2018 IEEE Conference on Decision and Control (CDC)
Miami Beach, FL, USA, Dec. 17-19, 2018

Networked Multi-Agent Reinforcement Learning in Continuous Spaces

Kaiqing Zhang, Zhuoran Yang, and Tamer Başar

Abstract— **Many real-world tasks on practical control systems involve the learning and decision-making of multiple agents, under limited communications and observations. In this paper, we study the problem of networked multi-agent reinforcement learning (MARL), where multiple agents perform reinforcement learning in a common environment, and are able to exchange information via a possibly time-varying communication network. In particular, we focus on a collaborative MARL setting where each agent has individual reward functions, and the objective of all the agents is to maximize the network-wide averaged long-term return. To this end, we propose a *fully decentralized* actor-critic algorithm that only relies on neighbor-to-neighbor communications among agents. To promote the use of the algorithm on practical control systems, we focus on the setting with continuous state and action spaces, and adopt the newly proposed expected policy gradient to reduce the variance of the gradient estimate. We provide convergence guarantees for the algorithm when linear function approximation is employed, and corroborate our theoretical results via simulations.**

I. Introduction

Reinforcement learning (RL) has been widely advocated in applications of sequential-decision making under uncertainty. One great challenge in applying RL algorithms to practical systems is that usually the systems involve more than one decision-maker, i.e., multiple agents that interact with each other. This multi-agent setting finds broad applications in practical control systems, including the power grid [1], robotics [2], and unmanned vehicles [3]. In this work, we focus on developing RL algorithms for such a setting, i.e., the problem of multi-agent reinforcement learning (MARL).

In particular, we study the collaborative MARL setting, where the agents operate in a common environment, and the joint actions of all agents influence the state of the environment. The agents are allowed to have different reward functions possibly from various tasks, but each agent can only observe its own reward. Then the objective is to collectively maximize the globally averaged return of all agents in the environment. To this end, one attractive protocol to foster this collaboration is through a centralized architecture. Specifically, in such a protocol, there would exist a central controller that can collect the rewards of all agents, and thus determine the optimal action for each agent. In fact, this centralized/hierarchical architecture has been employed in the design of some existing MARL algorithms; see [4], [5] as two recent examples. One more generally considered collaborative MARL setting is that the agents are restricted to have a common reward [6], [7]. This, in some way, is equivalent to the centralized setting, since each agent now has the knowledge of the global return to maximize.

However, in many real-world scenarios, such as the management of load in the power grid [8] and the control of intelligent transportation systems [9], [10], it may be costly or even unrealistic to have a central controller. Moreover, since a central controller has to communicate with all agents to transmit data and decisions, such a protocol is not scalable due to the incessantly growing communication overhead at such a single controller. This not only deteriorates the scalability of the system to large number of agents, but also increases the vulnerability in the presence of potential malicious attacks. In light of these disadvantages of the centralized architecture, a *fully decentralized* one becomes more favorable. By fully decentralized MARL, we mean that the agents are connected by a possibly time-varying communication network, through which they are able to exchange information without any central controller. In addition, each agent makes individual decisions, based on the local observation and the message sent from its neighbors. This decentralized architecture has been advocated in recent advances on distributed/consensus optimization algorithms, e.g., [11], [12], but has barely been incorporated in the framework of multi-agent reinforcement learning.

Another major challenge that impedes the applicability of RL algorithms is how to handle enormous, or even continuous state and action spaces in practical control systems. One popular family of algorithms that overcome this challenge is actor-critic (AC) algorithms, which are built upon value function approximation and parametrized policies [13], [14]. The technical core of AC algorithms is the well-known *stochastic policy gradient* (SPG) theorem [15], which establishes the closed form of the return with respect to the policy parameters. However, it is argued both theoretically [16] and empirically [17] that this SPG suffers from very high variance gradient estimates, especially with continuous action spaces. This flaw motivates the alternative deterministic policy gradient (DPG) theorem and the corresponding AC algorithms, which are especially designed for continuous action spaces. Unfortunately, DPG-based AC algorithms require off-policy exploration, which does not apply in our decentralized MARL setting, since the policies of other agents are not known to each agent during the learning process. To design an on-policy AC algorithm in continuous action spaces while enjoying smaller gradient variance, we resort to the newly proposed *expected policy gradient* (EPG)

K. Zhang and T. Başar are with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign ({kzhang66, basar1}@illinois.edu), and the work was supported in part by US Army Research Office (ARO) Grant W911NF-16-1-0485, and in part by Office of Naval Research (ONR) MURI Grant N00014-16-1-2710. Z. Yang is with the Department of Operations Research and Financial Engineering, Princeton University (zy6@princeton.edu).

978-1-5386-1395-5/18/$31.00 ©2018 IEEE

[16], which unifies SPG and DPG to some extent and is applicable to continuous action spaces. Therefore, it is worth extending the EPG to our MARL setting and developing AC algorithms accordingly.

In this work, we develop a fully decentralized MARL algorithm in continuous state and action spaces, for networked agents. The main contribution of the work is three-fold: first, we extend the form of expected policy gradient to the MARL setting; second, we develop a fully decentralized AC algorithm that only relies on neighbor-to-neighbor communications for agents over a network; and third, we establish convergence guarantees when linear function approximation is used, which provides theoretical support for the proposed MARL for networked agents. To the best of our knowledge, this appears to be the first attempt to bridge the decentralized architecture and MARL. We note that the settings in [18] and [19] are the most similar ones to ours with a networked structure; however, the former only works for the tabular-case MARL with no function approximation, and a remote/central controller is required, while the latter solves multiple independent MDPs (instead of a MARL problem), and does not have complete convergence analysis.

