Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Review:

Decentralized multi-agent reinforcement learning with networked agents: recent advances*

Kaiqing ZHANG‡[1], Zhuoran YANG[2], Tamer BAŞAR[1]

[1]*Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, IL 61801, USA*

[2]*Department of Operations Research and Financial Engineering, Princeton University, NJ 08544, USA*

E-mail: kzhang66@illinois.edu; zy6@princeton.edu; basar1@illinois.edu

Received Nov. 30, 2019; Revision accepted Aug. 17, 2020; Crosschecked Apr. 29, 2021

Abstract: Multi-agent reinforcement learning (MARL) has long been a significant research topic in both machine learning and control systems. Recent development of (single-agent) deep reinforcement learning has created a resurgence of interest in developing new MARL algorithms, especially those founded on theoretical analysis. In this paper, we review recent advances on a sub-area of this topic: decentralized MARL with networked agents. In this scenario, multiple agents perform sequential decision-making in a common environment, and without the coordination of any central controller, while being allowed to exchange information with their neighbors over a communication network. Such a setting finds broad applications in the control and operation of robots, unmanned vehicles, mobile sensor networks, and the smart grid. This review covers several of our research endeavors in this direction, as well as progress made by other researchers along the line. We hope that this review promotes additional research efforts in this exciting yet challenging area.

Key words: Reinforcement learning; Multi-agent systems; Networked systems; Consensus optimization; Distributed optimization; Game theory

https://doi.org/10.1631/FITEE.1900661 **CLC number:** TP18

1 Introduction

Reinforcement learning (RL) has achieved tremendous success recently in many sequential decision-making problems, especially when associated with the deep neural networks for function approximation (Mnih et al., 2015). Preeminent examples include playing the game of Go (Silver et al., 2016, 2017), robotics (Kober et al., 2013; Lillicrap et al., 2016), autonomous driving (Shalev-Shwartz et al., 2016), etc. Interestingly, most of the applications involve more than one single agent/player (hereafter, we will interchangeably use agent and player), placing these applications in the realm of multi-agent reinforcement learning (MARL). In particular, MARL models the sequential decision-making of multiple autonomous agents in a common environment, where each agent's objective and system evolution are both affected by the joint decision made by all agents. MARL algorithms can be generally categorized into three groups, according to the settings they address: fully cooperative, fully competitive, and a mix of the two (Busoniu et al., 2008; Zhang KQ et al., 2019). Specifically, fully cooperative MARL agents aim to optimize a long-term return common to all, whereas fully competitive MARL agents usually have completely misaligned returns that sum up to zero. Agents in the mixed MARL setting can be both fully cooperative and competitive. In this review, for simplicity, we

‡ Corresponding author

* Project supported in part by the US Army Research Laboratory (ARL) Cooperative Agreement (No. W911NF-17-2-0196), and in part by the Air Force Office of Scientific Research (AFOSR) Grant (No. FA9550-19-1-0353)

ORCID: Kaiqing ZHANG, https://orcid.org/0000-0002-7446-7581

refer to the first one as cooperative MARL, and the second and third ones as non-cooperative MARL.

There exist several long-standing challenges in both cooperative and non-cooperative MARL, especially in the associated theoretical analyses. First, because the agents' objectives may be misaligned with each other, the learning goals in MARL are not just one-dimensional, which introduces the challenge of handling equilibrium points, and several performance criteria other than return-optimization (e.g., the communication/coordination efficiency and the robustness against potential adversaries). Second, it is well-known that the environment faced by each agent is non-stationary in MARL, as it is affected not only by the underlying system evolution, but also by the decisions made by other agents who are concurrently improving their policies. This non-stationarity invalidates the framework of most theoretical analyses of single-agent RL, which are stationary and Markovian. Third, because the joint action-space increases exponentially with the number of agents, MARL algorithms can suffer from scalability issues by nature. Fourth, the information structure that dictates information availability to each agent becomes more complicated in multi-agent settings, as some of the observations may not be sharable with each other, and could sometimes be maintained in a decentralized fashion. Therefore, the theoretical analysis of MARL algorithms is relatively lacking in the literature.

Aside from the earlier works on MARL as summarized in Busoniu et al. (2008), there has been a resurgent interest in this area, especially with recent advances in single-agent RL (Foerster et al., 2016; Zazo et al., 2016; Gupta et al., 2017; Lowe et al., 2017; Omidshafiei et al., 2017; Zhang KQ et al., 2018c). Most of these studies involving applications of deep neural networks for function approximation are not placed under rigorous theoretical foundations, due to the limited understanding of even single-agent deep RL theories, where only one agent interacts with the environment and uses deep neural networks for function approximation. However, a relatively new paradigm for MARL, i.e., decentralized MARL with networked agents, has gained increasing research attention (Kar et al., 2013; Wai et al., 2018; Zhang KQ et al., 2018c; Doan et al., 2019a). This is partly due to the fact that the algorithms under this paradigm require no existence of any central controller; i.e., they can be implemented in a decentralized fashion. This can partially address the scalability issue, one of the aforementioned challenges, and is more amenable to a decentralized information structure common in practical multi-agent systems (Rabbat and Nowak, 2004; Corke et al., 2005; Dall'Anese et al., 2013). The second reason for its popularity is that most algorithms under this paradigm are accompanied by theoretical analysis for convergence/sample complexity, as they are closely related to, and inspired by the recent development of distributed/consensus optimization with networked agents, across the areas of control (Nedić and Ozdaglar, 2009), operations research (Nedić et al., 2017), signal processing (Sayed, 2014; Shi et al., 2015), and statistical learning (Boyd et al., 2011; Fan et al., 2015).

Specifically, we focus on the MARL setting where the agents, mostly cooperative, are connected by a communication network allowing the information exchange with each other. The setting is decentralized in the sense that each agent makes its own decision, based on only local observations and information transmitted from its neighbors, and without coordination by a central controller. Such a setting finds broad applications in practice, such as robotics (Corke et al., 2005), unmanned vehicles (Qie et al., 2019), mobile sensor networks (Rabbat and Nowak, 2004), intelligent transportation systems (Adler and Blue, 2002; Zhang KQ et al., 2018a), and the smart grid (Dall'Anese et al., 2013; Zhang KQ et al., 2018a), and enjoys several advantages over a centralized setting, in terms of either cost, scalability, or robustness. For example, it might be costly to even establish a central controller for coordination for some systems (Adler and Blue, 2002; Dall'Anese et al., 2013), which could also easily suffer from malicious attacks and high communication traffic, as the malfunctioning of the central controller will take down the overall system. This is a drawback of communication being concentrated in one place, between the controller and agents. Additionally, such a decentralized setting can help spare the computation cost at the central controller to all agents, especially when the number of agents is large (Bertsekas D, 2019). As a result, it is imperative to summarize the theories and algorithms on this topic, in order to both highlight the boundary of existing research endeavors and stimulate future research directions.

In this paper, we review recent advances in decentralized MARL with networked agents, in conjunction with our recent review (Zhang KQ et al., 2019) on general MARL algorithms. Indeed, Zhang KQ et al. (2019) have provided a comparatively complete overview of general MARL algorithms backed by theoretical analysis, and represented a big-picture introduction for the present review. Interested readers are referred to Zhang KQ et al. (2019) for a detailed review, not limited to the decentralized networked setting. The present review summarizes several of our earlier works on this decentralized MARL setting (Zhang KQ et al., 2018b, 2018c, 2018d), along with recent progress by other researchers along the line. We expect that this review will provide continuing stimulus for researchers with similar interest in working on this exciting yet challenging area.

