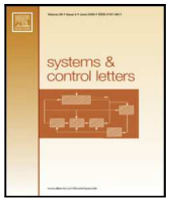
Systems & Control Letters 187 (2024) 105782

Contents lists available at ScienceDirect

Systems & Control Letters

journal homepage: www.elsevier.com/locate/sysconle



A Q-learning algorithm for Markov decision processes with continuous state spaces☆

Jiaqiao Hu [a], Xiangyu Yang [b,*], Jian-Qiang Hu [c], Yijie Peng [d]

[a] Department of Applied Mathematics & Statistics, State University of New York, Stony Brook, NY 11794-3600, USA
[b] School of Management, Shandong University, Jinan 250100, China
[c] School of Management, Fudan University, Shanghai 200433, China
[d] Guanghua School of Management, Peking University, Beijing 100871, China

ARTICLE INFO

Keywords:
Stochastic optimal control
Optimization algorithms
Markov processes
Statistical learning

ABSTRACT

We propose an online algorithm for solving a class of continuous-state Markov decision processes. The algorithm combines classical Q-learning with an asynchronous averaging procedure, which allows Q-function estimates at sampled state–action pairs to be adaptively updated based on observations collected along a single sample trajectory. These estimates are then used to iteratively construct an interpolation-based function approximator of the Q-function. We prove the convergence of the algorithm and provide numerical results to illustrate its performance.

1. Introduction

Markov decision processes (MDPs) provide an important framework to study sequential decision making problems arising in a variety of disciplines. However, when modeled as MDPs, due to the size and complexity of many practical problems, it is often not feasible to explicitly specify some of the model parameters (e.g., transition dynamics and random rewards). This has led to the development of model-free reinforcement learning (RL) techniques [1–5] that aim to approximate optimal solutions by using knowledge gained from simulation samples or system trajectories. Arguably one of the most popular and successful RL techniques is Q-learning [6]. The method can be viewed as a simulation-based approach for solving the well-known Bellman's equation and forms the foundation for many other algorithms in the field; see, e.g., [1,7–9] and references therein. Since classical Q-learning maintains a lookup table to store function estimates and requires all state–action pairs to be visited infinitely often, the applications of the method and its extensions are mostly centered around problems with finite state spaces.

In this paper, we propose a generalization of Q-learning for solving a class of infinite-horizon discounted MDPs with continuous state spaces but small (finite) action spaces. Such problems are sometimes termed "discrete decision processes" and arise frequently in industrial applications such as inventory control, optimal machine maintenance, financial derivative pricing, and many others; see, e.g., [10]. Our algorithm replaces the table-based representation of the Q-function with an interpolation-based function approximator. In particular, to achieve the desired transition from a (discrete) finite to an uncountable state space setting (where the probability of revisiting a previously encountered state is typically zero), the construction of the function approximator is coupled with a technique adapted from the simulation optimization literature called the shrinking ball method [11]. Such a technique allows the algorithm to learn the Q-value at a generated state–action pair by averaging estimates obtained at all other pairs that are close to it, avoiding the need for expending a significant amount of simulation effort at every visited state–action pair. These Q-value estimates are then retained at each step and used online in an interpolation-based strategy to update the function approximator. Under appropriate conditions, we show that the sequence of function approximators converges uniformly to the optimal Q-function with probability one.

☆ The work of Jiaqiao Hu was supported by the U.S. National Science Foundation under grant CMMI-2027527. The work of Xiangyu Yang was supported in part by the major project of National Natural Science Foundation of China (NSFC) under Grant 72293582, in part by the China Postdoctoral Science Foundation under Grant 2023M732054, in part by the Shandong Provincial Natural Science Foundation under Grant ZR2023QG159, and in part by the Shandong Postdoctoral Science Foundation under Grant SDCX-RS-202303004. The work of Jian-Qiang Hu was supported by National Natural Science Foundation of China (NSFC) under Grants 72033003 and 71720107003. The work of Yijie Peng was supported by National Natural Science Foundation of China (NSFC) under Grants 72250065, 72022001, and 71901003.

* Corresponding author.
E-mail addresses: jiaqiao.hu.1@stonybrook.edu (J. Hu), yangxiangyu@email.sdu.edu.cn (X. Yang), hujq@fudan.edu.cn (J.-Q. Hu), pengyijie@pku.edu.cn (Y. Peng).

https://doi.org/10.1016/j.sysconle.2024.105782
Received 4 July 2022; Received in revised form 31 May 2023; Accepted 20 March 2024

Perhaps the most studied value function approaches for solving continuous-state MDPs are the fitted value and Q-iterations [8,12–16]. These methods are typically off-line and use a pre-selected batch of transition samples under a supervised learning framework to compute an approximation to the value/Q-function. An online alternative is the soft-state aggregation method of [17], which maps the state space into a small number of clusters. The method generalizes the usual state aggregation in the sense that each visited state can belong to multiple clusters with certain clustering probabilities. Another online method is the interpolation-based Q-learning proposed in [18], which considers function approximators that locally interpolate Q-value estimates obtained on a given set of basis points. Melo and Ribeiro [19] also study a version of Q-learning based on linear function approximation and show the (local) convergence of the algorithm under the geometric ergodicity assumption on the underlying Markov chain. Of particular relevance to our work is the nearest neighbor regression method of [20], which estimates the Q-value at a given state–action pair using observations that lie in its neighborhood, a strategy that is very similar to our proposed shrinking ball method. Their approach uses a finite-state discretization of the original MDP and updates the Q-values over the discretized space all at once in a roughly synchronous manner.

We remark that with the exception of [19], all aforementioned approaches resort to some forms of state space discretization, whereas our algorithm is discretization-free and asynchronously approximates the Q-function based on a single sample trajectory. In addition, the convergence analysis of existing approaches are almost all based on the non-expansiveness property of the function approximator. Our approach, on the other hand, does not require the approximator to be a non-expansion and thus allows the use of more flexible function approximation tools.

The rest of this paper is structured as follows. We present the proposed algorithm in Section 2 and analyze its convergence in Section 3. A simple illustrative example is provided in Section 4. We conclude the paper in Section 5.

2. Q-learning for continuous-state MDPs

We consider an infinite-horizon discounted MDP model (S, A, p, R, β), where the state space S is a compact connected subset of $\mathfrak{R}^d$, the action space A is a finite set, $p(s'|s,a)$, $s, s' \in S$, $a \in A$ is a Markov transition density, $R(\cdot,\cdot) : S \times A \to \mathfrak{R}^+ \cup \{0\}$ is a non-negative reward function, and $\beta \in (0,1)$ is the discount factor. For simplicity, we assume that all actions $a \in A$ are admissible at all states $s \in S$.

Denote by Π the set of stationary deterministic Markovian policies, where each $\pi \in \Pi$ is a mapping from S to A with $\pi(s)$ signifying the action taken at state s. Let s_t be the state of the system at time t and a_t be the action applied at s_t. For an initial state $s_0 = s$, the value function associated with a policy π is given by $V^\pi(s) := E\left[\sum_{t=0}^{\infty} \beta^t R(s_t, \pi(s_t))|s_0 = s\right]$. The goal is to find an optimal policy $\pi^* \in \Pi$ that attains the supremum of V^π, i.e.,

$$V^*(s) := V^{\pi^*}(s) = \sup_{\pi \in \Pi} V^\pi(s)$$

for all initial states $s \in S$.

It is well-known that under mild assumptions, the optimal value function exists and is given by the unique solution to the Bellman's equation, which, when stated in terms of the Q-function $Q^*(s,a) := R(s,a) + \beta \int_S V^*(s')p(s'|s,a)ds'$, can be put in the following equivalent form:

$$Q^*(s,a) = R(s,a) + \beta \int_S \max_{b \in A} Q^*(s',b)p(s'|s,a)ds'. \tag{1}$$

In a model-free setting, the transition density p and/or the reward R are unknown. RL algorithms such as Q-learning (when S is finite) often work with a randomized (learning) policy $\pi_t(s,a)$, $a \in A$ and incrementally compute an approximate solution to (1) based on transition samples generated from π_t. Such a policy can be viewed as an action selection distribution that specifies the chance of taking an action $a \in A$ at time t when state s is encountered. Throughout the paper, we consider the case where the reward takes the form of an expectation $R(s_t,a_t) = E[r(s_t,a_t,\omega_t)]$, which cannot be evaluated exactly. Instead, only the sample reward $r(s_t,a_t,\omega_t)$ is available, where ω_t's are i.i.d. random vectors taking values from some common set.