II. Preliminaries

In this section, we first introduce the formulation of the networked MARL problem, and then provide the expected policy gradient theorem for MARL.

A. Networked MARL in Continuous Spaces

Consider a network of N agents operating in a common environment, where the set of agents is denoted by $\mathcal{N} = [N]$. Our focus is on the *fully decentralized* setting, where there exists no central controller that can collects rewards from agents or makes decisions on behalf of all the agents. In contrast, a communication network, denoted by $\mathcal{G}_t = (\mathcal{N}, \mathcal{E}_t)$, connects all the agents in $\mathcal{N}$, where $\mathcal{E}_t$ represents the set of communication links at time $t \in \mathbb{N}$. In words, $\mathcal{G}_t$ is a time-varying and undirected graph with vertex set $\mathcal{N}$ and edge set $\mathcal{E}_t \subseteq \{(i,j)\colon i,j \in \mathcal{N}, i \neq j\}$ at time t, where an edge $(i,j) \in \mathcal{E}_t$ means that agents i and j can share information at time t. We now define the networked multi-agent Markov decision process as follows.

Definition 1. *[Networked Multi-Agent MDP in Continuous Spaces] Let $\{\mathcal{G}_t = (\mathcal{N}, \mathcal{E}_t)\}_{t\geq 0}$ be a time-varying communication network connecting $|\mathcal{N}| = N$ agents. A networked multi-agent MDP is described by a tuple $(\mathcal{S}, \{\mathcal{A}^i\}_{i\in\mathcal{N}}, p, \{R^i\}_{i\in\mathcal{N}}, \{\mathcal{G}_t\}_{t\geq 0})$, where $\mathcal{S} \subseteq \mathbb{R}^d$ denotes the global state space, and $\mathcal{A}^i \subseteq \mathbb{R}^{c^i}$ denotes the action set of agent i. Note that both spaces $\mathcal{S}$ and $\mathcal{A}$ are continuous. Moreover, denoting the joint action space of all agents as $\mathcal{A} = \prod_{i=1}^N \mathcal{A}^i$, we let $R^i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ denote the local reward function of agent i, and $p(ds' \,|\, s, a)$ denote the state transition kernel of the MDP given action a and state s.*

The transition of the multi-agent MDP proceeds as follows. At time step t, all agents can observe the global state $s_t \in \mathcal{S}$, and then execute joint actions $a_t = (a_t^1, \ldots, a_t^N) \in \mathcal{A}$. As a result, each agent $i \in \mathcal{N}$ receives a reward r_{t+1}^i, a random variable whose conditional expectation is $\mathbb{E}(r_{t+1}^i \,|\, s_t, a_t) = R^i(s_t, a_t)$. Furthermore, the next state $s_{t+1} \in \mathcal{S}$ is sampled from the kernel $p(\cdot \,|\, s_t, a_t)$. All agents are assumed to choose their actions conditionally independent given the current state. Specifically, let $\pi_{\theta^i}(a^i \,|\, s)$ be the local policy of agent i parametrized by θ^i, where $\theta^i \in \Theta^i$ for some compact $\Theta^i \in \mathbb{R}^{m^i}$. The policy $\pi_{\theta^i}(a^i \,|\, s)$ represents the probability density over the action set $\mathcal{A}^i$ given state s. The parameters θ^i are concatenated as $\theta = [(\theta^1)^\top, \cdots, (\theta^N)^\top]^\top \in \Theta$, where $\Theta = \prod_{i=1}^N \Theta^i$. Then, the probability density over the joint action set $\mathcal{A}$ can be written as $\pi_\theta(a \,|\, s) = \prod_{i\in\mathcal{N}} \pi_{\theta^i}(a_i \,|\, s)$. Since the reward is received locally and the action is executed individually by each agent, we refer to our model as a *fully decentralized* one.

For any $\theta \in \Theta$, let p_θ be the transition kernel of the Markov chain $\{s_t\}_{t\geq 0}$ induced by policy π_θ, namely,

$$p_\theta(ds' \,|\, s) = \int_{\mathcal{A}} d\pi_\theta(a \,|\, s) p(ds' \,|\, s, a), \quad \forall s, s' \in \mathcal{S}. \quad (1)$$

For notational simplicity, we let $d\pi_\theta(a \,|\, s) = \pi_\theta(a \,|\, s)\nu(da)$, where $\nu(da)$ is a fixed measure on the action space $\mathcal{A}$. We first make the following standard regularity assumption on the MDP and policy functions as in the classical work [13].