2 Background

In this section, we provide the necessary background on MARL, especially in the decentralized setting with networked agents.

2.1 Single-agent RL

A general RL agent is modeled to perform sequential decision-making in a Markov decision process (MDP), as formally defined below:

Definition 1 An MDP is defined by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, respectively, $\mathcal{P}: \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ denotes the transition probability from any state $s \in \mathcal{S}$ to any state $s' \in \mathcal{S}$ for any given action $a \in \mathcal{A}$, $R: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function that determines the immediate reward received by the agent for a transition from (s, a) to s', and $\gamma \in [0, 1]$ is the discount factor that trades off the instantaneous and future rewards.

At each time t, the agent chooses to execute an action a_t in face of the system state s_t, which causes the system to transition to $s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)$. Moreover, the agent receives an instantaneous reward $R(s_t, a_t, s_{t+1})$. The goal of the agent is to find a policy $\pi: \mathcal{S} \to \Delta(\mathcal{A})$ so that $a_t \sim \pi(\cdot|s_t)$ maximizes the discounted accumulated reward

$$\mathbb{E}\left[\sum_{t\geq 0} \gamma^t R\left(s_t, a_t, s_{t+1}\right) | a_t \sim \pi\left(\cdot|s_t\right), s_0\right].$$

Due to the Markovian property, the optimal policy can be calculated by dynamic-programming/backward induction, such as value iteration and policy iteration (Bertsekas DP, 2005), which requires the full knowledge of the model. RL, on the other hand, is devised to find the optimal policy without knowing the model, but by learning from experiences collected by interacting with either the environment or the simulator. In general, RL algorithms can be categorized into two types, value- and policy-based methods.

1. Value-based methods

Value-based methods aim to find an estimate of the state-action value/Q-function, which leads to the optimal policy by taking the greedy action with respect to the estimate. Classical value-based RL algorithms include Q-learning (Watkins and Dayan, 1992) and SARSA (Singh S et al., 2000). Another important task in RL that is related to value functions is to estimate the value function of a fixed policy (not necessarily the optimal one). This task is referred to as policy evaluation, and can be addressed by standard algorithms such as temporal difference (TD) learning (Tesauro, 1995; Tsitsiklis and van Roy, 1997) and gradient TD methods (Sutton et al., 2008, 2009; Meai et al., 2009; Liu et al., 2015).

2. Policy-based methods

Policy-based methods propose to directly search for the optimal one over the policy space, while the space is generally parameterized by function approximators like neural networks, i.e., parameterizing $\pi(\cdot|s) \approx \pi_\theta(\cdot|s)$. Hence, it is straightforward to improve the policy following the gradient direction of the long-term return, known as the policy gradient (PG) method. Sutton et al. (2000) have derived the closed-form of PG as

$$\nabla J(\theta) = \mathbb{E}_{a\sim\pi_\theta(\cdot|s), s\sim\eta_{\pi_\theta}(\cdot)}\left[Q_{\pi_\theta}(s, a)\nabla \log \pi_\theta(a|s)\right],$$

where $J(\theta)$ and Q_{π_θ} are the return and Q-function under policy π_θ, respectively, $\nabla \log \pi_\theta(a|s)$ is the score function of the policy, and η_{π_θ} is the state occupancy measure, either discounted or ergodic, under policy π_θ. Other standard PG methods include REINFORCE (Williams, 1992), G(PO)MDP (Baxter and Bartlett, 2001), actor-critic (Konda and Tsitsiklis, 1999), and deterministic PGs (Silver et al., 2014).

2.2 Multi-agent RL framework

Multi-agent RL also addresses the sequential decision-making problems, but with more than one agent involved. Specifically, both the system state evolution and the reward received by each agent are influenced by the joint actions of all agents. Moreover, each agent has its own long-term reward to optimize, which now becomes a function of the policies of all other agents. Though various MARL frameworks exist in the literature (Busoniu et al., 2008; Zhang KQ et al., 2019), we here focus on two examples that are either representative or pertinent to our decentralized MARL setting.

1. Markov/Stochastic games

As a direct generalization of MDPs to the multi-agent setting, Markov games (MGs), also known as stochastic games (Shapley, 1953), has long been treated as a classical framework of MARL (Littman, 1994). A formal definition of MGs is introduced as follows:

Definition 2 A Markov game is defined by a tuple $(\mathcal{N}, \mathcal{S}, \{\mathcal{A}^i\}_{i\in\mathcal{N}}, \mathcal{P}, \{R^i\}_{i\in\mathcal{N}}, \gamma)$, where $\mathcal{N} = \{1, 2, \dots, N\}$ $(N > 1)$ denotes the set of agents, $\mathcal{S}$ the state space observed by all agents, and $\mathcal{A}^i$ the action space of agent i. Letting $\mathcal{A} := \mathcal{A}^1 \times \mathcal{A}^2 \times \cdots \times \mathcal{A}^N$, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ denotes the transition probability from any state $s \in \mathcal{S}$ to any state $s' \in \mathcal{S}$ for any joint action $a \in \mathcal{A}$. $R^i : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function that determines the immediate reward received by agent i for a transition from (s, a) to s', and $\gamma \in [0, 1]$ is the discount factor.

At time t, each agent $i \in \mathcal{N}$ chooses an action a_t^i, according to the system state s_t. The joint chosen action $\boldsymbol{a}_t = (a_t^1, a_t^2, \dots, a_t^N)$ then makes the system transition to state s_{t+1}, and assigns to each agent i a reward $r_t^i = R^i(s_t, \boldsymbol{a}_t, s_{t+1})$. Agent i's goal is to find the policy $\pi^i : \mathcal{S} \to \Delta(\mathcal{A}^i)$ such that its own long-term return is optimized. Accordingly, agent i's value-function $V^i : \mathcal{S} \to \mathbb{R}$ becomes a function of the joint policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ with $\pi(a|s) := \prod_{i\in\mathcal{N}} \pi^i(a^i|s)$, which is defined as

$$V^i_{\pi^i,\pi^{-i}}(s) := \mathbb{E}_{a_t^i\sim\pi^i(\cdot|s_t)}\left[\sum_{t\geq 0}\gamma^t r_t^i | s_0 = s\right], \quad (1)$$

where $-i$ represents the indices of all agents in $\mathcal{N}$ except agent i. Owing to this coupling of polices, the solution concept of MGs is not simply an optimum, but an equilibrium among all agents. The most common one, named Nash equilibrium (NE) in MGs, is defined below (Başar and Olsder, 1999):

Definition 3 A Nash equilibrium of the MG $(\mathcal{N}, \mathcal{S}, \{\mathcal{A}^i\}_{i\in\mathcal{N}}, \mathcal{P}, \{R^i\}_{i\in\mathcal{N}}, \gamma)$ is a joint policy $\boldsymbol{\pi}^* = (\pi^{1,*}, \pi^{2,*}, \dots, \pi^{N,*})$, such that for any $s \in \mathcal{S}$ and $i \in \mathcal{N}$, there is $V^i_{\pi^{i,*},\pi^{-i,*}}(s) \geq V^i_{\pi^i,\pi^{-i,*}}(s)$, for any π^i.

The NE describes a point $\boldsymbol{\pi}^*$, from which no agent has any incentive to deviate. Most MARL algorithms are devised to converge to such an equilibrium point, making MGs the most standard framework in MARL.