2.1. Algorithm description

Our algorithm aims at approximating the solution to (1) by using a sequence of function approximators. In particular, for each $a \in A$, we let $\mathbb{Q}_t(\cdot,a)$ be the function approximator of $Q^*(\cdot,a)$ constructed at time t and $Q_t(s_t,a_t)$ be the (point) estimate of $Q^*(s_t,a_t)$ at the state–action pair (s_t,a_t) obtained at time t. Denote by Λ_t the collection of all state–action pairs visited up to time t. Let $B(s,r)$ be an open ball centered at s with radius $r > 0$ and $\{r_t\}$ be the radiuses of a sequence of shrinking balls. For two state–action pairs (s_l,a_l) and (s_t,a_t) visited at distinct times $l < t$, we use

$$I_t(s_l,a_l) = \begin{cases} 1 & \text{if } s_t \in B(s_l,r_t) \text{ and } a_t = a_l; \\ 0 & \text{otherwise} \end{cases}$$

to indicate whether the pair (s_t,a_t) lies in the vicinity of (s_l,a_l). The detailed algorithmic steps are then presented below.

Q-learning for Continuous-State MDPs

Step 0: Select a policy $\{\pi_t\}$, an initial state s_0, learning rates $\alpha_t(s,a) \in (0,1)$ $\forall s \in S, \forall a \in A$, and $\forall t$, shrinking ball radiuses $\{r_t\}$, and a sequence of positive indices $\{i_t\}$. Set $\mathbb{Q}_0(s,a) = 0$ $\forall s \in S, \forall a \in A$. Set $\Lambda_0 = \emptyset$ and the iteration counter $t = 0$.

Step 1: Select an action $a_t \sim \pi_t(s_t,a)$, observe the next state $s_{t+1} \sim p(s|s_t,a_t)$, and obtain the random reward $r(s_t,a_t,\omega_t)$. Set $\Lambda_{t+1} = \Lambda_t \cup \{(s_t,a_t)\}$.

Step 2: Compute an estimate of $Q^*(s_t,a_t)$ at (s_t,a_t) as

$$Q_t(s_t,a_t) = r(s_t,a_t,\omega_t) + \beta \max_{b \in A} \mathbb{Q}_t(s_{t+1},b); \tag{2}$$

For each previously visited state–action pair $(s_l,a_l) \in \Lambda_t$, update the point estimate as

$$Q_t(s_l,a_l) = (1 - \alpha_t(s_l,a_l)I_t(s_l,a_l))Q_{t-1}(s_l,a_l) + \alpha_t(s_l,a_l)I_t(s_l,a_l)Q_t(s_t,a_t). \tag{3}$$

Step 3: If $a_t = a$, then update $\mathbb{Q}_t(\cdot,a)$ to obtain a new approximator $\mathbb{Q}_{t+1}(\cdot,a)$ that interpolates the data $\left\{\left((s',a'),Q_t(s',a')\right) : (s',a') \in \Lambda_{i_t},\ a' = a\right\}$. Set $t = t+1$ and go to Step 1.

The algorithm requires a separate function approximator for every action $a \in A$. These are primarily used as predictors to predict the Q-values at unsampled locations. At each iteration t, a point estimate of $Q^*(s_t,a_t)$ at the current state–action pair (s_t,a_t) is formed in (2) based on the predicted value at the sampled next state. This step is essentially a simulation-based version of (1) with the approximators $\mathbb{Q}_t(\cdot,b)$ replacing $Q^*(\cdot,b)$. To estimate the integral involved in (1), in Eq. (3) of Step 2, we have used an improved version of the shrink ball method proposed in [11] for solving noisy optimization problems. The key idea is not to allocate a large amount of simulation replications to each visited state–action pair, but to resort to a form of asynchronous recursion, so that the estimate at a given pair can be continuously updated by averaging the performance at all other pairs that lie within a certain distance from it. In particular, for each (s_l,a_l) generated prior to time t, if $a_t = a_l$ and s_t falls in the $B(x_l,r_t)$ neighborhood of s_l, then the current estimate $Q_{t-1}(s_l,a_l)$ is adjusted in (3) by taking into account the new information $Q_t(s_t,a_t)$. The hope is that the simulation noises (due to the uncertainties in the random rewards and transitions) will average out as the number of iterations increases, whereas the bias

(due to the difference between the Q-values at two different pairs) can be eliminated by gradually sending the radius r_t to zero.

Since (3) is an asynchronous procedure, the Q-value estimates at sampled pairs are updated at different frequencies. In particular, the estimates at pairs generated in more recent iterations tend to be updated less frequently, and hence are less reliable than those obtained at pairs sampled in early iterations. Consequently, the construction of the function approximator $\mathbb{Q}_{t+1}$ at Step 3 is only based on data at pairs collected prior to a certain time $i_t < t$. In Section 3, we provide sufficient conditions on the increasing rate of i_t to ensure that estimates at all pairs in Λ_{i_t} are updated sufficiently often to yield reasonable Q-value estimates.

Note that in a finite state space setting, each ball $B(s_l, r_t)$ will only contain the state s_l itself when the radius r_t becomes sufficiently small. Thus, if the approximator $\mathbb{Q}_t$ is replaced with a full state–action table, then the two Eqs. (2) and (3), when combined, turn out to be identical to the classical Q-learning method. From this viewpoint, the algorithm can be regarded as a natural generalization of Q-learning for solving continuous-state MDPs.

3. Convergence analysis

Define σ-fields $\bar{\mathscr{F}}_t = \sigma\{s_0, a_0, \omega_0, \dots, s_t, a_t, \omega_t\}$ and $\mathscr{F}_t = \sigma\{s_0, a_0, \omega_0, s_1, a_1, \omega_1, \dots, s_t, a_t\}$. For a state–action pair (s_l, a_l) visited at iteration $l < t$, we let $N_t(s_l, a_l) = \sum_{j=l+1}^{t} I_j(s_l, a_l)$, indicating the number of times the neighborhoods $B(s_l, r_j) \cap \{a_j = a_l\}$ of (s_l, a_l) have been visited between times $l+1$ and t. We also let $\Lambda_t(a) = \{s' : (s', a') \in \Lambda_t, a' = a\}$ be the set of states sampled up to t at which action a is taken. For two states $s, s' \in S$, their Euclidean distance is denoted by $d(s, s')$, and for a set of states $C \subset S$, the distance between s and C is given by $d(s, C) := \inf_{s' \in C} d(s, s')$. For two sequences of real numbers $\{u_t\}$ and $\{v_t\}$, we say that $u_t = \Omega(v_t)$ if there exist constants $c, K > 0$ such that $u_t \geq c v_t$ for all $t \geq K$. Finally, to simplify the analysis, we assume that the learning rate is a function of $N_t(s_l, a_l)$, i.e., $\alpha_t(s_l, a_l) = f(N_t(s_l, a_l))$ for some function f. We impose the following regularity conditions on the MDP model and algorithm parameters:

Assumptions:

A1. *$R_{max} := \sup_{s,a,\omega} r(s, a, \omega) < \infty$ and there exists a constant $K_R < \infty$ such that $|R(s, a) - R(s', a)| \leq K_R d(s, s')$ $\forall\, a \in A$.*

A2. *For each $a \in A$, the transition density $p(s'|s, a)$ is continuous in both s' and s, and $p(s'|s, a) > 0$ $\forall\, s, s' \in S$, $a \in A$. In addition, $|p(s''|s, a) - p(s''|s', a)| \leq K_p(s'') d(s, s')$ $\forall\, s, s', s'' \in S$, $a \in A$, and $K_p := \int_S K_p(s) ds < \infty$.*

A3. *For each $a \in A$, $\mathbb{Q}_t(s, a)$'s are Lipschitz continuous with their Lipschitz constants uniformly bounded by $L(a) < \infty$ w.p.1.*

A4. *$f(i) \in (0, 1)$ $\forall\, i$, $\sum_{i=1}^{\infty} f(i) = \infty$, and $\sum_{i=1}^{\infty} f^2(i) < \infty$.*

A5. *$i_t = \lfloor t^{\gamma_1} \rfloor$ for $\gamma_1 \in (0, 1)$, where $\lfloor \cdot \rfloor$ is the rounding operator.*

A6. *The shrinking ball radius is non-increasing in t and satisfies $r_t = \Omega(t^{-\gamma_2})$ for some constant $\gamma_2 \in (0, \frac{1}{2d})$.*

A7. *There exists a constant $\gamma_3 > 0$ with $\gamma_3 + \gamma_2 d < \frac{1}{2}$ such that the action selection probability satisfies $\inf_{s \in S} \pi_t(s, a) = \Omega(t^{-\gamma_3})$ $\forall\, a \in A$ w.p.1.*

We briefly comment on these assumptions. A1 and A2 guarantee that the Q-function is sufficiently smooth, which in turn justifies the use of the shrinking ball method. A3 requires the function approximators to be globally Lipschitz. Intuitively, this allows an easy quantification of their prediction errors at unvisited state–action pairs. Note that this condition does not require any prior knowledge about the Lipschitz constants and is weaker than the typical non-expansiveness assumption used in the existing literature. A4–A6 are conditions on the algorithm input parameters. A7 suggests the learning policy should be persistently exploratory so that every action will be sampled with a strictly positive probability at each visited state. The degree of exploration may decay with time, which permits policies that are greedy in the limit [21]. In fact, the condition, together with A2, ensures that the Markov chain under the learning policy is Harris recurrent (e.g., [22]).

