for all k. This implies that $\{x_k\}$ converges to x^*. □

We can now prove Theorem 2.1 on the convergence of the deterministic versions of our algorithms. Denote $x_k = (J_k, Q_k)$. The algorithm (2.10)–(2.11) can be written equivalently as

$$x_{k+1}(\ell) = \begin{cases} (L^{\nu_k} x_k)(\ell) & \text{if } \ell \in S_k \cup R_k, \\ x_k(\ell) & \text{if } \ell \notin S_k \cup R_k. \end{cases} \tag{2.20}$$

Its convergence is the consequence of the general fact that any asynchronous iteration of this form converges, if every component is updated infinitely often and if the mappings involved together with their associated $\overline{L}$, $\underline{L}$ mappings possess the properties listed in Proposition 2.2. The proof we give below bears similarity to those in the literature for asynchronous distributed dynamic programming and related algorithms (see e.g., Bertsekas 1982, 1983; see also Bertsekas and Tsitsiklis 1989, 1996; Tsitsiklis 1994). We also note that the convergence analysis of deterministic asynchronous versions of our algorithms with delays, which we did not introduce, follows an essentially identical line.

Proof of Theorem 2.1 Let $x_k = (J_k, Q_k)$ and $x^* = (J^*, Q^*)$, and consider the equivalent expression (2.20) of the algorithm (2.10)–(2.11). We want to show $\{x_k\}$ converges to x^*. To this end, let $c = \|x_0 - x^*\|_\infty$ and define two sequences $\{y_t\}$, $\{z_t\}$ by

$$y_t = \overline{L}^t\bigl(x^* + ce\bigr), \qquad z_t = \underline{L}^t\bigl(x^* - ce\bigr), \quad t \geq 0,$$

where $e \in \Re^{|S \cup R|}$ denotes the vector of all ones. By Proposition 2.2, $\overline{L}$ and $\underline{L}$ are monotone nonexpansive mappings and have x^* as their unique fixed point. Therefore, by Lemma 2.1 and its proof, $\{y_t\}$ and $\{z_t\}$ converge to x^* monotonically with

$$z_t \leq z_{t+1} \leq x^* \leq y_{t+1} \leq y_t.$$

We show by induction that for all $t \geq 0$, there exists a time k_t such that

$$z_t \leq x_k \leq y_t, \quad \forall k \geq k_t. \tag{2.21}$$

This will imply the desired convergence of $\{x_k\}$ to x^*.

For $t = 0$, let $k_0 = 0$. By induction on k we show $z_0 \leq x_k \leq y_0$ for all k. The choice of c ensures that $x^* - ce \leq x_0 \leq x^* + ce$, so the inequality (2.21) holds trivially for $k = 0$. Now suppose $z_0 \leq x_k \leq y_0$ for some k. Consider each component of x_{k+1}. For $\ell \notin S_k \cup R_k$, $x_{k+1}(\ell) = x_k(\ell)$, so $z_0(\ell) \leq x_{k+1}(\ell) \leq y_0(\ell)$ by the induction hypothesis. For $\ell \in S_k \cup R_k$, $x_{k+1}(\ell) = (L^{\nu_k} x_k)(\ell)$. Using the definition of $\overline{L}$, $\underline{L}$ and their monotonicity property (Proposition 2.2), and using also the monotonicity property of $\{y_t\}$ and of $\{z_t\}$, we have

$$\bigl(L^{\nu_k} x_k\bigr)(\ell) \leq (\overline{L} x_k)(\ell) \leq (\overline{L} y_0)(\ell) = y_1(\ell) \leq y_0(\ell), \tag{2.22}$$

$$\bigl(L^{\nu_k} x_k\bigr)(\ell) \geq (\underline{L} x_k)(\ell) \geq (\underline{L} z_0)(\ell) = z_1(\ell) \geq z_0(\ell), \tag{2.23}$$

so $z_0(\ell) \leq x_{k+1}(\ell) \leq y_0(\ell)$ holds also for $\ell \in S_k \cup R_k$. This shows $z_0 \leq x_{k+1} \leq y_0$, completes the induction on k, and proves the inequality (2.21) for $t = 0$.

We now use induction on t. Suppose the inequality (2.21) holds for some t. Consider each $\ell \in S \cup R$. By assumption, ℓ belongs to $S_k \cup R_k$ for infinitely many k, so there exists a time $k(\ell) \geq k_t$ such that $\ell \in S_{k(\ell)} \cup R_{k(\ell)}$. Then, similar to the derivation of Eq. (2.22), we have for $k = k(\ell)$,

$$x_{k+1}(\ell) = \bigl(L^{\nu_k} x_k\bigr)(\ell) \leq (\overline{L} x_k)(\ell) \leq (\overline{L} y_t)(\ell) = y_{t+1}(\ell), \tag{2.24}$$

where in the second inequality we also used the induction hypothesis that $x_k \leq y_t$ for all $k \geq k_t$. We show by induction on k that $x_{k+1}(\ell) \leq y_{t+1}(\ell)$ for all $k \geq k(\ell)$. Indeed, suppose this is true for some $\bar{k} \geq k(\ell)$. Then for $k = \bar{k} + 1$, if $\ell \in S_k \cup R_k$, the same reasoning leading to (2.24) shows $x_{k+1}(\ell) \leq y_{t+1}(\ell)$, whereas if $\ell \notin S_k \cup R_k$, then $x_{k+1}(\ell) = x_k(\ell) \leq y_{t+1}(\ell)$ by the induction hypothesis. This establishes $x_{k+1}(\ell) \leq y_{t+1}(\ell)$ for all $k \geq k(\ell)$. By a similar, symmetric argument, we have that for $k = k(\ell)$,

$$x_{k+1}(\ell) = \bigl(L^{\nu_k} x_k\bigr)(\ell) \geq (\underline{L} x_k)(\ell) \geq (\underline{L} z_t)(\ell) = z_{t+1}(\ell),$$

and for all $k \geq k(\ell)$, $x_{k+1}(\ell) \geq z_{t+1}(\ell)$. Hence $z_{t+1}(\ell) \leq x_{k+1}(\ell) \leq y_{t+1}(\ell)$ for all $k \geq k(\ell)$. Letting $k_{t+1} = 1 + \max_{\ell \in S \cup R} k(\ell)$, it follows that inequality (2.21) holds for $t + 1$ and the chosen k_{t+1}. This completes the induction on t and proves that the inequality (2.21) holds for all t. The proof is complete. □

2.4.1 Special SSP models and contraction properties

We now consider the special class of SSP models where all policies are proper. For these models, the various mappings associated with our algorithms also possess uniform sup-norm contraction properties, which we now derive. These properties are stronger than those given in Proposition 2.2. They will be useful in the convergence analysis and approximation-related error bounds. Qualitatively, they also suggest that the algorithms can converge faster for this special class of SSP than for general SSP which have improper policies.

To derive the uniform contraction properties, we note that when all policies in Π_{SD} are proper, by Bertsekas and Tsitsiklis (1996, Proposition 2.2, pp. 23–24), there exist a positive vector $\xi \in \Re^n$ and a positive scalar $\beta < 1$ such that

$$\sum_{j\in S} p_{ij}(u)\xi(j) \le \beta\xi(i), \quad \forall (i,u) \in R. \tag{2.25}$$

Let ξ^x denote the extension of ξ to the space of Q given by $\xi^x(i,u) = \xi(i)$ for all $(i,u) \in R$, and let $\|\cdot\|_\xi$ and $\|\cdot\|_{\xi^x}$ denote the weighted sup-norm on $\Re^n$ and $\Re^{|R|}$ with weights ξ and ξ^x, respectively:

$$\|J\|_\xi = \max_{i\in S} \frac{|J(i)|}{\xi(i)}, \qquad \|Q\|_{\xi^x} = \max_{(i,u)\in R} \frac{|Q(i,u)|}{\xi(i)}. \tag{2.26}$$

Lemma 2.2 *Assume that all policies in Π_{SD} are proper and let $\nu \in \Pi_{\mathrm{SR}}$. Then for any $(J,Q), (\hat{J},\hat{Q})$,*

$$\|F_{J,\nu}Q - F_{\hat{J},\nu}\hat{Q}\|_{\xi^x} \le \beta \max\{\|J-\hat{J}\|_\xi, \|Q-\hat{Q}\|_{\xi^x}\}, \tag{2.27}$$

$$\|MQ - M\hat{Q}\|_\xi \le \|Q-\hat{Q}\|_{\xi^x}, \tag{2.28}$$

where ξ, $\|\cdot\|_\xi$, $\|\cdot\|_{\xi^x}$, and $\beta \in (0,1)$, are given by Eqs. (2.25)–(2.26) and independent of ν.

Proof For all $(i,u) \in R$,

$$\begin{aligned} &\big|(F_{J,\nu}Q)(i,u) - (F_{\hat{J},\nu}\hat{Q})(i,u)\big| \\ &\quad \le \sum_{j\in S} p_{ij}(u) \sum_{v\in U(j)} \nu(v \mid j)\big|\min\{J(j), Q(j,v)\} - \min\{\hat{J}(j), \hat{Q}(j,v)\}\big| \\ &\quad \le \sum_{j\in S} p_{ij}(u) \sum_{v\in U(j)} \nu(v \mid j) \max\{|J(j)-\hat{J}(j)|, |Q(j,v)-\hat{Q}(j,v)|\} \end{aligned}$$

(where we used the fact in Footnote 3 to obtain the second inequality). For all $(j,v) \in R$,

$$\begin{aligned} &\max\{|J(j)-\hat{J}(j)|, |Q(j,v)-\hat{Q}(j,v)|\} \\ &\quad = \xi(j)\max\left\{\frac{|J(j)-\hat{J}(j)|}{\xi(j)}, \frac{|Q(j,v)-\hat{Q}(j,v)|}{\xi(j)}\right\} \\ &\quad \le \xi(j)\max\{\|J-\hat{J}\|_\xi, \|Q-\hat{Q}\|_{\xi^x}\}. \end{aligned}$$

Hence, for all $(i,u) \in R$,

$$\begin{aligned} \big|(F_{J,\nu}Q)(i,u) - (F_{\hat{J},\nu}\hat{Q})(i,u)\big| &\le \left(\sum_{j\in S} p_{ij}(u)\xi(j)\right)\max\{\|J-\hat{J}\|_\xi, \|Q-\hat{Q}\|_{\xi^x}\} \\ &\le \beta\,\xi(i)\max\{\|J-\hat{J}\|_\xi, \|Q-\hat{Q}\|_{\xi^x}\}, \end{aligned}$$

where the last inequality follows from Eq. (2.25). This implies Eq. (2.27).

For all $i \in S$, since $|\min_{u\in U(i)} Q(i,u) - \min_{u\in U(i)} \hat{Q}(i,u)| \le \max_{u\in U(i)} |Q(i,u) - \hat{Q}(i,u)|$ [cf. Footnote 3],

$$\frac{|\min_{u\in U(i)} Q(i,u) - \min_{u\in U(i)} \hat{Q}(i,u)|}{\xi(i)} \le \max_{u\in U(i)} \frac{|Q(i,u)-\hat{Q}(i,u)|}{\xi(i)}.$$

Taking maximum over i on both sides gives Eq. (2.28). □

Using the preceding lemma, we can construct a weighted sup-norm with respect to which a contraction property holds uniformly for all L^{ν}. This will be useful in establishing the convergence of our stochastic Q-learning algorithm for the special class of SSP models (Proposition 3.2).

Proposition 2.3 *Assume that all policies in Π_{SD} are proper. Then there exist a weighted sup-norm $\|\cdot\|_\zeta$ on the space of (J,Q) and a positive scalar $\bar\beta < 1$ such that for all $\nu \in \Pi_{\mathrm{SR}}$, the mappings L^{ν} given by Eq. (2.16) are contractions of modulus $\bar\beta$ with respect to $\|\cdot\|_\zeta$,*

$$\big\|L^{\nu}(J,Q) - L^{\nu}(\hat J,\hat Q)\big\|_\zeta \le \bar\beta \big\|(J,Q)-(\hat J,\hat Q)\big\|_\zeta, \quad \forall (J,Q),(\hat J,\hat Q),$$

with the same fixed point (J^*,Q^*).

Proof The proof is similar to that of Proposition 4.1 in Bertsekas and Yu (2012). Let the weighted sup-norms $\|J\|_\xi$, $\|Q\|_{\xi^x}$, and the scalar $\beta \in (0,1)$ be as in Lemma 2.2. We define a weighted sup-norm $\|\cdot\|_\zeta$ on the space of (J,Q) by

$$\big\|(J,Q)\big\|_\zeta = \max\big\{\|J\|_\xi, c\|Q\|_{\xi^x}\big\}, \tag{2.29}$$

where c is any positive scalar with $1 < c < 1/\beta$. Define $\bar\beta = \max\{c\beta, 1/c\}$. Then,

$$\beta < \bar\beta < 1, \qquad c\beta \le \bar\beta, \qquad \bar\beta c \ge 1.$$

(Take, for instance, $c = 1/\sqrt{\beta}$ to have $\bar\beta = \sqrt{\beta}$.) Consider any $\nu \in \Pi_{\mathrm{SR}}$ and any two pairs $(J,Q),(\hat J,\hat Q)$. By the definition of L^{ν} [cf. Eq. (2.16)],

$$L^{\nu}(J,Q) - L^{\nu}(\hat J,\hat Q) = (MQ - M\hat Q, F_{J,\nu}Q - F_{\hat J,\nu}\hat Q).$$

Using Eq. (2.27) of Lemma 2.2 and the fact $c\beta \le \bar\beta$, $\beta < \bar\beta$, we have

$$c\|F_{J,\nu}Q - F_{\hat J,\nu}\hat Q\|_{\xi^x} \le \max\big\{c\beta\|J-\hat J\|_\xi, c\beta\|Q-\hat Q\|_{\xi^x}\big\} \le \bar\beta\big\|(J,Q)-(\hat J,\hat Q)\big\|_\zeta,$$

and using Eq. (2.28) of Lemma 2.2 and the fact $\bar\beta c \ge 1$, we have

$$\begin{aligned}\|MQ - M\hat Q\|_\xi \le (\bar\beta c)\|Q-\hat Q\|_{\xi^x} &\le \max\big\{\bar\beta\|J-\hat J\|_\xi, \bar\beta c\|Q-\hat Q\|_{\xi^x}\big\} \\ &\le \bar\beta\big\|(J,Q)-(\hat J,\hat Q)\big\|_\zeta.\end{aligned}$$

From the preceding two relations, we obtain $\|L^{\nu}(J,Q) - L^{\nu}(\hat J,\hat Q)\|_\zeta \le \bar\beta\|(J,Q) - (\hat J,\hat Q)\|_\zeta$. Finally, that (J^*,Q^*) is the fixed point of L^{ν} is established in Proposition 2.2. □

Let us mention two other uses of Lemma 2.2. First, it implies that the basic synchronous algorithm (2.8)–(2.9) has in the worst case a geometric rate of convergence when all policies of the SSP are proper. This is similar to the algorithm behavior in discounted MDP (Bertsekas and Yu 2012). Another use of Lemma 2.2 will be in deriving error bounds for approximation algorithms for the special class of SSP models in Sect. 4 (Proposition 4.1).