Assumption 1. *For any $i \in \mathcal{N}$, $s \in \mathcal{S}$, and $\theta^i \in \Theta^i$, the conditional probability density of a^i is positive, that is, $\pi_{\theta^i}(a^i \,|\, s) > 0$. Also, $\pi_{\theta^i}(a^i \,|\, s)$ is continuously differentiable with respect to the parameter θ^i. Moreover, the Markov chains $\{s_t\}_{t\geq 0}$ and $\{(s_t, a_t)\}_{t\geq 0}$ induced by π_θ are both geometrically ergodic, with unique invariant measures denoted by $\rho_\theta(ds)$ and $\widetilde{\rho}_\theta(ds, da) = \rho_\theta(ds) d\pi_\theta(a \,|\, s)$, respectively.*

The collective goal of the networked agents is to maximize the *globally* averaged long-term return over the network, using only *local* information available to each agent. Specifically, the goal is to solve the following optimization problem

$$\begin{aligned}\max_\theta \quad J(\theta) &= \lim_{T\to\infty} \frac{1}{T}\mathbb{E}\bigg(\sum_{t=0}^{T-1} \frac{1}{N}\sum_{i\in\mathcal{N}} r_{t+1}^i\bigg) \\ &= \int_{\mathcal{S}} \rho_\theta(ds) \int_{\mathcal{A}} d\pi_\theta(a \,|\, s) \cdot \overline{R}(s,a), \quad (2)\end{aligned}$$

where $\overline{R}(s,a) = N^{-1} \cdot \sum_{i\in\mathcal{N}} R^i(s,a)$ denotes the globally averaged reward function, and it holds that $\overline{R}(s,a) = \mathbb{E}[\overline{r}_{t+1} \,|\, s_t = s, a_t = a]$, with $\overline{r}_t = N^{-1} \cdot \sum_{i\in\mathcal{N}} r_t^i$. The form of (2) is well defined if the function $\overline{R}$ is bounded [20]. Accordingly, we define the global relative action-value function [21] under policy π_θ as

$$Q_\theta(s,a) = \sum_{t=0}^{\infty} \mathbb{E}\big[\overline{r}_{t+1} - J(\theta) \,|\, s_0 = s, a_0 = a, \pi_\theta\big]. \quad (3)$$

For simplicity, we hereafter refer to Q_θ as *action-value* function only.

B. Expected Policy Gradient for MARL

To find the optimal joint policy that maximizes the return (2), we need to establish the closed-form expression for the gradient of $J(\theta)$. As compared in Section I, EPG inherits the advantage of both SPG and DPG, i.e., it can be used for *on-policy* learning in continuous action spaces with *lower gradient variance* [16]. Hence, it is worth extending the EPG to our networked MARL setting. We thus establish the form of expected policy gradient for MARL as follows.

Theorem 1. *[Expected Policy Gradient for MARL] Let π_θ be a joint policy and $J(\theta)$ be the globally long-term averaged return defined in* (2)*. Also, let Q_θ be the action-value function. Then, the gradient of $J(\theta)$ with respect to θ^i is given by*

$$\nabla_{\theta^i} J(\theta) = \int_{\mathcal{S}\times\mathcal{A}^{-i}} \rho_\theta(ds) d\pi_{\theta^{-i}}(a^{-i} \,|\, s) I^Q_{\theta^i}(s, a^{-i}), \quad (4)$$

where a^{-i} and $\pi_{\theta^{-i}}$ denote the actions and policies of all agents except for agent i, and $\mathcal{A}^{-i} = \prod_{j\neq i} \mathcal{A}^j$. Moreover, $I^Q_{\theta^i}(s, a^{-i})$ denotes the integral

$$I^Q_{\theta^i}(s, a^{-i}) = \int_{\mathcal{A}^i} d\pi_{\theta^i}(a^i \,|\, s) \nabla_{\theta^i} \log \pi_{\theta^i}(a^i \,|\, s) Q_\theta(s, a^i, a^{-i}).$$

The proof of Theorem 1 is straightforward and thus relegated to [22]. By Theorem 1, we can explicitly use the estimates of the integral $I^Q_{\theta^i}$ as samples of the policy gradient. Compared with the conventional SPG [15], one can greatly reduce the variance of gradient estimation [16] with EPG. Also, this general EPG unifies the SPG and the DPG in the sense that they correspond to different choices of calculating the quadrature $I^Q_{\theta^i}$. With EPG, a deterministic off-policy approach is no longer required to obtain a method with low variance [16], especially in continuous action spaces.

In fact, DPG with off-policy explorations should be preferred here, since it was developed particularly to handle continuous action spaces, and has achieved empirical success [17]. However, this is not viable in our MARL setting when the off-policies of other agents are unrevealed to each agent. Thus we use the on-policy EPG here instead. Nevertheless, we show in the following corollary that under Gaussian policies, a common family of parametrized policies for continuous actions, the quadrature $I^Q_{\theta^i}$ can be related to the off-policy DPG.

Corollary 1. *[Gaussian Policy Gradient for MARL] Let the policy for each agent i be Gaussian, i.e., $\pi_{\theta^i}(\cdot \,|\, s) \sim \mathcal{N}(\eta_{\theta^i}(s), \Sigma^i)$ with $\eta_{\theta^i}(s) \colon \mathcal{S} \to \mathcal{A}^i \in \mathbb{R}^{c^i}$ parametrized by θ^i, and the global action-value function Q_θ has the form $Q_\theta(s, a) = a^\top E(s) a + a^\top F(s) + c$ for some symmetric matrix $E(s)$ and constant c. Then, the quadrature $I^Q_{\theta^i}(s, a^{-i})$ has the following form*

$$\begin{aligned} I^Q_{\theta^i}(s, a^{-i}) &= \nabla_{\theta^i} \eta_{\theta^i}(s) \big[2E^{i,i}(s)\eta_{\theta^i}(s) + \widetilde{E}^{i,i}(s) a^{-i} + F^i(s) \big], \\ &= \nabla_{\theta^i} \eta_{\theta^i}(s) \cdot \nabla_{a^i} Q_\theta(s, a^i, a^{-i}) \Big|_{a^i = \eta_{\theta^i}(s)} \quad (5) \end{aligned}$$

where $-i$ denotes all the indices except agent i, and $E^{i,j}(s)$ denotes the submatrix of E whose rows and columns correspond to the indices of agent i and j, respectively. $\widetilde{E}^{i,i}(s) = E^{i,-i}(s) + [E^{-i,i}(s)]^\top$, and F^i are the rows of F that correspond to the indices of agent i.