We begin by stating a number of preliminary results (Lemmas 1–4) that will be used in our convergence analysis. The first result shows that for an MDP characterized by A1 and A2, the optimal Q-function is Lipschitz continuous. In our subsequent analysis, we will denote its associated Lipschitz constant by L_Q.

Lemma 1. *If Assumptions A1 and A2 hold, then for every fixed $a \in A$, the optimal Q-function $Q^*(s, a)$ is Lipschitz continuous with Lipschitz constant $L_Q := K_R + \frac{\beta K_p R_{max}}{1-\beta}$.*

Proof. Fix an $a \in A$ and write (1) as $Q^*(s, a) = R(s, a) + \beta \int_S V^*(s') p(s'|s, a) ds'$. By A1, it is easy to see that $V^*(s) \leq \frac{R_{max}}{1-\beta}$, $\forall\, s \in S$. Consequently, for any two states s, $s' \in S$, it follows from A1 and A2 that

$$\begin{aligned}
|Q^*(s, a) - Q^*(s', a)| &\leq |R(s, a) - R(s', a)| \\
&\quad + \beta \int_S V^*(s'') |p(s''|s, a) - p(s''|s', a)| ds'' \\
&\leq K_R d(s, s') + \frac{\beta R_{max}}{1-\beta} \int_S K_p(s'') d(s, s') ds'' \\
&= \Big(K_R + \frac{\beta K_p R_{max}}{1-\beta} \Big) d(s, s'),
\end{aligned}$$

which shows the Lipschitz continuity of $Q^*(s, a)$.

The following result implies that as the number of iterations $t \to \infty$, the shrinking ball neighborhoods of all state–action pairs sampled prior to time i_t will all be visited infinitely often (i.o).

Lemma 2. *If A2, A5, A6, and A7 hold, then for any state–action pair $(s_l, a_l) \in \Lambda_{i_t}$, $P(\lim_{t \to \infty} N_t(s_l, a_l) = \infty) = 1$.*

Proof. By A2, since S is compact and $p(s'|s, a)$ is continuous in both s' and s, from the extreme value theorem, $p(s'|s, a)$ attains its minimum for each $a \in A$. In addition, because $p(s'|s, a) > 0$ $\forall\, s, s' \in S$, $a \in A$ and A is finite, we must have that $\delta := \min_{a \in A} \inf_{s, s' \in S} p(s'|s, a) > 0$. It thus follows from A6 that $P(s_t \in B(s_l, r_t) | s_{t-1} = s, a) = \int_{B(s_l, r_t)} p(s'|s, a) ds' \geq \delta c_B t^{-\gamma_2 d}$ for some constant $c_B > 0$ when t is sufficiently large. On the other hand, by A7, there exist constants $c_\pi > 0$, $K > 0$ such that $\inf_s \pi_t(s, a) \geq \frac{c_\pi}{t^{\gamma_3}}$ for all $t \geq K$. Let $\rho_t > 0$ satisfy $\rho_t \leq (t - i_t) \delta c_B c_\pi t^{-(\gamma_2 d + \gamma_3)}$ and $I\{\cdot\}$ be the indicator function. We have for a sufficiently large t that

$$\begin{aligned}
&P(N_t(s_l, a_l) \leq \rho_t) \\
&= P\Big(\sum_{i=l+1}^{t} I\{s_i \in B(s_l, r_i) \cap a_i = a_l\} \leq \rho_t \Big) \\
&\leq P\Big(\sum_{i=i_t+1}^{t} I\{s_i \in B(s_l, r_i) \cap a_i = a_l\} \leq \rho_t \Big) \\
&= P\Big(\sum_{i=i_t+1}^{t} I\{s_i \notin B(s_l, r_i) \cup a_i \neq a_l\} \geq t - i_t - \rho_t \Big) \\
&= P\Big(e^{\lambda \sum_{i=i_t+1}^{t} I\{s_i \notin B(s_l, r_i) \cup a_i \neq a_l\}} \geq e^{\lambda(t - i_t - \rho_t)} \Big) \qquad (4)
\end{aligned}$$

for any given constant $\lambda > 0$, where the inequality above is due to the fact that $l \leq i_t$. Next, by applying Markov's inequality and noticing that the shrinking ball radius is non-increasing (A6), we obtain the following bound on (4):

$$\begin{aligned}
(4) &\leq e^{-\lambda(t - i_t - \rho_t)} E\Big[e^{\lambda \sum_{i=i_t+1}^{t} I\{s_i \notin B(s_l, r_i) \cup a_i \neq a_l\}} \Big] \\
&\leq e^{-\lambda(t - i_t - \rho_t)} E\Big[e^{\lambda \sum_{i=i_t+1}^{t} I\{s_i \notin B(s_l, r_t) \cup a_i \neq a_l\}} \Big]. \qquad (5)
\end{aligned}$$

Note that

$$
\begin{aligned}
&E\left[e^{\lambda I\{s_i\notin B(s_l,r_t)\cup a_i\neq a_l\}}|\tilde{\mathscr{F}}_{i-1}\right]\\
&=(e^\lambda-1)[1-P(s_i\in B(s_l,r_t)\cap a_i=a_l)|\tilde{\mathscr{F}}_{i-1}]+1\\
&=(e^\lambda-1)[1-P(a_i=a_l|s_i\in B(s_l,r_t),\tilde{\mathscr{F}}_{i-1})\\
&\quad\times P(s_i\in B(s_l,r_t)|\tilde{\mathscr{F}}_{i-1})]+1\\
&\le(e^\lambda-1)\left[1-\delta c_B c_\pi t^{-\gamma_2 d} i^{-\gamma_3}\right]+1, \qquad (6)
\end{aligned}
$$

where the inequality follows because $\inf_s \pi_i(s,a)\ge\frac{c_\pi}{i^{\gamma_3}}$ and $P(s_i\in B(s_l,r_t)|\tilde{\mathscr{F}}_{t-1})\ge\delta c_B t^{-\gamma_2 d}$. Thus, the expectation in (5) can be bounded through a repeated application of (6) as follows:

$$
\begin{aligned}
&E\left[e^{\lambda\sum_{i=i_t+1}^{t} I\{s_i\notin B(s_l,r_t)\cup a_i\neq a_l\}}\right]\\
&=E\left[E\left[e^{\lambda I\{s_t\notin B(s_l,r_t)\cup a_t\neq a_l\}}|\tilde{\mathscr{F}}_{t-1}\right]\right.\\
&\quad\left.\times e^{\lambda\sum_{i=i_t+1}^{t-1} I\{s_i\notin B(s_l,r_t)\cup a_i\neq a_l\}}\right]\\
&\le\left((e^\lambda-1)\left[1-\delta c_B c_\pi t^{-\gamma_2 d} t^{-\gamma_3}\right]+1\right)\\
&\quad\times E\left[e^{\lambda\sum_{i=i_t+1}^{t-1} I\{s_i\notin B(s_l,r_t)\cup a_i\neq a_l\}}\right]\\
&\cdots\\
&\le\prod_{i=i_t+1}^{t}\left[(e^\lambda-1)(1-\delta c_B c_\pi t^{-\gamma_2 d} i^{-\gamma_3})+1\right]\\
&=\exp\Big(\sum_{i=i_t+1}^{t}\ln\left[(e^\lambda-1)(1-\delta c_B c_\pi t^{-\gamma_2 d} i^{-\gamma_3})+1\right]\Big)\\
&\le\exp\Big((e^\lambda-1)\left[t-i_t-\delta c_B c_\pi t^{-\gamma_2 d}\sum_{i=i_t+1}^{t} i^{-\gamma_3}\right]\Big)\\
&\le\exp\big((e^\lambda-1)\left[t-i_t-\delta c_B c_\pi t^{-\gamma_2 d}(t-i_t)t^{-\gamma_3}\right]\big)\\
&=\exp\big((e^\lambda-1)(t-i_t)\left[1-\delta c_B c_\pi t^{-(\gamma_3+\gamma_2 d)}\right]\big),
\end{aligned}
$$