3 A new asynchronous stochastic Q-learning algorithm

The classical Q-learning algorithm of Watkins (1989) is a stochastic approximation-type of algorithm. It replaces expected values in the formula of VI by sample values obtained with simulation and thus does not require an explicit model of the MDP. In a simple version of the algorithm for SSP problems (using simplified notation), at each iteration we select some state-control pairs (i, u), and for each of them, we simulate a transition from state i with control u to obtain a random successor state $s \in S_o$, and then update $Q_k(i, u)$ by

$$Q_{k+1}(i,u) = (1 - \gamma_{iu,k}) Q_k(i,u) + \gamma_{iu,k} \Bigl(\hat{g}(i,u,s) + \min_{v \in U(s)} Q_k(s,v) \Bigr), \qquad (3.1)$$

where $\gamma_{iu,k} \geq 0$ is a stepsize parameter and $\hat{g}(i,u,s)$ is the transition cost. The relation with VI is revealed when the above iteration is equivalently expressed as

$$Q_{k+1}(i,u) = (1 - \gamma_{iu,k}) Q_k(i,u) + \gamma_{iu,k} (F Q_k)(i,u) + \gamma_{iu,k} \omega_{iu,k},$$

where $\omega_{iu,k}$ is a stochastic noise term with zero conditional mean, and F is the Bellman equation mapping [cf. Eq. (2.5)]. The classical Q-learning algorithm, in a more complex form than the above, is analyzed by Tsitsiklis (1994) in the context of asynchronous stochastic approximation algorithms. Its convergence for SSP under Assumption 1.1 and some mild algorithmic conditions is established by the results of Tsitsiklis (1994), Yu and Bertsekas (2011).

Our stochastic asynchronous Q-learning algorithm can also be viewed as a stochastic approximation algorithm. It is obtained, briefly speaking, by using simulated samples in place of expected values in the deterministic asynchronous algorithm (2.10)–(2.11) or in its more general variants that involve delays. In a simple version which parallels iteration (3.1), we update the cost vectors J_k for some selected states as in iteration (2.11), whereas we update the Q-factors Q_k for each selected state-control pair (i, u) with the iteration

$$Q_{k+1}(i,u) = (1 - \gamma_{iu,k}) Q_k(i,u) + \gamma_{iu,k} \bigl(\hat{g}(i,u,s) + \min\bigl\{ J_k(s), Q_k(s,v) \bigr\} \bigr), \qquad (3.2)$$

where $v \in U(s)$ is a control generated according to a randomized policy ν_k, and $\gamma_{iu,k}$ and s are the stepsize parameter and random successor state, respectively, as in the classical Q-learning iteration (3.1). Iteration (3.2) is much simpler than (3.1): it does not involve minimization of Q-factors over the full control set of the successor state, and instead, it compares just two quantities $J_k(s)$ and $Q_k(s, v)$. So if cost updates (2.11) are performed infrequently, the per-iteration overhead of our algorithm is lower than that of the classical Q-learning algorithm, and the computational saving can be considerable when the control sets are large.

In what follows, we introduce formally our stochastic asynchronous Q-learning algorithm for SSP, and prove its convergence under Assumption 1.1 (Theorem 3.1). To make our results comparable with those for the classical Q-learning algorithm, we will use the same asynchronous computational model with communication delays as the one in Tsitsiklis (1994) (see also Baudet 1978; Bertsekas and Tsitsiklis 1989), and we will introduce this model shortly.

In this section, the termination state 0 appears explicitly in the stochastic algorithms due to their simulation character [as it did already in the iterations (3.1) and (3.2) above]. Following our notational convention described in Sect. 2.1, we regard J_k and Q_k both as vectors on $\Re^{|S|}$ and $\Re^{|R|}$, respectively, and as vectors on the embedded subspaces of $\Re^{|S_o|}$ and $\Re^{|R_o|}$, respectively, with $J_k(0) = Q_k(0, 0) = 0$ for all $k \geq 0$.

3.1 Algorithm

Like its deterministic asynchronous version, our stochastic asynchronous Q-learning algorithm generates a sequence $\{(J_k, Q_k)\}$ by updating a subset of Q-factors and/or costs at each iteration. It involves many variables due to the presence of both simulation and asynchronous computation. To facilitate the presentation, let us first introduce the notion of "communication delays" and the related notation in an asynchronous computing framework, using intuitive terms such as "processors" and "distributed computation" (our interest, however, is in the mathematical model rather than the physical computing system). Imagine that a computation task of an iterative nature is distributed in a network of processors that operate asynchronously. Each processor updates the value of a particular component and processors exchange the results with each other. There are communication delays, so a component update is based on possibly outdated information and involves values that were calculated at earlier times. For our Q-learning algorithm, set in such a computation framework, each $\ell \in S \cup R$ is associated with a (imaginary) processor, which updates the ℓth component of (J, Q). In its computation at iteration/time k, the processor ℓ uses the value of the ℓ'th component that was computed by the processor ℓ' at time $\tau^{\ell}_{\ell',k} \le k$, where $\ell' \in S \cup R$. In other words, we may regard the difference $k - \tau^{\ell}_{\ell',k}$ as the communication delay between processors ℓ and ℓ', and regard $\{J_{\tau^{\ell}_{i,k}}(i) \mid i \in S\}$ and $\{Q_{\tau^{\ell}_{iu,k}}(i,u) \mid (i,u) \in R\}$ as the information available to the processor ℓ at time k for performing its task.

We refer to the variables $0 \le \tau^{\ell}_{\ell',k} \le k$, where $\ell, \ell' \in S \cup R$, as "delayed times." They are integer-valued and need to satisfy certain mild, minimal conditions to be specified later. For notational convenience, we also define the delayed times $\tau^{\ell}_{\ell',k}$ for $\ell' = 0$ and $\ell' = (0,0)$, albeit arbitrarily.

We now describe a general form of our asynchronous Q-learning algorithm. For $k \ge 0$, given (J_τ, Q_τ), $\tau \le k$, the kth iteration of the algorithm is as follows:

- For each $i \in S$, let $\gamma_{i,k} \ge 0$ be a stepsize parameter, and let

$$J_{k+1}(i) = (1 - \gamma_{i,k}) J_k(i) + \gamma_{i,k} \min_{u \in U(i)} Q_{\tau^{i}_{iu,k}}(i,u). \tag{3.3}$$

- For each $(i,u) \in R$, let $\gamma_{iu,k} \ge 0$ be a stepsize parameter, let $j^{iu}_k \in S_o$ be the successor state of a random transition from state i using control u, generated according to the transition probabilities $\{p_{ij}(u) \mid j \in S_o\}$, and let v^{iu}_k be a random control generated according to $\nu^{iu}_k(\cdot \mid j^{iu}_k)$, where $\nu^{iu}_k \in \Pi_{\mathrm{SR}}$ and its choice can possibly depend on j^{iu}_k. Then, let

$$Q_{k+1}(i,u) = (1 - \gamma_{iu,k}) Q_k(i,u) + \gamma_{iu,k}\big(\hat{g}(i,u,s) + \min\big\{J_{\tau^{iu}_{s,k}}(s), Q_{\tau^{iu}_{sv,k}}(s,v)\big\}\big), \tag{3.4}$$

  where we use the shorthand notation $s = j^{iu}_k$ and $v = v^{iu}_k$, and $\hat{g}(i,u,s)$ is the cost of transition from state i to j^{iu}_k with control u.

The variables in iteration (3.3)–(3.4) need to satisfy several conditions, without which the algorithm as just described is, in fact, imprecise. We will specify these conditions shortly, after a few remarks.

In the above algorithm, the subset of cost/Q-factor components which are selected for an update at time k, is implicitly specified by the positive stepsize parameters: the value of $J_{k+1}(i)$ or $Q_{k+1}(i,u)$ remains unchanged when $\gamma_{i,k} = 0$ or $\gamma_{iu,k} = 0$. The random transition cost, $\hat{g}(i,u,j^{iu}_k)$, is treated as a function of the transition and control. This is for notational simplicity; the case where the transition cost depends on some additional random disturbance is covered by our subsequent analysis.

The algorithm (3.3)–(3.4) is stated in general terms, leaving open the choices of components to be updated and the randomized policies to be employed in each Q-factor update. In applications, the selection may be random, but it may also be deliberate to supplement other selection schemes that suit the task at hand. For example, in a real-time learning setting without the help of simulators, the components to be updated are naturally determined by the state that the learning agent is currently in and the control that is being applied; that control can in turn be chosen based on the randomized policies ν_k^{iu}. (See Bertsekas and Yu 2012, Sects. 4 and 5 for some examples in this vein.) The choice of the randomized policies can be based on factors concerning optimality and exploration needs. To make the algorithm resemble stochastic modified/optimistic PI, the cost updates (3.3) for selected states are done infrequently, relative to the Q-factor updates (3.4), and the policies ν_k^{iu} are chosen based on the deterministic policies μ_k, which are maintained and updated at selected states together with the cost updates (3.3) by

$$\mu_{k+1}(i) \in \arg\min_{u\in U(i)} Q_k(i,u), \quad \text{if } \gamma_{i,k} \neq 0; \qquad \mu_{k+1}(i) = \mu_k(i), \quad \text{otherwise.}$$

The randomized policies ν_k^{iu} can be a mixture of μ_k with some randomized policy for exploration, for instance. This is similar to the deterministic asynchronous algorithm (2.12)–(2.14) mentioned in Sect. 2.3.

With the notation for stepsizes and delayed times just described, a general form of the classical Q-learning algorithm is given by

$$Q_{k+1}(i,u) = (1-\gamma_{iu,k})Q_k(i,u) + \gamma_{iu,k}\Big(\hat g(i,u,s) + \min_{v\in U(s)} Q_{\tau_{sv,k}^{iu}}(s,v)\Big), \tag{3.5}$$

where as in iteration (3.4), s is a shorthand for the random successor state j_k^{iu} of state i with control u. Every iteration here involves minimization over the full control set. By contrast, the Q-factor update (3.4) in our algorithm is computationally simpler, as we discussed earlier.

We now state the conditions on the algorithm (3.3)–(3.4) and the convergence theorem.

3.1.1 Algorithmic conditions and convergence theorem

We regard all the variables in our Q-learning algorithm as random variables on a common probability space $(\Omega, \mathcal{F}, \mathbf{P})$. In addition to those variables appearing in the iteration (3.3)–(3.4), there can be auxiliary variables that the algorithm uses to determine, for instance, the values of delayed times or stepsizes, including which components to update at each time k. Thus, to summarize rigorously the dependence relation between the variables, it is convenient to introduce a family $\{\mathcal{F}_k\}$ of increasing sub-σ-fields of $\mathcal{F}$ and to require the following *information structure condition*: (J_0, Q_0) is $\mathcal{F}_0$-measurable, and

for every $\ell, \ell' \in S\cup R$ and $k \geq 0$, $\gamma_{\ell,k}$ and $\tau_{\ell',k}^{\ell}$ are $\mathcal{F}_k$-measurable,
and for every $(i,u) \in R$ and $k \geq 0$, j_k^{iu} and v_k^{iu} are $\mathcal{F}_{k+1}$-measurable.

This condition means in practice that in iteration (3.3)–(3.4), the algorithm either chooses the stepsizes $\gamma_{\ell,k}$ and the delayed times $\tau_{\ell',k}^{\ell}$ before generating the random successor state j_k^{iu} and control v_k^{iu}, or it chooses the values of the former variables in a way that does not use the information of j_k^{iu}, v_k^{iu}. We note that although this condition seems abstract, it can be satisfied naturally by the algorithm.

In probabilistic terms, the way the successor states/controls are generated and used in the algorithm is described more precisely by the following relations of the random variables: for all $(i,u) \in R$ and $k \geq 0$,

$$\mathbf{P}\big(j_k^{iu} = j \mid \mathcal{F}_k\big) = p_{ij}(u), \quad \forall j \in S_o, \tag{3.6}$$

$$\mathrm{E}\big[\hat g\big(i,u,j_k^{iu}\big) \mid \mathcal{F}_k\big] = g(i,u), \qquad \mathrm{E}\big[\big(\hat g\big(i,u,j_k^{iu}\big) - g(i,u)\big)^2 \mid \mathcal{F}_k\big] \leq C, \tag{3.7}$$

where C is some deterministic constant and the conditional expectation is over j_k^{iu}; and

$$\mathbf{P}\big(j_k^{iu} = j,\ v_k^{iu} = v \mid \mathcal{F}_k\big) = 0, \quad \forall v \notin U(j),\ j \in S_o, \tag{3.8}$$

i.e., v_k^{iu} is feasible control at the successor state j_k^{iu}.