Due to space limitation, the proof of Corollary 1 is deferred to the full version [22]. Corollary 1 shows that with Gaussian policies and quadratic form of action-value function in action a, the EPG $I^Q_{\theta^i}(s, a^{-i})$ evaluated at each agent i is equivalent to the DPG with Gaussian exploration policy (see Section 4.2 in [17]), if all agents use this on-policy EPG update.

III. Actor-Critic for Networked Agents

In this section, we propose a multi-agent actor-critic algorithm for networked agents, based on the expected policy gradient for MARL from Theorem 1.

As shown in Theorem 1, an unbiased estimate of the integral $I^Q_{\theta^i}(s, a^{-i})$ can serve as a sample of global policy gradient with respect to the local policy parameter θ^i. This way, the joint policy improvement can be implemented in a *fully decentralized* fashion. However, the estimation of $I^Q_{\theta^i}(s, a^{-i})$ requires not only the local policy π_{θ^i}, but also the global value function Q_θ, while the latter cannot be updated locally with only the reward r^i_t of agent i. This motivates us to develop an MARL algorithm that uses consensus updates for each agent to estimate a common global Q_θ.

To handle continuous state-action spaces, the action-value function Q_θ is usually approximated by $Q(\cdot, \cdot; \omega) \colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, which is a class of functions parametrized by $\omega \in \mathbb{R}^K$. Each agent i keeps the local estimate $Q(\cdot, \cdot; \omega^i)$ of Q_θ by updating the parameter ω^i. Since the evaluation of Q_θ in (3) involves the globally averaged reward $\overline{r}_t$, a global information, we let each agent i share the local parameter ω^i with its neighbors on the network. This way, a consensual estimate of Q_θ may be reached, and each agent can perform its own policy improvement via Theorem 1.

In particular, we propose an actor-critic algorithm with two steps that operate on different time scales. The critic step is updated on a faster time scale, where each agent first performs temporal-difference (TD) learning for policy evaluation [23], i.e., to estimate the action-value function parameter ω^i. Then, the estimated parameter is shared with its neighbors by a consensus update. At time t, we use the weight matrix $\mathrm{C}_t = [c_t(i,j)]_{N\times N}$ with entries $c_t(i,j)$ to denote the weight on the message sent from agent i to j. More details on the design of C_t are provided in Section IV. Thus, the critic step has the following update rule

$$\mu^i_{t+1} = (1 - \beta_{\omega,t}) \cdot \mu^i_t + \beta_{\omega,t} \cdot r^i_{t+1}, \quad (6)$$

$$\widetilde{\omega}^i_t = \omega^i_t + \beta_{\omega,t} \cdot \delta^i_t \cdot \nabla_\omega Q_t(\omega^i_t), \quad (7)$$

$$\omega^i_{t+1} = \sum_{j\in\mathcal{N}} c_t(i,j) \cdot \widetilde{\omega}^j_t, \quad (8)$$

where $\beta_{\omega,t} > 0$ is the stepsize and μ^i_t tracks the long-term return of agent i. To simplify the notation, we use $Q_t(\omega)$ to denote $Q(s_t, a_t; \omega)$ for any $\omega \in \mathbb{R}^K$. The local TD-error δ^i_t in (6) is computed as

$$\delta^i_t = r^i_{t+1} - \mu^i_t + Q_{t+1}(\omega^i_t) - Q_t(\omega^i_t). \quad (9)$$

The actor step, which is motivated by (4), performs on a slower time scale. Each agent i improves its policy following

$$\theta^i_{t+1} = \theta^i_t + \beta_{\theta,t} \cdot \widehat{I}^{Q,i}_t, \quad (10)$$

where

$$\widehat{I}_t^{Q,i} = \widehat{I}_t^{Q,i}(s_t, a_t^{-i}) \tag{11}$$
$$= \int_{\mathcal{A}^i} d\pi_{\theta_t^i}(a^i \mid s_t)\nabla_{\theta^i}\log\pi_{\theta_t^i}(a^i \mid s_t)Q(s_t, a^i, a_t^{-i}; \omega_t^i),$$

and $\beta_{\theta,t} > 0$ is the stepsize. We use $\widehat{I}_t^{Q,i}$ instead of $I_t^{Q,i}$ here since the action-value function is approximated by $Q(\cdot,\cdot;\omega_t^i)$. Under the conditions for Corollary 1, i.e., the polices are Gaussian $\pi_{\theta^i}(\cdot \mid s) \sim \mathcal{N}(\eta_{\theta^i}(s), \Sigma^i)$, and $Q(s, a; \omega)$ is quadratic with respect to a, the update (11) has the form

$$\widehat{I}_t^{Q,i} = \nabla_{\theta^i}\eta_{\theta_t^i}(s_t) \cdot \nabla_{a^i} Q(s_t, a^i, a_t^{-i}; \omega_t^i) \,\big|_{\, a^i = \eta_{\theta_t^i}(s_t)}.$$

Details of the proposed actor-critic algorithm are summarized in Algorithm 1, whose pseudocode can be found in [22].

We note that Algorithm 1 is applicable to general function approximators, e.g., deep neural networks. Furthermore, with linear function approximation, the convergence guarantees can be established as in Section IV.

