Available online at www.sciencedirect.com
ScienceDirect
IFAC PapersOnLine 52-20 (2019) 363–368

Distributed Multi-Agent Reinforcement Learning by Actor-Critic Method

Paulo C. Heredia, Shaoshuai Mou

Purdue University, West Lafayette, IN 47906 USA
(e-mail: pheredia@ purdue.edu, mous@purdue.edu)

Abstract: We investigate the problem of multi-agent reinforcement learning, in which each agent only has access to its local reward and can only communicate with its nearby neighbors. A distributed algorithm based on actor-critic method has been developed to enable all agents to cooperatively learn a control policy that maximizes the global objective function. Simulations are also provided to validate the proposed algorithm.

1. INTRODUCTION

Multi-agent reinforcement learning (MARL) has recently gained a lot research attention with extensive applications into mobile sensor networks, robotics, UAV swarms, cybersecurity, and so on Malialis et al. (2015). Research challenges in MARL mainly come from the fact that each agent has its own local and private reward, and can only coordinate with nearby agents, which usually result in conflicts with other agents in credit assignment and coordinating actions Sunehag et al. (2018). This has led to a recently booming area of developing distributed algorithms for MARL, in which there is no centralized coordinator and only local coordination among nearby neighbors are allowed. Early results in the direction of distributed MARL usually assume finite states and actions to allow them to implement a tabular form of reinforcement learning Schneider et al. (1999); Tham and Renaud (2005), which are not applicable to situations requiring infinite states and actions. Further progress has been achieved in Wai et al. (2018); Lee et al. (2018); Kar et al. (2013); Mathkar and Borkar (2017) which only consider evaluation of fixed policies and cannot be immediately used to develop optimal control policies. Recently researchers have started to develop distributed MARL based on actor-critic methods in single-agent case in Sutton et al. (2000); Konda and Tsitsiklis (2000). It has recently been shown that critic training could be reformulated as a primal-dual optimization problem in single-agent case in Dai et al. (2018), with further generalization to distributed MARL algorithm in the worst-case by Wai et al. (2018), followed by a finite sample analysis in Yang et al. (2018). Perhaps one of the most significant progress in distributed MARL based on actor-critic method are algorithms developed in Zhang et al. (2018b,a), in which each gent makes its own decision only based on locally observed information and communication among nearby neighbors, and the network connecting agents are time-varying.

Motivated by Zhang et al. (2018b,a), we in this paper also develop a distributed algorithm for MARL, based on actor-critic methods. With this framework each agent is tasked with training an actor to generate a control input given the state, and a critic to output a scalar value for the performance of the current policy, given a state and input pair. In addition, we consider continuous states/actions as in Zhang et al. (2018a) but with a different variation of the actor-critic algorithm. Different from Zhang et al. (2018b), which considers the expected time-averaged reward and finite spaces for states/actions, we in this paper consider the expected sum of discounted rewards over an infinite time horizon. Under results developed in this paper, the policy evaluation algorithm proposed in Wai et al. (2018) can be used for action-value functions as well as state-value functions, which in turn implies that such policy evaluation algorithm can potentially be used in a distributed actor-critic framework based on Zhang et al. (2018b).

Notation Let ∇_a denote the gradient with respect to a parameter a. To indicate the transpose of a matrix A, we use $A^\top$. Furthermore, by $\{a(t)\}$ we mean a sequence of $a(t)$ and by $a \sim d$ we mean "a is sampled from the distribution d". We also use $col\{a_1, a_2, .., a_n\}$ to denote the column-wise stacking of $a_1, .., a_n$.

2. PROBLEM FORMULATION

Consider the case in which a network of m autonomous agents operate in an unknown environment (or plant). Let $x(t) \in \mathbb{R}^n$ denote the state of the plant at time t. For each control input $u_i(t)$ from agent i to the plant, a local reward $r_i(x(t), u(t))$ is produced, where $u(t) = col\{u_1(t), u_2(t), ..., u_m(t)\} \in \mathbb{R}^{\bar{n}}$. Here, each $r_i(\cdot)$ is the private reward locally accessible to only agent i, and is not shared with other agents. Let

$$R(x(t), u(t)) = \sum_{i=1}^{m} \frac{1}{m} r_i(x(t), u(t)) \tag{1}$$

which represents the average reward of all agents in the network. Let π denote a stochastic control policy such that $u \sim \pi(x, u)$. Let Q_π denote the corresponding objective function, which is assumed to be a sum of discounted rewards $R(x(t), u(t))$ when a stochastic control policy π is applied to the plant. Namely,

⋆ This research work was supported by funding from Northrop Grumman Corporation (NGC-REALM and NGCRC).

10.1016/j.ifacol.2019.12.182

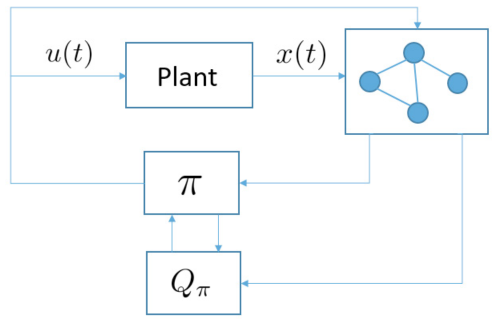

Fig. 1. Distributed Multi-Agent Reinforcement Learning

$$Q_\pi(x(t), u(t)) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R(x(k), u(k)) \middle| x(0) = x(t), u(0) = u(t)\right]$$

where $\gamma \in (0, 1)$ is a discount factor. The goal of MARL in this paper is to achieve a globally optimal control policy π^* to maximize the objective function Q_π.