where the second last inequality follows from the fact that $\ln(1+x)\le x$ for $x\ge 0$. Substituting the above into (5) and optimizing the bound with respect to λ, we get $P(N_t(s_l,a_l)\le\rho_t)\le e^{\mathcal{B}_t\left(1-\frac{\mathcal{A}_t}{\mathcal{B}_t}+\ln\frac{\mathcal{A}_t}{\mathcal{B}_t}\right)}$, where we have defined $\mathcal{A}_t:=(t-i_t)\left[1-\delta c_B c_\pi t^{-(\gamma_3+\gamma_2 d)}\right]$ and $\mathcal{B}_t=t-i_t-\rho_t$. Since $\rho_t\le(t-i_t)\delta c_B c_\pi t^{-(\gamma_2 d+\gamma_3)}$, it is clear that $0<\frac{\mathcal{A}_t}{\mathcal{B}_t}\le 1$. Thus, by applying the inequality $\ln x\le(x-1)-\frac{1}{2}(x-1)^2$ for $x\in(0,1]$, we obtain $P(N_t(s_l,a_l)\le\rho_t)\le e^{-\frac{(\mathcal{B}_t-\mathcal{A}_t)^2}{2\mathcal{B}_t}}$. Next, setting $e^{-\frac{(\mathcal{B}_t-\mathcal{A}_t)^2}{2\mathcal{B}_t}}=\frac{1}{t^2}$ and solving for ρ_t, we get $\rho_t=(t-i_t)\delta c_B c_\pi t^{-(\gamma_2 d+\gamma_3)}-2\ln t-2\sqrt{\mathcal{A}_t\ln t+\ln^2 t}$. This shows that $\sum_{t=l+1}^{\infty}P(N_t(x_l,a_l)\le\rho_t)\le\sum_{t=l+1}^{\infty}\frac{1}{t^2}<\infty$, which indicates that $P(\{N_t(x_l,a_l)\le\rho_t\}\ i.o.)=0$ by applying Borel–Cantelli lemma. Finally, from A5, A6, and A7, it is not hard to observe that $\rho_t\to\infty$ as $t\to\infty$. This completes the proof of the lemma.

Next, we show that for each fixed $a\in A$, the collection of states in $\Lambda_{i_t}(a)$ visited up to time i_t will become dense in S as $t\to\infty$.

Lemma 3. *If Assumptions A2, A5, and A7 hold, then for every action $a\in A$, $P\left(\lim_{t\to\infty}\sup_{s\in S} d(s,\Lambda_{i_t}(a))=0\right)=1$.*

Proof. Let $\varepsilon>0$ be a positive constant. The set $\cup_{s\in S}B(s,\frac{\varepsilon}{2})$ forms an open cover of S. Since S is compact, there exists a finite collection of states $\{v_1,\dots,v_n\}$ such that $S\subseteq\cup_{j=1}^{n}B(v_j,\frac{\varepsilon}{2})$. By A2 and A7, we can find a finite $K>0$ and constants $c_B>0$ and $c_\pi>0$ such that $P(s_t\in B(v_j,\varepsilon/2)|s_{t-1}=s,a)\ge\delta c_B(\varepsilon/2)^d$ and $\inf_s\pi_t(s,a)\ge\frac{c_\pi}{t^{\gamma_3}}$ for all $t\ge K$, where $\delta=\min_{a\in A}\inf_{s,s'\in S}p(s'|s,a)$. It follows that when t is large,

$$
\begin{aligned}
P\left(\sup_s d(s,\Lambda_{i_t}(a))>\varepsilon\right)&=P\left(\exists s'\in S,\ d(s',\Lambda_{i_t}(a))>\varepsilon\right)\\
&=P\left(\exists s'\in S,\ B(s',\varepsilon)\cap\Lambda_{i_t}(a)=\emptyset\right)\\
&\le P\left(\exists j=1,\dots,n,\ B(v_j,\varepsilon/2)\cap\Lambda_{i_t}(a)=\emptyset\right)\\
&\le\sum_{j=1}^{n}P\Big(B(v_j,\varepsilon/2)\cap\Lambda_{i_t}(a)=\emptyset\Big)\\
&=\sum_{j=1}^{n}P\Big[\left(s_0\notin B(v_j,\varepsilon/2)\cup a_0\neq a\right)\cap\dots\cap\\
&\quad\left(s_{i_t}\notin B(v_j,\varepsilon/2)\cup a_{i_t}\neq a\right)\Big]\\
&=\sum_{j=1}^{n}\Big[1-P\left(s_{i_t}\in B(v_j,\varepsilon/2)\cap a_{i_t}=a\Big|s_{i_t-1}\right.\\
&\quad\left.\notin B(v_j,\varepsilon/2)\cup a_{i_t-1}\neq a,\dots\right)\Big]\\
&\quad\times P\Big(\left(s_{i_t-1}\notin B(v_j,\varepsilon/2)\cup a_{i_t-1}\neq a\right)\cap\dots\cap\\
&\quad\left(s_0\notin B(v_j,\varepsilon/2)\cup a_0\neq a\right)\Big)\\
&\le\sum_{j=1}^{n}\left[1-\delta c_B c_\pi(\varepsilon/2)^d i_t^{-\gamma_3}\right]\\
&\quad\times P\Big(\left(s_{i_t-1}\notin B(v_j,\varepsilon/2)\cup a_{i_t-1}\neq a\right)\cap\dots\cap\\
&\quad\left(s_0\notin B(v_j,\varepsilon/2)\cup a_0\neq a\right)\Big)\\
&\cdots\\
&\le\sum_{j=1}^{n}\prod_{i=K+1}^{i_t}\left[1-\delta c_B c_\pi(\varepsilon/2)^d i^{-\gamma_3}\right]\\
&=\sum_{j=1}^{n}\exp\Big(\sum_{i=K+1}^{i_t}\ln\left(1-\delta c_B c_\pi(\varepsilon/2)^d i^{-\gamma_3}\right)\Big)\\
&\le\sum_{j=1}^{n}\exp\left(-\delta c_B c_\pi(\varepsilon/2)^d\sum_{i=K+1}^{i_t} i^{-\gamma_3}\right)\\
&\le n\exp\left(-\delta c_B c_\pi(\varepsilon/2)^d(i_t-K)i_t^{-\gamma_3}\right).
\end{aligned}
$$

Since $i_t=\lfloor t^{\gamma_1}\rfloor$, $\gamma_1\in(0,1)$ (A5) and $0<\gamma_3<1/2$ (A7), it can be verified that $\sum_{t=1}^{\infty}P\left(\sup_s d(s,\Lambda_{i_t}(a))>\varepsilon\right)<\infty$. It thus follows from Borel–Cantelli lemma that $P\left(\{\sup_s d(s,\Lambda_{i_t}(a))>\varepsilon\}\ i.o.\right)=0$, and because ε is arbitrary, we must have $P\left(\lim_{t\to\infty}\sup_{s\in S} d(s,\Lambda_{i_t}(a))=0\right)=1$.

Lemma 4 below states that both the point estimate Q_t and the function approximator $\mathbb{Q}_t$ constructed at Steps 2 and 3 of the algorithm remain bounded at all time.

Lemma 4. *If A1 and A3 hold, then $\max_{(s,a)\in\Lambda_{t+1}}Q_t(s,a)$ and $\sup_{s\in S}\max_{a\in A}\mathbb{Q}_t(s,a)$ are bounded for all t w.p.1.*

Proof. Define $D_t=\max_{(s,a)\in\Lambda_{t+1}}|Q_t(s,a)|$, and let D be the diameter of S. It is clear from A3 that for any states $s,s'\in S$, w.p.1, $|\mathbb{Q}_t(s,a)|\le|\mathbb{Q}_t(s',a)|+LD$, where $L:=\max_{a\in A}L(a)<\infty$. In addition, since $\mathbb{Q}_t(s,a)$ interpolates $\left\{\left((s',a'),Q_{t-1}(s',a')\right):(s',a')\in\Lambda_{i_{t-1}},a'=a\right\}$ and $\max_{(s,a)\in\Lambda_{i_{t-1}}}|Q_{t-1}(s,a)|\le\max_{(s,a)\in\Lambda_t}|Q_{t-1}(s,a)|=D_{t-1}$, we have $\sup_{s\in S}\max_a|\mathbb{Q}_t(s,a)|\le D_{t-1}+LD$. Therefore, the point estimate obtained from (2) satisfies $|Q_t(s_t,a_t)|\le R_{max}+\beta(D_{t-1}+LD)$, and from (3), this further indicates that $|Q_t(s_l,a_l)|\le\max\left\{D_{t-1},R_{max}+\beta(D_{t-1}+LD)\right\}$ for all $(s_l,a_l)\in\Lambda_t$. Taking together, we have

$$
D_t\le\max\left\{D_{t-1},R_{max}+\beta(D_{t-1}+LD)\right\}. \qquad (7)
$$

Note that by construction, $\mathbb{Q}_0(s,a)=0$ for all $s\in S$ and $a\in A$. It is thus obvious from (2) that $D_0\le R_{max}\le\frac{R_{max}+\beta LD}{1-\beta}$, and a simple inductive argument using (7) shows that $D_t\le\frac{R_{max}+\beta LD}{1-\beta}$ for all t. Furthermore, we also obtain $\sup_{s\in S}\max_a|\mathbb{Q}_t(s,a)|\le D_{t-1}+LD\le\frac{R_{max}+\beta LD}{1-\beta}+LD=\frac{R_{max}+LD}{1-\beta}$ for all t.