Indeed, this framework of MGs is generally enough to cover both cooperative and non-cooperative MARL settings. For the former one, all agents share a common reward function, i.e., $R^1 = R^2 = \cdots = R^N = R$. Such a model is also known as multi-agent MDPs (MMDPs) (Boutilier, 1996; Lauer and Riedmiller, 2000) or Markov teams (Wang and Sandholm, 2003; Mahajan and Teneketzis, 2008). In this setting, the value functions are identical to all agents, enabling the use of single-agent RL algorithms, provided that all agents are coordinated as one decision maker. The latter setting with non-cooperative agents corresponds to the MGs with either zero-sum or general-sum reward functions. Such misaligned objectives of self-interested agents necessitate the use of NE as the solution concept.

2. Networked MMDPs

As a generalization of the above common reward cooperative model, the following one of networked MMDPs plays an essential role in decentralized MARL with networked agents.

Definition 4 A networked MMDP is defined by a tuple $(\mathcal{N}, \mathcal{S}, \{\mathcal{A}^i\}_{i\in\mathcal{N}}, \mathcal{P}, \{R^i\}_{i\in\mathcal{N}}, \gamma, \{\mathcal{G}_t\}_{t\geq 0})$, where the first six elements are identical to those in Definition 2 for MGs, and $\mathcal{G}_t = (\mathcal{N}, \mathcal{E}_t)$ denotes the time-varying communication network that connects all agents with $\mathcal{E}_t$ being the set of communication links at time t (i.e., an edge (i, j) for agents $i, j \in \mathcal{N}$ belongs to $\mathcal{E}_t$ if agents i and j can communicate with each other at time t).

The system evolution of networked MMDPs is identical to that of MGs, but with one difference in terms of the objective: all agents aim to cooperatively optimize the long-term return corresponding to the team-average reward $\bar{R}(s, a, s') :=$

$N^{-1}\cdot\sum_{i\in\mathcal{N}}R^i(s,a,s')$ for any $(s,a,s')\in\mathcal{S}\times\mathcal{A}\times\mathcal{S}$. Moreover, each agent makes decisions using only the local information, including the information transmitted from its neighbors over the network. The networked MMDP model allows agents to cooperate, but with different reward functions/preferences. This model is able to not only capture more heterogeneity and privacy among agents (compared to conventional MMDPs), but also facilitate the development of decentralized MARL algorithms with only neighbor-to-neighbor communications (Kar et al., 2013; Wai et al., 2018; Zhang KQ et al., 2018c). In addition, such heterogeneity necessitates the incorporation of more efficient communication protocols into MARL, an important while relatively open problem in MARL that naturally arises in networked MMDPs (Chen et al., 2018; Lin et al., 2019; Ren and Haupt, 2019).

3 Algorithms

This section provides a review of MARL algorithms under the frameworks introduced in Section 2.2. Specifically, we categorize the algorithms, which are amenable to the decentralized setting with networked agents, according to the tasks they address, such as learning the optimal/equilibrium policies and policy evaluation. Besides, we will mention several algorithms aiming to achieve other learning goals in this setting.

3.1 Learning policies

We first review the algorithms for the task of control in RL, namely, learning the optimal/equilibrium polices for the agents. Algorithms for both cooperative and non-cooperative settings exist in the literature.

3.1.1 Cooperative setting

Consider a team of agents cooperating under the framework of networked MMDPs introduced in Definition 4. Including the framework of MMDPs/Markov teams as a special case, this one generally requires more coordination, since the global value function cannot be estimated locally without knowing the other agents' reward functions. This challenge becomes more severe when no central controller, but only neighbor-to-neighbor communication over a network, is available for coordination.

Such an information structure has appeared frequently in the prolific studies on decentralized/distributed algorithms (note that hereafter we use decentralized and distributed interchangeably to describe this structure, to respect some conventions from distributed optimization literature), such as average consensus (Xiao et al., 2007) and distributed/consensus optimization (Nedić and Ozdaglar, 2009; Shi et al., 2015). Nevertheless, relatively few efforts have been devoted to address this structure in MARL. In fact, most existing results in distributed/consensus optimization can be viewed as solving static/one-stage decision-making problems (Nedić and Ozdaglar, 2009; Agarwal and Duchi, 2011; Jakovetic et al., 2011; Tu and Sayed, 2012), which is easier to analyze compared to RL, a sequential decision-making setting where the decisions made at current time will have a long-term effect.

The idea of decentralized MARL over networked agents dates back to Varshavskaya et al. (2009), describing their use for controlling distributed robotic systems. The algorithm therein uses the idea of average consensus, and is policy-based. However, no theoretical analysis was provided in the work. To the best of our knowledge, under this setting, the first MARL algorithm with provable convergence guarantees was proposed by Kar et al. (2013), which combines the idea of consensus + innovation (Kar and Moura, 2013) to the standard Q-learning algorithm, leading to the $\mathcal{QD}$-learning algorithm that is updated as follows:

$$\begin{aligned}Q^i_{t+1}(s,a)\leftarrow Q^i_t(s,a)+\alpha_{t,s,a}\Big[R^i(s,a)\\+\gamma\min_{a'\in\mathcal{A}}Q^i_t(s',a')-Q^i_t(s,a)\Big]\\-\beta_{t,s,a}\sum_{j\in\mathcal{N}^i_t}\big[Q^i_t(s,a)-Q^j_t(s,a)\big],\end{aligned}$$

where $\alpha_{t,s,a},\ \beta_{t,s,a}>0$ denote the step sizes, and $\mathcal{N}^i_t$ denotes agent i's set of neighboring agents, at time t. Compared to the Q-learning update (Watkins and Dayan, 1992), $\mathcal{QD}$-learning adds an innovation term, which is the difference between the agent's Q-value estimate and its neighbors'. Under some standard conditions for the step sizes, $\mathcal{QD}$-learning has been proven to converge to the optimum Q-function, for the tabular setting with finite state-action spaces.

As the joint action space increases exponentially with the number of agents, function approximation becomes especially pivotal to the scalability of MARL algorithms. To establish convergence analysis in the function approximation regime, we have resorted to policy-based algorithms, specifically, actor-critic algorithms, for this setting (Zhang KQ et al., 2018c). Specifically, each agent i's policy is parameterized as $\pi^i_{\boldsymbol{\theta}^i}:\mathcal{S}\to\mathcal{A}^i$ by some $\boldsymbol{\theta}^i\in\mathbb{R}^{m^i}$, and the joint policy is thus defined as $\pi_{\boldsymbol{\theta}}(a|s):=\prod_{i\in\mathcal{N}}\pi^i_{\boldsymbol{\theta}^i}(a^i|s)$. Let $Q_{\boldsymbol{\theta}}$ be the global value function corresponding to the team-average reward $\bar{R}$ under the joint policy $\pi_{\boldsymbol{\theta}}$. Then, we first establish the policy gradient of the return with respect to each agent i's parameter $\boldsymbol{\theta}^i$ as

$$\nabla_{\boldsymbol{\theta}^i}J(\boldsymbol{\theta})=\mathbb{E}\left[\nabla_{\boldsymbol{\theta}^i}\log\pi^i_{\boldsymbol{\theta}^i}(s,a^i)Q_{\boldsymbol{\theta}}(s,a)\right]. \tag{2}$$