In a multi-agent network, each agent i usually can only communicate with certain neighboring agents denoted by $\mathcal{N}_i$, which includes agent i. The neighbor relations can be modeled by a connected undirected graph $\mathbb{G}$ such that there is an edge between i and j if and only if i and j are neighbors. Suppose each agent i controls π_i (an estimate to the optimal control policy π^*) and Q_i (the Q_{π_i} corresponding to π_i). The **problem** of interest, as indicated in Fig. 1, is to develop a distributed algorithm such that each agent achieves an ϵ approximation of the optimal policy π^* (denoted by $\pi^* \pm \epsilon$), as well as its corresponding action value function $Q_{\pi^* \pm \epsilon}$, using only coordination with its nearby neighbors, namely,

$$\pi_i \to \pi^* \pm \epsilon \tag{2}$$
$$Q_i \to Q_{\pi^* \pm \epsilon}. \tag{3}$$

Here, $\pi^* \pm \epsilon$ denotes a policy value in the interval $[\pi^* - \epsilon, \pi^* + \epsilon]$.

3. THE UPDATE

In this section we will develop a distributed algorithm for MARL by introducing an actor and a critic at each agent. That is, each agent is tasked with training an actor to generate a control input given the state (control policy) and a critic to output a scalar value for the performance of the current policy given a state and input pair (action-value function). In the following we will present the updates for both critic training and actor training.

3.1 Critic Training

We first assume π is fixed, and so the proposed approach is to train each agent's critic to converge to Q_π

As is well known, the Bellman equation for reinforcement learning can be described in terms of Q_π as follows Sutton and Barto (2018):

$$Q_\pi(x(t), u(t)) = R(x(t), u(t)) + \gamma \mathbb{E}_{x|x(t),u(t)}\left[V_\pi(x(t+1))\right],$$

where

$$V_\pi(x(t+1)) = \mathbb{E}_{u(t+1)|\pi}\left[Q_\pi(x(t+1), u(t+1))\right] \tag{4}$$

is the state-value function at $t + 1$. The above Bellman equation can be used to directly compute the entries in Q_π, which is however not directly applicable to continuous space of actions and states. To address this, we approximate Q_π as linear combination of given basis functions Tadić (2001), that is,

$$Q_w(x, u) = w^\top \phi(x, u), \tag{5}$$

where $w \in \mathbb{R}^{q_1}$ is unknown and $\phi(x, u) \in \mathbb{R}^{q_1}$ is a column vector of basis functions.

Similarly, the control policy π can also be approximated as a parameterized function π_θ, where $\theta \in \mathbb{R}^p$ This can be achieved by defining π_θ as a normal distribution with mean and standard deviation as functions of θ.

Let $\mathbf{R}$, $\mathbf{Q_w}$ and $\mathbf{V}_\pi$ denote the vectors from stacking all R in (1), Q_w in (5), and V_π in (4), respectively, for every (x, u) pair. To ease notation we refer to $\mathbb{E}_{u|\pi}$ as $\mathbb{E}_u$ and $\mathbb{E}_{x|x(t),u(t)}$ as $\mathbb{E}_x$. Then a nice estimate of Q_π can be achieved by minimizing the following mean squared projected bellman error(MSPBE) with respect to w Wai et al. (2018), namely,

$$\begin{aligned} &\min_w \text{MSPBE}(w) = \\ &\min_w \frac{1}{2}\|\mathbf{\Pi_\Phi}(\mathbf{Q_w} - \mathbf{R} - \gamma\mathbb{E}_x\left[\mathbf{V}_\pi\right])\|^2_{\mathbf{D}} + \rho\|w\|^2, \end{aligned} \tag{6}$$

where $\mathbf{D} = \text{diag}[\{\mu_{\pi_\theta}(x)\forall x \in \mathbb{R}^n\}]$ is a diagonal matrix with the stationary distribution of π_θ on the diagonal; $\mathbf{\Pi_\Phi} = \mathbf{\Phi}(\mathbf{\Phi}^\top\mathbf{D}\mathbf{\Phi})^{-1}\mathbf{\Phi}^\top\mathbf{D}$ is the projection onto the subspace $\{\mathbf{\Phi}w : w \in \mathbb{R}^{q_1}\}$; $\mathbf{\Phi}$ is the stacking of $\phi(x, u)$ for every (x, u) pair,and ρ is a free parameter for regularization of w. From Wai et al. (2018), and assuming $\mathbf{A}$ is invertible, we know this can also be rewritten as