IV. Convergence Results

In this section, convergence results of Algorithm 1 are presented. The complete proofs of the results are relegated to [22, Section V]. We start with the following assumptions.

Assumption 2. *The update of the policy parameter θ_t^i includes a local projection operator, $\Gamma^i : \mathbb{R}^{m_i} \to \Theta^i \subset \mathbb{R}^{m_i}$, that projects any θ_t^i onto a compact set Θ^i. Also, $\Theta = \prod_{i=1}^N \Theta^i$ is large enough to include at least one local minimum of $J(\theta)$.*

Assumption 3. *The instantaneous reward r_t^i is uniformly bounded for any $i \in \mathcal{N}$ and $t \geq 0$.*

This projection technique is standard in the analysis of stochastic approximation (SA) for RL algorithms [14], [24].

Then, we need some conditions on the weight matrix $\{\mathrm{C}_t\}_{t\geq 0}$ for the consensus update, which have been justified in our previous work on MARL over networks [24].

Assumption 4. *The sequence of nonnegative random matrices $\{\mathrm{C}_t\}_{t\geq 0} \subseteq \mathbb{R}^{N\times N}$ satisfies:*

(a.1) C_t is row stochastic and $\mathbb{E}(\mathrm{C}_t)$ is column stochastic. That is, $\mathrm{C}_t\mathbb{1} = \mathbb{1}$ and $\mathbb{1}^\top\mathbb{E}(\mathrm{C}_t) = \mathbb{1}^\top$. Furthermore, there exists a constant $\eta \in (0, 1)$ such that, for any $c_t(i, j) > 0$, we have $c_t(i, j) \geq \eta$.

(a.2) C_t respects the communication graph $\mathcal{G}_t$, i.e., $c_t(i, j) = 0$ if $(i, j) \notin \mathcal{E}_t$. . Moreover, the spectral norm of $\mathbb{E}[\mathrm{C}_t^\top \cdot (\mathrm{I} - \mathbb{1}\mathbb{1}^\top/N) \cdot \mathrm{C}_t]$ is strictly smaller than one.

(a.3) Given the σ-algebra generated by the random variables before time t, C_t is conditionally independent of r_{t+1}^i for any $i \in \mathcal{N}$.

Moreover, we need the following assumption on the action-value function approximation, which has been used previously to establish the convergence of actor-critic algorithms [14], [24].

Assumption 5. *For each agent i, the action-value function is approximated by linear functions, i.e., $Q(s, a; \omega) = \omega^\top\phi(s, a)$ where $\phi(s, a) = [\phi_1(s, a), \cdots, \phi_K(s, a)]^\top \in \mathbb{R}^K$ is the feature associated with (s, a), and $\phi_k : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the feature function for each $k \in [K]$. The feature $\phi_k(\cdot,\cdot)$ is bounded for any $s \in \mathcal{S}, a \in \mathcal{A}$. Furthermore, the feature functions $\{\phi_k\}_{k\in[K]}$ are linearly independent, and for any $u \in \mathbb{R}^K$ and $u \neq 0$, $u^\top\phi$ is not constant over $\mathcal{S} \times \mathcal{A}$.*

Additionally, we need the assumption on the stepsizes $\beta_{\omega,t}$ and $\beta_{\theta,t}$, which separates the actor and critic steps in two time scales as in single-agent AC algorithms.

Assumption 6. *The stepsizes $\beta_{\omega,t}$ and $\beta_{\theta,t}$ satisfy*

$$\sum_t \beta_{\omega,t} = \sum_t \beta_{\theta,t} = \infty, \quad \sum_t \beta_{\omega,t}^2 + \beta_{\theta,t}^2 < \infty.$$

Also, $\beta_{\theta,t} = o(\beta_{\omega,t})$ and $\lim_{t\to\infty} \beta_{\omega,t+1} \cdot \beta_{\omega,t}^{-1} = 1$.

In view of this two-time-scale update, we first present the convergence of the critic step. To this end, we assume the joint policy π_θ to be fixed on this faster time scale. Based on that, we establish convergence of the actor step, i.e., the policy parameter θ_t, on the slower time scale. For notational simplicity, we define P_θ as the operator over any matrix of functions $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{q\times p}$ as

$$(P_\theta Q)(s, a) = \int_{\mathcal{S}\times\mathcal{A}} Q(s', a')p(ds' \mid s, a)d\pi_\theta(a' \mid s'),$$

and also the operator T_θ as

$$(T_\theta Q)(s, a) = \overline{R}(s, a) - J(\theta) + (P_\theta Q)(s, a), \tag{12}$$

for any $(s, a) \in \mathcal{S} \times \mathcal{A}$. Note that both operators are defined for each element of their argument, i.e., each element in the matrix Q. Thus, it requires $\overline{R} \in \mathbb{R}^{q\times p}$ and $(P_\theta Q)(\cdot,\cdot), (T_\theta Q)(\cdot,\cdot) \in \mathbb{R}^{q\times p}$. Moreover, recall that $\widetilde{\rho}_\theta(ds, da)$ is the invariant measure of the Markov chain $\{(s_t, a_t)\}_{t\geq 0}$. Then we define the inner product with respect to this measure between any two matrices of functions $F_1 : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{q_1\times p}$ and $F_2 : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{p\times q_2}$ as

$$\langle F_1, F_2\rangle_\theta = \int_{\mathcal{S}\times\mathcal{A}} \widetilde{\rho}_\theta(ds, da)F_1(s, a) \cdot F_2(s, a). \tag{13}$$

Now we are ready to present the convergence result for the critic step (6)-(8).

