2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

Distributed Multi-Agent Gradient Based Q-Learning with Linear Function Approximation

Miloš S. Stanković, Marko Beko and Srdjan S. Stanković

Abstract— **In this paper we propose a novel distributed gradient-based two-time-scale algorithm for multi-agent off-policy learning of linear approximation of the optimal action-value function (Q-function) in Markov decision processes (MDPs). The algorithm is composed of: 1) local parameter updates based on an off-policy gradient temporal difference learning algorithm with target policy belonging to either the greedy or the Gibbs distribution class and stationary behavior policies possibly different for each agent, and 2) a linear stochastic time-varying consensus scheme. It is proved, under general assumptions, that the parameter estimates generated by the proposed algorithm weakly converge to a bounded invariant set of the corresponding ordinary differential equation (ODE). Simulation results illustrate effectiveness of the proposed algorithm.**

I. Introduction

Reinforcement learning (RL) provides a widely accepted framework for *decision making* in unknown and stochastic environments, e.g. [1], [2]. Numerous undoubtedly successful solutions to practical problems, ranging from robotics to board games, have been reported, e.g., [3]. RL problems are mostly formulated using Markov Decision Processes (MDPs) with unknown transition probabilities. The goal is to find an *optimal policy* so that the total discounted future reward is maximized. One of the most important contributions to the RL field is the *temporal-difference* (TD) learning, typically used to approximate the *value function* of a given MDP [1], [4]. It is often desirable to evaluate a given *target policy* by implementing different *behavior policies* (*off-policy learning*, e.g., [5]–[7]). Among other methods, *Q-learning* has been recognized as a promising tool for finding the optimal policy in RL problems [1], [8]. It provides estimates of the optimal *Q-function* (*action-value function*), wherefrom the *optimal policy* itself can be simply computed. The Q-learning algorithm is basically an *off-policy method*, since it learns the optimal policy using data generated by arbitrary non-optimal *behavior policies*. Q-learning methodology has been extensively studied in the literature, e.g. [9]–[13]. However, applications have remained limited to the problems with relatively small state and action spaces. In order to overcome this, Q-learning with *function approximation* has been treated in numerous papers, e.g., [10]–[12], [14], [15]. However, there is still a gap between theory and practice, due possible divergence in off-policy learning involving function approximation and bootstrapping [1], [4], [13]–[15].

The focus of this paper is on *distributed multi-agent Q-learning based on linear function approximation*, following the approach to the single agent problems in [11]. In general, distributed, decentralized and multi-agent RL methods have attracted recently a lot of attention due to their high potential for solving diverse problems within complex, intelligent and networked systems (see e.g., [16] and numerous references therein). The problem of distributed multi-agent state-value function approximation has attracted great attention, e.g., [17]–[21], often within the *actor-critic* algorithms, e.g., [22]–[25]. However, to the authors knowledge, multi-agent distributed Q-learning *based on linear function approximation* has not yet been treated in the literature. Our main motivation has been, in general, to provide: a) a new tool for efficient *collaborative exploration* of possibly large state-action spaces with *provable convergence under fairly general conditions* and b) *variance reduction* owing to the collaborative function implemented by the consensus scheme. A specific system topology has been adopted, in which each agent can observe transitions of a given MDP independently, using a carefully chosen local behavior policy (Strict Information Structure Constraint (SISC) [21]). The main line of thought can be considered as an extension to distributed Q-learning of the approaches based on consensus from [17], [19]–[21], [25], [26].

We propose in this paper a new algorithm for distributed multi-agent off-policy gradient temporal difference learning of linear approximation to the optimal Q-function using linear dynamic *consensus iterations*. The target policy is assumed to belong to either *the greedy or the Gibbs class*, and the behavior policies are assumed to be stationary and different for each agent. In this way, the proposed algorithm becomes a learning tool for distributed off-policy control (not only for policy evaluation), similar to *A3C* or *A2C* [3], [11]. Assuming a general stochastic time-varying dynamic consensus scheme and practically mild assumptions, a *proof of the weak convergence of parameter estimates to consensus* is provided, based on appropriately defined ordinary differential equations (ODE's) [7], [27]–[29]. The proof is based on the properties of distributed stochastic approximation introduced in [27] and the arguments related to stability from [11], [30].

M. S. Stanković is with Singidunum University, Belgrade, Serbia; and COPELABS, Universidade Lusófona, Lisboa, Portugal; e-mail: milstank@gmail.com

M. Beko is with Instituto de Telecomunicações, Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal; and COPELABS, Universidade Lusófona, Lisboa, Portugal; e-mail: beko.marko@gmail.com

S. S. Stanković is with School of Electrical Engineering, University of Belgrade, Serbia; e-mail: stankovic@etf.rs

This research was supported by the Science Fund of the Republic of Serbia, Grant No. 7502, Intelligent Multi-Agent Control and Optimization applied to Green Buildings and Environmental Monitoring Drone Swarms - ECOSwarm, and by Fundação para a Ciência e a Tecnologia under Grant 2022.07530.CEECIND and Project UIDB/04111/2020.

Simulation results illustrate characteristic properties of the proposed algorithm.

The paper is organized as follows. In Section II we formulate the problem and define the algorithm. Section III is devoted to the description of the global algorithm model at the network level. Section IV provides a convergence analysis. Section V contains some illustrative simulation results.