$$\begin{aligned} &\min_w \text{MSPBE}(w) = \\ &\min_w \frac{1}{2}\|\mathbf{\Phi}^\top\mathbf{D}(\mathbf{Q_w} - \mathbf{R} - \gamma\mathbb{E}_x\left[\mathbf{V}_\pi\right])\|^2_{(\mathbf{\Phi}^\top\mathbf{D}\mathbf{\Phi})^{-1}} + \rho\|w\|^2 \\ &= \min_w \frac{1}{2}\|\mathbf{A}w - \mathbf{b}\|^2_{\mathbf{A}^{-1}} + \rho\|w\|^2, \end{aligned}$$

where

$$\mathbf{A} = \mathbb{E}[A(t)], \mathbf{b} = \mathbb{E}[b(t)]$$

with

$$\begin{aligned} A(t) &= \phi(x(t), u(t))\phi(x(t), u(t))^\top \\ b(t) &= (R(x(t), u(t)) + \gamma V_\pi(x(t+1)))\phi(x(t), u(t)) \end{aligned}$$

and $\|v\|_M = \sqrt{v^\top M v}$ for any vector v. Note that $\mathbf{A}$ and $\mathbf{b}$ are usually not available in practice since they are all computed with respect to the stationary distribution of π_θ, which denoted by μ_{π_θ} usually requires the knowledge of state dynamics of the plant. Thus instead of solving (7), we will solve its equivalent problem, as shown in Wai et al. (2018):

$$\min_w \frac{1}{m}\sum_{i=1}^{m} \text{MSPBE}_i(w) \tag{7}$$

where

$$\text{MSPBE}_i(w) = \frac{1}{2}\|\hat{\mathbf{A}}w - \hat{\mathbf{b}}_i\|^2_{\hat{\mathbf{A}}^{-1}} + \rho\|w\|^2, \tag{8}$$

$\hat{\mathbf{A}} = \frac{1}{T}\sum_{t=1}^{T} A(t)$, $\hat{\mathbf{b}}_i = \frac{1}{T}\sum_{t=1}^{T} b_i(t)$, and

$$b_i(t) = (r_i(x(t), u(t)) + \gamma V_\pi(x(t+1)))\phi(x(t), u(t)). \tag{9}$$

Then the problem of learning a good estimate to Q_π can be achieved by solving the optimization problem in (7).

Similar to Wai et al. (2018), we employ the following update at each agent i:

$$w_i(t+1) = \sum_{j=1}^{N} W_{i,j} w_i(t) - \alpha_1 s_i(A(t), t)$$
$$\nu_i(t+1) = \nu_i(t) + \alpha_2 d_i(A(t), b_i(t), t),$$

Here, ν_i is the dual variable of agent i with Metropolis weights $W_{i,j}$ given by

$$W_{i,j} = \begin{cases} \dfrac{1}{\max\{e_i, e_j\}}, \text{ if } j \in \mathcal{N}_i, j \neq i \\ 1 - \displaystyle\sum_{k \in \mathcal{N}_i, k \neq i} W_{i,j} \text{ , if } i = j \\ 0, \text{ if } j \notin \mathcal{N}_i \end{cases},$$

where e_i is the number of neighbors of agent i, which by our definition of $\mathcal{N}_i$ includes agent i. Here, s_i is a surrogate for the gradient of the objective function in (7) with respect to w_i, and likewise d_i is a surrogate for the gradient of the same objective function with respect to ν_i. Through these gradient surrogates, each agent attempts to track the actual gradients of the objective function by using only local information and the estimates of its neighbors. As such, the updates of these surrogates use gradients on the locally available function given by (8), plus the local average of previous estimates (in the case of s_i only). Please refer to Wai et al. (2018) for more details on the definition and updates of these gradient surrogates.

Note that computing $b_i(t)$ at each agent i requires $V_\pi(x)$ as shown in (9), which is related to $Q_w(x, u)$ by (4). Since the expectation $\mathbb{E}_{u(t+1)|\pi}$ in (4) cannot be calculated by each agent i without access to π, we employ a linear function approximation for the state-value function, namely,

$$V_\upsilon(x) = \upsilon^\top \eta(x), \tag{10}$$

where $\upsilon \in \mathbb{R}^{q_2}$ and $\eta(x) \in \mathbb{R}^{q_2}$ is a vector of basis functions. Then we need to find proper parameters υ such that $V_\upsilon \to V_\pi$, for which we employ the following updates to improve our estimates of V_υ:

$$\upsilon_i(t+1) = \sum_{j=1}^{N} W_{i,j} \upsilon_i(t) - \alpha_3 h_i(C(t), t)$$
$$\kappa_i(t+1) = \kappa_i(t) + \alpha_4 l_i(C(t), D(t), f_i(t), t).$$

Here,

$$C(t) = \eta(x(t))(\eta(t) - \gamma\eta(x(t+1)))^\top$$
$$D(t) = \eta(x(t))\eta(x(t))^\top$$
$$f_i(t) = r_i(x(t), u(t))\eta(x(t)),$$

and κ_i is the corresponding dual variable. In addition, we have that h_i is the gradient surrogate with respect to υ_i and l_i is the gradient surrogates with respect to κ_i. The definition of gradient surrogates is discussed above.