There are other mild conditions on the algorithm. The totally asynchronous computation framework has the following minimal requirement on the delayed times: with probability 1 (w.p.1),

$$\lim_{k\to\infty} \tau^{i}_{jv,k} = \infty, \qquad \lim_{k\to\infty} \tau^{iu}_{jv,k} = \infty, \quad \forall i \in S,\ (i,u),(j,v) \in R. \tag{3.9}$$

This is referred to in the literature as "continuous information renewal," and it guarantees that outdated information about the updates will eventually be purged from the computation. Another condition is on the stepsize variables. Besides the requirement that w.p.1,

$$\gamma_{\ell,k} \in [0,1] \quad \text{eventually}, \quad \forall \ell \in S \cup R, \tag{3.10}$$

as usual in stochastic approximation algorithms, the standard stepsize condition is required for the Q-factor updates (3.4): w.p.1,

$$\sum_{k\geq 0} \gamma_{iu,k} = \infty, \qquad \sum_{k\geq 0} \gamma^2_{iu,k} < \infty, \quad \forall (i,u) \in R. \tag{3.11}$$

However, for the cost components involving "noiseless" updates (3.3), a weaker stepsize condition is sufficient: w.p.1,

$$\sum_{k\geq 0} \gamma_{i,k} = \infty, \quad \forall i \in S. \tag{3.12}$$

These conditions imply that every cost/Q-factor component is updated infinitely often.

Our subsequent analysis of the Q-learning algorithm (3.3)–(3.4) assumes all the algorithmic conditions given above. Let us collect them in one assumption:

Assumption 3.1 (Algorithmic conditions) The information structure condition holds, and w.p.1, Eqs. (3.6)–(3.12) are satisfied.

We will establish the following convergence theorem in the rest of this section. (A separate shorter proof for the special SSP models with all policies assumed proper will also be given.)

Theorem 3.1 *Under Assumptions 1.1, 3.1, for any given initial* (J_0, Q_0), *the sequence* $\{(J_k, Q_k)\}$ *generated by the iteration* (3.3)–(3.4) *converges to* (J^*, Q^*) *w.p.*1.

Note that the conditions of the theorem allow for a wide range of algorithmic parameters. So like the convergence analysis of deterministic algorithms given in Sect. 2.4, our subsequent analysis also has a worst-case character.

3.2 Preliminaries for convergence analysis

Before going into the main convergence proof, we discuss some preliminary facts. First, we show that to prove Theorem 3.1, it is sufficient to prove a weaker version of it, Proposition 3.1 below, which assumes in addition that the stepsizes are bounded by some constant, a condition that is technically more convenient. Once this weaker version of the theorem is proved, we can apply the result to the case of general stepsizes.

Proposition 3.1 *Suppose Assumptions 1.1, 3.1 hold and in addition, for some* (*deterministic*) *constant* D,

$$\gamma_{\ell,k} \leq D \quad w.p.1, \quad \forall \ell \in S \cup R,\ \forall k \geq 0. \tag{3.13}$$

Then, for any given initial (J_0, Q_0), *the sequence* $\{(J_k, Q_k)\}$ *generated by the iteration* (3.3)–(3.4) *converges to* (J^*, Q^*) *w.p.*1.

Indeed, suppose that Proposition 3.1 has been proved. Then, the additional condition (3.13) can be removed and the main convergence theorem, Theorem 3.1, will immediately follow as a consequence:

Proof of Theorem 3.1 For each positive integer m, let $\tilde{\gamma}^m_{\ell,k} = \min\{\gamma_{\ell,k}, m\}$ for all $\ell \in S \cup R$ and $k \geq 0$, and let $\{(\tilde{J}^m_k, \tilde{Q}^m_k)\}$ be given by the recursions (3.3)–(3.4) with $\tilde{\gamma}^m_{\ell,k}$ in place of $\gamma_{\ell,k}$. In particular, let $\tilde{J}^m_0 = J_0$, $\tilde{Q}^m_0 = Q_0$, and for $k \geq 0$, let

$$\tilde{J}^m_{k+1}(i) = \big(1 - \tilde{\gamma}^m_{i,k}\big)\tilde{J}^m_k(i) + \tilde{\gamma}^m_{i,k} \min_{u \in U(i)} \tilde{Q}^m_{\tau^i_{iu,k}}(i,u), \qquad \forall i \in S,$$

$$\tilde{Q}^m_{k+1}(i,u) = \big(1 - \tilde{\gamma}^m_{iu,k}\big)\tilde{Q}^m_k(i,u) + \tilde{\gamma}^m_{iu,k}\big(\hat{g}(i,u,s) + \min\big\{\tilde{J}^m_{\tau^{iu}_{s,k}}(s), \tilde{Q}^m_{\tau^{iu}_{sv,k}}(s,v)\big\}\big),$$
$$\forall (i,u) \in R,$$

where s, v are shorthand notation for j^{iu}_k, v^{iu}_k, respectively, and the variables j^{iu}_k, v^{iu}_k and $\tau^\ell_{\ell',k}, \ell, \ell' \in S \cup R$, are the same random variables that appear in the iteration (3.3)–(3.4). With the stepsizes $\{\tilde{\gamma}^m_{\ell,k}\}$ in place of $\{\gamma_{\ell,k}\}$, condition (3.13) is now satisfied with $D = m$, and Assumption 3.1 is also satisfied. So by Proposition 3.1, $\{(\tilde{J}^m_k, \tilde{Q}^m_k)\}$ converges to (J^*, Q^*) w.p.1. Now let Ω' be the set of sample paths on which $\{(\tilde{J}^m_k, \tilde{Q}^m_k)\}$ converges to (J^*, Q^*) for all $m \geq 1$ and condition (3.10) is satisfied by the stepsizes $\gamma_{\ell,k}$. This set Ω' has probability one. For each sample path in Ω', in view of condition (3.10), there exists some integer m such that $\gamma_{\ell,k} \leq m$ for all ℓ and k, and consequently, $\gamma_{\ell,k} = \tilde{\gamma}^m_{\ell,k}$ for all ℓ and k. This implies that on that sample path, $\{(J_k, Q_k)\}$ coincides with $\{(\tilde{J}^m_k, \tilde{Q}^m_k)\}$ and hence converges to (J^*, Q^*). Since Ω' has probability one, this shows that $\{(J_k, Q_k)\}$ converges to (J^*, Q^*) w.p.1. □

Henceforth, we will assume the conditions of Proposition 3.1 and prove that proposition. (Viewed in another way, what we will be doing is actually to prove for each $m \geq 1$, the convergence of the process $\{(\tilde{J}^m_k, \tilde{Q}^m_k)\}$ defined in the above proof of Theorem 3.1, but we will simply use the notation $\{(J_k, Q_k)\}$ for this process.)

To prepare for the convergence proof, we express the iteration (3.3)–(3.4) explicitly in terms of the mappings L^ν of Eq. (2.16), thereby casting it in a form amenable for stochastic approximation-based analysis. To this end, we identify the particular mapping L^ν or equivalently the randomized policy ν that is associated with each Q-factor update in the algorithm. For each $(i,u) \in R$, we define an $\mathcal{F}_k$-measurable Π_{SR}-valued random variable $\bar{\nu}^{iu}_k = \{\bar{\nu}^{iu}_k(v \mid j) \mid v \in U(j),\ j \in S\}$, which is the conditional distribution of v^{iu}_k corresponding to the joint distribution $\mathbf{P}(j^{iu}_k = j,\ v^{iu}_k = v \mid \mathcal{F}_k)$ of (j^{iu}_k, v^{iu}_k), i.e.,

$$\mathbf{P}\big(j^{iu}_k = j,\ v^{iu}_k = v \bigm| \mathcal{F}_k\big) = p_{ij}(u)\bar{\nu}^{iu}_k(v \mid j), \qquad \forall v \in U(j),\ j \in S_o; \tag{3.14}$$

cf. Eqs. (3.6) and (3.8). [If (i,u) and j are such that $p_{ij}(u) = 0$, we have $\mathbf{P}(j^{iu}_k = j,\ v^{iu}_k = v \mid \mathcal{F}_k) = 0$ for all $v \in U(j)$, and we may define $\bar{\nu}^{iu}_k(\cdot \mid j)$ to be any distribution over $U(j)$, for example, the uniform distribution.] If in the Q-factor update (3.4), the algorithm chooses the randomized policy ν^{iu}_k before it generates the successor state j^{iu}_k, then the randomized policy $\bar{\nu}^{iu}_k$ coincides with ν^{iu}_k. We associate $L^{\bar{\nu}^{iu}_k}$ with the Q-factor update (3.4) at (i,u) and iteration k.

To simplify notation, denote $x_k = (J_k, Q_k)$, and for each $\ell \in S \cup R$, let $x^{(\ell)}_k = (J^{(\ell)}_k, Q^{(\ell)}_k)$ where $J^{(\ell)}_k$ and $Q^{(\ell)}_k$ denote the vectors of costs and Q-factors respectively, with components

$$J^{(\ell)}_k(i) = J_{\tau^\ell_{i,k}}(i), \qquad Q^{(\ell)}_k(i,u) = Q_{\tau^\ell_{iu,k}}(i,u), \qquad i \in S_o,\ (i,u) \in R_o.$$

(Note that $J_k^{(\ell)}(0) = Q_k^{(\ell)}(0,0) = 0$.) In the terminology we used earlier, $x_k^{(\ell)}$ may be viewed as the information available to the processor ℓ at time k for updating the ℓth component. Denote by $x_k(\ell)$ the ℓth component of x_k, and by L_ℓ^ν the ℓth component mapping of L^ν.

Using the above definitions as well as the definition of L^ν, the iteration (3.3)–(3.4) can be equivalently and compactly written as

$$x_{k+1}(\ell) = (1 - \gamma_{\ell,k}) x_k(\ell) + \gamma_{\ell,k} L_\ell^{\bar{\nu}_k^\ell} x_k^{(\ell)} + \gamma_{\ell,k}\omega_{\ell,k}, \quad \ell \in S \cup R, \tag{3.15}$$

where, if $\ell = (i,u) \in R$ and $\gamma_{\ell,k} > 0$, $\bar{\nu}_k^\ell$ is the randomized policy $\bar{\nu}_k^{iu}$ defined above and $\omega_{\ell,k}$ is a noise term given by

$$\omega_{\ell,k} = \hat{g}\bigl(i,u,j_k^{iu}\bigr) + \min\bigl\{J_k^{(\ell)}\bigl(j_k^{iu}\bigr), Q_k^{(\ell)}\bigl(j_k^{iu}, v_k^{iu}\bigr)\bigr\} - \bigl(F_{J_k^{(\ell)},\bar{\nu}_k^\ell} Q_k^{(\ell)}\bigr)(i,u); \tag{3.16}$$

and if $\ell = i \in S$, then $\omega_{\ell,k} = 0$ and $\bar{\nu}_k^\ell$ is immaterial (we can let it be a fixed $\nu \in \Pi_{\mathrm{SR}}$, for instance). It can be seen that because $\bar{\nu}_k^{iu}$ is $\mathcal{F}_k$-measurable, $L_\ell^{\bar{\nu}_k^\ell} x_k^{(\ell)}$ is $\mathcal{F}_k$-measurable and $\omega_{\ell,k}$ is $\mathcal{F}_{k+1}$-measurable, for every k and ℓ. This fact is important in the analysis.

Using the equivalent expression (3.15), we can analyze the convergence of our algorithm (3.3)–(3.4) in the general framework given in Tsitsiklis (1994) for asynchronous stochastic approximation algorithms involving sup-norm contractions or monotone nonexpansive mappings. The analysis of Tsitsiklis (1994) concerns a single fixed point mapping, but with proper modifications, it can be applied in our context where a set of mappings, $\{L^\nu \mid \nu \in \Pi_{\mathrm{SR}}\}$, is involved. We give a detailed account of the necessary modifications in Appendix C (a reproduction of the proofs of Tsitsiklis 1994 with modifications is also available Yu 1994). Here we explain why the analysis of Tsitsiklis (1994) is applicable in our case, in order to set the groundwork for our convergence proofs.

The technical conditions in Tsitsiklis (1994) may be separated into two groups. One group (Assumptions 1–3 in Tsitsiklis 1994) consists of algorithmic conditions: conditions on the information structure, on the delayed times, on the stepsizes, and on the variance of the noise terms $\omega_{\ell,k}$ conditional on $\mathcal{F}_k$. Our algorithm (3.3)–(3.4) satisfies these conditions in Tsitsiklis (1994) or some slightly different versions of them which do not affect the validity of the analysis of Tsitsiklis (1994). In particular, for our algorithm, the condition on the delayed times [cf. Eq. (3.9)] matches that in Tsitsiklis (1994) (Assumption 1); the information structure specified in Sect. 3 [together with the definition of $\omega_{\ell,k}$ in Eq. (3.15)] implies the corresponding condition in Tsitsiklis (1994) (Assumption 2(a)–(c)). The stepsize conditions are slightly different from those in Tsitsiklis (1994) (Assumption 3), but do not prevent the analysis of Tsitsiklis (1994) from going through; detailed explanations are given in Appendix C. Finally, regarding the noise terms $\omega_{\ell,k}$ in Eq. (3.15), besides their being $\mathcal{F}_{k+1}$-measurable by definition, it can be verified by a direct calculation (see Appendix B) that they satisfy the conditions required in Tsitsiklis (1994) (Assumption 2(d)–(e)), namely, that for every $\ell \in S \cup R$ and $k \geq 0$,

$$\mathrm{E}[\omega_{\ell,k} \mid \mathcal{F}_k] = 0, \quad \text{w.p.1},$$

and that there exist (deterministic) constants A and B such that for every $\ell \in S \cup R$ and $k \geq 0$,

$$\mathrm{E}\bigl[\omega_{\ell,k}^2 \bigm| \mathcal{F}_k\bigr] \leq A + B \max_{\ell' \in S \cup R} \max_{\tau \leq k} \bigl|x_\tau\bigl(\ell'\bigr)\bigr|^2, \quad \text{w.p.1}.$$

We mention that it is here that the additional condition (3.13) on stepsizes is used.

The other group of conditions in Tsitsiklis (1994) (Assumptions 4–6) consists of conditions on the underlying fixed point mapping, such as contraction or nonexpansiveness, monotonicity and the existence of a unique fixed point. Since our algorithm involves a set of mappings instead of a single one, these conditions will be replaced by conditions on the associated set of mappings, which are the properties of L^{ν} stated by Proposition 2.2 or Proposition 2.3 for general or special SSP models, respectively. Correspondingly, some modifications of its arguments are needed, in order to apply the analysis of Tsitsiklis (1994) to our case; but the changes required are nonessential and will be described in Appendix C.

We are now ready to proceed to convergence proofs, focusing on the main arguments.