II. Problem Formulation. Definition of the Algorithm

Consider N *autonomous agents*, attached one-to-one to N Markov Decision Processes (MDPs), denoted as MDP$^{(i)}$, $i = 1, \dots, N$. All the MDPs are characterized by the quadruplets $\{\mathcal{S},\ \mathcal{A},\ p(s'|s,a), R_i(s,a,s')\}$, where $\mathcal{S}$ is a finite set of states, $\mathcal{A}$ is a finite set of actions, $p(s'|s,a)$ defines probabilities of moving from $s \in \mathcal{S}$ to $s' \in \mathcal{S}$ by applying action $a \in \mathcal{A}$, and $R_i(s,a,s')$ is a local random reward with distribution $q(\cdot|s,a,s')$. Each MDP$^{(i)}$, applies a fixed *stationary behavior policy* $\pi^{(i)}(a|s)$ (probability of taking action a at state s), so that the state processes $\{S_i(n)\}$ and the state-action processes $\{S_i(n), A_i(n)\}$ represent time-homogenous Markov chains ($n \geq 0$ is an integer denoting transition time). We shall assume that $(S_i(n), A_i(n))$ is in the steady state and that μ_i denotes the underlying distribution.

Notice that, in general, in a hypothetical single agent case, the *target policy* is characterized by a stationary distribution $\pi(a|s)$. The *value function* associated with $\pi(a|s)$ is defined by

$$V^{\pi}(s) = E_{\pi}\left\{\sum_{n=0}^{\infty}\gamma^n r(S(n),A(n))|S(0)=s\right\}, \quad (1)$$

where $\gamma \in [0,1)$ is a *discount factor*. The *action-value function* under policy $\pi(a|s)$ is defined as

$$Q^{\pi}(s,a) = E_{\pi}\left\{\sum_{n=0}^{\infty}\gamma^n r(S(n),A(n))|S(0)=s, A(0)=a\right\}. \quad (2)$$

The *optimal Q-function* $Q^*(s,a)$ satisfies the following Bellman equation

$$Q^*(s,a) = r(s,a) + \gamma\sum_{s'} p(s'|s,a)\max_{a'} Q^*(s',a'), \quad (3)$$

where $r(s,a)$ denotes the one-step expected reward. The optimal policy π^* is defined by $\pi^* = \arg\max_{\pi} Q^*(s,a)$. The optimal functions V^* and Q^* can be computed using dynamic programming. Alternatively, if the MDP model is unknown, it can be computed by stochastic approximation. The so-called *Q-learning* algorithm directly provides the optimal Q-values in a tabular form [8].

Let $\phi : \mathcal{S}\times\mathcal{A} \to \mathcal{R}^p$ be a function that maps each state-action pair (s,a) to a feature vector $\phi(s,a)$. We shall use the *linear action-value function approximation* in the form $Q_{\theta}(s,a) = \theta^T\phi(s,a)$, $\|\phi(s,a)\| < \infty$, $(s,a) \in \mathcal{S}\times\mathcal{A}$, where $\theta \in \mathcal{R}^p$ is a parameter vector ($p << |\mathcal{S}\times\mathcal{A}|$). Following [11], we shall employ two classes of stationary stochastic target policies $\pi_{\theta}(\cdot|s)$: a) the greedy class, when the action is given by $\arg\max_{a'\in\mathcal{A}} Q_{\theta}(s,a')$, and b) the Gibbs class, when $\pi_{\theta}(a|s) \sim \exp\{\kappa(Q_{\theta}(s,a))\}$, with an appropriately defined differentiable function $\kappa(x)$ (see, e.g., [11] and Section V).

Remark 1: Application of the greedy policy has the obvious advantage of providing convergence to *optimality for any behavior policy*. Notice, however, that the Gibbs class is a "soft-max"-type solution (see the simulation results in Section V).

Introduce the global parameter vector $\Theta = [\theta_1^T \cdots \theta_N^T]^T$, where $\theta_i \in \mathcal{R}^p$, $i = 1, \dots, N$, is the parameter vector attached to MDP$^{(i)}$ and define the following *optimization problem*

$$J(\Theta) = \textstyle\sum_{i=1}^N q_i J_i(\theta_i) \quad (4)$$
$$\text{Subject to } \ \theta_1 = \cdots = \theta_N = \theta,$$

where $J_i(\theta_i)$ is the objective function attached to i-th agent and $q_i > 0$ *a priori* defined weighting coefficients. The main idea is to locally minimize the projected Bellman error

$$J_i(\theta_i) = \|\Pi_i T^{\pi_{\theta_i}} Q_{i;\theta_i} - Q_{i;\theta_i}\|_{\mu_i}^2 \quad (5)$$

using the stochastic gradient descent, where $\|Q_{i;\theta_i}\|_{\mu_i}^2 = \sum_{s,a} Q_{i;\theta_i}^2(s,a)\mu_i(s,a)$ and Π_i is the projection operator that projects Q-functions into the linear space $\mathcal{F}_i = \{Q_{i;\theta_i} : \theta_i \in R^p\}$ w.r.t. $\|\cdot\|_{\mu_i}$, i.e., $\Pi_i\hat{Q}_i = \arg\min_{f_i\in\mathcal{F}_i}\|\hat{Q}_i - f_i\|_{\mu_i}$.

We shall use the arguments from [5], [9], [11] and rewrite J_i as

$$\begin{aligned} J_i(\theta_i) =& E\{\delta_i(n+1;\theta_i)\phi_i(n)\}^T[E\{\phi_i(n)\phi_i(n)^T\}]^{-1} \\ &\times E\{\delta_i(n+1;\theta_i)\phi_i(n)\} \end{aligned} \quad (6)$$

where $\phi_i(n) = \phi(S_i(n), A_i(n))$,

$$\delta_i(n+1;\theta_i) = R_i(n+1) + \gamma\bar{V}_i(n+1;\theta_i) - \theta^T\phi_i(n) \quad (7)$$

is the *temporal difference* and $\bar{V}_i(n+1;\theta_i) = \bar{V}_{i;\theta_i}(S_i(n+1))$ is the expected value of the next state under π_{θ_i}, i.e.,

$$\bar{V}_{i;\theta_i}(s) = \sum_{a\in\mathcal{A}} \theta_i^T\phi(s,a)\pi_{\theta_i}(a|s). \quad (8)$$