3.2 Actor Training

Now based on the convergence of the critic, we train each agent's actor to converge on the globally optimal control policy. Similar to Zhang et al. (2018b), we will also utilize the policy gradient method for the actor training in this section. A policy best for the whole network will be achieved based on the advantage function

$$A_\pi(t) = Q_\pi(x(t), u(t)) - V_\pi(s),$$

Sutton et al. (2000); Konda and Tsitsiklis (2000); Sutton and Barto (2018). Though each agent does not know the exact value to this advantage function, we allow each agent to use

$$A_i(t) = Q_{w_i}(x(t), u(t)) - V_{\upsilon_i}(x(t)),$$

which can be looked at as a local estimate to the global advantage function $A_\pi(t)$. Motivated by this we employ the following updates for actor training:

$$\theta_i(t+1) = \Gamma(\theta_i(t) + \beta(t) A_i(t) \psi_i(x(t), u(t))),$$

where Γ is a projection operator and we have:

$$\psi_i(x(t), u(t)) = \frac{\nabla_{\theta_i}(\pi_{\theta_i}(x(t), u(t)))}{\pi_{\theta_i}(x(t), u(t)))},$$

which comes from the gradient of $log(\pi_{\theta_i}(x(t), u(t)))$ with respect to θ_i.

To summarize, the proposed distributed update at each agent i is given as follows:

Critic Update:

$$w_i(t+1) = \sum_{j=1}^{N} W_{i,j} w_i(t) - \alpha_1 s_i(t) \tag{11}$$
$$\nu_i(t+1) = \nu_i(t) + \alpha_2 d_i(t), \tag{12}$$
$$\upsilon_i(t+1) = \sum_{j=1}^{N} W_{i,j} \upsilon_i(t) - \alpha_3 h_i(t) \tag{13}$$
$$\kappa_i(t+1) = \kappa_i(t) + \alpha_4 l_i(t). \tag{14}$$

Actor Update:

$$\theta_i(t+1) = \Gamma(\theta_i(t) + \beta(t) A_i(t) \psi_i(x(t), u(t))). \tag{15}$$

4. MAIN RESULT

We will now go over the main result of our paper where we describe what the proposed algorithm can achieve under the following assumptions:

(A1)- The function approximation of the policy , i.e π_θ, is greater than 0 for any θ. This is a standard assumption used in Zhang et al. (2018b); Bhatnagar et al. (2009); Konda and Tsitsiklis (2000)

(A2)- $\pi_\theta(x, u)$ is continuously differentiable in θ, as is assumed in Bhatnagar et al. (2009)

(A3)-The projection operator Γ, which is used in the proposed update, projects any $\theta_i(t)$ onto a compact set. Furthermore, we assume that the compact set Θ is large enough to include a least one local minimum of V_{π_θ}.

(A4)-The reward function $r_i(t)$ is uniformly bounded for each agent and for all time. This assumption has been made in works such as Zhang et al. (2018b,a)

(A5)- The step-sizes $\alpha_1(t)$,$\alpha_3(t)$, $\beta(t)$ satisfy:

$$\sum_{t=1}^{\infty} \alpha_1(t) \to \infty \qquad \sum_{t=1}^{\infty} \alpha_3(t) \to \infty \qquad \sum_{t=1}^{\infty} \beta(t) \to \infty$$
$$\beta(t) = o(\alpha_1(t)) \qquad \alpha_1(t) = o(\alpha_3(t)), \text{ as } t \to \infty,$$

where $f(t) = o(g(t))$ means for every constant ϵ there exists a constant N such that $|f(t)| \leq \epsilon g(t)$, for all $t \geq N$. In addition, we assume $\sum_t^\infty (\alpha_1(t)^2 + \alpha_3(t)^2 + \beta(t)^2)$ is bounded.

(A6)- Each data sample is selected at least once every T iterations of parameter updates.

(A7)-The matrices $\hat{A}$ and $\hat{C}$ are full rank for large T

(A8)- The sequence of states produced by any policy π is a Markov chain that is irreducible and aperiodic.

(A9)- $\mathbf{\Phi}$ is full rank , $q_1 \leq n$, and $\phi(x(t), u(t))$ is uniformly bounded for all x, u pairs.

Theorem 1. We denote $w_{\pi_\theta} = w_\theta$ and $\upsilon_{\pi_\theta} = \upsilon_\theta$, where w_θ and υ_θ are the target parameters such that $|Q_{w_\theta}(x, u) - Q_{\pi_\theta}(x, u)|$ and $|V_{\upsilon_\theta}(x) - V_{\pi_\theta}(x)|$ are minimized for all (x, u) and some parametrized policy π_θ. Given assumptions (A1)-(A9) we have the following: For each agent i , given a fixed parametrized policy π_θ, w_i and υ_i converge with consensus to parameters w_θ and υ_θ in a linear rate, such that $Q_{w_i} \rightarrow Q_{w_\theta}$ and $V_{\upsilon_i} \rightarrow V_{\upsilon_\theta}$. Furthermore, given $Q_{w_i} \rightarrow Q_{w_\theta}$ and $\epsilon > 0$, there exists a $\delta > 0$ such that if $\sup_{\theta(t)} \|e_{\theta(t)}\| < \delta$, then the proposed actor updates on $\theta_i(t)$ converge almost surely to an ϵ neighborhood of a local optimum of Q_{π_θ}. Where the local optimum is defined as a θ such that $\nabla_\theta Q_{\pi_\theta} = 0$ and furthermore:

$$\begin{aligned} e_{\theta(t)} =& \mathbb{E}_x \left[\mathbb{E}_{u|x} \left[((Q_{\pi_\theta}(x, u) - Q_{w_\theta}(x, u)) + \right.\right. \\ & \left.\left. (V_{\pi_\theta}(x) - V_{\upsilon_\theta}(x)))\psi_i(x, u) \right]\right]. \end{aligned}$$