Our main result is to show the uniform convergence of the sequence $\{\mathbb{Q}_t\}$ to the optimal Q-function. Since $\mathbb{Q}_t$ is constructed using estimates Q_{t-1} obtained for pairs in $\Lambda_{i_{t-1}}$, we proceed by studying the convergence properties of the iterates produced by (3). For notational convenience, we define $\epsilon_t(s_l,a_l)=Q_t(s_l,a_l)-Q^*(s_l,a_l)$ and let $\eta_t(s_l,a_l)=$

$\alpha_t(s_l,a_l)I_t(s_l,a_l)$. Then, subtracting both sides of (3) by $Q^*(s_l,a_l)$, we obtain the following recursion:

$$\begin{aligned}\epsilon_t(s_l,a_l) &= (1-\eta_t(s_l,a_l))\epsilon_{t-1}(s_l,a_l)+\eta_t(s_l,a_l)\\ &\quad\times\big[r(s_t,a_t,\omega_t)+\beta\max_b \mathbb{Q}_t(s_{t+1},b)-Q^*(s_l,a_l)\big]\\ &= (1-\eta_t(s_l,a_l))\epsilon_{t-1}(s_l,a_l)\\ &\quad+\eta_t(s_l,a_l)\big(B_t(s_l,a_l)+W_t(s_t,a_t)+H_t(s_{t+1})\big),\end{aligned}\tag{8}$$

where $B_t(s_l,a_l):=Q^*(s_t,a_t)-Q^*(s_l,a_l)$ is the bias caused by the use of the shrinking ball strategy, $W_t(s_t,a_t):=r(s_t,a_t,\omega_t)+\beta\max_b Q^*(s_{t+1},b)-Q^*(s_t,a_t)$ is a noise term, and $H_t(s_{t+1}) = \beta\max_b\mathbb{Q}_t(s_{t+1},b)-\beta\max_b Q^*(s_{t+1},b)$ is the approximation error of $\mathbb{Q}_t$. An expansion of (8) then yields

$$\epsilon_t(s_l,a_l)=U_\epsilon(t:l)+U_B(t:l)+U_W(t:l)+U_H(t:l),\tag{9}$$

where we have defined

$$U_\epsilon(t:l):=\Big[\prod_{i=l+1}^{t}(1-\eta_i(s_l,a_l))\Big]\epsilon_l(s_l,a_l),\tag{10}$$

$$U_B(t:l):=\sum_{i=l+1}^{t}\Big[\prod_{j=i+1}^{t}(1-\eta_j(s_l,a_l))\Big]\eta_i(s_l,a_l)B_i(s_l,a_l),\tag{11}$$

$$U_W(t:l):=\sum_{i=l+1}^{t}\Big[\prod_{j=i+1}^{t}(1-\eta_j(s_l,a_l))\Big]\eta_i(s_l,a_l)W_i(s_i,a_i),\tag{12}$$

$$U_H(t:l):=\sum_{i=l+1}^{t}\Big[\prod_{j=i+1}^{t}(1-\eta_j(s_l,a_l))\Big]\eta_i(s_l,a_l)H_i(s_{i+1}).\tag{13}$$

The convergence properties of terms (10)–(12) are given in Lemmas 5, 7, and 8, respectively.

As shown in the following lemma, the influence of the initial point estimation error $\epsilon_l(s_l,a_l)=Q_l(s_l,a_l)-Q^*(s_l,a_l)$ at the state–action pair (s_l,a_l) sampled at time l will become negligible as the number of iterations increases.

Lemma 5. *If conditions A1–A7 hold, then for any state–action pair $(s_l,a_l)\in\Lambda_{i_l}$, $U_\epsilon(t:l)\to 0$ as $t\to\infty$ w.p.1.*

Proof. From the proof of Lemma 4, it is obvious that $|\epsilon_l(s_l,a_l)|\le\frac{2R_{max}+\beta LD}{1-\beta}$. We have

$$\begin{aligned}|U_\epsilon(t:l)| &= \Big[\prod_{i=l+1}^{t}(1-\eta_i(s_l,a_l))\Big]|\epsilon_l(s_l,a_l)|\\ &= \exp\Big(\sum_{i=l+1}^{t}\ln(1-\eta_i(s_l,a_l))\Big)|\epsilon_l(s_l,a_l)|\\ &\le \exp\Big(-\sum_{i=l+1}^{t}\eta_i(s_l,a_l)\Big)\frac{2R_{max}+\beta LD}{1-\beta}.\end{aligned}$$

By the definition of $\eta_i(s_l,a_l)$, we obtain $\sum_{i=l+1}^{t}\eta_i(s_l,a_l)=\sum_{i=l+1}^{t}\alpha_i(s_l,a_l)I_i(s_l,a_l)=\sum_{i=l+1}^{t}f(N_i(s_l,a_l))I_i(s_l,a_l)=\sum_{j=1}^{N_t(s_l,a_l)}f(j)$. We know from Lemma 2 that $N_i(s_l,a_l)\to\infty$ w.p.1. This, together with the condition $\sum_{j=1}^{\infty}f(j)=\infty$ (A4), implies that $|U_\epsilon(t:l)|\to 0$ as $t\to\infty$ w.p.1.

The analysis of the terms (11) and (12) relies on the following intermediate result, whose proof follows from a straightforward inductive argument and is hence omitted.

Lemma 6. *For a given integer $l>0$, $\sum_{i=l+1}^{t}\big[\prod_{j=i+1}^{t}(1-\eta_j(s,a))\big]\eta_i(s,a)\le 1$ for all $t>l$.*

Next regarding term (11), we have the following result, indicating that as the sequence of shrinking ball radiuses decreases, the cumulative effect of the estimation bias at a sampled pair arising from averaging points within its neighborhoods vanishes.

Lemma 7. *If conditions A1–A7 hold, then for any state–action pair $(s_l,a_l)\in\Lambda_{i_l}$, $U_B(t:l)\to 0$ as $t\to\infty$ w.p.1.*

Proof. By Lemma 1, we have

$$\begin{aligned}I_i(s_l,a_l)|B_i(s_l,a_l)| &= I_i(s_l,a_l)|Q^*(s_i,a_l)-Q^*(s_l,a_l)|\\ &\le L_Q d(s_i,s_l)I_i(s_l,a_l)\le L_Q r_i,\end{aligned}$$

which tends to zero as $i\to\infty$ by condition A5. Therefore, for any $\varepsilon>0$, there exists some $N>0$ such that $I_i(s_l,a_l)|B_i(s_l,a_l)|\le\varepsilon/2$ for all $i>N$. We thus obtain for a sufficiently large t that w.p.1,

$$\begin{aligned}|U_B(t:l)| &= \sum_{i=l+1}^{t}\Big[\prod_{j=i+1}^{t}(1-\eta_j(s_l,a_l))\Big]\eta_i(s_l,a_l)|B_i(s_l,a_l)|\\ &= \sum_{i=l+1}^{N}\Big[\prod_{j=i+1}^{t}(1-\eta_j(s_l,a_l))\Big]\eta_i(s_l,a_l)|B_i(s_l,a_l)|\\ &\quad+\sum_{i=N+1}^{t}\Big[\prod_{j=i+1}^{t}(1-\eta_j(s_l,a_l))\Big]\eta_i(s_l,a_l)|B_i(s_l,a_l)|\\ &\le \sum_{i=l+1}^{N}\Big[\prod_{j=i+1}^{t}(1-\eta_j(s_l,a_l))\Big]\eta_i(s_l,a_l)|B_i(s_l,a_l)|\\ &\quad+\frac{\varepsilon}{2}\sum_{i=N+1}^{t}\Big[\prod_{j=i+1}^{t}(1-\eta_j(s_l,a_l))\Big]\eta_i(s_l,a_l)\\ &\le \sum_{i=l+1}^{N}\Big[\prod_{j=i+1}^{t}(1-\eta_j(s_l,a_l))\Big]\eta_i(s_l,a_l)|B_i(s_l,a_l)|+\frac{\varepsilon}{2}\\ &\le \sum_{i=l+1}^{N}\Big[\prod_{j=i+1}^{t}(1-\eta_j(s_l,a_l))\Big]|B_i(s_l,a_l)|+\frac{\varepsilon}{2},\end{aligned}\tag{14}$$

where the second inequality follows from Lemma 6 and the last step follows because $0\le\eta_i(s_l,a_l)<1$. Next, using the inequality $\prod_{j=i+1}^{t}(1-x_j)\le e^{-\sum_{j=i+1}^{t}x_j}$ for all $x_j\in[0,1)$ and noting that $|B_i(s_l,a_l)|=|Q^*(s_i,a_l)-Q^*(s_l,a_l)|\le\frac{2R_{max}}{1-\beta}$, it can be seen that (14) is bounded by

$$\begin{aligned}(14) &\le \sum_{i=l+1}^{N}\exp\Big(-\sum_{j=i+1}^{t}\eta_j(s_l,a_l)\Big)|B_i(s_l,a_l)|+\frac{\varepsilon}{2}\\ &\le (N-l)\frac{2R_{max}}{1-\beta}e^{-\sum_{j=N+1}^{t}\eta_j(s_l,a_l)}+\frac{\varepsilon}{2}.\end{aligned}$$