The Fréchet sub-gradient of $J_i(\theta_i)$ w.r.t. θ_i (denoted as $\partial J_i(\theta_i)$) can be obtained following [11]. If $\hat{\phi}_i(n+1;\theta_i)$ is an unbiased estimate of the sub-gradient of $\bar{V}_i(n+1;\theta_i)$ (given $S_i(n+1)$), we have, after denoting $d_i(n+1;\theta_i) = \gamma\hat{\phi}_i(n+1;\theta_i) - \phi_i(n)$, the following expression

$$\begin{aligned} \partial J_i(\theta_i) =& E\{\delta_i(n+1;\theta_i)\phi_i(n)\} \\ &+ \gamma E\{\hat{\phi}_i(n+1;\theta_i)\phi_i(n)^T\}w_i^*(\theta_i), \end{aligned} \quad (9)$$

where

$$w_i^*(\theta_i) = E\{\phi_i(n)\phi_i(n)^T\}^{-1}E\{\delta_i(n+1;\theta_i)\phi_i(n)\}. \quad (10)$$

Remark 2: In the case of the greedy policy class, an appropriate choice for $\hat{\phi}_i(n+1;\theta_i)$ is $\hat{\phi}_i(n+1;\theta_i) = \phi(S_i(n+1), A_i'(n+1))$, where $A_i'(n+1)$ is a maximizing action of $Q_{\theta^i}(S_i(n+1),\cdot)$ [11].

When $\pi_{\theta_i}(a|s)$ is differentiable w.r.t. θ_i, we have

$$\nabla_{\theta_i}\bar{V}_{i;\theta_i}(s) = \sum_{a\in\mathcal{A}}[\phi(s,a) + Q_{\theta_i}(s,a)\nabla\log\pi_{\theta_i}(a|s)]\pi_{\theta_i}(a|s). \quad (11)$$

Consequently, we can sample $A_i'(n+1) \sim \pi_{\theta_i(n)}(\cdot|S_i(n+1))$ and use $\hat{\phi}(n+1;\theta_i(n)) = \phi(S_i(n+1), A_i'(n+1)) + Q_{\theta_i}(S_i(n+1), A_i'(n+1))\nabla \log \pi_{\theta_i}(A_i'(n+1)|S_i(n+1))$ [11].

Coming back to (4), we come to the condition $\sum_{i=1}^N q_i \partial J_i(\theta_i) = 0$ subject to $\theta_1 = \cdots = \theta_N = \theta$, or $\sum_{i=1}^N q_i \partial J_i(\theta) = 0$. The update equations given below are aimed at following the negative sub-gradient to $J(\Theta)$ for $\Theta = [\theta^T \cdots \theta^T]^T$. The new distributed algorithm is composed of *two main parts*: 1) *local parameter updates* based on the *gradient descent* methodology using local state transition and reward observations from MDPs and 2) *convexification* of current parameter estimates based on inter-agent communications. The algorithm represents a multi-agent version of the Greedy-GQ algorithm proposed in [11]. The updates are given by

$$\begin{aligned}\theta_i'(n) =& \theta_i(n) + \alpha_i(n) q_i [\delta_i(n+1;\theta_i(n))\phi_i(n) \\ &- \gamma \hat{\phi}_i(n+1;\theta_i(n))\phi_i(n)^T w_i(n)] \qquad (12)\\ w_i'(n) =& w_i(n) + \beta_i(n)[\delta_i(n+1;\theta_i(n)) \\ &- \phi_i(n)^T w_i(n)]\phi_i(n). \qquad (13)\end{aligned}$$

The initial values are chosen arbitrarily. The step size sequences $\{\alpha_i(n)\}$ and $\{\beta_i(n)\}$ are composed of positive numbers which satisfy $\alpha_i(n) << \beta_i(n)$, introducing two time-scales, see [7]. The second part of the algorithm is given by

$$\theta_i(n+1) = \sum_{j=1}^N a_{ij}(n)\theta_j'(n); \quad w_i(n+1) = w_i'(n). \qquad (14)$$

We shall assume that $a_{ij}(n) \geq 0$ are random variables, elements of a time-varying random matrix $A(n) = [a_{ij}(n)]$ [29], [31], [32].

If one adopts that the agents are connected by communication links in accordance with a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{E}$ the set of arcs, then matrix $A(n)$ has zeros at the same places as the graph adjacency matrix $A_{\mathcal{G}}(n) = A_{\mathcal{G}}$ and is *row-stochastic,* i.e. $\sum_{j=1}^N a_{ij}(n) = 1$, $i = 1,\ldots,N$, $\forall n \geq 0$.

Remark 3: In (14) convexification is applied to the θ-iterates only. It can be extended to the w-iterates, providing additional smoothing at the expense of slower convergence rate.

III. Global Model

Define $Z_i(n) = (S_i(n),\ S_i(n+1))$. After denoting $z = (s, s')$, we introduce the following functions

$$g_i(\theta, w, z) = \phi(s)\bar{\delta}_i(s, s', \theta) - \gamma\hat{\phi}(s')\phi(s)^T w, \qquad (15)$$

and

$$k_i(\theta, w, z) = \phi(s)\bar{\delta}_i(s, s', \theta) - \phi(s)\phi(s)^T w, \qquad (16)$$

where $\bar{\delta}_i(s, s', \theta) = r(s, a, s') + \gamma\bar{V}_i(s', \theta) - \phi(s)^T\theta$ is the expected temporal difference error [7]. For the mean values we have

$$\begin{aligned}\bar{g}_i(\theta, w) =& b_i - A_i(\theta)\theta - \gamma B_i(\theta)w \qquad (17)\\ \bar{k}_i(\theta, w) =& b_i - A_i(\theta)\theta - C_i w \qquad (18)\end{aligned}$$

where $b_i = \sum_{s,a,s'} R_i(s, a, s')p(s'|s, a)\mu_i(s, a)$, $C_i = E_i\{\phi_i(n)\phi_i(n)^T\}$, $A_i(\theta) = C_i - \gamma\sum_{s,a,s',b}\phi(s', b)\phi(s, a)^T\pi_{\theta_i}(b|s)p(s'|s, a)\mu_i(s, a)$, $B_i(\theta) = E_i\{\hat{\phi}_i(n+1, \theta_i)\phi_i(n)^T\}$ ($E_i\{\cdot\}$ denotes the expectation according to the probability law induced in MDP$^{(i)}$).