Remark: We note that $e_{\theta(t)}$ expresses the bias due to linear function approximation of Q_{π_θ} and V_{π_θ}. Therefore, as long as this bias is small enough the proposed algorithm can achieve convergence to ϵ neighborhood of the optimal policy, and so we achieve the goal of our paper described by: $\pi_{\theta_i} \rightarrow \pi^* \pm \epsilon$ and $Q_{w_i} \rightarrow Q_{\pi^* \pm \epsilon}$.

In order to prove Theorem 1, we need the following lemmas, where lemma 1 is used to prove lemma 2.

Lemma 1.
Given assumption (A5),(A6),(A7), a sufficiently small α_3 with $\alpha_4 = \iota_1 \alpha_3$ where

$$\iota_1 = 8(\rho + \lambda_{max}(\hat{\mathbf{C}}^\top \hat{\mathbf{D}}^{-1} \hat{\mathbf{C}}))/\lambda_{min}(\hat{\mathbf{D}}),$$

and a policy π_θ, then for each agent i the updates on υ_i from (13) converge to network consensus on υ_θ such that $V_{\upsilon_i} \rightarrow V_{\upsilon_\theta}$.

From Lemma 1 we have that υ_i converges such that $V_\upsilon \rightarrow V_\theta$ and that the network reaches consensus on υ_i, furthermore from (A5) we know that υ_i converges in a faster time-scale than w_i. Therefore, using two-timescale stochastic approximation Borkar (2008) we have that $V_\upsilon = V_\theta$ for our analysis of updates on w_i. By following the same proof in Wai et al. (2018), one then has the following lemma:

Lemma 2. If assumptions (A5), (A6), and (A7) hold and the primal step size α_1 is sufficiently small with $\alpha_2 = \iota_2 \alpha_1$ where $\iota_2 = 8(\rho + \lambda_{max}(\hat{\mathbf{A}}))/\lambda_{min}(\hat{\mathbf{A}})$, then for a given policy π_θ the critic algorithm converges to the optimal parameters w_θ, ν_i^*, and $\frac{1}{m}\sum_{i=1}^m \|w_i(t) - \bar{w}(t)\|)$ converges to zero, all at a linear rate. More formally we have:

$$\|\bar{w}(t) - w_\theta\|^2 + \frac{1}{\iota_2 m} \sum_{i=1}^m \|\nu_i - \nu_i^*\|^2 = \mathcal{O}(\sigma^t)$$

$$\frac{1}{m} \sum_{i=1}^m \|w_i(t) - \bar{w}(t)\| = \mathcal{O}(\sigma^t),$$

where $\bar{w}(t) = \frac{1}{m}\sum_{i=1}^m w_i(t)$ and $0 < \sigma < 1$.

Proof of Theorem 1: For convenience we denote $Q_{\pi_\theta} = Q_\theta$, $w_{\pi_\theta} = w_\theta$, $V_{\pi_\theta} = V_\theta$, $\upsilon_{\pi_\theta} = \upsilon_\theta$, $\psi_i(x(t), u(t))$as $\psi_i(t)$, $\mathbb{E}_x[\mathbb{E}_{u|x}[\cdot]]$ as $\mathbb{E}[\cdot]$, and for simplicity we denote $\theta(t) = \theta$. From our problem formulation we have that each agent maintains an estimate of the global optimal policy π_i, however we also have that each agent only executes a control input u_i.This u_i is only an element of the global control input estimate u_{π_i} sampled from π_i. We now define an effective global policy π such that when u is sampled from π, we get the actual global control input vector $col\{u_1, u_2, ..., u_m\}$. In addition we define π_θ as the parameterized form of π, given some parameter vector θ .

Given the above definitions, we begin by writing out the actor update :

$$\theta_i(t+1) = \Gamma(\theta_i(t) + \beta(t) A_i(t) \psi_i(t)).$$

Now let $\mathcal{F}(t) = \sigma(\theta_i(\tau), \tau \leq t)$ be a σ-field (also called σ-algebra). We can then define the following:

$$\begin{aligned} \xi_1(t+1) &= A_i(t)\phi(t) - \mathbb{E}[A_i(t)\psi_i(t)|\mathcal{F}(t)] \\ \xi_2(t+1) &= \mathbb{E}[(A_i(t) - A_\theta(t)\psi_i(t)|\mathcal{F}(t)], \end{aligned}$$

where $A_\theta(t) = Q_{w_\theta}(x(t), u(t)) - V_{\upsilon_\theta}(x(t))$ is the advantage function after the critic converges to $w_\theta, \upsilon_\theta$ for a given π_θ, whereas $A_i(t)$ is a current estimate using the critic of agent i. With this we can rewrite the actor updates as:

$$\begin{aligned} \theta_i(t+1) = \Gamma(&\theta_i(t) + \beta(t)\mathbb{E}[A_\theta(t)\psi_i(t)] \\ &+ \beta(t)\xi_1(t) + \beta(t)\xi_2(t)). \end{aligned}$$