Theorem 2. *Under Assumptions 1, and 3-6, for any given policy π_θ, if $\{\mu_t^i\}$ and $\{\omega_t^i\}$ are generated from (6)-(8), then it follows that $\lim_t \sum_{i\in\mathcal{N}} \mu_t^i \cdot N^{-1} = J(\theta)$ and $\lim_t \omega_t^i = \omega_\theta$ almost surely (a.s.) for any $i \in \mathcal{N}$, where $J(\theta)$ is the globally long-term averaged return as defined in (2), and ω_θ is the unique solution to*

$$\Big\langle \phi, T_\theta\big(\omega_\theta^\top\phi\big) - \omega_\theta^\top\phi \Big\rangle_\theta = 0. \tag{14}$$

Theorem 2 characterizes the convergent point of our consensus-based critic step. In fact, it is precisely the point that the TD(0) algorithm [23] converges to, for action-value function evaluation. Note that in the setting with continuous spaces, one cannot write the convergence condition (14) in a compact matrix form as in [14]. But still, we claim that the solution to (14) is the solution to the fixed-point problem

$$\omega_\theta^\top\phi = \Pi T_\theta\big(\omega_\theta^\top\phi\big), \tag{15}$$

where Π denotes the projection of the argument onto the span of the basis functions $\{\phi_k\}_{k\in[K]}$. To see this, consider the projection of any $J\colon \mathcal{S}\times\mathcal{A}\to\mathbb{R}$ as ΠJ, which is defined as the solution to

$$\Pi J = \underset{\bar{J}\in\{\omega^\top\phi\},\omega\in\mathbb{R}^K}{\operatorname{argmin}} \|J-\bar{J}\|_\theta, \tag{16}$$

where $\|\cdot\|_\theta$ is the norm[1] induced by the inner product, $\|\cdot\|_\theta = \sqrt{\langle\cdot,\cdot\rangle_\theta}$. Then an optimal ω^* makes $(\omega^*)^\top\phi$ the solution to (16) if and only if it satisfies

$$\big\langle \phi, J-(\omega^*)^\top\phi\big\rangle_\theta = 0. \tag{17}$$

Comparing (17) and (14), we obtain that $\omega_\theta^\top\phi$ is essentially the projection $\Pi T_\theta\big(\omega_\theta^\top\phi\big)$, the same result as in the finite space setting [23].

To show the convergence of the actor step, we define the quantity $\widehat{I}^{Q,i}_{t,\theta}$ as

$$\begin{aligned}&\widehat{I}^{Q,i}_{t,\theta} = \widehat{I}^{Q,i}_{t,\theta}(s_t,a_t^{-i}) &(18)\\ &= \int_{\mathcal{A}^i} d\pi_{\theta^i}(a^i\,|\,s_t)\nabla_{\theta^i}\log\pi_{\theta^i}(a^i\,|\,s_t)\cdot\omega_\theta^\top\phi(s_t,a^i,a_t^{-i}),\end{aligned}$$

where we substitute the explicit form of $Q(\cdot,\cdot;\omega)$ in the definition of $I_t^{Q,i}$ in (11), and replace θ_t and ω_t^i as θ and ω_θ, respectively. Also, we define a vector $\hat{\Gamma}^i(\cdot)$ as

$$\hat{\Gamma}^i[g(\theta)] = \lim_{0<\eta\to 0}\big\{\Gamma^i[\theta^i+\eta\cdot g(\theta)]-\theta^i\big\}/\eta \tag{19}$$

for any $\theta\in\Theta$ and any continuous function $g\colon\Theta\to\mathbb{R}^{\sum_{i\in\mathcal{N}}m_i}$. Note that $\hat{\Gamma}^i[g(\theta)]$ can be a set of all limit points of (19). Then the convergence of Algorithm 1 can be stated as follows.

Theorem 3. *Under Assumptions 1-6, if the policy parameter $\{\theta_t^i\}$ is generated from* (10)*, then it follows that θ_t^i converges almost surely to a point in the set of asymptotically stable equilibria of*

$$\dot{\theta}^i = \hat{\Gamma}^i\bigg[\int_{\mathcal{S}\times\mathcal{A}^{-i}}\rho_\theta(ds_t)d\pi_{\theta^{-i}}(a_t^{-i}\,|\,s_t)\cdot\widehat{I}^{Q,i}_{t,\theta}\bigg], \tag{20}$$

for any $i\in\mathcal{N}$.

Theorem 3 shows that Algorithm 1 converges to the projection of some stationary point onto the compact set Θ. Since we do not restrict our feature functions to be the *compatible* features as in [23], [15], the equilibria of (20) is the best one can hope for any single-agent AC algorithms that use general linear function approximators [14]. See more discussions on this in the full version [22].