Analogous to the single-agent PG given in Section 2.1, the PG in Eq. (2) involves the expectation of the product between the global Q-function $Q_{\boldsymbol{\theta}}$ and the local score function $\nabla_{\boldsymbol{\theta}^i}\log\pi^i_{\boldsymbol{\theta}^i}(s,a^i)$. The former quantity, however, cannot be estimated locally at each agent. Therefore, by parameterizing each local copy of $Q_{\boldsymbol{\theta}}(\cdot,\cdot)$ as $Q_{\boldsymbol{\theta}}(\cdot,\cdot;\omega^i)$, a consensus-based TD learning update was proposed for the critic step, i.e., for estimating $Q_{\boldsymbol{\theta}}(\cdot,\cdot)$ given $\pi_{\boldsymbol{\theta}}$:

$$\tilde{\omega}^i_t=\omega^i_t+\beta_{\omega,t}\delta^i_t\nabla_\omega Q_t(\omega^i_t), \tag{3}$$

$$\omega^i_{t+1}=\sum_{j\in\mathcal{N}}c_t(i,j)\tilde{\omega}^j_t, \tag{4}$$

where $\beta_{\omega,t}>0$ denotes the step size, and δ^i_t is the local TD-error calculated using $Q_{\boldsymbol{\theta}}(\cdot,\cdot;\omega^i)$. Eq. (3) is the standard TD learning update at agent i, while Eq. (4) is a weighted combination step of the neighbors' estimates $\tilde{\omega}^j_t$. The weights $c_t(i,j)$ are determined by the topology of the communication network; namely, it only has non-zero values if the two agents i and j are connected at time t, i.e., $(i,j)\in\mathcal{E}_t$. The weights also need to satisfy the doubly stochastic property in expectation, so that ω^i_t reaches a consensual value for all $i\in\mathcal{N}$ as $t\to\infty$. Then, in the actor step, each agent i updates its policy following stochastic policy gradient (2), using its own Q-function estimate $Q_{\boldsymbol{\theta}}(\cdot,\cdot;\omega^i_t)$. In addition, motivated by the fact that the temporal difference can be used in policy gradient to replace the Q-function (Bhatnagar et al., 2009), we proposed a variant algorithm that relies on not the Q-function, but the state-value function approximation (Zhang KQ et al., 2018c), in order to reduce the variance in the PG update.

When linear functions are used for value function approximation, we can establish the almost sure convergence of the decentralized actor-critic updates (Zhang KQ et al., 2018c). The proof techniques therein are based on the two-timescale stochastic approximation approach in Borkar (2008). Later in Zhang KQ et al. (2018d), we extended the similar ideas to the setting specifically with continuous spaces, where the deterministic policy gradient (DPG) method is usually used. For DPG methods, off-policy exploration using a stochastic behavior policy is generally required, as the deterministic on-policy might not be sufficiently explorative. Nevertheless, as the policies of other agents are unknown in the multi-agent setting, the standard off-policy approach (Silver et al., 2014, Section 4.2) is not applicable. As a result, we developed an actor-critic algorithm (Zhang KQ et al., 2018d), which is still on-policy, using the recent development of the expected policy gradient (EPG) method (Ciosek and Whiteson, 2018). EPG unifies stochastic PG (SPG) and DPG, but reduces the variance of general SPGs. Specifically, the critic step remains identical to Eqs. (3) and (4), whereas the actor step is replaced by the multi-agent version of EPG we newly derived. When linear function approximation is used, we can also establish the almost sure convergence of the algorithm. In the same vein, the extension of the algorithms of Zhang KQ et al. (2018c) to an off-policy setting has been investigated in Suttle et al. (2019), which was built upon the emphatic temporal differences (ETD) method for the critic (Sutton et al., 2016). Convergence can also be established using the stochastic approximation approach, by incorporating the analysis of ETD(λ) (Yu, 2015) into the analysis in Zhang KQ et al. (2018c). In addition, another off-policy algorithm for the same setting was proposed in a concurrent work by Zhang Y and Zavlanos (2019). Deviating from the previously described studies, in their algorithms, agents do not share/exchange their estimates of the value function. In contrast, the agents' goal is to reach consensus over the global optimal policy estimation. This yields a local critic and a consensus actor update, which also enjoys provably asymptotic convergence. More recently, Cassano et al. (2019) have considered a team policy-learning setting, where the

reward differs at each agent, whereas all agents aim to learn a single joint policy that maximizes the return induced by the average reward. This allows all agents to reach a consensual estimate of the policy. However, no theoretical convergence guarantees were provided therein.

We note that aforementioned convergence guarantees are asymptotic; namely, the algorithms are guaranteed to converge only as the number of iterations goes to infinity. More importantly, these convergence results are restricted to the case with linear function approximations. These two drawbacks make it imperative, while challenging, to quantify the performance when a finite number of iterations and/or samples are used, and when nonlinear functions such as deep neural networks are used in practice. Serving as an initial step towards the finite-sample analyses in this setting with more general function approximation, we studied in Zhang KQ et al. (2018b) the batch RL algorithms (Lange et al., 2012) in the multi-agent setting. In particular, we proposed decentralized variants of the fitted-Q iteration (FQI) algorithm (Riedmiller, 2005; Antos et al., 2008a). We focused on FQI as it motivates the celebrated deep Q-learning algorithm (Mnih et al., 2015) that has achieved great empirical success. All agents collaborate to update the global Q-function estimate iteratively, by fitting nonlinear least squares with the target values as the responses. Let $\mathcal{F}$ denote the function class for Q-function approximation, $\{(s_j, \{a_j^i\}_{i\in\mathcal{N}}, s_j')\}_{j\in[n]}$ be the batch transition dataset of size n available to all agents, and $\{r_j^i\}_{j\in[n]}$ be the local reward samples private to each agent. Then, the local target value at each agent i is calculated as $y_j^i = r_j^i + \gamma \cdot \max_{a\in\mathcal{A}} Q_t^i(s_j', a)$, where Q_t^i is agent i's Q-function estimate at iteration t. As a consequence, all agents aim to collaboratively find a common Q-function estimate by solving

$$\min_{f\in\mathcal{F}} \frac{1}{N}\sum_{i\in\mathcal{N}} \frac{1}{2n}\sum_{j=1}^{n} \left[y_j^i - f\left(s_j, a_j^1, a_j^2, \ldots, a_j^N\right)\right]^2. \tag{5}$$

As r_j^i, and thus y_j^i, is available only to agent i, the problem in (5) fits in the standard formulation of distributed/consensus optimization (Nedić and Ozdaglar, 2009; Agarwal and Duchi, 2011; Jakovetic et al., 2011; Tu and Sayed, 2012; Hong and Chang, 2017; Nedić et al., 2017). If $\mathcal{F}$ makes $\sum_{j=1}^n [y_j^i - f(s_j, a_j^1, a_j^2, \ldots, a_j^N)]^2$ convex for each i, then the global optimum can be achieved by the algorithms in these references. For the special case where $\mathcal{F}$ is a linear function class, this is indeed the case.

Unfortunately, with only a finite iteration of distributed optimization algorithms performed at each agent, the agents may not reach exact consensus. This results in an error in each agent's Q-function estimate, compared with the actual optimum of problem (5). When nonlinear function approximation is used, this error is even more obvious, as the actual global optimum can hardly be obtained in general. By accounting for this error due to decentralized computation, we derived the error propagation results following those for the single-agent batch RL (Munos, 2007; Antos et al., 2008a, 2008b; Munos and Szepesvári, 2008; Farahmand et al., 2010), to establish the finite-sample performance of the proposed algorithms. Specifically, we established the dependence of the accuracy of the algorithms' output on the function class $\mathcal{F}$, the number of samples within each iteration n, and the number of iterations for t.