From lemma 2 we know that the critic converges, and from our time-step assumptions we know that it converges in a faster time scale than the actor. Therefore, in the actor update time-scale we have that $A_i(t) \rightarrow A_\theta(t)$, and so ξ_2 is in $o(1)$. Furthermore, let $M(t) = \sum_{t=1}^\infty \beta(t)\xi_1(t)$, we also note that the sequence $\{M(t)\}$ is a martingale sequence. We also know that from assumption the sequences $\{w_i(t)\}$, $\{\psi_i(t)\}$, and $\{\phi(x(t), u(t))\}$ are bounded, and so $\{\xi_1(t)\}$ must also be bounded. Using our step-size assumption we then have the following almost surely:

$$\sum_{t=1}^\infty \mathbb{E}[\|M(t+1) - M(t)\|^2 | \mathcal{F}(t)] = \sum_{t\geq 1}^\infty \|\beta(t)\xi_1(t+1)\|^2 < \infty.$$

From the martingale convergence theorem we know that $M(t)$ conveges almost surely, and so we have :

$$\lim_{t\rightarrow\infty} \mathbb{P}\left(\sup_{n\geq t} \|\sum_{\tau=t}^n \beta(\tau)\xi_1(\tau)\| \geq \epsilon \right) = 0$$

for some $\epsilon > 0$.

We now look at the quantity $\mathbb{E}[A_{\theta_i}(t)\psi_i(t)]$, which can be rewritten as the following:

$$\begin{aligned} \mathbb{E}[A_\theta(t)\psi_i(t)] &= \int_{-\infty}^\infty \mu_\theta(x) \int_{-\infty}^\infty \pi_\theta(x, u)\psi_i(t) A_\theta(t)\, du\, dx \\ &= \int_{-\infty}^\infty \mu_\theta(x) \int_{-\infty}^\infty \pi_\theta(x, u)\psi_i(x, u)(w_\theta^\top \phi(x, u) - \upsilon_\theta^\top \eta(x))\, du\, dx. \end{aligned}$$

From the above we can show that $\mathbb{E}[A_\theta(t)\psi_i(t)]$ is continuous in θ_i.It is important to note that θ is the parameter vector of π_θ, which when sampled produces the same u_i extracted from each agents $u \sim \pi_{\theta_i}$. Therefore, as long as the parametrization of π can represent any viable(stochastic) policies, then θ can be seen as a continuous function of each agent's θ_i.This observation implies that that if a function is continuous in θ, then it is also continuous in θ_i.

With the above observations of each term in the proposed update equation, the Kushner-Clark lemma Kushner and Clark (2012) tells us that the update converges almost surely to the set of asymptotically stable equilibria of the following ODE:

$$\dot{\theta}_i(t) = \Gamma(\mathbb{E}[A_\theta \psi_i(t)]). \tag{16}$$

From Sutton et al. (2000); Konda and Tsitsiklis (2000); Sutton and Barto (2018) we know that in order to update the policy towards the optimal policy we must compute $\nabla_\theta V_\theta(x)$, which is the policy gradient, usually expressed as $\nabla J(\theta)$. In Sutton and Barto (2018); Bhatnagar et al. (2009) we find that:

$$\nabla_\theta V_\theta(x) = \mathbb{E}[(Q_\theta(x,u) - V_\theta(x))\psi_i(x,u)].$$

We can then rewrite $\mathbb{E}[A_\theta \psi_i(t)]$ similar to Bhatnagar et al. (2009); Zhang et al. (2018b),in the following way :

$$\begin{aligned}\mathbb{E}[A_\theta \psi_i(t)] =& \nabla_\theta V_\theta(x) + (\mathbb{E}[A_\theta \psi_i(t)] \\ &- \mathbb{E}[(Q_\theta(x,u) - V_\theta(x))\psi_i(t)]).\end{aligned}$$

By rearranging terms and using the linearity of expectation we then get:

$$\begin{aligned}\mathbb{E}[A_\theta \psi_i(t)] =& \nabla_\theta V_\theta(x) + \mathbb{E}[((Q_{w_\theta}(x,u) - Q_\theta(x,u)) \\ &+ (V_{v_\theta} - V_\theta(x)))\psi_i(t)],\end{aligned}$$

where $\mathbb{E}[(Q_{w_\theta}(x,u) - Q_\theta(x,u)) + (V_{v_\theta} - V_\theta(x))\psi_i(t)]$ expresses the bias due to linear function approximation of V_θ and Q_θ. Therefore, if

$$\sup_{\theta(t)} ||\mathbb{E}[((Q_{w_\theta}(x,u) - Q_\theta(x,u)) + (V_{v_\theta} - V_\theta(x)))\psi_i(t)]|| < \delta$$

for some $\delta > 0$, then (16) converges almost surely to an ϵ neighborhood of $\nabla_\theta V_\theta = 0$, which is a local optimum of V_θ and , since $V_\theta(x) = \mathbb{E}_{u\sim\pi_\theta}[Q_\theta(x,u)]$, a local optimum of Q_θ. Bhatnagar et al. (2009). ■