Let $X(n) = [\Theta(n)^T \vdots W(n)^T]^T$, $\Theta(n) = [\theta_1(n)^T \cdots \theta_N(n)^T]^T$, $W(n) = [w_1(n)^T \cdots w_N(n)^T]^T$ and $X'(n) = [\Theta'(n)^T \vdots W'(n)^T]^T$. Then, we have

$$\begin{aligned}X'(n) &= X(n) + \Gamma(n)F(X(n), n),\\ X(n+1) &= \mathrm{diag}\{(A(n)\otimes I_p), I_{Np}\}X'(n), \qquad (19)\end{aligned}$$

$X(0) = X_0$, where $\otimes$ denotes the Kronecker's product, while $\Gamma(n) = \mathrm{diag}\{\alpha_1(n),\ldots,\ \alpha_N(n), \beta_1(n),\ldots,\beta_N(n)\}$ $\otimes I_p$, $F(X(n), n) = [F^\theta(X(n), n)^T \vdots F^w(X(n), n)^T]^T$, $F^\theta(X(n), n) = [F_1^\theta(X(n), n)^T \cdots F_N^\theta(X(n), n)^T]^T$, $F^w(X(n), n) = [F_1^w(X(n), n)^T \cdots F_N^w(X(n), n)^T]^T$, with $F_i^\theta(X(n), n) = q_i g_i(\theta_i(n), w_i(n), Z_i(n+1)) + \phi_i(n)\omega_i(n+1)$ and $F_i^w(X(n), n) = k_i(\theta_i(n), w_i(n), Z_i(n+1)) + \phi_i(n)\omega_i(n+1)$, where $\omega_i(n+1)$ is a zero-mean noise term modeling a stochastic component in $R_i(S_i(n), A_i(n), S_i(n+1))$, see [17].

Also, we introduce $\bar{F}(X) = [\bar{F}^\theta(X)^T \vdots \bar{F}^w(X)^T]^T$, where $\bar{F}_i^\theta(X) = q_i\bar{g}_i(\theta, w)$ and $\bar{F}_i^w(X) = \bar{k}_i(\theta, w)$, $i = 1,\ldots,N$.

1) Consensus Part: Define $\Psi(n|k) = A(n)\cdots A(k)$ for $n \geq k$, $\Psi(n|n+1) = I_N$. Let $\mathcal{F}_n$ be an increasing sequence of σ-algebras, such that $\mathcal{F}_n$ measures $\{X(k), k \leq n, A(k), k < n\}$.

(A1) There is a scalar $\alpha_0 > 0$ such that $a_{ii}(n) \geq \alpha_0$, and, for $i \neq j$, either $a_{ij}(n) = 0$ or $a_{ij}(n) \geq \alpha_0$.

(A2) Graph $\mathcal{G}$ is strongly connected.

(A3) There are a scalar $p_0 > 0$ and an integer n_0 such that $P_{\bar{\mathcal{F}}_n}$(agent j communicates to agent i on the interval $[n, n+n_0]$) $\geq p_0$, for all n and $i, j = 1,\ldots N$ such that (i, j)-th element of $A_{\mathcal{G}} \neq 0$.

Lemma 1 ([27], [29]): Let (A1)–(A3) hold. Then $\Psi(k) = \lim_n \Psi(n|k)$ exists with probability 1 (w.p.1) and its rows are all equal; moreover, $E\{|\Psi(n|k) - \Psi(k)|\}$ and $E_{\mathcal{F}_k}\{|\Psi(n|k) - \Psi(k)|\} \to 0$ geometrically as $n - k \to \infty$, uniformly in k (w.p.1); also, $E_{\mathcal{F}_k}\{\Psi(n|k)\}$ converges to $\Psi(k)$ geometrically, uniformly in k, as $n \to \infty$ ($|\cdot|$ denotes the infinity norm).

(A4) There is a $N \times N$ matrix $\bar{\Psi}$ such that $E\{|E_{\mathcal{F}_k}\{\Psi(n)\} - \bar{\Psi}|\} \to 0$ as $n - k \to \infty$, which, according to Lemma 1 and [27], has the form $\bar{\Psi} = [\hat{\Psi}^T \cdots \hat{\Psi}^T]^T$, where $\hat{\Psi} = [\bar{\psi}_1 \cdots \bar{\psi}_N]^T$.

Specific values of $\hat{\Psi}$ follow from the network properties and the weights of the arcs. In the following, we shall adopt that $\bar{\psi}_i = 1/N$, $i = 1,\ldots,N$, in order to avoid ambiguities w.r.t. q_i (see also [17]). On the other hand, for $q_i = 1$ we can obtain arbitrary desired values of $\bar{\psi}_i$, $i = 1, \ldots, N$, by appropriate definition of matrices $A(n)$ [29].

(A5) Sequence $\{A(n)\}$ is independent of the processes in MDP$^{(i)}$, $i = 1, \ldots, N$.

IV. Convergence Analysis

(A6) Sequence $\{X(n)\}$ is tight (bounded in probability), see, e.g., [28].

Remark 4: Assumption (A6) is frequent for weak convergence proofs, in general. As stated in [27], [28], one can achieve, w.l.o.g., that $\{X(n)\}$ is tight by an adequate projection or truncation of the estimates (see [27], [28, Section IV.A]).