3.3 Convergence proofs

In this subsection we prove Proposition 3.1 on the convergence of our Q-learning algorithm (3.3)–(3.4). We will give separate proofs for two SSP model classes: first for the special class of SSP where all policies in Π_{SD} are proper, and then for the general class of SSP that satisfy Assumption 1.1. The analyses for the two are based on different arguments. While the special SSP model is covered by the general one, it allows for a simpler contraction-based convergence proof, similar to the one for discounted problems given in Bertsekas and Yu (2012, Sect. 4.3). For the general class of SSP, the analysis is based on the monotonicity and nonexpansiveness of the mappings L^{ν}, $\nu \in \Pi_{\mathrm{SR}}$ with respect to the sup-norm (cf. Proposition 2.2). We will use a convergence result of Tsitsiklis (1994) on asynchronous stochastic approximation methods involving such mappings, together with the results of Yu and Bertsekas (2011) on the boundedness of the classical Q-learning iterates for SSP under Assumption 1.1.

3.3.1 SSP models with all policies being proper

When all policies in Π_{SD} are proper, by Proposition 2.3, the mappings $\{L^{\nu} \mid \nu \in \Pi_{\mathrm{SR}}\}$ are contraction mappings with modulus $\bar{\beta} < 1$, with respect to a common weighted sup-norm, and with the same fixed point (J^*, Q^*). Then, by combining Proposition 2.3 with (Tsitsiklis 1994, Theorems 1 and 3) (and its appropriate modifications described in Appendix C), the convergence of $\{(J_k, Q_k)\}$ to (J^*, Q^*) follows.

Proposition 3.2 *Suppose all policies in Π_{SD} are proper. Then, under Assumption 3.1 and condition (3.13), for any given initial (J_0, Q_0), the sequence $\{(J_k, Q_k)\}$ generated by the iteration (3.3)–(3.4) converges to (J^*, Q^*) w.p.1.*

Note that condition (3.13) can be removed in the same way as we did in the proof of Theorem 3.1, without affecting the conclusion of the preceding proposition.

3.3.2 General SSP models—proof of Proposition 3.1

In this subsection, we prove Proposition 3.1. For SSP models satisfying Assumption 1.1, there may exist an improper policy in Π_{SD}, and the uniform sup-norm contraction mapping argument used in the preceding convergence proof for special SSP models does not apply. However, by Proposition 2.2, the set of mappings $\{L^{\nu} \mid \nu \in \Pi_{\mathrm{SR}}\}$, as well as the mappings $\overline{L}$ and $\underline{L}$ associated with this set, are monotone and nonexpansive with respect to the sup-norm, and have (J^*, Q^*) as the unique fixed point. Combining Proposition 2.2 with the proof arguments of Theorem 2 in Tsitsiklis (1994) for nonexpansive mappings, and taking into account also the modifications to the latter proof described in Appendix C, we have the following lemma.

Lemma 3.1 *Suppose the conditions of Proposition 3.1 hold. Then, for any given initial* (J_0, Q_0), *the sequence* $\{(J_k, Q_k)\}$ *generated by the iteration* (3.3)–(3.4) *converges to* (J^*, Q^*) *w.p.*1 *if* $\{(J_k, Q_k)\}$ *is bounded w.p.*1.

In the rest of this section, we prove the boundedness of $\{(J_k, Q_k)\}$ required as a condition of convergence in Lemma 3.1, thereby establishing Proposition 3.1. We will show first that $\{(J_k, Q_k)\}$ is bounded above w.p.1, and then that it is bounded below w.p.1. The proof for the former will use a contraction property associated with a proper policy, and the proof for the latter will use a lower-boundedness result from Yu and Bertsekas (2011) on the iterates of the classical Q-learning algorithm.

In both boundedness proofs, we will start with a certain process $\{(\bar{J}_k, \bar{Q}_k)\}$, which is easier to work with. It is defined on the same probability space as $\{(J_k, Q_k)\}$ and defined by the recursions (3.3)–(3.4), which define $\{(J_k, Q_k)\}$, except that it uses different stepsizes and different initial value $\bar{J}_0$. More specifically, $\{(\bar{J}_k, \bar{Q}_k)\}$ is defined as follows. (By our convention $\bar{J}_k(0) = \bar{Q}_k(0, 0) = 0$ for all k.)

Let $\bar{Q}_0 = Q_0$ and

$$\bar{J}_0(i) = \min_{u \in U(i)} \bar{Q}_0(i, u), \quad \forall i \in S. \tag{3.17}$$

For $k \ge 0$, let $(\bar{J}_{k+1}, \bar{Q}_{k+1})$ be given by: for each $i \in S$,

$$\bar{J}_{k+1}(i) = (1 - \alpha_{i,k})\bar{J}_k(i) + \alpha_{i,k} \min_{u \in U(i)} \bar{Q}_{\tau^i_{iu,k}}(i, u), \tag{3.18}$$

and for each $(i, u) \in R$,

$$\bar{Q}_{k+1}(i, u) = (1 - \alpha_{iu,k})\bar{Q}_k(i, u) + \alpha_{iu,k}\big(\hat{g}(i, u, s) + \min\big\{\bar{J}_{\tau^{iu}_{s,k}}(s), \bar{Q}_{\tau^{iu}_{sv,k}}(s, v)\big\}\big), \tag{3.19}$$

where the stepsize $\alpha_{\ell,k}$, for $\ell = i$ or (i, u), in Eq. (3.18) or (3.19), is given by

$$\alpha_{\ell,k} = \begin{cases} \gamma_{\ell,k}, & \text{if } \gamma_{\ell,k} \le 1, \\ 0, & \text{if } \gamma_{\ell,k} > 1, \end{cases}$$

with $\gamma_{\ell,k}$ being the same stepsize variables that appear in the iteration (3.3)–(3.4). In the above, s and v in Eq. (3.19) are shorthand for the successor state j^{iu}_k and control v^{iu}_k, respectively; and the variables j^{iu}_k, v^{iu}_k, and the delayed times $\tau^{\ell}_{\ell',k}$, $\ell, \ell' \in S \cup R$, are also the same random variables that appear in the iteration (3.3)–(3.4).

Lemma 3.2 *Under the stepsize condition* (3.10), *w.p.*1, $\{(J_k, Q_k)\}$ *is bounded if and only if* $\{(\bar{J}_k, \bar{Q}_k)\}$ *is bounded.*

Proof Consider any sample path from a set of probability 1 on which the stepsize condition (3.10) holds. In view of the latter condition, there exists some time $\bar{k} < \infty$ such that the stepsizes $\gamma_{\ell,k} \in [0, 1]$ for all $k \ge \bar{k}$ and $\ell \in S \cup R$. This implies that after time $\bar{k}$, the stepsizes of $\{(\bar{J}_k, \bar{Q}_k)\}$ and $\{(J_k, Q_k)\}$ coincide: for every $\ell \in S \cup R$, $\gamma_{\ell,k} = \alpha_{\ell,k}$ for all $k \ge \bar{k}$.

Let $\Delta = \max_{\tau \le \bar{k}} \|(J_\tau, Q_\tau) - (\bar{J}_\tau, \bar{Q}_\tau)\|_\infty$. By the definition of Δ, we have for all $k \le \bar{k}$,

$$(\bar{J}_k, \bar{Q}_k) - \Delta e \le (J_k, Q_k) \le (\bar{J}_k, \bar{Q}_k) + \Delta e, \tag{3.20}$$

where $e \in \Re^{|S \cup R|}$ denotes the vector with all components equal to one. We prove by induction that the relation (3.20) holds also for all $k \ge \bar{k}$. Suppose that for some $k \ge \bar{k}$, it holds for all τ with $0 \le \tau \le k$. Then, since $\alpha_{\ell,k} = \gamma_{\ell,k} \in [0, 1]$ for all $\ell \in S \cup R$, using the definition of (J_{k+1}, Q_{k+1}), $(\bar{J}_{k+1}, \bar{Q}_{k+1})$, and the induction hypothesis, we obtain that for all $i \in S$,

$$\begin{aligned} J_{k+1}(i) &\le (1-\gamma_{i,k})\big(\bar{J}_k(i)+\Delta\big)+\gamma_{i,k}\min_{u\in U(i)}\big(\bar{Q}_{\tau^i_{iu,k}}(i,u)+\Delta\big),\\ &= \bar{J}_{k+1}(i)+\Delta, \end{aligned}$$

and for all $(i,u)\in R$,

$$\begin{aligned} Q_{k+1}(i,u) &\le (1-\gamma_{iu,k})\big(\bar{Q}_k(i,u)+\Delta\big)\\ &\quad +\gamma_{iu,k}\Big(\hat{g}(i,u,s)+\min\Big\{\bar{J}_{\tau^{iu}_{s,k}}(s)+\Delta,\ \bar{Q}_{\tau^{iu}_{sv,k}}(s,v)+\Delta\Big\}\Big)\\ &= \bar{Q}_{k+1}(i,u)+\Delta, \end{aligned}$$

where s and v are shorthand j^{iu}_k, v^{iu}_k, respectively. This proves $(J_{k+1},Q_{k+1})\le(\bar{J}_{k+1},\bar{Q}_{k+1})+\Delta e$. A symmetric argument yields $(J_{k+1},Q_{k+1})\ge(\bar{J}_{k+1},\bar{Q}_{k+1})-\Delta e$, and hence, by induction, Eq. (3.20) holds for all $k\ge\bar{k}$. This implies that $\{(J_k,Q_k)\}$ is bounded if and only if $\{(\bar{J}_k,\bar{Q}_k)\}$ is bounded. □

Two properties of the process $\{(\bar{J}_k,\bar{Q}_k)\}$ that we will exploit in the boundedness proofs below are the form of its initial $\bar{J}_0$ given by Eq. (3.17), and the fact that its stepsizes satisfy

$$\alpha_{\ell,k}\in[0,1],\quad \forall\ell\in S\cup R,\ \forall k\ge 0. \tag{3.21}$$

Note also that in view of their relation with $\gamma_{\ell,k}$, the sequences $\{\alpha_{\ell,k},k\ge 0\}$, $\ell\in S\cup R$, satisfy the stepsize conditions, Eqs. (3.10)–(3.13).

In what follows, we refer to a sequence of iterates generated by the classical Q-learning algorithm (3.5) as a sequence of ordinary Q-learning iterates. The convergence of classical Q-learning under the SSP model assumption, Assumption 1.1, is analyzed by Tsitsiklis (1994), Yu and Bertsekas (2011); the required algorithmic conditions are the same as stated in Assumption 3.1 for the respective variables. In our proofs, we will exploit the use of the delayed times and purposely choose them to define certain ordinary Q-learning iterates, which we will relate to the processes we want to bound.

Proposition 3.3 *Under Assumptions* 1.1(i), 3.1 *and condition* (3.13), *for any given initial* (J_0,Q_0), *the sequence* $\{(J_k,Q_k)\}$ *generated by the iteration* (3.3)–(3.4) *is bounded above w.p.*1.

Proof By Lemma 3.2, it is sufficient to show that the sequence $\{(\bar{J}_k,\bar{Q}_k)\}$ is bounded above w.p.1. In turn, it is sufficient to show that $\{\bar{Q}_k\}$ is bounded above w.p.1 [since this clearly implies that the same holds for $\{\bar{J}_k\}$, in view of Eq. (3.21)]. To this end, we introduce another process $\{\hat{Q}_k\}$ with the property that $\hat{Q}_k\ge\bar{Q}_k$ for all $k\ge 0$, and we show that $\{\hat{Q}_k\}$ is bounded above w.p.1.

First, for each $(\bar{i},\bar{u})$ and $(i,u)\in R$, define a new sequence of delayed times $\{\hat{\tau}^{iu}_{\bar{i}\bar{u},k},k\ge 0\}$ such that

$$\hat{\tau}^{iu}_{\bar{i}\bar{u},k}\in\arg\max_{\tau\le k}\bar{Q}_\tau(\bar{i},\bar{u}). \tag{3.22}$$

Note that $\hat{\tau}^{iu}_{\bar{i}\bar{u},k}$ is $\mathcal{F}_k$-measurable.

We now define $\{\hat{Q}_k\}$. (By our convention $\hat{Q}_k(0,0)=0$ for all $k\ge 0$.) Let $\hat{Q}_0=\bar{Q}_0$. Let μ be any proper policy in Π_{SD}, which exists by Assumption 1.1(i), and let $\mu(i)$ denote the control applied by the deterministic policy μ at state i. For each $(i,u)\in R$ and $k\ge 0$, define

$$\hat{Q}_{k+1}(i,u)=(1-\alpha_{iu,k})\hat{Q}_k(i,u)+\alpha_{iu,k}\big(\hat{g}(i,u,s)+\hat{Q}_{\hat{\tau}^{iu}_{s\mu(s),k}}\big(s,\mu(s)\big)\big), \tag{3.23}$$

where s is a shorthand for j_k^{iu}, and $j_k^{iu}, \alpha_{iu,k}$ are the same random variables that appear in Eq. (3.19), which defines $\bar{Q}_{k+1}$.