5. SIMULATIONS

As in in Zhang et al. (2018a), we also consider the following nonlinear system:

$$x(t+1) = \varphi|x(t)| + v^\top u + (\sqrt{1-\varphi^2})\varrho(t)$$

where $\varphi = 0.9$, $\varrho(t) \sim \mathcal{N}(0,1)$, and $v \in \mathbb{R}^m$ is selected randomly from $[0,1]^m$.We use $\mathcal{N}(0,1)$ to denote the normal distribution with zero mean and standard deviation of one.

Consider a small network of $m = 4$ agents. Each agent's π_i is approximated by a normal distribution $\mathcal{N}(\zeta_{\theta_i}(x), \sigma)$, where $\zeta_{\theta_i}(x) = \theta_i^\top \chi(x)$ and $\sigma = 0.5$. We have that $\chi(x) \in \mathbb{R}^5$ is a vector of Gaussian radial basis functions(RBF) with means randomly selected from $[0,1]$ and a standard deviation of 0.001. Furthermore, each agent observes a reward $r_i(x,u) = k_{0,i} + k_{1,i}u_i^2 + k_{2,i}x^2$, where u_i is the scalar control input of agent i. The coefficients $k_{0,i}, k_{1,i}, k_{2,i}$ are selected randomly from the range $[0,1]$ for each agent.

In order to approximate the state-value function $V(x)$, we use a scalar basis function $\eta(x)$ which we implement as a Gaussian radial basis function with mean selected randomly from the interval $[-2,5]$ and standard deviation of 0.1. For the approximation of the action-value function $Q(x,u)$ we use the following structure:

$$Q_{w_i}(x,u) = w_{1,i}u^\top E(x)u + u^\top F(x)w_{2:q_1-1,i} + w_{q_1,i}$$

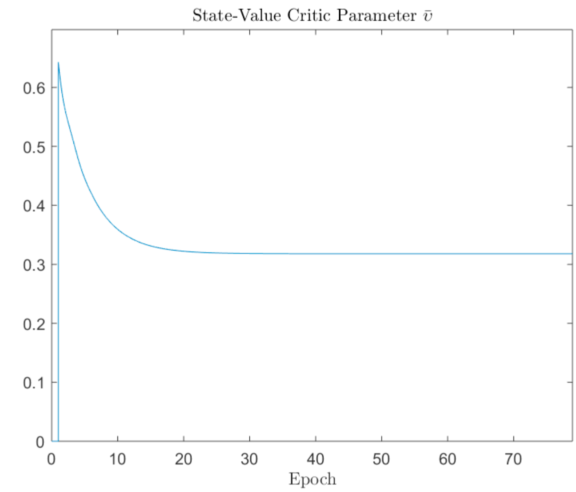

Fig. 2. Averaged State-Value Parameter for the Critic

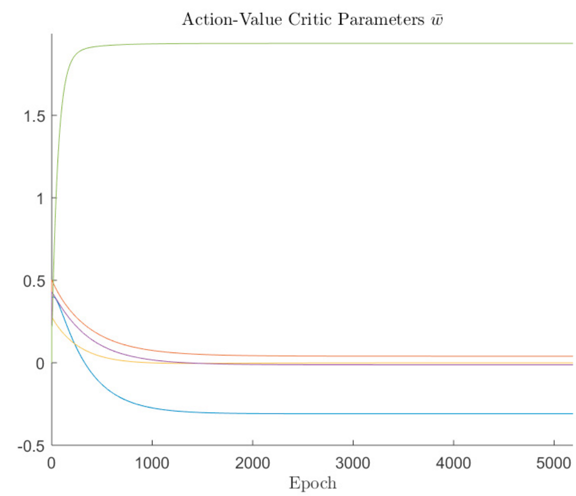

Fig. 3. Averaged Action-Value Parameters for the Critic

where $w_i = col\{w_{1,i}, w_{2:q_1-1,i}, w_{q_1,i}\}$ with $q_1 = 5$. The basis functions $E(x)$ and $F(x)$ are also selected as Gaussian radial basis functions with means randomly selected from $[0,1]$ and standard deviation of 0.1 for both.

The plots of our simulation results, shown in figures $(2,3,4)$, show the time evolution of the network average of the parameters of interest, namely $\bar{v}, \bar{w}, \bar{\theta}$. Where $\bar{v} = \frac{1}{m}\sum_{i=1}^m v_i$, $\bar{w} = \frac{1}{m}\sum_{i=1}^m w_i$, $\bar{\theta} = \frac{1}{m}\sum_{i=1}^m \theta_i$.We label each x-axis as "epochs". We define an "epoch" as the time step t divided by the number of data samples in memory M, where data samples are the sequences of states and control inputs that have been observed and recorded. For our simulations we used a memory of $M = 1500$ data samples.

From figures (2) and (3) we can see that the proposed updates on v_i and w_i both converge for every agent in the network, and that by design the critic converges much faster for v_i then for w_i.