(A7) Matrix C_i is nonsingular, $i = 1, \ldots, N$.

(A8) For any θ_i, the policy $\pi^{\infty}_{\theta_i}(a|s) = \lim_{c \to \infty} \pi^{\infty}_{c\theta_i}(a|s)$ exists and its convergence is uniform on compact sets, $(s, a) \in \mathcal{S} \times \mathcal{A}$.

(A9) Matrix $\sum_{i=1}^{N} q_i[C_i - \gamma \sum_{s,a,s',b} \phi(s', b) \phi(s, a)^T \pi(b|s') p(s'|s, a) \mu_i(s, a)]$ is nonsingular for any $\pi(b|s') \in \mathcal{L}$, where $\mathcal{L} = \{\pi^{\infty}_{\theta} : \theta \in \mathcal{R}^d\}$ is bounded w.p.1 [11].

Assume that $\alpha_i(n) = \alpha$ and $\beta_i(n) = \beta$. According to [27], let n_α be a sequence tending to ∞ and satisfying $\alpha^{\frac{1}{2}} n_\alpha \to 0$ as $\alpha \to 0$. For $t \geq 0$, $t \in \mathcal{R}$, define $X^{\alpha,\beta}(\cdot)$ as $X^{\alpha,\beta}(t) = X(n)$ for $t \in [(n - n_\alpha)\alpha, (n - n_\alpha + 1)\alpha)$ (for details, see [27]).

Theorem 1: Let (A1)–(A9) hold. Let $X^{\alpha,\beta}(n)$ be generated by (12), (13) and (14), with $\alpha_i(n) = \alpha > 0$, $\beta_i(n) = \beta > 0$, $\beta >> \alpha$. Define $X^{\alpha,\beta}(0) = [\theta_0^T \cdots \theta_0^T w_{1,0}^T \cdots w_{N,0}^T]^T$. Then $X^{\alpha,\beta}(\cdot)$ is *tight and converges weakly* at the *fast time-scale* to a process $W(\cdot) = [w_1(\cdot)^T \cdots w_N(\cdot)^T]^T$ generated by

$$\dot{w}_i = \bar{k}_i(\theta_i, w_i), \tag{20}$$

($i = 1, \ldots, N$), for any given $\theta_1, \ldots, \theta_N$, and at the *slow time-scale* to $\Theta(\cdot) = [\theta(\cdot)^T \cdots \theta(\cdot)^T]^T$, where

$$\dot{\theta} = \frac{1}{N} \sum_{i=1}^{N} q_i \bar{g}_i(\theta, \bar{w}_i^*(\theta)), \tag{21}$$

with $\bar{w}_i^*(\theta)$ obtained as the unique solution (w.r.t. w_i) of

$$\bar{k}_i(\theta, w_i) = b_i - A_i(\theta)\theta - C_i w_i = 0. \tag{22}$$

Proof: At the start, it is essential to verify the basic assumptions from [27, Theorem 3.1]. We conclude in a straightforward way that the assumptions C(3.2) and C(3.3) from [27] are satisfied for our algorithm, and that C(3.4) holds for the communications within the network. Therefore, one can show that $\sup_{\alpha, n \geq n_\alpha} \frac{1}{\alpha^2} E\{|X(n+1) - X(n)|^2\} < \infty$, $\{\frac{1}{\alpha}|X(n+1) - X(n)|, n \geq n_\alpha\}$ is uniformly integrable, $\{X^{\alpha,\beta}(\cdot)\}$ is tight and the limit paths are Lipschitz continuous [27, Theorem 3.1, Part 1].

According to [27], the formulation of the asymptotic mean ODE (21) follows from the demonstration that the $M_f(t)$ defined by

$$\begin{aligned} M_f(t) =& f(X(t)) - f(X(0)) \\ &+ \int_0^t f'_X(X(s)) \mathrm{diag}\{\bar{\Psi} \otimes I_p, I_{Np}\} \bar{F}(X(s)) ds, \end{aligned} \tag{23}$$

is a Lipschitz-continuous martingale, where $f(\cdot)$ a real valued function with compact support and continuous second derivatives [27]. The technical part of the derivation is based on the Skorokhod embedding [28]. As $X(\cdot)$ is Lipschitz continuous and $M_f(0) = 0$, it follows that $M_f(t) = 0$. This implies that $\dot{X} = \mathrm{diag}\{\bar{\Psi} \otimes I_p, I_{Np}\} \bar{F}(X)$. By (A1)–(A5), all the rows of $\bar{\Psi}$ are equal. It follows that N p-dimensional vector components of Θ must be equal, i.e., we obtain that $\Theta(\cdot) = [\theta(\cdot)^T \cdots \theta(\cdot)^T]$ and that $\theta(\cdot)$ satisfies the ODE from (21).

ODE in (20) follows directly from the two-time-scale property of the algorithm. Existence and uniqueness of the solution to $\bar{k}_i(\theta, w_i) = 0$ w.r.t. w_i follows from (A7), so that the ODE $\dot{w}_i = \bar{k}_i(\theta, w_i)$ admits a unique, globally asymptotically stable equilibrium $\bar{w}_i^*$ for any fixed θ. ■

Theorem 2: Let the assumptions of Theorem 1 be satisfied. Then, for any integers $n'_{\alpha,\beta}$ such that $\alpha n'_{\alpha,\beta} \to \infty$ as $\alpha \to 0$, there exist positive numbers $\{T_{\alpha,\beta}\}$ with $T_{\alpha,\beta} \to \infty$ as $(\beta, \alpha/\beta) \to 0$, such that for any $\epsilon > 0$

$$\limsup_{\beta \to 0, \alpha/\beta \to 0} P\{(\theta_i^{\alpha,\beta}(n'_{\alpha,\beta} + k)) \notin N_\epsilon(\bar{\Sigma}_{\bar{\theta}})\} = 0 \tag{24}$$

for some $k \in [0, T_{\alpha,\beta}/\alpha]$, $i = 1, \ldots, N$, where $\bar{\Sigma}_\theta$ is a *bounded set* of stationary points $\bar{\theta} \in \mathcal{R}^p$ of the criterion $J(\theta)$, satisfying $\sum_{i=1}^{N} q_i g_i(\bar{\theta}, \bar{w}_i^*(\bar{\theta})) = 0$.