3.1.2 Non-cooperative setting

The networked MMDP model can also be considered in a non-cooperative setting, which though has not been extensively studied in the literature. In Zhang KQ et al. (2018b), we also considered one type of non-cooperative setting, where two teams of networked agents, teams 1 and 2, form a zero-sum MG as introduced in Definition 2. Such a setting can be viewed as a mixed one with both cooperative (within each team) and competitive (against the opponent team) agents. We then established finite-sample analysis for a decentralized variant of FQI for this setting.

In particular, by instantiating the definition of NE in a two-player zero-sum case, for a given Q-value $Q(s,\cdot,\cdot): \mathcal{A}\times\mathcal{B}\to\mathbb{R}$, one can define a `Value` operator at any state $s\in\mathcal{S}$ as

$$\texttt{Value}[Q(s,\cdot,\cdot)] = \max_{u\in\Delta(\mathcal{A})} \min_{v\in\Delta(\mathcal{B})} \mathbb{E}_{a\sim u, b\sim v}[Q(s,a,b)],$$

where $\mathcal{A}$ and $\mathcal{B}$ are the joint action spaces, and u and v are the one-stage strategies of agents in teams 1 and 2, respectively. Additionally, if some function Q satisfies the following fixed-point equation for any $s\in\mathcal{S}$:

$$Q(s,a,b) = \bar{R}(s,a,b) + \gamma\cdot\texttt{Value}[Q(s,\cdot,\cdot)], \tag{6}$$

where $\bar{R}$ is the team-average reward of team 1 (thus $-\bar{R}$ is that of team 2), then such a $\texttt{Value}\big[Q(s,\cdot,\cdot)\big]$ defines the value of the game at any state $s\in\mathcal{S}$. In comparison to the single-agent case, the max min operator, instead of the max one, is used to define the optimal/equilibrium value function.

Therefore, to solve the MARL problem in this setting, it suffices to find a good estimate of the Q-function satisfying Eq. (6). Hence, similar to the single-team cooperative setting, all agents within one team now aim to collaboratively find a common Q-function estimate, by solving problem (5), but replace the local target value at each agent i by $y_j^i=r_j^i+\gamma\cdot\texttt{Value}\big[Q_t^i(s_j',a,b)\big]$, and the fitting function $f(s_j,a_j)$ by $f(s_j,a_j,b_j)$, a function over the joint action spaces of both teams. Then, such an optimization problem is solved in a distributed fashion as problem (5). Similar error-propagation analysis can be performed in this setting, leading to the finite-sample error bounds of the decentralized FQI algorithm. To the best of our knowledge, this appears to be the first finite-sample analysis for decentralized batch MARL in non-cooperative settings.

3.2 Policy evaluation

Besides control, a great number of algorithms have been developed to address the policy evaluation task in this decentralized MARL setting. In particular, policy evaluation corresponds to the critic step of the aforementioned actor-critic algorithms only. With a fixed policy, this task enjoys a neater formulation, because the sampling distribution now becomes stationary. Moreover, as linear function approximation is commonly used for this task, the objective is mostly convex. This makes the finite-time/sample analyses easier, in comparison to many control algorithms with only asymptotic convergence guarantees.

Specifically, under joint policy π, suppose each agent parameterizes the value function by $\big\{V_\omega(s):=\boldsymbol{\phi}^{\mathrm{T}}(s)\boldsymbol{\omega}:\boldsymbol{\omega}\in\mathbb{R}^d\big\}$, where $\boldsymbol{\phi}(s)\in\mathbb{R}^d$ is the feature vector at $s\in\mathcal{S}$, and $\boldsymbol{\omega}\in\mathbb{R}^d$ is the parameter vector. For notational convenience, let $\boldsymbol{\Phi}:=\big(\cdots;\boldsymbol{\phi}^{\mathrm{T}}(s);\cdots\big)\in\mathbb{R}^{|\mathcal{S}|\times d}$, $\boldsymbol{D}=\mathrm{diag}\big(\{\eta_\pi(s)\}_{s\in\mathcal{S}}\big)\in\mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ be a diagonal matrix constructed using the state-occupancy measure η_π, $\bar{R}^\pi(s)=N^{-1}\cdot\sum_{i\in\mathcal{N}}R^{i,\pi}(s)$, where $R^{i,\pi}(s)=\mathbb{E}_{a\sim\pi(\cdot|s),s'\sim P(\cdot|s,a)}[R^i(s,a,s')]$, and $\boldsymbol{P}^\pi\in\mathbb{R}^{|S|\times|S|}$ with the (s,s') element being $[P^\pi]_{s,s'}=\sum_{a\in\mathcal{A}}\pi(a|s)P(s'|s,a)$. The objective of all agents is to jointly minimize the mean square projected Bellman error (MSPBE) associated with the team-average reward, i.e.,

$$\begin{aligned}\min_{\boldsymbol{\omega}}\mathrm{MSPBE}(\boldsymbol{\omega}):&=\big\|\Pi_\Phi\left(V_\omega-\gamma\boldsymbol{P}^\pi V_\omega-\bar{R}^\pi\right)\big\|_{\boldsymbol{D}}^2\\&=\|\boldsymbol{A}\boldsymbol{\omega}-\boldsymbol{b}\|_{\boldsymbol{C}^{-1}}^2,\end{aligned}\tag{7}$$

where $\Pi_\Phi:=\boldsymbol{\Phi}(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{D}$ is the projection operator onto subspace $\{\boldsymbol{\Phi}\boldsymbol{\omega}:\boldsymbol{\omega}\in\mathbb{R}^d\}$, $\boldsymbol{A}:=\mathbb{E}\{\boldsymbol{\phi}(s)[\boldsymbol{\phi}(s)-\gamma\boldsymbol{\phi}(s')]^{\mathrm{T}}\}$, $\boldsymbol{C}:=\mathbb{E}[\boldsymbol{\phi}(s)\boldsymbol{\phi}^{\mathrm{T}}(s)]$, and $\boldsymbol{b}:=\mathbb{E}[\bar{R}^\pi(s)\boldsymbol{\phi}(s)]$. Using Fenchel duality, and replacing the expectation with samples, the finite-sum version of Eq. (7) can be re-formulated as a distributed saddle-point problem:

$$\begin{aligned}\min_{\boldsymbol{\omega}}\max_{\boldsymbol{\lambda}^i}\frac{1}{Nn}\sum_{i\in\mathcal{N}}\sum_{j=1}^{n}&2\left(\boldsymbol{\lambda}^i\right)^{\mathrm{T}}\boldsymbol{A}_j\boldsymbol{\omega}\\&-2\left(\boldsymbol{b}_j^i\right)^{\mathrm{T}}\boldsymbol{\lambda}^i-\left(\boldsymbol{\lambda}^i\right)^{\mathrm{T}}\boldsymbol{C}_j\boldsymbol{\lambda}^i,\end{aligned}$$

where n is the data size, $\boldsymbol{A}_j$, $\boldsymbol{C}_j$, and $\boldsymbol{b}_j^i$ are empirical estimates of $\boldsymbol{A}$, $\boldsymbol{C}$, and $\boldsymbol{b}^i:=\mathbb{E}[R^{i,\pi}(s)\boldsymbol{\phi}(s)]$ using sample j, respectively. The objective above is convex in $\boldsymbol{\omega}$ and concave in $\{\boldsymbol{\lambda}^i\}_{i\in\mathcal{N}}$. The use of MSPBE as an objective is standard in multi-agent policy evaluation (Macua et al., 2015; Lee et al., 2018; Wai et al., 2018; Doan et al., 2019a), and the idea of saddle-point reformulation has been adopted in Macua et al. (2015), Cassano et al. (2018), Lee et al. (2018), Wai et al. (2018), Ding et al. (2019), and Sha et al. (2020).