Since $\sum_{j=N+1}^{t}\eta_j(s_l,a_l)=\sum_{j=N+1}^{t}\alpha_j(s_l,a_l)I_j(s_l,a_l)=\sum_{j=N+1}^{t}f(N_j(s_l,a_l))I_j(s_l,a_l)$, we know from Lemma 2 and condition A4 that $\sum_{j=N+1}^{t}\eta_j(s_l,a_l)\to\infty$ as $t\to\infty$. This in turn implies that $(N-l)\frac{2R_{max}}{1-\beta}e^{-\sum_{j=N+1}^{t}\eta_j(s_l,a_l)}$ can be made smaller than $\varepsilon/2$ for t sufficiently large. Finally, because ε is arbitrary, we have $U_B(t:l)\to 0$ as $t\to\infty$ w.p.1.

On the other hand, because the shrinking ball neighborhoods of each sample state–action pair are visited i.o. as $t\to\infty$ (see Lemma 2), the estimation noise will be averaged out over the course of the iterations. This intuition is formalized in the result below.

Lemma 8. *If Assumptions A1–A7 hold, then for every $(s_l,a_l)\in\Lambda_{i_l}$, $U_W(t:l)\to 0$ as $t\to\infty$ w.p.1.*

Proof. Recall that $W_t(s_t,a_t)=r(s_t,a_t,\omega_t)+\beta\max_b Q^*(s_{t+1},b)-Q^*(s_t,a_t)$. We consider the sequence $M_t:=\sum_{i=l+1}^{t}\eta_i(s_l,a_l)W_i(s_i,a_i)$. Since $\eta_t(s_l,a_l)$ is $\mathscr{F}_t$-measurable and $E[W_t(s_t,a_t)|\mathscr{F}_t]=0$, we have

$$\begin{aligned}E[M_t|\mathscr{F}_t] &= \sum_{i=l+1}^{t-1}\eta_i(s_l,a_l)W_i(s_i,a_i)\\ &\quad+\eta_t(s_l,a_l)E[W_t(s_t,a_t)|\mathscr{F}_t]\\ &= M_{t-1}.\end{aligned}$$

In addition, $E[M_t^2]=E\big[\big(\sum_{i=l+1}^{t}\eta_i(s_l,a_l)W_i(s_i,a_i)\big)^2\big]=E\big[\sum_{i=l+1}^{t}\eta_i^2(s_l,a_l)W_i^2(s_i,a_i)\big]$ because the cross terms $E[\eta_i(s_l,a_l)\eta_j(s_l,a_l)$

$W_i(s_i,a_i)W_j(s_j,a_j)] = E[\eta_i(s_l,a_l)\eta_j(s_l,a_l)W_i(s_i,a_i)E[W_j(s_j,a_j)|\mathcal{F}_j]] = 0$, $\forall\, i < j$. Under A1, we have $|W_t(s,a)| \le R_{max} + (\beta+1)\frac{R_{max}}{1-\beta} = \frac{2R_{max}}{1-\beta}$. It follows that $E[M_t^2] \le (\frac{2R_{max}}{1-\beta})^2 E[\sum_{i=l+1}^{t}\eta_i^2(s_l,a_l)] = (\frac{2R_{max}}{1-\beta})^2 E[\sum_{j=1}^{N_t(s_l,a_l)} f^2(j)] < \infty$ due to the condition $\sum_{i=1}^{\infty} f^2(i) < \infty$ (A4). Consequently, $\{M_t\}$ is an L^2-bounded martingale and converges to a finite random variable M_∞ w.p.1. Next, we note that

$$
\begin{aligned}
U_W(t:l) &= \sum_{i=l+1}^{t}\Big[\prod_{j=i+1}^{t}(1-\eta_j(s_l,a_l))\Big]\eta_i(s_l,a_l)W_i(s_i,a_i)\\
&= \prod_{j=l+1}^{t}(1-\eta_j(s_l,a_l))\\
&\quad\times\sum_{i=l+1}^{t}\frac{1}{\prod_{j=l+1}^{i}(1-\eta_j(s_l,a_l))}\eta_i(s_l,a_l)W_i(s_i,a_i)\\
&= \frac{1}{b_t}\sum_{i=l+1}^{t} b_i\eta_i(s_l,a_l)W_i(s_i,a_i),
\end{aligned}
$$

where $b_i := \frac{1}{\prod_{j=l+1}^{i}(1-\eta_j(s_l,a_l))}$. Clearly, $0 < b_i \le b_{i+1}$ and due to Lemma 2, $b_t \to \infty$ as $t \to \infty$ w.p.1. Using the fact that $M_t \to M_\infty$ w.p.1 and applying Kronecker's lemma (e.g., [23]) in a path-wise manner, we finally obtain $U_W(t:l) \to 0$ w.p.1 as required.

Finally, we arrive at the following main convergence theorem.

Theorem 1. *Suppose all conditions A1–A7 are satisfied. Then*

$$\lim_{t\to\infty}\sup_{s\in S}\max_{a\in A}|\mathbb{Q}_t(s,a) - Q^*(s,a)| = 0 \quad \textit{w.p.1.}$$

Proof. In the proof of Lemma 4, we have shown that $\sup_{s\in S}\max_a|\mathbb{Q}_t(s,a)| \le \frac{R_{max}+LD}{1-\beta}$. Thus, it is easy to see that $\sup_{s\in S}\max_{a\in A}|\mathbb{Q}_t(s,a) - Q^*(s,a)| \le \frac{2R_{max}+LD}{1-\beta}$ for all $t \ge 0$.

Let $\xi \in (0, 1-\beta)$ be a given constant. We proceed by using an idea similar to that of [24]. In particular, suppose there exist a constant G and time $\tau_k \ge 0$ (with $\tau_0 := 0$) satisfying $\sup_{s\in S}\max_a|\mathbb{Q}_t(s,a) - Q^*(s,a)| \le G$ for all $t \ge \tau_k$. Then we show there must be another time $\tau_{k+1} \ge \tau_k$ such that $\sup_{s\in S}\max_a|\mathbb{Q}_t(s,a) - Q^*(s,a)| \le (\beta+\xi)G$ for all $t \ge \tau_{k+1}$. Since $\beta+\xi < 1$, this guarantees the convergence of $\sup_{s\in S}\max_{a\in A}|\mathbb{Q}_t(s,a) - Q^*(s,a)|$ to zero.

Let $r = \frac{\xi G}{4(L+L_Q)}$ and define $\Omega_a = \{\lim_{t\to\infty}\sup_s d(s,\Lambda_{i_t}(a)) = 0\}$, $a \in A$. For every $\omega \in \cap_{a\in A}\Omega_a$, there exists some τ' such that $S \subseteq \cup_{s\in\Lambda_{i_{\tau'}}(a)}B(s,r)$ for all $a \in A$. Take $\tau = \max\{\tau',\tau_k\}$. Clearly $S \subseteq \cup_{s\in\Lambda_{i_\tau}(a)}B(s,r)$ for all $a \in A$.

For each state–action pair $(s,a) \in \Lambda_{i_\tau}$, we let $\epsilon_t(s,a) = Q_t(s,a) - Q^*(s,a)$ and consider the recursion

$$
\begin{aligned}
\epsilon_t(s,a) &= (1-\eta_t(s,a))\epsilon_{t-1}(s,a) + \eta_t(s,a)\big[r(s_t,a_t,\omega_t)\\
&\qquad + \beta\max_b\mathbb{Q}_t(s_{t+1},b) - Q^*(s,a)\big],\ \forall\, t \ge \tau+1.
\end{aligned}
$$

As in (9), this can be written as

$$\epsilon_t(s,a) = U_\epsilon(t:\tau) + U_B(t:\tau) + U_W(t:\tau) + U_H(t:\tau),$$

where $U_\epsilon(t:\tau)$, $U_B(t:\tau)$, $U_W(t:\tau)$, and $U_H(t:\tau)$ are defined respectively in the same way as in (10), (11), (12), and (13) with (s,a) replacing (s_l,a_l).