Proof: According to Theorem 1, there exists a set $M = M(\omega) \subset \mathcal{R}^p$ which represents a compact connected invariant set to $\dot{\theta} = \frac{1}{N} \sum_{i=1}^{N} q_i \bar{g}_i(\theta, \bar{w}_i^*(\theta))$, such that $(\theta_i(n), w_i(n)) \to \{(\bar{\theta}, \bar{w}_i^*(\bar{\theta})) : \bar{\theta} \in M\}$, $i = 1, \ldots, N$.

We shall apply the ODE methodology proposed by Borkar and Meyn [30]. We want to show that the following limit exists $\lim_{c \to \infty} \frac{1}{cN} \sum_{i=1}^{N} q_i \bar{g}_i(c\theta, \bar{w}_i^*(c\theta)) = \bar{g}^{\infty}(\theta)$, that the convergence is uniform and that zero is the unique global exponentially stable equilibrium to the ODE $\dot{\theta} = \bar{g}^{\infty}(\theta)$. Following [11], we obtain that

$$\begin{aligned} \bar{g}_{\infty}(\theta) = \tfrac{1}{N} \textstyle\sum_{i=1}^{N} q_i \partial \| \tfrac{1}{c}(b_i - \gamma B_i(c\theta) \bar{w}_i^*(\theta)) \\ - A_i(c\theta)\theta \|^2_{C_i^{-1}}. \end{aligned} \tag{25}$$

Owing to (A8), the limit $A_i^{\infty}(\theta) = \lim_{c \to \infty} A_i(c\theta)$ exists and the convergence is uniform. Let $N_{\infty}(\theta) = 2 \sum_{i=1}^{N} q_i A_i^{\infty T}(\theta) A_i^{\infty}(\theta)$, and also let $\{N_1, \ldots, N_K\} = \{N_{\infty}(\theta) : \theta \in \mathcal{R}^p\}$ be constant values of $N_{\infty}(\theta)$ in K non-overlapping regions of $\mathcal{R}^p$, denoted as $\bar{R}_j, j = 1, \ldots, K$ (in accordance with the regions in which $A_i^{\infty}(\theta)$ are constant). Consequently, $N_j = N_{\infty}(\theta)$, $\forall \theta \in R^p$, for some j. It follows that

$$\dot{\theta} = -\sum_{j=1}^{K} I\{\theta \in \bar{R}_j\} N_j \theta, \tag{26}$$

where $I\{\cdot\}$ is the indicator function.

Adopting $W(\theta(t)) = \frac{1}{2}\theta(t)^T \theta(t)$ as a Lyapunov function, where $\theta(t)$ is a solution of (26), we can obtain, by using the arguments from [11], that

$$\dot{W} = \frac{1}{2}(\theta(t)^T \dot{\theta}(t) + \dot{\theta}(t)^T \theta(t)) \leq -\rho \|\theta(t)\|^2,$$

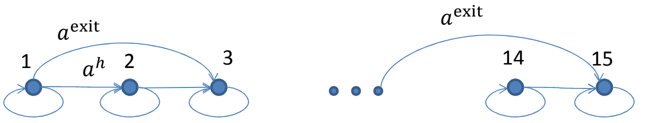

Fig. 1. Diagram of the simulated MDPs.

where $N_j + N_j^T \geq 2\rho I$, $\rho > 0$, having in mind that matrices N_j are normal, $j = 1, \dots, K$. Therefore, zero is the unique globally asymptotically stable equilibrium to $\dot{\theta} = \bar{g}^{\infty}(\theta)$, so that the result follows. ■

Remark 5: Notice that the multi-agent environment introduces a relaxation of the condition (A9) w.r.t. the single agent case treated in [11], due to freedom in choosing weights q_i.

Remark 6: Theorem 2 shows that in the case when the algorithm utilizes projection of the parameter estimates to a ball at the origin $B(\rho_B) \subset \mathcal{R}^p$ (ensuring (A6)), one can always find such a radius $\rho_B < \infty$ that $\bar{\Sigma}_\theta \subset B(\rho_B)$.

Remark 7: Asymptotic rate of convergence of the algorithm can be expressed in terms of a stochastic differential equation (SDE) [24], [25], [28]. It is possible to show that the proposed algorithm ensures lower error covariance than in the case of alternative single-agent methods, due to averaging over the multi-agent network [27].

V. Simulation Results

In this section we demonstrate the algorithm's convergence properties using simulations. The underlying MDP is assumed to be a class of the Boyan chain [5], [17], [21], [33]. Fig. 1 depicts the MDP diagram.

There are two possible actions in this scenario: either to take action a^h, which means staying on the current main route or road, or to take action a^{exit}, which involves exiting and using alternative route. State 15 is the goal state where the process terminates. When a^{exit} action is chosen, the constant reward $R(s, a^{\text{exit}}, s') = -2.5$ is received for all the states involved and there is the probability $p^{\text{exit}}_{\text{stuck}} = 0.2$ of staying in the same state. If action a^h is chosen, a reward of -1 is received for any transition between states and the probability of remaining in the same state increases as $1 - \frac{1}{s}$. The discount factor of $\gamma = 0.9$ is applied. The probability of choosing action a^{exit} in state s, denoted as $\pi(a^{\text{exit}}|s)$, is the control policy that needs to be optimized. Feature vectors used in Q-function approximation have dimensionality $p = 14$, 7 for each of two possible actions, represented by Gaussian radial basis functions defined as $e^{-\frac{(s-z_i)^2}{2\sigma^2}}$, where i ranges from 1 to 7, z_i takes values from the set $\{1, 3, 5, 7, 9, 11, 13\}$ and $\sigma^2 = 2$. Simulations are carried out over multiple episodes since State 15 is the absorbing state.