V. Numerical Results

Consider in total $N=20$ agents, operating as a nonlinear multi-agent system that is governed by $s_{t+1} = \varphi|s_t| + B^\top a + (1-\varphi^2)^{1/2}\xi_{t+1}$, where $|\varphi|<1$, $\xi_t\sim\mathcal{N}(0,1)$, and $B\in\mathbb{R}^N$ is selected randomly from $[0,1]^N$. We see that without control, i.e., when $B=0$, the Markov chain $\{s_t\}_{t\geq 0}$ has a unique stationary distribution [25]. Here we choose $\varphi=0.9$. The action of each agent is continuous and is sampled from a parametrized policy $\mathcal{N}\big(\eta_{\theta^i}(s),(\sigma^i)^2\big)$, where $\eta_{\theta^i}(s) = (\theta^i)^\top\psi^i(s)$ and $\psi^i(s)\in\mathbb{R}^{m^i}$ is the policy feature function of agent i. We let $m^1=\cdots=m^N=5$ for all $i\in\mathcal{N}$ and select ψ^i to be the Gaussian radial basis functions (RBF) with their means randomly selected from $[0,1]$, and their variances selected as 0.1. The exploration noise $\sigma^i=0.1,\forall i\in\mathcal{N}$. The reward function of each agent is $R^i(s,a)=k_0^i+k_a^i\cdot(a^i)^2+k_s^i\cdot s^2$, where the coefficient-tuple (k_0^i,k_a^i,k_s^i) is selected randomly from $[0,1]^3$, and is thus different for all agents i over the network. The instantaneous reward r_t^i is sampled from a uniform distribution $[R^i(s,a)-0.5,R^i(s,a)+0.5]$. As suggested by Corollary 1, we approximate the action-value function by a quadratic function in a, and also linear in the parameter ω following Assumption 5, i.e., $Q(s,a;\omega)=\omega_1\cdot a^\top E(s)a + a^\top F(s)\cdot\omega_{2:K-1}+\omega_K$, where $\omega_{2:K-1}$ denotes the 2 to $K-1$ rows of ω, and thus $\omega=(\omega_1,\omega_{2:K-1}^\top,\omega_K)^\top\in\mathbb{R}^K$. We choose $K=5$. This way, the EPG can be evaluated efficiently by Corollary 1. The feature functions $E(s)\in\mathbb{R}^{N\times N}$ and $F(s)\in\mathbb{R}^{N\times(K-2)}$ are also Gaussian RBFs with their means randomly selected from $[0,1]$, and their variances set as 0.1.

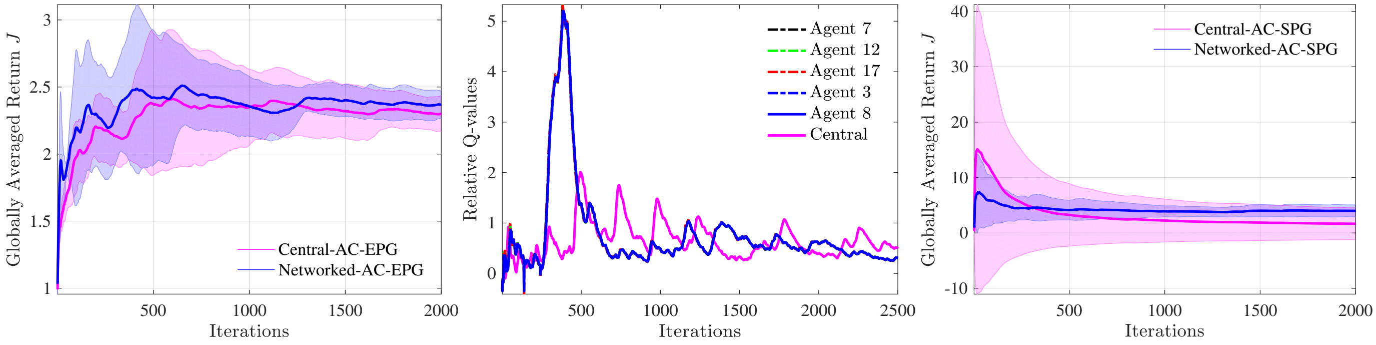
Moreover, we design the consensus weight matrix C_t by normalizing the absolute Laplacian matrix of an i.i.d. random graph $\mathcal{G}_t$ to be doubly stochastic. We generate the random graph $\mathcal{G}_t$ by randomly placing communication links among agents such that $\mathcal{G}_t$ has connectivity ratio[2] $4/N$. To satisfy Assumption 6, the stepsizes are selected as $\beta_{\omega,t}=1/t^{0.65}$ and $\beta_{\theta,t}=1/t^{0.85}$.

We first compare the performance of the fully decentralized algorithm with that of the centralized counterpart, where the rewards r_t^i are available at a certain central controller. The controller then updates the global policy from the rewards collected, using a single-agent AC algorithm. Here we also use EPG-based actor step in the single-agent AC, and the critic step is the basic TD(0) update. We run each algorithm 20 times and plot the mean-variance figure of the globally averaged reward in Fig. 1a. We see that our fully decentralized algorithm successfully converges to the globally averaged return achieved by the centralized AC algorithm, although with a slightly greater variance. This is possibly due to the additional process of reaching consensus among the agents. The convergence of the action-value function at 5 randomly selected agents is illustrated in Fig. 1b. It shows that the estimate of Q_θ indeed reaches consensus and converges to the estimate obtained by the central controller, at a relatively slower rate. This is possibly due to the delay of information diffusion across the network. It is also corroborated that the action-value function reaches consensus much faster than the AC algorithm converges, thanks to the separated time scales of the algorithm.

We then compare the performance with the algorithms

[1] Note that the definition of $\langle\cdot\rangle$ in (13) also applies to F_1 and F_2 with different dimensions. To obtain a valid norm $\|\cdot\|_\theta$, when defining the norm, we restrict the function only to have dimension 1, i.e., $q_1=p=1$ for $F_1\colon\mathcal{S}\times\mathcal{A}\to\mathbb{R}^{q_1\times p}$ and thus $\|F_1\|_\theta=\sqrt{\langle F_1,F_1\rangle_\theta}$.