We show by induction that $\hat{Q}_k \geq \bar{Q}_k$ for all $k \geq 0$. This holds for $k = 0$ by the definition of $\hat{Q}_0$. Assuming it holds for all $k' \leq k$, we prove that it holds for $k + 1$. Consider any $\bar{i} \in S$ and $\bar{k} \leq k$. Denote $[\bar{k} - 1]^+ = \max\{0, \bar{k} - 1\}$. By the definition of $\bar{J}_0, \dots, \bar{J}_k$ [cf. Eqs. (3.17) and (3.18)] and the fact that all stepsizes are in $[0, 1]$ [cf. Eq. (3.21)], it can be seen that $\bar{J}_{\bar{k}}(\bar{i})$ is a convex combination of the terms

$$\min_{\bar{v} \in U(\bar{i})} \bar{Q}_{\tau^{\bar{i}}_{\bar{i}\bar{v},k'}}(\bar{i}, \bar{v}), \quad k' \leq [\bar{k} - 1]^+,$$

and is therefore no greater than the maximal of these terms. Hence, for any $\bar{u} \in U(\bar{i})$ and $(i, u) \in R$,

$$\begin{aligned}\bar{J}_{\bar{k}}(\bar{i}) &\leq \max_{k' \leq [\bar{k}-1]^+} \min_{\bar{v} \in U(\bar{i})} \bar{Q}_{\tau^{\bar{i}}_{\bar{i}\bar{v},k'}}(\bar{i}, \bar{v}) \leq \max_{k' \leq [\bar{k}-1]^+} \bar{Q}_{\tau^{\bar{i}}_{\bar{i}\bar{u},k'}}(\bar{i}, \bar{u}) \\ &\leq \max_{k' \leq k} \bar{Q}_{k'}(\bar{i}, \bar{u}) = \bar{Q}_{\hat{\tau}^{iu}_{\bar{i}\bar{u},k}}(\bar{i}, \bar{u}),\end{aligned}$$

where the last equality follows from the definition of $\hat{\tau}^{iu}_{\bar{i}\bar{u},k}$ given by Eq. (3.22). The preceding inequality implies by the induction hypothesis $\bar{Q}_{k'} \leq \hat{Q}_{k'}$ for all $k' \leq k$ that

$$\bar{J}_{\bar{k}}(\bar{i}) \leq \hat{Q}_{\hat{\tau}^{iu}_{\bar{i}\bar{u},k}}(\bar{i}, \bar{u}), \quad \forall \bar{k} \leq k,\ (\bar{i}, \bar{u}), (i, u) \in R.$$

This in turn implies the following relation for the term appearing in the definition of $\bar{Q}_{k+1}(i, u)$ [cf. Eq. (3.19)]:

$$\min\bigl\{\bar{J}_{\tau^{iu}_{s,k}}(s), \bar{Q}_{\tau^{iu}_{sv,k}}(s, v)\bigr\} \leq \bar{J}_{\tau^{iu}_{s,k}}(s) \leq \hat{Q}_{\hat{\tau}^{iu}_{s\mu(s),k}}\bigl(s, \mu(s)\bigr), \tag{3.24}$$

where s and v are shorthand notation for j_k^{iu} and v_k^{iu}, respectively. Comparing the definitions of $\bar{Q}_{k+1}$ and $\hat{Q}_{k+1}$ [cf. Eqs. (3.19), (3.23)], using inequality (3.24) and the induction hypothesis $\bar{Q}_k \leq \hat{Q}_k$, and using also the fact $\alpha_{iu,k} \in [0, 1]$, we then obtain $\bar{Q}_{k+1} \leq \hat{Q}_{k+1}$. By induction, this establishes $\hat{Q}_k \geq \bar{Q}_k$ for all $k \geq 0$.

Now $\{\hat{Q}_k\}$ is a sequence of ordinary Q-learning iterates for the case of a SSP with a single proper policy μ and involving the mapping F_μ, which is a weighted sup-norm contraction [cf. Eq. (2.6) and the discussion immediately following it]. Such sequences are analyzed in Tsitsiklis (1994), the results of which can be applied here. In particular, under Assumption 3.1, in view of the fact that the stepsizes $\alpha_{\ell,k}$ satisfy conditions (3.10)–(3.13), $\{\hat{Q}_k\}$ satisfies all but one algorithmic conditions required in the analysis of Tsitsiklis (1994) (with its extension regarding the stepsize condition; see Appendix C). The condition $\{\hat{Q}_k\}$ may violate is the one on the delayed times: for each (i, u) and $(\bar{i}, \bar{u}) \in R$, the sequence $\{\hat{\tau}^{iu}_{\bar{i}\bar{u},k}, k \geq 0\}$ should converge to infinity w.p.1. Indeed, it is possible that $\{\hat{\tau}^{iu}_{\bar{i}\bar{u},k}, k \geq 0\}$ as defined by Eq. (3.22) is bounded. However, the unboundedness condition on the delayed times is not needed for establishing the boundedness of the iterates, as the proof of Theorem 1 of Tsitsiklis (1994) shows (although this condition is needed in proving convergence). Therefore, we can apply the latter theorem to obtain that $\{\hat{Q}_k\}$ is bounded w.p.1. Since $\hat{Q}_k \geq \bar{Q}_k$, this implies that $\{\bar{Q}_k\}$ is bounded above w.p.1. The proof is complete. □

We now show that $\{(J_k, Q_k)\}$ is bounded below w.p.1. In the proof, we construct a sequence of ordinary Q-learning iterates that lies below $\{\bar{Q}_k\}$ (cf. Lemma 3.2), and we then use the result (Yu and Bertsekas 2011, Proposition 3.3) on the boundedness of the ordinary Q-learning iterates for SSP problems to complete the proof.

Proposition 3.4 *Under the conditions of Proposition* 3.1, *for any given initial* (J_0, Q_0), *the sequence* $\{(J_k, Q_k)\}$ *generated by the iteration* (3.3)–(3.4) *is bounded below w.p.*1.

Proof Similar to the proof of Proposition 3.3, by Lemma 3.2, it is sufficient to show that the sequence $\{(\bar J_k, \bar Q_k)\}$ is bounded below w.p.1. In turn, it is sufficient to show that $\{\bar Q_k\}$ is bounded below w.p.1 [since this clearly implies that the same holds for $\{\bar J_k\}$, in view of Eq. (3.21)]. To this end, we introduce another process $\{\hat Q_k\}$ with the property $\hat Q_k \le \bar Q_k$ for all $k \ge 0$, and we will show that $\{\hat Q_k\}$ is bounded below w.p.1.

First, for each $(\bar i, \bar u)$ and $(i, u) \in R$, define a new sequence of delayed times $\{\hat\tau^{iu}_{\bar i\bar u,k},\ k \ge 0\}$ such that

$$\hat\tau^{iu}_{\bar i\bar u,k} \in \arg\min_{\tau \le k} \bar Q_\tau(\bar i, \bar u). \tag{3.25}$$

Note that $\hat\tau^{iu}_{\bar i\bar u,k}$ is $\mathcal{F}_k$-measurable. We then define $\{\hat Q_k\}$ as follows. [By our convention $\hat Q_k(0,0) = 0$ for all $k \ge 0$.] Let $\hat Q_0 = \bar Q_0$. For each $(i, u) \in R$ and $k \ge 0$, define

$$\hat Q_{k+1}(i,u) = (1-\alpha_{iu,k})\hat Q_k(i,u) + \alpha_{iu,k}\Big(\hat g(i,u,s) + \min_{\bar v \in U(s)} \hat Q_{\hat\tau^{iu}_{s\bar v,k}}(s,\bar v)\Big), \tag{3.26}$$

where s is a shorthand for j^{iu}_k; and $\alpha_{iu,k}$ and j^{iu}_k are the same random variables that appear in Eq. (3.19), which defines $\bar Q_{k+1}$.

We show by induction that $\hat Q_k \le \bar Q_k$ for all $k \ge 0$. This holds for $k = 0$ by the definition of $\hat Q_0$. Assuming it holds for all $k' \le k$, we prove that it holds for $k+1$. Consider any $\bar i \in S$ and $\bar k \le k$. Denote $[\bar k - 1]^+ = \max\{0, \bar k - 1\}$. As argued in the proof of Proposition 3.3, $\bar J_{\bar k}(\bar i)$ is a convex combination of the terms

$$\min_{\bar v \in U(\bar i)} \bar Q_{\tau^{\bar i}_{\bar i\bar v,k'}}(\bar i, \bar v), \qquad k' \le [\bar k - 1]^+,$$

so it is no less than the minimal of these terms. Therefore, for any $\bar i \in S$ and $\bar k \le k$,

$$\bar J_{\bar k}(\bar i) \ge \min_{k' \le [\bar k - 1]^+} \min_{\bar v \in U(\bar i)} \bar Q_{\tau^{\bar i}_{\bar i\bar v,k'}}(\bar i, \bar v) \ge \min_{\bar v \in U(\bar i)} \min_{k' \le k} \bar Q_{k'}(\bar i, \bar v). \tag{3.27}$$

We now compare the definitions of $\bar Q_{k+1}$ and $\hat Q_{k+1}$ [cf. Eqs. (3.19), (3.26)]. Using inequality (3.27) and the induction hypothesis $\bar Q_{k'} \ge \hat Q_{k'}$ for all $k' \le k$, we have that for each $(i, u) \in R$, the following relation holds for the term appearing in the definition of $\bar Q_{k+1}(i,u)$ [cf. Eq. (3.19)]:

$$\begin{aligned}\min\big\{\bar J_{\tau^{iu}_{s,k}}(s), \bar Q_{\tau^{iu}_{sv,k}}(s,v)\big\} &\ge \min_{\bar v \in U(s)} \min_{k' \le k} \bar Q_{k'}(s, \bar v) = \min_{\bar v \in U(s)} \bar Q_{\hat\tau^{iu}_{s\bar v,k}}(s, \bar v)\\ &\ge \min_{\bar v \in U(s)} \hat Q_{\hat\tau^{iu}_{s\bar v,k}}(s, \bar v),\end{aligned}$$

where s and v are shorthand notation for j^{iu}_k and v^{iu}_k, respectively, and the equality relation follows from the definition of $\hat\tau^{iu}_{s\bar v,k}$ [cf. Eq. (3.25)]. Combining the preceding inequality with the definition of $\bar Q_{k+1}$, using the induction hypothesis $\bar Q_k \ge \hat Q_k$ and the fact $\alpha_{iu,k} \in [0,1]$, it follows that $\bar Q_{k+1} \ge \hat Q_{k+1}$. Hence, by induction, $\hat Q_k \le \bar Q_k$ for all $k \ge 0$.

Now $\{\hat Q_k\}$ is a sequence of the ordinary Q-learning iterates for the SSP problem, whose model satisfies Assumption 1.1. Similar to the proof of Proposition 3.3, under Assumption 3.1, the sequence $\{\hat Q_k\}$ meets all the conditions required in the convergence analysis of Q-learning iterates (Tsitsiklis 1994; Yu and Bertsekas 2011), except that for each $(\bar i, \bar u)$, $(i, u) \in R$, the sequence of delayed times $\{\hat\tau^{iu}_{\bar i\bar u,k},\ k \ge 0\}$ defined by Eq. (3.25) need not converge to infinity w.p.1. However, the unboundedness of the delayed times, while being necessary for the convergence of the iterates, is not needed for the boundedness of the iterates, an analysis of which under Assumption 1.1 is given by Yu and Bertsekas (2011). Applying Yu and Bertsekas (2011, Proposition 3.3), we obtain that $\{\hat{Q}_k\}$ is bounded below w.p.1. Since $\hat{Q}_k \leq \bar{Q}_k$, this implies that $\{\bar{Q}_k\}$ is bounded below w.p.1. The proof is complete. □

Proposition 3.1 is now implied by Lemma 3.1 and Propositions 3.3 and 3.4.

4 Function approximation and error bound

To apply VI and PI in large-scale MDP problems, one approach is to combine these methods with cost function approximation (see e.g., Bertsekas and Tsitsiklis 1996; Sutton and Barto 1998). For our method, a direct way to incorporate function approximation is based on the prototype algorithm (2.8)–(2.9) of Sect. 2.2. Recall that the main idea of this algorithm is to solve a certain optimal stopping problem at the policy evaluation phase and then update the stopping costs, to be used in defining the next optimal stopping problem, at the policy improvement/cost minimization phase. By carrying out the computation of these two steps approximately using function approximation, we obtain approximate versions of our method.

We may describe abstractly such an approximation algorithm as follows: at iteration k, given (J_k, Q_k), for a chosen policy $\nu_k \in \Pi_{\mathrm{SR}}$ and integer $m_k \geq 1$, find (J_{k+1}, Q_{k+1}) such that

$$Q_{k+1} \approx F^{m_k}_{J_k,\nu_k} Q_k, \qquad J_{k+1} \approx M Q_{k+1}. \tag{4.1}$$

From a slightly different viewpoint, we may also describe it as follows: for given (J_k, ν_k), find (J_{k+1}, Q_{k+1}) such that

$$Q_{k+1} \approx F_{J_k,\nu_k} Q_{k+1}, \qquad J_{k+1} \approx M Q_{k+1}, \tag{4.2}$$

i.e., Q_{k+1} is an approximation of the unique solution Q_{J_k,ν_k} (cf. Proposition 2.1(i)) to the Bellman equation $Q = F_{J_k,\nu_k} Q$ for the optimal stopping problem defined by (J_k, ν_k). Similar to the prototype algorithm (2.8)–(2.9), if we choose the randomized policies ν_k based on the policies that attain, within some tolerance, the minima in MQ_k, and if m_k is relatively large in (4.1), then the above algorithms resemble approximate PI.

In what follows, we first describe an implementation of the abstract algorithm (4.2) with simulation, in which the optimal stopping problem at each iteration is approximately solved by using a special Q-learning algorithm for optimal stopping problems, proposed by Tsitsiklis and Van Roy (1999). We then derive an error bound for approximation algorithms of the above forms, under the assumption that all policies in the SSP are proper policies. The discussion in this section is similar to the one of our earlier work (Bertsekas and Yu 2012, Sects. 6 and 7) for discounted MDP; however, there are some subtle technical differences in SSP problems, which we will address.

4.1 An implementation with simulation and function approximation

We consider implementing the abstract algorithm (4.2) with function approximation and with the Q-learning algorithm of Tsitsiklis and Van Roy (1999) as a subroutine to find approximate solutions of the optimal stopping problems

$$Q_{k+1} \approx F_{J_k,\nu_k} Q_{k+1}. \tag{4.3}$$

We approximate Q-factors using a linear approximation architecture and restrict $\{Q_k\}$ in a chosen low-dimensional subspace $\mathcal{H} \subset \Re^{|R|}$, given by

$$\mathcal{H} = \left\{ Q \;\middle|\; Q(i,u) = \phi(i,u)'r,\ \forall (i,u) \in R;\ r \in \Re^d \right\},$$

where $\phi(i,u)$ is a column vector of d "features" of the state-control pair (i,u), r is a column vector of d weights, and $'$ denotes transpose. To approximate the costs, we can use a nonlinear approximation architecture, restricting $\{J_k\}$ in some parametric family of functions. The approximation problem of finding $J_{k+1} \approx MQ_{k+1}$, where Q_{k+1} is given, can be solved by optimization, which we will not go into. We focus on the approximation problems (4.3) of the policy evaluation phases.