6. CONCLUSION

In this paper we have looked at the problem of distributed multi-agent reinforcement learning where agents only observe their own local rewards. We have presented an actor-

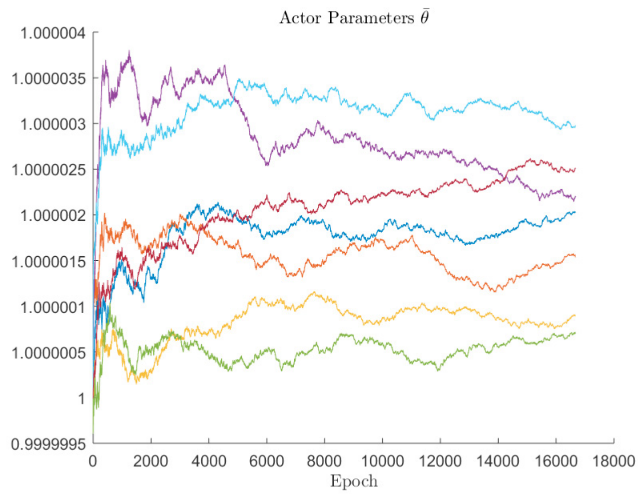

Fig. 4. Averaged Policy Parameters for the Actor

critic algorithm that allows agents to use information from their neighbors in order to improve their policies so that the globally averaged reward is maximized. The algorithm has been analyzed based on the two-timescale method used in stochastic approximation problems, and conditions for its convergence have been provided.

REFERENCES

Bhatnagar, S., Sutton, R.S., Ghavamzadeh, M., and Lee, M. (2009). Natural actor–critic algorithms. *Automatica*, 45(11), 2471–2482.

Borkar, V. (2008). Stochastic approximation: a dynamical systems view. *Hindustan Publ. Co., New Delhi, India and Cambridge Uni. Press, Cambridge, UK*.

Dai, B., Shaw, A., Li, L., Xiao, L., He, N., Liu, Z., Chen, J., and Song, L. (2018). Sbeed: Convergent reinforcement learning with nonlinear function approximation. In *ICML 2018*.

Kar, S., Moura, J.M., and Poor, H.V. (2013). Qd-learning: A collaborative distributed strategy for multi-agent reinforcement learning through consensus+innovations. *IEEE Transactions on Signal Processing*, 61(7), 1848–1862.

Konda, V.R. and Tsitsiklis, J.N. (2000). Actor-critic algorithms. In *Advances in neural information processing systems*, 1008–1014.

Kushner, H.J. and Clark, D.S. (2012). *Stochastic approximation methods for constrained and unconstrained systems*, volume 26. Springer Science & Business Media.

Lee, D., Yoon, H., and Hovakimyan, N. (2018). Primal-dual algorithm for distributed reinforcement learning: distributed gtd. In *2018 IEEE Conference on Decision and Control (CDC)*, 1967–1972. IEEE.

Malialis, K., Devlin, S., and Kudenko, D. (2015). Distributed reinforcement learning for adaptive and robust network intrusion response. *Connection Science*, 27(3), 234–252.

Mathkar, A. and Borkar, V.S. (2017). Distributed reinforcement learning via gossip. *IEEE Transactions on Automatic Control*, 62(3), 1465–1470.

Schneider, J., Wong, W.K., Moore, A., and Riedmiller, M. (1999). Distributed value functions. In *ICML*, 371–378.

Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W.M., Zambaldi, V., Jaderberg, M., Lanctot, M., Sonnerat, N., Leibo, J.Z., Tuyls, K., et al. (2018). Value-decomposition networks for cooperative multi-agent learning based on team reward. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 2085–2087. International Foundation for Autonomous Agents and Multiagent Systems.

Sutton, R.S. and Barto, A.G. (2018). *Reinforcement learning: An introduction*. MIT press.

Sutton, R.S., McAllester, D.A., Singh, S.P., and Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, 1057–1063.

Tadić, V. (2001). On the convergence of temporal-difference learning with linear function approximation. *Machine learning*, 42(3), 241–267.

Tham, C.K. and Renaud, J.C. (2005). Multi-agent systems on sensor networks: A distributed reinforcement learning approach. In *2005 International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 423–429. IEEE.

Wai, H.T., Yang, Z., Wang, P.Z., and Hong, M. (2018). Multi-agent reinforcement learning via double averaging primal-dual optimization. In *Advances in Neural Information Processing Systems*, 9649–9660.

Yang, Z., Zhang, K., Hong, M., and Başar, T. (2018). A finite sample analysis of the actor-critic algorithm. In *2018 IEEE Conference on Decision and Control (CDC)*, 2759–2764. IEEE.

Zhang, K., Yang, Z., and Basar, T. (2018a). Networked multi-agent reinforcement learning in continuous spaces. In *2018 IEEE Conference on Decision and Control (CDC)*, 2771–2776. IEEE.

Zhang, K., Yang, Z., Liu, H., Zhang, T., and Basar, T. (2018b). Fully decentralized multi-agent reinforcement learning with networked agents. In J. Dy and A. Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 5872–5881. PMLR, Stockholmsmssan, Stockholm Sweden.