With formulation (7), Lee et al. (2018) developed a distributed variant of the gradient TD-based method (Sutton et al., 2009), and established asymptotic convergence using the ordinary differential equation (ODE) method. Wai et al. (2018) proposed a double averaging scheme that combines dynamic consensus (Qu and Li, 2018) and the stochastic average gradient (SAG) algorithm (Schmidt et al., 2017), in order to solve the saddle-point problem with a linear rate. In Cassano et al. (2018), the idea of variance-reduction, specifically, amortized variance-reduced gradient (AVRG) in Ying et al. (2018), has been incorporated into gradient TD-based policy evaluation. Achieving the same linear rate as in Wai et al. (2018), three advantages were claimed in Cassano et al. (2018), i.e., data-independent memory requirement, use of eligibility traces (Singh SP and Sutton, 1996), and no need for synchronization in sampling. More recently, standard TD learning (Tesauro, 1995), instead of

gradient-TD, has been generalized to this MARL setting, with special focuses on finite-sample analyses (Doan et al., 2019a, 2019b). By the proof techniques in Bhandari et al. (2018), Doan et al. (2019a) studied the distributed TD(0) algorithm. A projection operation was required on the iterates, and the data samples were assumed to be independent and identically distributed (i.i.d.). Then, following the recent advance in Srikant and Ying (2019), Doan et al. (2019b) provided finite-time performance of the more general distributed TD(λ) algorithm, without the need of any projection or i.i.d. noise assumption. More recently, Ding et al. (2019) used the tool of homotopy stochastic primal-dual method to improve the convergence rate of consensus-based gradient-TD, from $1/\sqrt{T}$ to $1/T$, which also explicitly takes into account the Markovian nature of the sampling process. Later, Sha et al. (2020) provided the first fully asynchronous scheme for policy evaluation in this decentralized setting. Specifically, each agent can communicate with its neighbors and compute their local variables using (possibly) delayed information at any time, which fully takes advantage of the decentralized setting.

3.3 Other learning goals

Several other learning goals have also been investigated in this setting. Zhang HG et al. (2017) considered the optimal consensus problem, where each agent tracks its neighbors' as well as a leader's states, so that the consensus error is minimized by the joint policy. Then, a policy iteration algorithm was devised, and made practical by introducing an actor-critic algorithm with neural networks for function approximation. Zhang QC et al. (2018) used a similar consensus error objective, with the name of cooperative multi-agent graphical games. Off-policy RL algorithms were developed, using a centralized-critic-decentralized-actor scheme.

As an essential ingredient in the algorithm design for the decentralized MARL settings, communication efficiency in MARL has drawn increasing attention recently (Chen et al., 2018; Lin et al., 2019; Ren and Haupt, 2019). In Chen et al. (2018), lazily aggregated policy gradient (LAPG), a distributed PG algorithm, was developed, which can reduce the number of communication rounds between the agents and a central controller. This was achieved by judiciously designing communication trigger rules. In Ren and Haupt (2019), the same policy evaluation problem as in Wai et al. (2018) was addressed, and a hierarchical distributed algorithm was developed by proposing a mixing matrix different from the doubly stochastic one used in Lee et al. (2018), Wai et al. (2018), and Zhang KQ et al. (2018c), which reduces communication by allowing unidirectional information exchange among agents. In comparison, Lin et al. (2019) proposed a distributed actor-critic algorithm, which reduces communication by transmitting only one scalar entry of its state vector at each iteration. The same convergence guarantee as that in Zhang KQ et al. (2018c) can be established.

We note that RL under this decentralized setting with networked agents has been studied beyond the multi-agent setting. Indeed, several works have modeled the setting for multi-task RL, where multiple cooperative agents are also connected by a communication network, without any coordination from a central controller. However, each agent is in face of an independent MDP, which is not influenced by other agents. Different agents may still have different reward functions, while the goal is to learn the optimal joint policy that optimizes the long-term return corresponding to the team-average reward. In some sense, this setting can be deemed a simplified version of our MARL setting, for the less coupling among agents. Under this setting, Pennesi and Paschalidis (2010) developed a distributed actor-critic algorithm, where each agent first conducts a local TD-based critic step, followed by a consensus-based actor step in which the gradient is calculated based on the neighbors' information exchanged. The gradient of the average return was then shown to converge to zero. In Macua et al. (2017), Diff-DAC, another distributed actor-critic algorithm, was developed from duality theory. The updates, which look similar to those of Zhang KQ et al. (2018c), are essentially an example of the dual ascent method to solve some linear programs. This provides additional insights into the actor-critic update for this setting. More recently, Assran et al. (2019) provided a gossip-based actor-learner architecture (GALA) for this networked independent MDP setting, in order to improve the stability and scalability of the multi-simulator learning processes. It proves that even with this asynchronous peer-to-peer update, the policy parameter of the GALA agent remains to be close to its neighbors' parameters during training.

More importantly, the work has provided extensive simulations to show that this networked decentralized setting outperforms the standard A2C and A3C training schemes, with a centralized controller. This has justified the great benefit of this decentralized networked setting in practice.

Policy evaluation has also been considered under this setting of networked agents interacting with independent MDPs. The early work (Macua et al., 2015) studied off-policy evaluation using the importance sampling technique. Without coupling among agents, there is no need for each agent to know the actions of the others. Then, a diffusion-based distributed gradient-TD method was proposed, which is proven to converge with a sublinear rate in the mean-square sense. Stanković and Stanković (2016) then proposed two other variants of the gradient-TD updates, i.e., GTD2 and TDC (Sutton et al., 2009), and proved weak convergence using the general stochastic approximation theory developed in Stanković et al. (2016). Stanković and Stanković (2016) specifically considered the case where agents are connected by a time-varying communication network. The aforementioned works (Cassano et al., 2018; Sha et al., 2020) also considered the independent MDP setting, with the same results established as those in the MARL setting.

4 Concluding remarks

Given the ubiquity of sequential decision-making in the presence of more than one agent, multi-agent RL has long been a significant but challenging research area. In this review, we have summarized the recent advances in a sub-area of MARL, that is, decentralized MARL with networked agents. In particular, we have focused on the MARL algorithms that address this setting and are supported by theoretical analysis. We hope that this review is appealing to the researchers of similar interest, and has provided stimulus for them to continue pursuing this direction. We highlight several interesting and challenging theoretical research directions in this decentralized setting. Several future directions are listed as follows:

1. Partial observability

Most of the described algorithms still fall into the regime of fully observable systems, where the state and actions of other agents are assumed to be observable to each agent. However, in many practical applications, each agent might be able to observe only part of the state, and/or part of the joint action. This will require each agent to maintain beliefs over the global state and actions of other agents, thereby making the problem more challenging to solve. In fact, even the cooperative setting, i.e., the decentralized partially observable MDP (Dec-POMDP), becomes difficult to address (Oliehoek and Amato, 2016).