To simplify notation, consider for given J and $\nu \in \Pi_{\mathrm{SR}}$, the problem of approximately solving the Bellman equation

$$Q = F_{J,\nu} Q$$

of the optimal stopping problem associated with (J,ν). The approximation approach of Tsitsiklis and Van Roy (1999) is to find the solution of a projected version of the Bellman equation,

$$Q = \Pi F_{J,\nu} Q, \tag{4.4}$$

where Π is the projection on $\mathcal{H}$ with respect to some weighted Euclidean norm $\|\cdot\|_w$. Although the Bellman equation has a unique solution due to the property of $F_{J,\nu}$ [see Proposition 2.1(i)] under our SSP model Assumption 1.1, the projected equation (4.4) need not have a solution. To ensure that it has a unique solution and that the Q-learning algorithm of Tsitsiklis and Van Roy (1999) will converge in our context, besides choosing the projection norm in a specific way as will be described below, we require that the policy ν is a proper policy. This means that the policies $\{\nu_k\}$ in our approximation algorithm are required to be proper policies, which can be satisfied easily by letting $\nu_k(u \mid i) > 0$ for all $(i,u) \in R$. Nevertheless, this condition is somewhat restrictive and one may be able to weaken it with further research.

We now describe the details of implementing the algorithm of Tsitsiklis and Van Roy (1999) to solve Eq. (4.4) in our context. This algorithm is a stochastic iterative algorithm, and it uses a trajectory of states of the unstopped process of the optimal stopping problem. In our case, the trajectory is a sequence of state-control pairs, $\{(i_t,u_t) \mid t = 0,\dots,T\}$, generated under the policy ν and with regeneration at state 0, where the length T is chosen sufficiently large for the accuracy requirements of the implementation. More specifically, the transition from (i_t,u_t) to (i_{t+1},u_{t+1}) occurs with probability

$$\begin{cases} p_{i_t i_{t+1}}(u_t)\nu(u_{t+1} \mid i_{t+1}), & \text{if } i_t \neq 0, \\ \sigma(i_{t+1},u_{t+1}), & \text{if } i_t = 0, \end{cases}$$

where σ can be any given probability distribution over the state-control space R such that $\sigma(i,u) > 0$ for all $(i,u) \in R$. The Markov chain $\{(i_t,u_t)\}$ regenerates according to σ whenever it reaches the state $(0,0)$. Using a trajectory of state-control pairs generated as above and using positive diminishing stepsizes $\{\gamma_t\}$, a sequence of weight vectors $\{\bar{r}_t\}$ is calculated iteratively by

$$\bar{r}_{t+1} = \bar{r}_t + \gamma_t \phi(i_t,u_t)\big(\hat{g}(i_t,u_t,i_{t+1}) + \min\big\{J(i_{t+1}), \phi(i_{t+1},u_{t+1})'\bar{r}_t\big\} - \phi(i_t,u_t)'\bar{r}_t\big).$$

In the above, at state 0, $\phi(0,0) = 0$ and $J(0) = 0$ by definition, so in particular, when $i_t = 0$ and the Markov chain regenerates, we have $\bar{r}_{t+1} = \bar{r}_t$. When applying the Q-learning algorithm of Tsitsiklis and Van Roy (1999) as above at iteration k of our approximation algorithm (4.2), the initial $\bar{r}_0$ can be naturally chosen as the weight vector that defines Q_k, and when the algorithm terminates with $\bar{r}_T$, we let $Q_{k+1}(i,u) = \phi(i,u)'\bar{r}_T$ for all $(i,u) \in R$.

The convergence of the sequence $\{\bar{r}_t, t \ge 0\}$ is established by Tsitsiklis and Van Roy (1999) for discounted problems. It can be similarly proved in the SSP context, assuming that ν is a proper policy. In particular, under standard stepsize conditions, one can show that as t goes to infinity, the vector $\tilde{Q}_t$ with components $\tilde{Q}_t(i,u) = \phi(i,u)'\bar{r}_t$ converges to the unique solution of the projected Bellman equation (4.4), where the weights defining the projection norm $\|\cdot\|_w$ in Π correspond to the steady-state probabilities of the Markov chain $\{(i_t, u_t)\}$ given above. The argument for convergence is based on the fact that $\Pi F_{J,\nu}$ is a contraction mapping (with respect to some norm)—hence Eq. (4.4) has a unique solution, and that the damped fixed point mapping $(1-\alpha)I + \alpha\Pi F_{J,\nu}$ with $\alpha \in (0,1)$ is a contraction mapping with respect to the weighted Euclidean norm $\|\cdot\|_w$ (this can be shown using the arguments in Bertsekas and Yu 2009, Example 5, Sect. 7). Using this fact with some fixed α, the convergence of $\{\tilde{Q}_t\}$ can be analyzed similar to Tsitsiklis and Van Roy (1999).

We mention that the Q-learning algorithm of Tsitsiklis and Van Roy (1999) has several variants, e.g., Choi and Van Roy (2006), Yu and Bertsekas (2007a, 2007b). They can also be used to solve the optimal stopping problems in our context, under the same conditions as discussed above.

4.2 Error bound

In this subsection, we consider the abstract approximation algorithm (4.1) and give a worst-case performance bound for certain deterministic policies obtained with such an algorithm, in terms of the approximation error the algorithm incurs at each iteration. Our result requires the strong condition that all policies in Π_{SD} are proper, and uses the uniform sup-norm contraction property of the mappings underlying the algorithm (Lemma 2.2). Our result applies to the case where $m_k = +\infty$ in each iteration of the algorithm (4.1), hence it applies also to approximation algorithms in the form of (4.2), including the one of the preceding subsection. Our bound is qualitative, as it depends on constants that are generally unknown.

To begin with, we define how approximation error will be measured. Recall that for SSP models in which all policies are proper, we have defined the weighted sup-norms $\|J\|_\xi$ and $\|Q\|_{\xi^x}$, which are given by Eqs. (2.25)–(2.26), and with respect to which the uniform sup-norm contraction property given by Lemma 2.2 is stated. We will use these norms to measure the error in cost/Q-factor approximation. In particular, we characterize the worst-case per iteration error in the approximation algorithm (4.1) as follows: for all $k \geq 0$,

$$\left\| Q_{k+1} - F^{m_k}_{J_k,\nu_k} Q_k \right\|_{\xi^x} \leq \delta, \tag{4.5}$$

and

$$\|J_{k+1} - M Q_{k+1}\|_\xi \leq \epsilon,$$

which is equivalently written as

$$\left| J_{k+1}(i) - \min_{u \in U(i)} Q_{k+1}(i,u) \right| \leq \epsilon \xi(i), \quad \forall i \in S. \tag{4.6}$$

The error terms δ and ϵ are in general unknown.

We consider the performance of the deterministic policies $\{\mu_k\}$, where μ_{k+1} satisfies

$$Q_{k+1}\big(i, \mu_{k+1}(i)\big) \leq \min_{u \in U(i)} Q_{k+1}(i,u) + \epsilon \xi(i), \quad \forall i \in S. \tag{4.7}$$

These policies are associated with the approximate minimization in Eq. (4.6) for obtaining J_{k+1} and would be the policies of interest when the algorithm is terminated at iteration $k+1$. We have the following asymptotic result.

Proposition 4.1 *Suppose that all policies in Π_{SD} are proper. Assume that for some $\delta, \epsilon \geq 0$ and each $k \geq 0$, Eq.* (4.5) *holds for some positive integer $m_k \geq 1$, and Eqs.* (4.6)–(4.7) *also hold. Then for any stationary policy μ that is a limit point of $\{\mu_k\}$, we have*

$$\|J_\mu - J^*\|_\xi \leq \frac{2(\delta + \epsilon)}{(1-\beta)^2},$$

where J_μ is the vector of total costs of μ, and $\beta \in [0, 1)$ is given by Eq. (2.25).

We omit the proof for the reason that it is a nearly verbatim repetition of the corresponding proof of Bertsekas and Yu (2012, Proposition 6.1) for the discounted case. In particular, the proof can be obtained from the latter analysis by replacing the discount factor by β and by replacing the sup-norms on the spaces of costs and Q-factors, which are used in the discounted case, by the weighted sup-norms $\|\cdot\|_\xi$, $\|\cdot\|_{\xi^x}$, respectively.

We mention that the bound of Proposition 4.1 is consistent with related bounds for approximate PI (Bertsekas and Tsitsiklis 1996, Proposition 6.2) as well as approximated modified PI (Thiery and Scherrer 2010a, 2010b) for discounted MDP and SSP with all policies assumed proper. The latter is a restrictive condition. For approximate PI, there is a similar bound under the general SSP model assumption (Bertsekas and Tsitsiklis 1996, Proposition 6.3). Appropriate extensions of our analysis to more general SSP models is a worthy subject for further research.

5 Concluding remarks

We have developed and analyzed new Q-learning algorithms for SSP problems, extending related algorithms for discounted MDP, given in our earlier paper (Bertsekas and Yu 2012). We have established the convergence of deterministic and stochastic implementations without cost function approximation under the same conditions as those for the classical Q-learning algorithm. The challenge of the convergence analysis is to deal with issues arising from the lack of a contraction property of the associated mappings, as well as the natural asynchronism of simulation-based implementations.

The algorithms may be applied, with guaranteed convergence, without cost function approximation to small-scale problems or to large-scale problems through the use of aggregation (see e.g., Jaakkola et al. 1994, 1995; Gordon 1995; Tsitsiklis and Van Roy 1996; Bertsekas 2011). The algorithms may also be applied with cost function approximation to large-scale problems, and for this case, their performance has been quantified through an error bound that we have obtained.

While we have not presented computational results with our Q-learning algorithms of Sects. 2 and 3, we expect that they have similar qualitative behavior to their discounted counterparts, which we have tested on several problems in Bertsekas and Yu (2012), including the counterexample of Williams and Baird (1993). In particular, compared to modified PI algorithms that use linear equation-based policy evaluation, we expect that our deterministic algorithms of Sect. 2 have a more reliable performance with theoretically guaranteed convergence in an asynchronous computational environment. Compared to the classical Q-learning algorithm, we expect that our stochastic algorithms of Sect. 3 require a comparable number of iterations, but a faster computation time, since they perform a minimization over all controls far less frequently.

Finally, we note that the general idea of Bertsekas and Yu (2012) and this paper about forming PI-like asynchronous algorithms, is applicable to a host of dynamic programming problems, including minimax control problems. This has been studied, in a deterministic setting, by the authors (Bertsekas and Yu 2010).

Acknowledgements We thank Prof. John Tsitsiklis for mentioning to us the technique to generalize the convergence result of Proposition 3.1 to Theorem 3.1.

Appendix A: A fixed point property of $F_{J,\nu}$

The mapping $F_{J,\nu}$ defined by Eq. (2.7) is a monotone and nonexpansive mapping with respect to the sup-norm, i.e.,

$$F_{J,\nu}Q \le F_{J,\nu}\hat{Q} \quad \text{if } Q \le \hat{Q}; \qquad \|F_{J,\nu}Q - F_{J,\nu}\hat{Q}\|_\infty \le \|Q - \hat{Q}\|_\infty, \quad \forall Q, \hat{Q}.$$

In this appendix we prove that $F_{J,\nu}$ has a unique fixed point, thereby establishing Proposition 2.1(i) in view of Lemma 2.1. For convenience we repeat Proposition 2.1(i) below.

Proposition A.1 *Under Assumption 1.1, for any J and $\nu \in \Pi_{\mathrm{SR}}$, the mapping $F_{J,\nu}$ given by Eq. (2.7) has a unique fixed point $Q_{J,\nu}$, and $\lim_{m\to\infty} F^m_{J,\nu}Q = Q_{J,\nu}$ for any $Q \in \Re^{|R|}$.*

We start with two lemmas that will be used in the proof. The first lemma is about a relation between the (long-run) average cost and the total cost of a stationary policy. (The total cost is as defined in Sect. 1; for the definition of the average cost, see e.g., Puterman 1994.) The second lemma is about an implication of our SSP model assumption on the average cost of an improper policy.

Lemma A.1 *Let σ be a stationary policy of a finite-space MDP. If the average cost of σ is nonpositive (strictly positive, respectively) for an initial state, then its total cost is less than (equal to, respectively) $+\infty$ for that initial state.*

Proof This follows from the inequalities of Puterman (1994, Theorem 9.4.1(a), p. 472) applied to a single policy σ. □

Lemma A.2 *Under Assumption 1.1, any improper policy in Π_{SD} has strictly positive average cost on every recurrent class that it induces, other than the recurrent class $\{0\}$.*

Proof The proof is by contradiction. By definition, for a stationary policy to be called improper, it must induce at least one recurrent class in S. Assume that the statement of the lemma were not true. Let μ be an improper policy in Π_{SD} and $C \subset S$ be a recurrent class that μ induces, such that the average cost of μ on C is no greater than 0. Let $\bar{\mu}$ be a proper policy in Π_{SD}, which exists under Assumption 1.1(i). Consider the policy $\hat{\mu} \in \Pi_{\mathrm{SD}}$ which coincides with the improper policy μ on C and with the proper policy $\bar{\mu}$ on the rest of the states. Then $\hat{\mu}$ is an improper policy and induces two recurrent classes, C and $\{0\}$; the rest of the states are transient under $\hat{\mu}$. (If there were another recurrent class C' under $\hat{\mu}$, then C' would also be a recurrent class under $\bar{\mu}$, which would contradict the fact $\bar{\mu}$ is proper.) On C, the average cost of $\hat{\mu}$ equals that of μ, which is no greater than 0. Hence, the average cost of $\hat{\mu}$ is no greater than 0 for all initial states in S_o. This, by Lemma A.1, implies that the total cost of $\hat{\mu}$ is less than $+\infty$ for all initial states. On the other hand, since $\hat{\mu}$ is an improper policy in Π_{SD}, by Assumption 1.1(ii) it must incur infinite cost for some initial state, a contradiction. □

We now consider the equation

$$Q = F_{J,\nu}Q \tag{A.1}$$

and show that it has a unique solution. As described in Sect. 2.2 (see also Footnote 2), $F_{J,\nu}$ is associated with an optimal stopping problem defined by (J, ν), and Eq. (A.1) corresponds to the Bellman equation for the Q-factors associated with the continuation action, for the optimal stopping problem. In what follows, we view this problem equivalently as an SSP problem, to which we apply the results of Bertsekas and Tsitsiklis (1991) on SSP to conclude that the Bellman equation (A.1) has a unique solution.