[2] The connectivity ratio here is defined as the ratio between the total degree of the graph and the degree of the complete graph, i.e., $2E/[N(N-1)]$, where E is the number of edges.



(a) Globally averaged return using EPG (b) Relative action-values using EPG (c) Globally averaged return using SPG

Fig. 1: Comparison of the convergence of several algorithms. Fig. 1a shows the convergence of the proposed decentralized AC algorithm, i.e., Algorithm 1, referred to as *Networked-AC-EPG*, and the centralized counterpart, both based on the EPG actor step. Fig. 1b shows the convergence of the estimated relative action-value functions Q_θ at 5 randomly selected agents. Fig. 1c shows the convergence of the decentralized AC algorithm, referred to as *Networked-AC-SPG*, and the centralized counterpart, both based on the SPG actor step.

built upon SPG, as developed in [24] for finite space MDPs. It is demonstrated in Fig. 1c that the SPG-based AC algorithms have much larger variance than the EPG-based ones. This justifies the effectiveness of our EPG-based AC algorithm for MARL in continuous spaces.

VI. Conclusions

In this paper, we have studied the problem of networked multi-agent reinforcement learning in continuous spaces. In particular, we have considered a *fully decentralized* setting where each agent makes individual decisions and receives local rewards, while exchanging information with neighbors over the network to accomplish optimal network-wide averaged return. To this end, we have developed an actor-critic algorithm for networked MARL in continuous state and action spaces, which can apply to practical multi-agent control systems. We have provided theoretical analysis on the convergence of the proposed algorithm, which is corroborated through numerical simulations.

References

[1] K. Zhang, W. Shi, H. Zhu, E. Dall'Anese, and T. Basar, "Dynamic power distribution system management with a locally connected communication network," *IEEE Journal of Selected Topics in Signal Processing*, 2018.

[2] P. Corke, R. Peterson, and D. Rus, "Networked robots: Flying robot navigation using a sensor net," *Robotics Research*, pp. 234–243, 2005.

[3] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1465–1476, 2004.

[4] J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative multi-agent control using deep reinforcement learning," in *International Conference on Autonomous Agents and Multi-agent Systems*, 2017, pp. 66–83.

[5] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *arXiv preprint arXiv:1706.02275*, 2017.

[6] X. Wang and T. Sandholm, "Reinforcement learning to play an optimal Nash equilibrium in team Markov games," in *Advances in Neural Information Processing Systems*, 2003, pp. 1603–1610.

[7] G. Arslan and S. Yüksel, "Decentralized Q-learning for stochastic teams and games," *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1545–1558, 2017.

[8] E. Dall'Anese, H. Zhu, and G. B. Giannakis, "Distributed optimal power flow for smart microgrids," *IEEE Transactions on Smart Grid*, vol. 4, no. 3, pp. 1464–1475, 2013.

[9] J. L. Adler and V. J. Blue, "A cooperative multi-agent transportation management and route guidance system," *Transportation Research Part C: Emerging Technologies*, vol. 10, no. 5, pp. 433–454, 2002.

[10] K. Zhang, L. Lu, C. Lei, H. Zhu, and Y. Ouyang, "Dynamic operations and pricing of electric unmanned aerial vehicle systems and power networks," *Transportation Research Part C: Emerging Technologies*, vol. 92, pp. 472–485, 2018.

[11] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[12] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4289–4305, 2012.

[13] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," in *Advances in Neural Information Processing Systems*, 2000, pp. 1008–1014.

[14] S. Bhatnagar, R. Sutton, M. Ghavamzadeh, and M. Lee, "Natural actor-critic algorithms," *Automatica*, vol. 45, no. 11, pp. 2471–2482, 2009.

[15] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in Neural Information Processing Systems*, 2000, pp. 1057–1063.

[16] K. Ciosek and S. Whiteson, "Expected policy gradients for reinforcement learning," *arXiv preprint arXiv:1801.03326*, 2018.

[17] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *International Conference on Machine Learning*, 2014, pp. 387–395.

[18] S. Kar, J. M. Moura, and H. V. Poor, "QD-learning: A collaborative distributed strategy for multi-agent reinforcement learning through consensus + innovations," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1848–1862, 2013.

[19] S. V. Macua, A. Tukiainen, D. G.-O. Hernández, D. Baldazo, E. M. de Cote, and S. Zazo, "Diff-dac: Distributed actor-critic for multitask deep reinforcement learning," *arXiv preprint arXiv:1710.10363*, 2017.

[20] S. P. Meyn and R. L. Tweedie, *Markov Chains and Stochastic Stability*. Springer Science & Business Media, 2012.

[21] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

[22] K. Zhang, Z. Yang, and T. Başar, "Networked multi-agent reinforcement learning in continuous spaces," Tech. Rep., 2018. [Online]. Available: https://www.dropbox.com/s/c0ng361kibqobk9/Tech_Report_Linked.pdf?dl=0

[23] J. N. Tsitsiklis and B. Van Roy, "Analysis of temporal-diffference learning with function approximation," in *Advances in Neural Information Processing Systems*, 1997, pp. 1075–1081.

[24] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Başar, "Fully decentralized multi-agent reinforcement learning with networked agents," in *International Conference on Machine Learning*, 2018, pp. 5872–5881.

[25] K.-S. Chan, "Consistency and limiting distribution of the least squares estimator of a threshold autoregressive model," *The Annals of Statistics*, pp. 520–533, 1993.