We analyzed performance of the proposed algorithm (12), (13), (14), with 10 agents involved, communicating according to a sparse time-invariant communication graph where each agent communicates with 2-3 randomly selected other agents, under the following three setups: 1) The agents use different state-invariant behavior policies such that, when initiated in State 1, they are all capable of reaching any other state with positive probability. The following behavior policies (exit probabilities in each state) are assumed: $\pi_b(a^{\text{exit}}|s) = [0.15, 0.24, 0.13, 0.38, 0.55, 0.89, 0.64, 0.97, 0.75, 0.69]$. The greedy target policy is selected. 2) The agents use the same non-restrictive behavior policies as in case 1), but the target policy is selected from the Gibbs class, when $\pi_\theta(a|s) = \frac{e^{\theta^T \phi(s,a)/\tau}}{\sum_{a' \in \mathcal{A}} e^{\theta^T \phi(s,a')/\tau}}$, with the "temperature" parameter τ set to a relatively low value of $1/50$. 3) The agents use the same state-invariant behavior policies, but they cannot visit all the states with positive probability, i.e. the following are each agents starting and stopping (absorbing) states $[(1, 9), (1, 5), (1, 7), (1, 6), (5, 13), (3, 14), (8, 14), (1, 6), (6, 6), (6, 15)]$ (the first agent always starts in State 1 and stops in State 9, etc.). The target policy is the same as in the case 2) (Gibbs with $\tau = 1/50$). The step sizes in all three cases are set to $\alpha = 0.001$ and $\beta = 1.5$ (respecting the need for two time-scales). Fig. 2 shows the average of the exact value functions corresponding to the agents' policy estimates (exactly calculated in each time step and averaged over all the agents and states) versus the number of iterations. It is interesting to observe that, although it is known that the optimal policy is deterministic, we get better results in case 2) (Gibbs policy) than in case 1) (greedy policy). A possible explanation is that the Gibbs stochastic policy is capable to better adapt to the imprecision in the Q-function estimates which are due to the assumed linear approximation. Of course, in the case of tabular features the greedy target policy converges to the optimal one. It is also evident that the agents can approach the optimal policy together even in the case 3), i.e., when the individual agents are not capable of learning due to their local behavior/exploration restrictions. Fig 3 shows the final value function approximations obtained by the algorithm, together with the true optimal value function and the average of the exact value functions corresponding to the final policies estimated by each agent (they have converged to almost the same values due to the consensus scheme). It is evident from both figures that case 2) shows the best performance, followed by the cases 1) and 3).

VI. Conclusion

In this paper, we have proposed a novel algorithm for distributed off-policy gradient based Q-learning algorithm with linear function approximation in a collaborative multi-agent setting. Under nonrestrictive assumptions, we have proved, after formulating asymptotic mean ODEs for the algorithm, that the parameter estimates weakly converge to consensus, as required. Efficiency of the proposed algorithm has been illustrated by simulation. It is shown that function parallelization introduced by the proposed multi agent algorithm leads to advantages w.r.t. the single agent case in the sense of much more efficient state exploration and reduced covariance of the parameter estimates.

In further work, the proposed algorithm could be extended to the cases of nonlinear value function approximations (such as those using deep neural networks [3]).

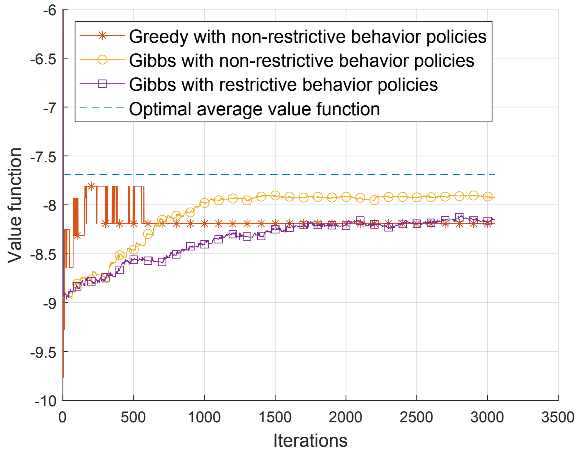

Fig. 2. Evolution of the average of the exact value functions corresponding to the agents' optimal policy estimates under the three described scenarios. The horizontal line is the optimal average value function.

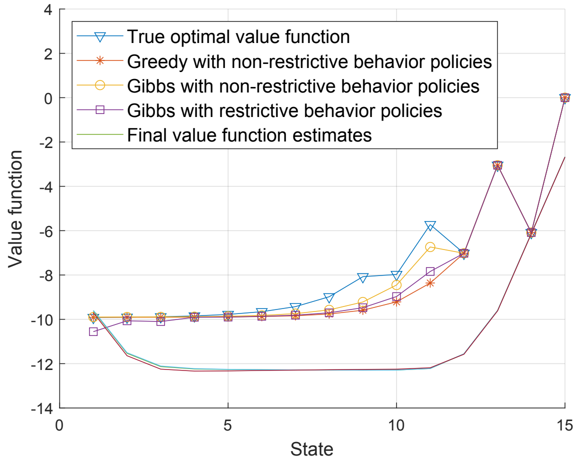

Fig. 3. The average of the exact value functions of the final policies estimated by the agents under the three described scenarios, together with the final value function approximations obtained by the algorithm and the true optimal value function.

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 2017.

[2] D. P. Bertsekas and J. Tsitsiklis, *Neuro-Dynamic Programming*. Athena Scientific, 1996.

[3] V. Mnih, K. Karavukcuoglu, D. Silver, A. Rusu, J. Veness, J. Bellemare, A. Graves, M. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–535, 2022.