More specifically, as described in Sect. 2.2, in the optimal stopping problem associated with $F_{J,\nu}$, the unstopped process is the Markov chain on the state space R_o that has the same distribution as the one induced by ν in the original SSP problem. We can assume without loss of generality that whenever the stopping action is applied, the stopping cost is incurred and the system then transitions to the absorbing, cost-free destination state (with its dummy control), $(0, 0)$. The optimal stopping problem can thus be viewed equivalently as an SSP problem with state space R_o and with two controls, to continue and to stop, for each state in R [where at $(i, u) \in R$, the expected one-stage cost of continuation is $g(i, u)$ and the stopping cost is $J(i)$]. We refer to this SSP problem as the optimal stopping/SSP problem. The total cost Bellman equation for the Q-factors of this SSP problem is a system of equations in the Q-factors associated with the continuation action (cf. Footnote 2), and it is the same as Eq. (A.1).

If we establish that the optimal stopping/SSP problem satisfies the SSP model Assumption 1.1, then based on the results of Bertsekas and Tsitsiklis (1991), its Bellman equation (A.1) would have a unique solution, and Proposition 2.1(i) would be proved. Now the optimal stopping/SSP problem satisfies Assumption 1.1(i), that is, it has a proper policy in Π_{SD}, because to stop at every state is a proper policy. Hence, to prove Proposition 2.1(i), it suffices to show that the optimal stopping/SSP problem satisfies Assumption 1.1(ii).

Lemma A.3 *The optimal stopping/SSP problem satisfies Assumption 1.1*(ii): *every improper policy in* Π_{SD} *incurs infinite cost for some initial state.*

Proof We use proof by contradiction. Assume that the optimal stopping/SSP problem has an improper policy $\sigma \in \Pi_{\mathrm{SD}}$ whose total cost for all initial states is less than $+\infty$. Then by Lemma A.1, the average cost of σ is nonpositive for every initial state.

Recall that for a stationary policy to be called improper, it must induce a recurrent class on the state space from which the desired, absorbing destination state is unreachable. Recall also that in the optimal stopping/SSP problem, the unstopped process is the same Markov chain (on R_o) induced by the policy ν in our original SSP problem. Therefore, the fact that σ is an improper policy of the optimal stopping/SSP problem implies that

(i) ν is an improper policy of the original SSP problem; and
(ii) σ induces a recurrent class $E \subset R$, and (hence) σ does not take the stopping action on E.

From (ii) it follows that E must also be a recurrent class induced by ν in the original SSP, and the Markov chain on E induced by ν is the same as the one induced by σ in the optimal stopping/SSP. Therefore, on E, the average costs of σ and ν are equal. Denote this average cost by η. Since the average cost of σ is nonpositive for all initial states as proved earlier, $\eta \le 0$.

From now on, we focus on the original SSP problem, its improper policy ν, and the recurrent Markov chain on E induced by ν. Let $\xi(x)$, $x \in E$ denote the steady-state probabilities of the latter Markov chain. Then, since η is the average cost of ν on E, we have

$$\eta = \sum_{x \in E} \xi(x) g(x) \le 0, \tag{A.2}$$

where g is the expected one-stage cost function. On the other hand, ν is a stationary randomized policy defined by $\nu(u \mid i)$, $(i,u) \in R$. Let $U_\nu(i) = \{u \in U(i) \mid \nu(u \mid i) > 0\}$ for every state $i \in S$, and let $D \subset S$ be the projection of E on S (i.e., $D = \{i \in S \mid \exists u \text{ with } (i,u) \in E\}$). Consider all deterministic policies $\mu \in \Pi_{\mathrm{SD}}$ (of the original SSP problem) such that $\mu(i) \in U_\nu(i)$ for all $i \in D$. There are a finite number of such policies; denote them by $\mu_1, \dots, \mu_m$. The definition of these policies, together with the fact that $E \subset R$ is recurrent under the stationary randomized policy ν, have the following implications:

(a) $E = \bigcup_{i \in D} \{(i,u) \mid u \in U_\nu(i)\}$.
(b) Under every policy μ_j, $j = 1, \dots, m$, D is closed (with respect to the Markov chain induced by μ_j on S_o).
(c) Every μ_j is an improper policy in Π_{SD} [since from any initial state in D, the state 0 is unreachable in view of (b)].
(d) Restricted to the state and control subsets, D and $U_\nu(i)$, $i \in D$, the original SSP problem is an MDP with state space D and state-control space E; in this MDP, $\mu_1, \dots, \mu_m$ (restricted to D) are all the deterministic stationary policies, and ν (restricted to D) is a randomized stationary policy.

Let us consider the MDP with state space D mentioned in (d) above. Fix some $\bar{i} \in D$. For $j = 1, \dots, m$, let $\xi_j(x)$, $x \in E$, denote the limiting average state-action frequencies of μ_j starting from initial state $\bar{i}$. (See e.g., Puterman 1994, Chap. 8.9.1 for the definition of these frequencies.) Note that the limiting average state-action frequencies of ν starting from $\bar{i}$ are $\xi(x), x \in E$. By Puterman (1994, Theorem 8.9.3, p. 400), ξ lies in the convex hull of $\{\xi_1, \dots, \xi_m\}$, so by Eq. (A.2), for some $\alpha_1, \dots, \alpha_m \in [0,1]$ with $\sum_{j=1}^m \alpha_j = 1$, we have

$$\eta = \sum_{j=1}^{m} \alpha_j \sum_{x \in E} \xi_j(x) g(x) \le 0.$$

Hence, there exists some j such that

$$\sum_{x \in E} \xi_j(x) g(x) \le 0. \tag{A.3}$$

Since the left-hand side equals the average cost of the policy μ_j for initial state $\bar{i}$, Eq. (A.3) implies that under μ_j, there exists a recurrent class $C \subset D$ that is reachable from $\bar{i}$, such that the average cost of μ_j on C is no greater than 0. But μ_j is an improper policy in Π_{SD} [see (c) above] and the original SSP problem satisfies Assumption 1.1, so by Lemma A.2, the average cost of μ_j must be strictly positive on C, a contradiction.

Hence, every improper policy in Π_{SD} of the optima stopping/SSP problem must incur infinite cost for some initial state. The proof is complete. □

We have now proved Proposition 2.1(i) (Proposition A.1).

Appendix B: Verifying convergence conditions on noise

In Sect. 3.2, as the first step to analyze the convergence of the iterates $x_k = (J_k, Q_k)$ generated by our Q-learning algorithm (3.3)–(3.4), we expressed the iterates equivalently and compactly as

$$x_{k+1}(\ell) = (1 - \gamma_{\ell,k}) x_k(\ell) + \gamma_{\ell,k} L_\ell^{\bar{\nu}_k^\ell} x_k^{(\ell)} + \gamma_{\ell,k} \omega_{\ell,k}, \quad \ell \in S \cup R, \tag{B.1}$$

where $x_k(\ell)$ denotes the ℓth component of x_k, and if $\ell = (i,u) \in R$ and $\gamma_{\ell,k} > 0$, then $\bar{\nu}_k^\ell$ is the randomized policy defined by Eq. (3.14) and $\omega_{\ell,k}$ is a noise term given by

$$\omega_{\ell,k} = \hat{g}\big(i,u,j_k^{iu}\big) + \min\big\{J_k^{(\ell)}\big(j_k^{iu}\big), Q_k^{(\ell)}\big(j_k^{iu}, v_k^{iu}\big)\big\} - \big(F_{J_k^{(\ell)},\bar{\nu}_k^\ell} Q_k^{(\ell)}\big)(i,u); \qquad \text{(B.2)}$$

whereas if $\ell = i \in S$, then $\omega_{\ell,k} = 0$ (and $\bar{\nu}_k^\ell$ is immaterial). [See Eqs. (3.15) and (3.16).] The stochastic approximation-based convergence analysis we used requires the following conditions on the conditional mean and variance of the noise terms $\omega_{\ell,k}$: (i) for every $\ell \in S \cup R$ and $k \geq 0$,

$$\mathrm{E}[\omega_{\ell,k} \mid \mathcal{F}_k] = 0, \quad \text{w.p.1},$$

and (ii) there exist (deterministic) constants A and B such that for every $\ell \in S \cup R$ and $k \geq 0$,

$$\mathrm{E}\big[\omega_{\ell,k}^2 \bigm| \mathcal{F}_k\big] \leq A + B \max_{\ell' \in S\cup R} \max_{\tau \leq k} \big|x_\tau(\ell')\big|^2, \quad \text{w.p.1}.$$

They are certainly satisfied for $\ell \in S$ (since $\omega_{\ell,k} = 0$). We verify below that they are satisfied for $\ell \in R$, under Assumption 3.1 and condition (3.13), which are the algorithmic conditions in Proposition 3.1.

First, we verify by induction on k that $\mathrm{E}[|J_k(i)|] < \infty$ and $\mathrm{E}[|Q_k(i,u)|] < \infty$ for all $i \in S$, $(i,u) \in R$, and $k \geq 0$. Since the initial (J_0, Q_0) is given, this is certainly true for $k = 0$. Suppose that for some $k \geq 0$, this is true for all $\tau \leq k$. Then, by Eqs. (3.3)–(3.4) and condition (3.13) on stepsizes, we have

$$\big|J_{k+1}(i)\big| \leq D\big|J_k(i)\big| + D \sum_{\tau \leq k} \sum_{u \in U(i)} \big|Q_\tau(i,u)\big|, \quad \forall i \in S,$$

$$\big|Q_{k+1}(i,u)\big| \leq D\big|Q_k(i,u)\big| + D\big|\hat{g}\big(i,u,j_k^{iu}\big)\big| + D \sum_{\tau \leq k} \sum_{\ell \in S\cup R} \big|x_\tau(\ell)\big|, \quad \forall (i,u) \in R,$$

where D is the constant in condition (3.13) and $x_\tau(\ell)$ is the ℓth component of (J_τ, Q_τ). In the right-hand sides of these two equations are sums of a finite (constant) number of random variables, each of which has finite expectation by the induction hypothesis and by condition (3.7). Therefore, $\mathrm{E}[|J_{k+1}(i)|] < \infty$ and $\mathrm{E}[|Q_{k+1}(i,u)|] < \infty$ for every i and (i,u), and by induction, the claim is true for all k.

Consider any $k \geq 0$ and $\ell = (i,u) \in R$. We now verify the required conditions on the conditional mean and variance of $\omega_{\ell,k}$. By the definitions of $\bar{\nu}_k^\ell$ and mapping $F_{J,\nu}$ [cf. Eqs. (3.14) and (2.7)], we have

$$g(i,u) + \mathrm{E}\big[\min\big\{J_k^{(\ell)}\big(j_k^{iu}\big), Q_k^{(\ell)}\big(j_k^{iu}, v_k^{iu}\big)\big\} \bigm| \mathcal{F}_k\big] = \big(F_{J_k^{(\ell)},\bar{\nu}_k^\ell} Q_k^{(\ell)}\big)(i,u), \qquad \text{(B.3)}$$

where the conditional expectation is over (j_k^{iu}, v_k^{iu}). Here, we have also used the following fact: since

$$\big|\min\big\{J_k^{(\ell)}\big(j_k^{iu}\big), Q_k^{(\ell)}\big(j_k^{iu}, v_k^{iu}\big)\big\}\big| \leq \sum_{\tau \leq k} \sum_{\ell' \in S\cup R} \big|x_\tau(\ell')\big|$$

and all the random variables in the right-hand side have finite expectation as we proved earlier, the random variable on the left-hand side has finite expectation and therefore, the conditional expectation in Eq. (B.3) is well-defined. By Eqs. (B.2) and (B.3),

$$\omega_{\ell,k} = Z_1 + Z_2$$

where

$$Z_1 = \hat{g}\big(i,u,j_k^{iu}\big) - g(i,u),$$
$$Z_2 = \min\big\{J_k^{(\ell)}\big(j_k^{iu}\big), Q_k^{(\ell)}\big(j_k^{iu}, v_k^{iu}\big)\big\} - \mathrm{E}\big[\min\big\{J_k^{(\ell)}\big(j_k^{iu}\big), Q_k^{(\ell)}\big(j_k^{iu}, v_k^{iu}\big)\big\} \bigm| \mathcal{F}_k\big].$$

By condition (3.7) $\mathrm{E}[Z_1 \mid \mathcal{F}_k] = 0$, so $\mathrm{E}[\omega_{\ell,k} \mid \mathcal{F}_k] = 0$.

We now bound the conditional variance of $\omega_{\ell,k}$. (Because $\omega_{\ell,k}^2$ is a nonnegative random variable, $\mathrm{E}[\omega_{\ell,k}^2 \mid \mathcal{F}_k]$ is well-defined always.) By the definition of Z_2, we have

$$|Z_2| \le 2 \max_{\ell' \in S \cup R} \max_{\tau \le k} \bigl|x_\tau(\ell')\bigr|.$$

Hence,

$$\begin{aligned}
\omega_{\ell,k}^2 &\le Z_1^2 + Z_2^2 + 2|Z_1||Z_2| \\
&\le \bigl(\hat{g}\bigl(i,u,j_k^{iu}\bigr) - g(i,u)\bigr)^2 + 4 \max_{\ell' \in S \cup R} \max_{\tau \le k} \bigl|x_\tau(\ell')\bigr|^2 \\
&\quad + 4\bigl|\hat{g}\bigl(i,u,j_k^{iu}\bigr) - g(i,u)\bigr| \cdot \max_{\ell' \in S \cup R} \max_{\tau \le k} \bigl|x_\tau(\ell')\bigr|,
\end{aligned}$$

and taking conditional expectation of both sides,

$$\begin{aligned}
\mathrm{E}\bigl[\omega_{\ell,k}^2 \mid \mathcal{F}_k\bigr] &\le \mathrm{E}\bigl[\bigl(\hat{g}\bigl(i,u,j_k^{iu}\bigr) - g(i,u)\bigr)^2 \mid \mathcal{F}_k\bigr] + 4 \max_{\ell' \in S \cup R} \max_{\tau \le k} \bigl|x_\tau(\ell')\bigr|^2 \\
&\quad + 4\mathrm{E}\bigl[\bigl|\hat{g}\bigl(i,u,j_k^{iu}\bigr) - g(i,u)\bigr| \mid \mathcal{F}_k\bigr] \cdot \max_{\ell' \in S \cup R} \max_{\tau \le k} \bigl|x_\tau(\ell')\bigr|.
\end{aligned}$$

Combining this with condition (3.7), it follows that

$$\mathrm{E}\bigl[\omega_{\ell,k}^2 \mid \mathcal{F}_k\bigr] \le A + B \max_{\ell' \in S \cup R} \max_{\tau \le k} \bigl|x_\tau(\ell')\bigr|^2,$$

for some (deterministic) constant A and B independent of k; since the set R is finite, A, B can be chosen to be independent of ℓ. This shows that the desired conditions hold.