Note that by our hypothesis for all $i > \tau \ge \tau_k$,

$$
\begin{aligned}
|H_i(s_{i+1})| &\le \beta\max_b|Q_i(s_{i+1},b) - Q^*(s_{i+1},b)|\\
&\le \beta\sup_{s\in S}\max_b|Q_i(s,b) - Q^*(s,b)| \le \beta G.
\end{aligned}
$$

In addition, since $\sum_{i=\tau+1}^{t}\big[\prod_{j=i+1}^{t}(1-\eta_j(s,a))\big]\eta_i(s,a) \le 1$ for all $t \ge \tau+1$ by Lemma 6, we have $|U_H(t:\tau)| = \sum_{i=\tau+1}^{t}\big[\prod_{j=i+1}^{t}(1-\eta_j(s,a))\big]\eta_i(s,a)|H_i(s_{i+1})| \le \beta G$. Consequently, $|\epsilon_t(s,a)| \le |U_\epsilon(t:\tau)| + |U_B(t:\tau)| + |U_W(t:\tau)| + \beta G$, $\forall\, t \ge \tau+1$. We further let $\Omega_\epsilon := \{U_\epsilon(t:\tau) \to 0\}$, $\Omega_B := \{U_B(t:\tau) \to 0\}$, and $\Omega_W := \{U_W(t:\tau) \to 0\}$. Since the number of state–action pairs in Λ_{i_τ} is finite, on each sample path $\omega \in \cap_{a\in A}\Omega_a \cap \Omega_\epsilon \cap \Omega_B \cap \Omega_W$, there exists a $\tau_{k+1} > \tau$ such that $|\epsilon_t(s,a)| \le \frac{\xi}{4}G + \frac{\xi}{4}G + \frac{\xi}{4}G + \beta G = (\beta+\frac{3}{4}\xi)G$, $\forall\,(s,a) \in \Lambda_{i_\tau}$, $\forall\, t \ge \tau_{k+1}$.

Next, for any $s \in S$ and $a \in A$, let $s_a = \arg\min_{s'\in\Lambda_{i_\tau}(a)} d(s,s')$. Because $S \subseteq \cup_{s'\in\Lambda_{i_\tau}(a)}B(s',r)$, we have $d(s,s_a) \le r$. It thus follows that for all $t > \tau_{k+1}$,

$$
\begin{aligned}
|\mathbb{Q}_t(s,a) - Q^*(s,a)| &\le |\mathbb{Q}_t(s,a) - \mathbb{Q}_t(s_a,a)|\\
&\quad + |\mathbb{Q}_t(s_a,a) - Q^*(s_a,a)| + |Q^*(s_a,a) - Q^*(s,a)|\\
&\le (L+L_Q)d(s,s_a) + |Q_{t-1}(s_a,a) - Q^*(s_a,a)|\\
&\le (L+L_Q)\frac{\xi G}{4(L+L_Q)} + (\beta+\frac{3}{4}\xi)G\\
&= (\beta+\xi)G,
\end{aligned}
$$

where in the second step above we have used the fact that $\mathbb{Q}_t(s_a,a) = Q_{t-1}(s_a,a)$ because $\mathbb{Q}_t$ is constructed by interpolating $\big\{\big((s',a'),Q_{t-1}(s',a')\big) : (s',a') \in \Lambda_{i_t},\ a' = a\big\}$, which contains $\big((s_a,a),Q_{t-1}(s_a,a)\big)$. In view of Lemmas 3, 5, 7, and 8, we have that $P(\cap_{a\in A}\Omega_a \cap \Omega_\epsilon \cap \Omega_B \cap \Omega_W) = 1$. This leads us to conclude that $\sup_{s\in S}\max_a|\mathbb{Q}_t(s,a) - Q^*(s,a)| \le (\beta+\xi)G$ for all $t > \tau_{k+1}$, w.p.1.

4. An illustrative example

We consider a four-dimensional inventory control problem with lost sales and zero order lead time. There are four types of commodities stored separately in four warehouses with respective capacities $\mathcal{L}_i$, $i = 1,2,3,4$. At each time $t = 0,1,\ldots$, the four inventory levels $(s_t^1,s_t^2,s_t^3,s_t^4)$ are reviewed, a decision $a_t \in \{0,1,2,3,4\}$ is then made whether to replenish inventory i (0 means "do nothing"), and the demands d_t^1, d_t^2, d_t^3 and d_t^4 for the four commodities are realized. For the ith commodity, let c_i be the per unit order cost, h_i be the per period per unit inventory holding cost, and p_i be the per period per unit penalty cost for unsatisfied demands. The transition functions for the inventory levels are given by $s_{t+1}^i = \big(s_t^i + (\mathcal{L}_i - s_t^i)I\{a_t = i\} - d_t^i\big)^+$, where $x^+ = \max\{x,0\}$. The goal is to minimize the expectation of the total discounted costs, which comprise order, holding, and penalty costs, i.e., $\sum_{t=0}^{\infty}\beta^t\big[\sum_{i=1}^{4}c_i(\mathcal{L}_i - s_t^i)I\{a_t = i\} + h_i\big(s_t^i + (\mathcal{L}_i - s_t^i)I\{a_t = i\} - d_t^i\big)^+ + p_i\big(d_t^i - s_t^i - (\mathcal{L}_i - s_t^i)I\{a_t = i\}\big)^+\big]$ for all initial inventory levels $(s_0^1,s_0^2,s_0^3,s_0^4)$. In our computational experiments, we set $\mathcal{L}_1 = \mathcal{L}_2 = \mathcal{L}_3 = \mathcal{L}_4 = 1$, $c_1 = 0.5, h_1 = 1.5, p_1 = 1.5, c_2 = 1, h_2 = 1, p_2 = 2, c_3 = 0.2, h_3 = 0.5, p_3 = 1, c_4 = 0.4, h_4 = 0.1, p_4 = 0.5, \beta = 0.9$. The demands d_t^1, d_t^2, d_t^3 and d_t^4 are assumed to be i.i.d. uniformly distributed between 0 and 1.

The proposed Q-learning (QL) algorithm is implemented with the following parameter values: $\gamma_1 = 0.98$, learning rate $\alpha_t(s_l) = N_t(s_l)^{-0.501}$, shrinking ball radius $r_t = \ln(100)/\ln(100+t)$. The function approximator is constructed using the stochastic kriging method with a Matérn kernel (see, e.g., [25,26]), and the learning policy is taken to be an ϵ-greedy policy with $\epsilon = 0.1$ (i.e., choose the action that minimizes the current $\mathbb{Q}_t$ with probability $1-\epsilon$ and select a random action with probability ϵ). The initial state is taken to be $(1,1,1,1)$.

In addition to QL, we have also applied four other methods: a discretization-based value iteration (DVI) algorithm, a non-parametric version of the fitted Q-iteration (FQI) algorithm presented in Chapter 3.4.3 of [8], the soft-state aggregation method of [17] (QL-SSA), and the nearest-neighbor Q-learning method (QL-NN) proposed in [20]. DVI is just the standard VI applied to a discrete version of the problem, obtained by discretizing the state space using a grid size of 0.1 along each dimension and then replacing the demand distributions with discrete uniform distributions over $\{0.1k : k = 0,1,\ldots,10\}$. This results in a discrete-state MDP, whose transition probabilities can be computed from the state transition functions. FQI requires transition samples to be generated and stored beforehand. In our implementation, we have used a set of 512 states, which are selected by using the Sobol sequence on the four-dimensional state space (cf., e.g., Chapter 5 of [27]). Each of the 512 states is repeatedly simulated 150 times (with 30 transition samples per action) and the updated Q-value at each state–action pair

Table 1
Weighted relative errors of comparison algorithms, means and standard errors (in parentheses) based on 30 independent replications.