[4] J. N. Tsitsiklis and B. V. Roy, "An analysis of temporal-difference learning with function approximation," *IEEE Trans. Autom. Control*, vol. 42, pp. 674–690, 1997.

[5] R. S. Sutton, H. R. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvári, and E. Wiewiora, "Fast gradient-descent methods for temporal-difference learning with linear function approximation," in *Proc. 26th Int. Conf. on Machine Learning*, 2009, pp. 993–1000.

[6] M. Geist and B. Scherrer, "Off-policy learning with eligibility traces: A survey," *Journal of Machine Learning Research*, vol. 15, pp. 289–333, 2014.

[7] H. Yu, "On convergence of some gradient-based temporal-differences algorithms for off-policy learning," *arXiv:1712.09652*, 2017.

[8] C. J. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, pp. 279–292, 1992.

[9] H. R. Maei and R. S. Sutton, "GQ(λ): A general gradient algorithm for temporal difference prediction learning with eligibility traces," in *Proc. 3rd Conf. Artificial General Intelligence*, 2010, pp. 91–96.

[10] F. S. Melo, S. P. Meyn, and M. I. Ribeiro, "An analysis of reinforcement learning with function approximation," in *25th Intern. Conf. Machine Learning*, 2008.

[11] H. R. Maei, C. Szepesvari, S. Bhatnagar, and R. S. Sutton, "Toward off policy learning control with function approximation," in *Proc. Intern. Conf. Machine Learning*, 2010, pp. 719–726.

[12] Z. Chen, S. Zhang, T. Doan, S. T. Maguluri, and J. Clarke, "Performance of Q-learning with linear function approximation," *arXiv:1905.11435*, 2019.

[13] H.-D. Lim, D. W. Kim, and D. Lee, "Regularized Q-learning," *arXiv:2202.05404v5*, 2022.

[14] D. S. Carvalho, F. S. Melo, and P. A. Santos, "A new convergent variant of Q-learning with linear function approximation," in *Proc. 34th Conf. NeurIPS*, 2020.

[15] D. Lee and N. He, "Performance of Q-learning with linear function approximation," *arXiv:1912.02270*, 2021.

[16] J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative multi-agent control using deep reinforcement learning," in *Autonomous Agents and Multiagent Systems*, G. Sukthankar and J. A. Rodriguez-Aguilar, Eds. Cham: Springer International Publishing, 2017, pp. 66–83.

[17] M. S. Stanković, M. Beko, and S. S. Stanković, "Distributed value function approximation for collaborative multiagent reinforcement learning," *IEEE Transactions on Control of Network Systems*, vol. 8, no. 3, pp. 1270–1280, 2021.

[18] T. Doan, S. Maguluri, and J. Romberg, "Finite-time analysis of distributed TD(0) with linear function approximation on multi-agent reinforcement learning," in *Proc. Int. Conf. Machine Learning*, 2019, pp. 1626–1635.

[19] D.Lee and J. Hu, "Primal-dual distributed temporal difference learning," *arXiv:1805.07198*, 2020.

[20] D. Ding, X. Wei, Z. Yang, Z.Wang, and M. Jovanović, "Fast multi-agent temporal-difference learning via homotopy stochastic primal-dual optimization," *arXiv:1908.02805*, 2020.

[21] M. S. Stanković, M. Beko, N. Ilić, and S. S. Stanković, "Distributed consensus-based multi-agent temporal-difference learning," *Automatica*, vol. 151, p. 110922, 2023.

[22] P. Pennesi and I. Paschalidis, "A distributed actor-critic algorithm and applications to mobile sensor network coordination problems," *IEEE Trans. Autom. Control*, vol. 55, pp. 492–497, 2010.

[23] Y. Zhang and M. M. Zavlanos, "Distributed off-policy actor-critic reinforcement learning with policy consensus," *arXiv:1903.09255*, 2019.

[24] M. S. Stanković, M. Beko, N. Ilić, and S. S. Stanković, "Multi-agent actor-critic multitask reinforcement learning based on GTD(1) with consensus," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, 2022, pp. 4591–4596.

[25] ——, "Multi-agent off-policy actor-critic algorithm for distributed multi-task reinforcement learning," *European Journal of Control*, vol. 74, p. 100853, 2023.

[26] S. V. Macua, J. Chen, S. Zazo, and A. H. Sayed, "Distributed policy evaluation under multiple behavior strategies," *IEEE Trans. Autom. Control*, vol. 60, no. 5, pp. 1260–1274, 2015.

[27] H. J. Kushner and G. Yin, "Asymptotic properties of distributed and communicating stochastic approximation algorithms," *SIAM J. Control Optim.*, vol. 25, pp. 1266–1290, 1987.

[28] ——, *Stochastic Approximation and Recursive Algorithms and Applications*. Springer, 2003.

[29] M. S. Stanković, N. Ilić, and S. S. Stanković, "Distributed stochastic approximation: Weak convergence and network design," *IEEE Trans. Autom. Control*, vol. 61, no. 12, pp. 4069–4074, 2016.

[30] V. Borkar and S. P. Meyn, "The ODE method for convergence of stochastic approximation and reinforcement learning," *SIAM Journal on Control And Optimization*, vol. 38, pp. 447–469, 2000.

[31] M. S. Stanković, S. S. Stanković, and D. M. Stipanović, "Consensus-based decentralized real-time identification of large-scale systems," *Automatica*, vol. 60, pp. 219 – 226, 2015.

[32] M. S. Stanković, M. Beko, and S. S. Stanković, "Nonlinear robustified stochastic consensus seeking," *Systems & Control Letters*, vol. 139, p. 104667, 2020.

[33] J. A. Boyan, "Technical update: Least-squares temporal difference learning," *Machine learning*, vol. 49, pp. 233–246, 2002.