Appendix C: Modifications and extensions of the analysis of Tsitsiklis (1994)

In Sect. 3.3, when proving Proposition 3.1 on the convergence of our stochastic Q-learning algorithm, we applied the results of Tsitsiklis (1994) with certain modifications and extensions to suit our problem (cf. Sect. 3.2). We explain what the necessary changes are in this appendix; we suggest that it be read side-by-side with the reference (Tsitsiklis 1994). A reproduction of the latter with modifications is available (Yu 1994), for readers who wish to verify all the proof details. Before we start, let us also remark that the changes required do not alter the analysis of Tsitsiklis (1994) in any essential way, nor are they technically complicated.

Recall from Sect. 3.2 that under the conditions of Proposition 3.1, our algorithm can be written equivalently as in Eq. (3.15),

$$x_{k+1}(\ell) = (1 - \gamma_{\ell,k}) x_k(\ell) + \gamma_{\ell,k} L_\ell^{\bar{v}_k^\ell} x_k^{(\ell)} + \gamma_{\ell,k}\omega_{\ell,k}, \quad \ell \in S \cup R$$

(where for every ℓ and k, $x_k, \gamma_{\ell,k}, L_\ell^{\bar{v}_k^\ell} x_k^{(\ell)}$ are $\mathcal{F}_k$-measurable, and $\omega_{\ell,k}$ is $\mathcal{F}_{k+1}$-measurable and has zero mean conditional on $\mathcal{F}_k$). There are three differences between the form and conditions of our algorithm and those of the stochastic approximation algorithm analyzed in Tsitsiklis (1994). Two of them are related to the stepsize conditions, and the third is related to the use of multiple mappings (instead of a single one) in our algorithm. We describe them one by one below, together with the modifications in the analysis of Tsitsiklis (1994) to accommodate them.

The stepsize condition in Tsitsiklis (1994) for each component ℓ is slightly different than condition (3.11); in addition to $\gamma_{\ell,k} \in [0,1]$, it requires $\sum_{k\ge 0} \gamma_{\ell,k} = \infty$, $\sum_{k \ge 0} \gamma_{\ell,k}^2 < C$ w.p.1, for some deterministic constant C, instead of C being ∞ (Tsitsiklis 1994, Assumption 3).

However, this difference only affects one technical lemma, Lemma 1 in Tsitsiklis (1994). By strengthening that lemma so that its conclusions hold under the weaker stepsize condition (3.11), (3.13), the rest of the analysis in Tsitsiklis (1994) remains essentially intact and all the conclusions in Tsitsiklis (1994) hold under the weaker condition. The additional analysis just mentioned for strengthening Lemma 1 of Tsitsiklis (1994) can be found in Bertsekas and Tsitsiklis (1996, Proposition 4.1 and Example 4.3, pp. 141–143) (see also Corollary 4.1 and Sect. 4.3.6 therein).

Instead of the stepsize condition (3.11), our algorithm uses the different stepsize condition (3.10), (3.12) for updating cost components $J(i)$, under which the stepsize sequence $\{\gamma_{i,k},\ k \geq 0\}$ can violate the square-summable condition $\sum_{k\geq 0} \gamma_{i,k}^2 < \infty$. The reason the square-summable condition is not needed for the cost updates is that these updates are "noiseless": $\omega_{i,k}$ is identically zero for all k [cf. Eqs. (3.3) and (3.15)]. For proof details, we note that the conclusions of Lemmas 1 and 2 of Tsitsiklis (1994) are trivially true for a zero noise sequence. Since in Tsitsiklis (1994) these two lemmas, which concern the cumulative effects of noises, are the only places where the square-summable condition is used, the analysis given in Tsitsiklis (1994) goes through and the conclusions therein hold under our stepsize conditions for those components with noiseless updates.

The results of Tsitsiklis (1994) are stated for an algorithm of a form similar to (3.15) but with a single mapping L, instead of multiple mappings L^ν as in our algorithms. To apply the analysis of Tsitsiklis (1994) to the case of multiple mappings, we replace the condition on the single mapping L by appropriate conditions on the set of mappings L^ν, and we also need to make some modifications in the proofs of Tsitsiklis (1994). We describe the details separately for the two cases: (i) all L^ν are contraction mappings, and (ii) all L^ν are monotone nonexpansive mappings.

The first case where all L^ν are contraction mappings is simpler to handle. In Tsitsiklis (1994), the boundedness and convergence of the iterates x_k when the associated mapping L is a contraction, are established by Theorems 1 and 3, respectively. The proofs of these theorems rely on the contraction property through two inequalities [cf. Eqs. (8)–(9) in Assumptions 5 and 6 in Tsitsiklis 1994]:

$$\|Lx\|_\zeta \leq \beta \|x\|_\zeta + D, \quad \forall x, \tag{C.1}$$

for some constant D, and

$$\left\|Lx - x^*\right\|_\zeta \leq \beta \left\|x - x^*\right\|_\zeta, \quad \forall x, \tag{C.2}$$

where L is a contraction with respect to some weighted sup-norm $\|\cdot\|_\zeta$, with modulus $\beta \in [0,1)$ and fixed point x^*. In our case, we place a uniform sup-norm contraction condition on the set of mappings involved; this is the same property stated in Proposition 2.3 for the set $\{L^\nu \mid \nu \in \Pi_{\mathrm{SR}}\}$ when all policies are proper in the SSP problem. Then, inequalities (C.1)–(C.2) are satisfied by every mapping involved, and the proofs of Theorems 1 and 3 of Tsitsiklis (1994) go through. (The details of this analysis can also be found in Bertsekas and Tsitsiklis (1996, Sects. 4.3.3, 4.3.6). This reference does not concern asynchronous algorithms with communication delays, but the analysis given there essentially applies to such algorithms.)

The second case where all L^ν are nonexpansive mappings involves more modifications. In Tsitsiklis (1994), the mapping L is required to be monotone and nonexpansive with respect to the sup-norm and to have a unique fixed point x^* (cf. Assumption 4 therein). In our case, we place this condition on all the mappings L^ν involved and require them to have the same fixed point x^*, and furthermore, we require that the same is true for the two mappings $\overline{L}$ and $\underline{L}$, which are defined by taking componentwise supremum and infimum, respectively, of $L^{\nu}x$ over ν. This is the same property stated in Proposition 2.2 for the set $\{L^{\nu} \mid \nu \in \Pi_{\mathrm{SR}}\}$ under our general SSP model assumption, Assumption 1.1. We then make several modifications in the proofs of Tsitsiklis (1994) as follows (where the equations and lemmas mentioned refer to those appearing in Tsitsiklis 1994):

(i) Replace the mapping L (F in the notation of Tsitsiklis 1994) by $\overline{L}$ ($\underline{L}$, respectively) in the definition of a sequence of upper (lower, respectively) bounds given by Eq. (16) [Eq. (17), respectively]. Make the same changes in the statement of Lemma 4 and its proof, as well as in the proof of Lemma 5.

(ii) For proving Lemmas 6–7 (which are about upper-bounding x_k), change the mapping L (F in the notation of Tsitsiklis 1994) in Eq. (21) to $\overline{L}$; change the equality sign in the last line of the proof of Lemma 6 to "$\leq$"; and change the mapping L that appears after the proof of Lemma 6, to $\overline{L}$.

(iii) For proving the symmetric counterparts of Lemmas 6–7 (which are about lower-bounding x_k), repeat the changes described in (ii) with $\underline{L}$ in place of $\overline{L}$ and with "$\geq$" in place of "$=$" in the last line of the proof of Lemma 6.

With the above modifications, the proof for Theorem 2 in Tsitsiklis (1994) applies to our case and shows that Lemma 3.1 in our Sect. 3.3.2 holds.

References

Baudet, G. M. (1978). Asynchronous iterative methods for multiprocessors. *Journal of the Association for Computing Machinery*, *25*, 226–244.

Bertsekas, D. P. (1982). Distributed dynamic programming. *IEEE Transactions on Automatic Control*, *27*, 610–616.

Bertsekas, D. P. (1983). Asynchronous distributed computation of fixed points. *Mathematical Programming*, *27*, 107–120.

Bertsekas, D. P. (2007). *Dynamic programming and optimal control* (Vol. II, 3rd ed.). Belmont: Athena Scientific.

Bertsekas, D. P. (2011). *Approximate dynamic programming*. Book chapter. On-line at: http://web.mit.edu/dimitrib/www/dpchapter.html.

Bertsekas, D. P., & Tsitsiklis, J. N. (1989). *Parallel and distributed computation: numerical methods*. Englewood Cliffs: Prentice-Hall. Republished by Athena Scientific, Belmont, 1997.

Bertsekas, D. P., & Tsitsiklis, J. N. (1991). An analysis of stochastic shortest path problems. *Mathematics of Operations Research*, *16*(3), 580–595.

Bertsekas, D. P., & Tsitsiklis, J. N. (1996). *Neuro-dynamic programming*. Belmont: Athena Scientific.

Bertsekas, D. P., & Yu, H. (2009). Projected equation methods for approximate solution of large linear systems. *Journal of Computational and Applied Mathematics*, *227*, 27–50.

Bertsekas, D. P., & Yu, H. (2010). Distributed asynchronous policy iteration in dynamic programming. In *Proc. of the 48th Allerton conference on communication, control and computing*, Allerton, IL (pp. 1368–1375).

Bertsekas, D. P., & Yu, H. (2012). Q-learning and enhanced policy iteration in discounted dynamic programming. *Mathematics of Operations Research*, *37*(1), 66–94. (Technical Report 2831). LIDS, MIT, April 2010.

Canbolat, P. G., & Rothblum, U. G. (2012). (Approximate) iterated successive approximations algorithm for sequential decision processes. *Annals of Operations Research*. doi:10.1007/s10479-012-1073-x.

Choi, D. S., & Van Roy, B. (2006). A generalized Kalman filter for fixed point approximation and efficient temporal-difference learning. *Discrete Event Dynamic Systems*, *16*, 207–239.

Derman, C. *Finite state Markovian decision processes*. New York: Academic Press (1970).

Eaton, J. H., & Zadeh, L. A. (1962). Optimal pursuit strategies in discrete state probabilistic systems. *Transactions of the ASME. Series D. Journal of Basic Engineering*, *84*, 23–29.

Feinberg, E. A. (1992). Stationary strategies in Borel dynamic programming. *Mathematics of Operations Research*, *17*, 392–397.

Gordon, G. J. (1995). Stable function approximation in dynamic programming. In *Proc. of the 12th int. conf. on machine learning*, San Francisco, CA (pp. 261–268).

Jaakkola, T. S., Jordan, M. I., & Singh, S. P. (1994). On the convergence of stochastic iterative dynamic programming algorithms. *Neural Computation*, *6*, 1185–1201.

Jaakkola, T. S., Singh, S. P., & Jordan, M. I. (1995). Reinforcement learning algorithm for partially observable Markov decision problems. *Advances in Neural Information Processing Systems*, *7*, 345–352.

Puterman, M. L. (1994). *Markov decision processes: discrete stochastic dynamic programming*. New York: Wiley.

Rothblum, U. G. (1979). Iterated successive approximation for sequential decision processes. In J. W. B. van Overhagen & H. C. Tijms (Eds.), *Stochastic control and optimization*. Amsterdam: Vrije University.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning*. Cambridge: MIT Press.

Thiery, C., & Scherrer, B. (2010a). Least-squares λ policy iteration: bias-variance trade-off in control problems. In *Proc. of the 27th int. conf. machine learning*, Haifa, Israel (pp. 1071–1078).

Thiery, C., & Scherrer, B. (2010b). *Performance bound for approximate optimistic policy iteration* (Technical report). INRIA.

Tsitsiklis, J. N. (1994). Asynchronous stochastic approximation and Q-learning. *Machine Learning*, *16*, 185–202.

Tsitsiklis, J. N., & Van Roy, B. (1996). Feature-based methods for large-scale dynamic programming. *Machine Learning*, *22*, 59–94.

Tsitsiklis, J. N., & Van Roy, B. (1999). Optimal stopping of Markov processes: Hilbert space theory, approximation algorithms, and an application to pricing financial derivatives. *IEEE Transactions on Automatic Control*, *44*, 1840–1851.

Veinott, A. F. Jr. (1969). Discrete dynamic programming with sensitive discount optimality criteria. *Annals of Mathematical Statistics*, *40*, 1635–1660.

Watkins, C. J. C. H. (1989). *Learning from delayed rewards*. Ph.D. thesis, Cambridge University, England.

Williams, R. J., & Baird, L. C. (1993). *Analysis of some incremental variants of policy iteration: first steps toward understanding actor-critic learning systems* (Report NU-CCS-93-11). College of Computer Science, Northeastern University.

Whittle, P. (1983). *Optimization over time* (Vol. 2) New York: Wiley.

Yu, H., & Bertsekas, D. P. (2007a). *A least squares Q-learning algorithm for optimal stopping problems* (Technical Report 2731). LIDS, MIT.

Yu, H., & Bertsekas, D. P. (2007b). Q-learning algorithms for optimal stopping based on least squares. In *Proc. European control conference (ECC)*, Kos, Greece (pp. 2368–2375).

Yu, H., & Bertsekas, D. P. (2011). *On boundedness of Q-learning iterates for stochastic shortest path problems* (Technical Report 2859). LIDS, MIT. Accepted by Mathematics of Operations Research.

Yu, H. (2011). *Some proof details for asynchronous stochastic approximation algorithms*. On-line at: http://www.mit.edu/~janey_yu/note_asaproofs.pdf.