2. Heterogeneous/Adversarial agents

Increased heterogeneity should be considered for multi-agent RL systems. Indeed, not all agents strictly follow the model. It is possible that in practice, some agents are compromised or attacked during operation. Therefore, it is crucial to develop safe and robust MARL algorithms that can accommodate these uncertainties and adversaries.

3. Deep MARL theory

Due to the tremendous empirical success of incorporating deep neural networks into RL, it is worth developing theoretical guarantees for deep RL algorithms. Recent progress has been made in this area, largely motivated by advances in both deep learning theory and RL theory. However, in the multi-agent regime, deep RL theory has not yet been developed.

More future directions for general MARL can be found in Zhang KQ et al. (2019).

Contributors

Kaiqing ZHANG, Zhuoran YANG, and Tamer BAŞAR have participated in all the three processes of drafting, organizing, and revising.

Compliance with ethics guidelines

Kaiqing ZHANG, Zhuoran YANG, and Tamer BAŞAR declare that they have no conflict of interest.

References

Adler JL, Blue VJ, 2002. A cooperative multi-agent transportation management and route guidance system. *Transp Res Part C Emerg Technol*, 10(5-6):433-454. https://doi.org/10.1016/S0968-090X(02)00030-X

Agarwal A, Duchi JC, 2011. Distributed delayed stochastic optimization. Proc 24th Int Conf on Neural Information Processing Systems, p.873-881.

Antos A, Szepesvári C, Munos R, 2008a. Fitted Q-iteration in continuous action-space MDPs. Advances in Neural Information Processing Systems, p.9-16.

Antos A, Szepesvári C, Munos R, 2008b. Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path.

Mach Learn, 71(1):89-129. https://doi.org/10.1007/s10994-007-5038-2

Assran M, Romoff J, Ballas N, et al., 2019. Gossip-based actor-learner architectures for deep reinforcement learning. Advances in Neural Information Processing Systems, p.13299-13309.

Başar T, Olsder GJ, 1999. Dynamic Noncooperative Game Theory. SIAM, Philadelphia.

Baxter J, Bartlett PL, 2001. Infinite-horizon policy-gradient estimation. *J Artif Intell Res*, 15:319-350. https://doi.org/10.1613/jair.806

Bertsekas D, 2019. Multiagent rollout algorithms and reinforcement learning. https://arxiv.org/abs/1910.00120

Bertsekas DP, 2005. Dynamic Programming and Optimal Control. Athena Scientific, Belmont, MA, USA.

Bhandari J, Russo D, Singal R, 2018. A finite time analysis of temporal difference learning with linear function approximation. Proc 31st Conf on Learning Theory, p.1691-1692.

Bhatnagar S, Sutton RS, Ghavamzadeh M, et al., 2009. Natural actor-critic algorithms. *Automatica*, 45(11):2471-2482. https://doi.org/10.1016/j.automatica.2009.07.008

Borkar VS, 2008. Stochastic Approximation: a Dynamical Systems Viewpoint. Cambridge University Press, Cambridge, UK.

Boutilier C, 1996. Planning, learning and coordination in multiagent decision processes. Proc 6th Conf on Theoretical Aspects of Rationality and Knowledge, p.195-210.

Boyd S, Parikh N, Chu E, et al., 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found Trends® Mach Learn*, 3(1):1-122. https://doi.org/10.1561/2200000016

Busoniu L, Babuska R, de Schutter B, et al., 2008. A comprehensive survey of multiagent reinforcement learning. *IEEE Trans Syst Man Cybern Part C Appl Rev*, 38(2):156-172. https://doi.org/10.1109/TSMCC.2007.913919

Cassano L, Yuan K, Sayed AH, 2018. Multi-agent fully decentralized value function learning with linear convergence rates. https://arxiv.org/abs/1810.07792

Cassano L, Alghunaim SA, Sayed AH, 2019. Team policy learning for multi-agent reinforcement learning. IEEE Int Conf on Acoustics, Speech and Signal Processing, p.3062-3066. https://doi.org/10.1109/ICASSP.2019.8683168

Chen TY, Zhang KQ, Giannakis GB, et al., 2018. Communication-efficient distributed reinforcement learning. https://arxiv.org/abs/1812.03239

Ciosek K, Whiteson S, 2018. Expected policy gradients for reinforcement learning. https://arxiv.org/abs/1801.03326

Corke P, Peterson R, Rus D, 2005. Networked robots: flying robot navigation using a sensor net. In: Dario P, Chatila R (Eds.), Robotics Research. Springer, Berlin, p.234-243. https://doi.org/10.1007/11008941_25

Dall'Anese E, Zhu H, Giannakis GB, 2013. Distributed optimal power flow for smart microgrids. *IEEE Trans Smart Grid*, 4(3):1464-1475. https://doi.org/10.1109/TSG.2013.2248175

Ding DS, Wei XH, Yang ZR, et al., 2019. Fast multi-agent temporal-difference learning via homotopy stochastic primal-dual optimization. https://arxiv.org/abs/1908.02805

Doan TT, Maguluri S, Romberg J, 2019a. Finite-time analysis of distributed TD(0) with linear function approximation for multi-agent reinforcement learning. Proc 36th Int Conf on Machine Learning, p.1626-1635.

Doan TT, Maguluri ST, Romberg J, 2019b. Finite-time performance of distributed temporal difference learning with linear function approximation. https://arxiv.org/abs/1907.12530

Fan JQ, Tong X, Zeng Y, 2015. Multi-agent inference in social networks: a finite population learning approach. *J Am Stat Assoc*, 110(509):149-158.

Farahmand AM, Munos R, Szepesvári C, 2010. Error propagation for approximate policy and value iteration. Advances in Neural Information Processing Systems, p.568-576.

Foerster JN, Assael YM, de Freitas N, et al., 2016. Learning to communicate with deep multi-agent reinforcement learning. Proc 30th Int Conf on Neural Information Processing Systems, p.2137-2145.

Gupta JK, Egorov M, Kochenderfer M, 2017. Cooperative multi-agent control using deep reinforcement learning. Int Conf on Autonomous Agents and Multiagent Systems, p.66-83. https://doi.org/10.1007/978-3-319-71682-4_5

Hong MY, Chang TH, 2017. Stochastic proximal gradient consensus over random networks. *IEEE Trans Signal Process*, 65(11):2933-2948. https://doi.org/10.1109/TSP.2017.2673815

Jakovetic D, Xavier J, Moura JMF, 2011. Cooperative convex optimization in networked systems: augmented Lagrangian algorithms with directed gossip communication. *IEEE Trans Signal Process*, 59(8):3889-3902. https://doi.org/10.1109/TSP.2011.2146776

Kar S, Moura JMF, 2013. Consensus + innovations distributed inference over networks: cooperation and sensing in networked systems. *IEEE Signal Process Mag*, 30(3):99-109. https://doi.org/10.1109/MSP.2012.2235193

Kar S, Moura JMF, Poor HV, 2013. *QD*-learning: a collaborative distributed strategy for multi-agent reinforcement learning through consensus + innovations. *IEEE Trans Signal Process*, 61(7):1848-1862. https://doi.org/10.1109/TSP.2013.2241057

Kober J, Bagnell JA, Peters J, 2013. Reinforcement learning in robotics: a survey. *Int J Rob Res*, 32(11):1238-1274. https://doi.org/10.1177/0278364913495721

Konda VR, Tsitsiklis JN, 1999. Actor-critic algorithms. Advances in Neural Information Processing Systems, p.1008-1014.