QL	FQI	QL-SSA	QL-NN
1.71% (7.8e−4)	4.93% (1.3e−3)	8.02% (5.1e−4)	3.80% (1.5e−3)

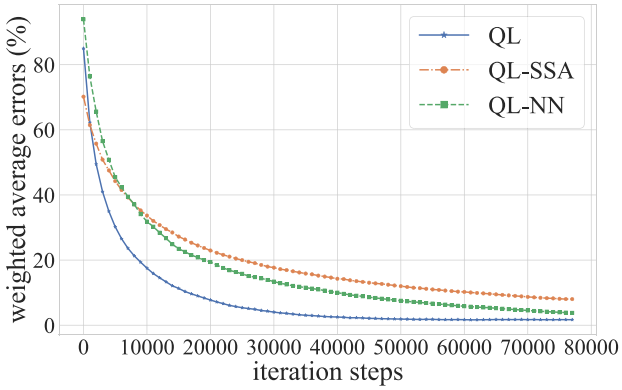


Fig. 1. Weighted Relative Errors of QL, QL-SSA, and QL-NN.

is obtained as the average of the 30 samples to reduce uncertainty. These Q-values are then used in the stochastic kriging method with a Matérn kernel to fit an approximation of the Q-function. The number of iterations for FQI is set to 100. Unlike FQI, QL-SSA is an asynchronous method that uses the transition samples generated from a learning policy to iteratively estimate the Q-function values at a given set of clusters (aggregate states). In the experiments, those clusters are taken to be the set of 512 states used in FQI, and each encountered state s belongs to the ith cluster with probability $P_{SSA}(i|s) = \frac{\exp(-\|s-i\|^2/0.001)}{\sum_j \exp(-\|s-j\|^2/0.001)}$. The Q-function estimator is then constructed in the form of a weighted sum $\sum_i P_{SSA}(i|s)\hat{Q}(i,a)$ for all (s,a), where $\hat{Q}(i,a)$ is an estimate of the Q-value at each cluster-action pair. The implementation of QL-NN is based on the same set of 512 discretized states. QL-NN differs from QL-SSA in that it updates the Q-values at the clusters in a roughly synchronous manner (i.e., after all neighborhoods of the discretized states are visited). The Q-values at neighboring states are then estimated at each step using a weighted sum, where the weighting function is taken to be a (truncated) Gaussian-type kernel $P_{NN}(i|s) = \frac{\exp(-\|s-i\|^2/2h^2)I\{\|s-i\|\le h\}}{\sum_j \exp(-\|s-j\|^2/2h^2)I\{\|s-j\|\le h\}}$ (see Appendix C in [20]) with the bandwidth parameter h set to $h = 1.7$ in our experiments. The learning policy and all other parameters in QL-SSA and QL-NN are taken to be the same as in QL, and to allow for a fair comparison with FQI, the numbers of iterations of QL, QL-SSA, and QL-NN are all set to 512×150 = 76800, which corresponds to the number of transition samples used by FQI.

Note that since a fine discretization is used, the solution returned by DVI provides a close approximation to the optimal value function, and hence can be used as a benchmark to gauge the performance of other comparison algorithms. In particular, we use the performance measure $\sum_{s\in S_D} \mu^*(s)\frac{|V(s)-V_D^*(s)|}{|V_D^*(s)|}$ to signify the weighted relative error of a value function V, where S_D and V_D^* are the state space and the optimal value function of the discrete-state MDP solved by DVI, and μ^* is the steady state distribution of the chain under the optimal policy. Table 1 shows the weighted relative errors of the value function approximations obtained by the four comparison algorithms upon termination. In Fig. 1, we also plotted the weighted relative errors (averaged over 30 runs) of all online methods (QL, QL-SSA, and QL-NN) as a function of the number of algorithm iterations (note that FQI is a one-shot approach that computes the value function estimate using the available samples all at once).

From the results, we see that QL yields the smallest weighted relative error and significantly outperforms QL-SSA and QL-NN. Compared with QL and FQI, the benefits of QL-SSA and QL-NN lie in their computation and memory efficiencies, as they work with a constant number of aggregate states at each step and do not require storage of the transition/historical data needed by QL and FQI. However, the use of the weighted average in the Q-function approximator may entail a large estimation bias, resulting in slow convergence. The performances of QL-SSA and QL-NN may be improved by finding good clustering probabilities tailored to the problem (e.g., using the adaptive procedure outlined in [17]) and/or fine-tuning the value of the bandwidth parameter h.

5. Conclusions

In this paper, we have proposed a Q-learning algorithm for solving continuous-state MDPs in a model-free setting. The algorithm uses a function approximator, in lieu of a tabular representation, to interpolate the historical data collected from a given learning policy. Unlike existing methods, the algorithm does not require the approximator to be a non-expansion and hence allows the use of more flexible function approximation tools. Another feature of the algorithm is that it employs an asynchronous averaging technique, which enables the construction of Q-value estimates to be conducted along a single sample trajectory. This further distinguishes the algorithm from many other approaches studied in the literature, which often resort to some forms of state-space discretization. We have analyzed the algorithm and shown the strong uniform convergence of the sequence of function approximators to the optimal Q-function. A simple inventory control example has also been provided to numerically illustrate the algorithm. An important line of future research will be to carry out a finite-time performance analysis, e.g., developing probability bounds along the lines of [20], to gain some insight into the computational complexity of the algorithm in relation to the problem size.

CRediT authorship contribution statement

Jiaqiao Hu: Conceptualization, Methodology, Writing – original draft. **Xiangyu Yang:** Software, Validation, Visualization, Writing – review & editing. **Jian-Qiang Hu:** Methodology, Resources, Writing – review & editing. **Yijie Peng:** Conceptualization, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] D.P. Bertsekas, J.N. Tsitsiklis, Neuro-Dynamic Programming, Athena Scientific, Belmont, MA, 1996.
[2] H. Chang, J. Hu, M. Fu, S. Marcus, Simulation-based algorithms for Markov decision processes, second ed., Springer, NY, 2013.
[3] A. Gosavi, Reinforcement learning: A tutorial survey and recent advances, INFORMS J. Comput. 21 (2) (2009) 178–192.
[4] W.B. Powell, Approximate Dynamic Programming: Solving the curses of dimensionality, Wiley-Interscience, NJ, USA, 2007.
[5] R. Sutton, A. Barto, Reinforcement Learning:An Introduction, MIT Press, 1998.
[6] C.J.C.H. Watkins, Learning from Delayed Rewards (Phd Thesis), Cambridge University, 1989.
[7] S. Bhatnagar, K.M. Babu, New algorithms of the Q-learning type, Automatica 44 (4) (2008) 1111–1119.

[8] L. Busoniu, R. Babuska, B. De Schutter, D. Ernst, Reinforcement Learning and Dynamic Programming using Function Approximators, vol. 39, CRC Press, 2010.
[9] J. Hu, H.S. Chang, Approximate stochastic annealing for online control of infinite horizon Markov decision processes, Automatica 48 (9) (2012) 2182—2188.
[10] J. Rust, Chapter 51 structural estimation of markov decision processes, in: Handbook of Econometrics, vol. 4, Elsevier, 1994, pp. 3081–3143.
[11] S. Baumert, R.L. Smith, Pure Random Search for Noisy Objective Functions, Technical Report 01-03, University of Michigan, 2002.
[12] A. Antos, R. Munos, C. Szepesvári, Fitted Q-iteration in continuous action-space MDPs, in: Advances in Neural Information Processing Systems, MIT Press, 2008, pp. 9–16.
[13] D. Ernst, P. Geurts, L. Wehenkel, Tree-based batch mode reinforcement learning, J. Mach. Learn. Res. 6 (2005) 503–556.
[14] G.J. Gordon, Stable function approximation in dynamic programming, in: Proceedings of the 20th International Conference on Machine Learning, 1995, pp. 261–268.
[15] T. Horiuchi, Fuzzy interpolation-based Q-learning with continuous states and actions, in: Proceedings of IEEE International Conference on Fuzzy Systems, 1996.
[16] M. Riedmiller, Neural fitted Q-iteration—first experiences with a data efficient neural reinforcement learning method, in: Proceedings 16th European Conference on Machine Learning, 2005, pp. 317–328.
[17] S. Singh, T. Jaakkola, M. Jordan, Reinforcement learning with soft state aggregation, in: G. Tesauro, D. Touretzky, T. Leen (Eds.), in: Advances in Neural Information Processing Systems, vol. 7, MIT Press, 1995, pp. 361–368.
[18] C. Szepesvári, W.D. Smart, Interpolation-based Q-learning, in: Proceedings of the 21st International Conference on Machine Learning, 2004, pp. 791–798.
[19] F.S. Melo, M.I. Ribeiro, Convergence of Q-learning with linear function approximation, in: 2007 European Control Conference, ECC, 2007, pp. 2671–2678.
[20] D. Shah, Q. Xie, Q-learning with nearest neighbors, in: Advances in Neural Information Processing Systems, Curran Associates, Inc., 2018, pp. 3111–3121.
[21] S. Singh, T. Jaakkola, M.L. Littman, Convergence results for single-step on-policy reinforcement-learning algorithms, Mach. Learn. 38 (3) (2000) p.287–308.
[22] S.P. Meyn, R.L. Tweedie, Markov Chains and Stochastic Stability, Springer-Verlag, London, 1993.
[23] A.N. Shiryaev, Probability, second ed., Springer-Verlag, New York, USA, 1996.
[24] J.N. Tsitsiklis, Asynchronous stochastic approximation and Q-learning, Mach. Learn. 16 (1994) 185–202.
[25] M.L. Stein, Interpolation of Spatial Data: Some Theory for Kriging, Springer Science & Business Media, 1999.
[26] B. Ankenman, B.L. Nelson, J. Staum, Stochastic kriging for simulation metamodeling, Oper. Res. 58 (2) (2010) 371–382.
[27] P. Glasserman, Monte Carlo methods in financial engineering, vol. 53, Springer, 2004.