Lange S, Gabel T, Riedmiller M, 2012. Batch reinforcement learning. In: Wiering M, van Otterlo M (Eds.), Reinforcement Learning. Adaptation, Learning, and Optimization. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-27645-3_2

Lauer M, Riedmiller MA, 2000. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. Proc 17th Int Conf on Machine Learning, p.535-542.

Lee D, Yoon H, Hovakimyan N, 2018. Primal-dual algorithm for distributed reinforcement learning: distributed GTD. IEEE Conf on Decision and Control, p.1967-1972. https://doi.org/10.1109/CDC.2018.8619839

Lillicrap TP, Hunt JJ, Pritzel A, et al., 2016. Continuous control with deep reinforcement learning. Proc 4th Int Conf on Learning Representations.

Lin YX, Zhang KQ, Yang ZR, et al., 2019. A communication-efficient multi-agent actor-critic algorithm for distributed reinforcement learning. Proc IEEE 58th Conf on Decision and Control, p.5562-5567. https://doi.org/10.1109/CDC40024.2019.9029257

Littman ML, 1994. Markov games as a framework for multi-agent reinforcement learning. Proc 11th Int Conf on Machine Learning, p.157-163.

Liu B, Liu J, Ghavamzadeh M, et al., 2015. Finite-sample analysis of proximal gradient TD algorithms. Proc 31st Conf on Uncertainty in Artificial Intelligence, p.504-513.

Lowe R, Wu Y, Tamar A, et al., 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. Proc 31st Int Conf on Neural Information Processing Systems, p.6379-6390.

Macua SV, Chen JS, Zazo S, et al., 2015. Distributed policy evaluation under multiple behavior strategies. *IEEE Trans Autom Contr*, 60(5):1260-1274. https://doi.org/10.1109/TAC.2014.2368731

Macua SV, Tukiainen A, Hernández DGO, et al., 2017. Diff-DAC: distributed actor-critic for average multitask deep reinforcement learning. https://arxiv.org/abs/1710.10363

Mahajan A, Teneketzis D, 2008. Sequential Decomposition of Sequential Dynamic Teams: Applications to Real-Time Communication and Networked Control Systems. University of Michigan, Ann Arbor, USA.

Meai HR, Szepesvári C, Bhatnagar S, et al., 2009. Convergent temporal-difference learning with arbitrary smooth function approximation. Proc 22nd Int Conf on Neural Information Processing Systems, p.1204-1212.

Mnih V, Kavukcuoglu K, Silver D, et al., 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529-533. https://doi.org/10.1038/nature14236

Munos R, 2007. Performance bounds in L_p-norm for approximate value iteration. *SIAM J Contr Optim*, 46(2):541-561. https://doi.org/10.1137/040614384

Munos R, Szepesvári C, 2008. Finite-time bounds for fitted value iteration. *J Mach Learn Res*, 9:815-857.

Nedić A, Ozdaglar A, 2009. Distributed subgradient methods for multi-agent optimization. *IEEE Trans Autom Contr*, 54(1):48-61. https://doi.org/10.1109/TAC.2008.2009515

Nedić A, Olshevsky A, Shi W, 2017. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM J Optim*, 27(4):2597-2633. https://doi.org/10.1137/16M1084316

Oliehoek FA, Amato C, 2016. A Concise Introduction to Decentralized POMDPs. Springer, Cham.

Omidshafiei S, Pazis J, Amato C, et al., 2017. Deep decentralized multi-task multi-agent reinforcement learning under partial observability. Proc 34th Int Conf on Machine Learning, p.2681-2690.

Pennesi P, Paschalidis IC, 2010. A distributed actor-critic algorithm and applications to mobile sensor network coordination problems. *IEEE Trans Autom Contr*, 55(2):492-497. https://doi.org/10.1109/TAC.2009.2037462

Qie H, Shi DX, Shen TL, et al., 2019. Joint optimization of multi-UAV target assignment and path planning based on multi-agent reinforcement learning. *IEEE Access*, 7:146264-146272. https://doi.org/10.1109/ACCESS.2019.2943253

Qu GN, Li N, 2018. Harnessing smoothness to accelerate distributed optimization. *IEEE Trans Contr Netw Syst*, 5(3):1245-1260. https://doi.org/10.1109/TCNS.2017.2698261

Rabbat M, Nowak R, 2004. Distributed optimization in sensor networks. Proc 3rd Int Symp on Information Processing in Sensor Networks, p.20-27. https://doi.org/10.1145/984622.984626

Ren J, Haupt J, 2019. A communication efficient hierarchical distributed optimization algorithm for multi-agent reinforcement learning. Real-World Sequential Decision Making Workshop at Int Conf on Machine Learning.

Riedmiller M, 2005. Neural fitted Q iteration—first experiences with a data efficient neural reinforcement learning method. Proc 16th European Conf on Machine Learning, p.317-328. https://doi.org/10.1007/11564096_32

Sayed AH, 2014. Adaptation, learning, and optimization over networks. *Found Trends® Mach Learn*, 7(4-5):311-801. https://doi.org/10.1561/2200000051

Schmidt M, Le Roux N, Bach F, 2017. Minimizing finite sums with the stochastic average gradient. *Math Program*, 162(1-2):83-112. https://doi.org/10.1007/s10107-016-1030-6

Sha XY, Zhang JQ, Zhang KQ, et al., 2020. Asynchronous policy evaluation in distributed reinforcement learning over networks. https://arxiv.org/abs/2003.00433

Shalev-Shwartz S, Shammah S, Shashua A, 2016. Safe, multi-agent, reinforcement learning for autonomous driving. https://arxiv.org/abs/1610.03295

Shapley LS, 1953. Stochastic games. *PNAS*, 39(10):1095-1100. https://doi.org/10.1073/pnas.39.10.1095

Shi W, Ling Q, Wu G, et al., 2015. Extra: an exact first-order algorithm for decentralized consensus optimization. *SIAM J Optim*, 25(2):944-966. https://doi.org/10.1137/14096668X

Silver D, Lever G, Heess N, et al., 2014. Deterministic policy gradient algorithms. Proc 31st Int Conf on Machine Learning, p.387-395.

Silver D, Huang A, Maddison CJ, et al., 2016. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484-489. https://doi.org/10.1038/nature16961

Silver D, Schrittwieser J, Simonyan K, et al., 2017. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354-359. https://doi.org/10.1038/nature24270

Singh S, Jaakkola T, Littman ML, et al., 2000. Convergence results for single-step on-policy reinforcement-learning algorithms. *Mach Learn*, 38(3):287-308. https://doi.org/10.1023/A:1007678930559

Singh SP, Sutton RS, 1996. Reinforcement learning with replacing eligibility traces. *Mach Learn*, 22(1-3):123-158. https://doi.org/10.1007/BF00114726

Srikant R, Ying L, 2019. Finite-time error bounds for linear stochastic approximation and TD learning. Proc 32nd Conf on Learning Theory, p.2803-2830.

Stanković MS, Stanković SS, 2016. Multi-agent temporal-difference learning with linear function approximation: weak convergence under time-varying network topologies. American Control Conf, p.167-172. https://doi.org/10.1109/ACC.2016.7524910

Stanković MS, Ilić N, Stanković SS, 2016. Distributed stochastic approximation: weak convergence and network design. *IEEE Trans Autom Contr*, 61(12):4069-4074. https://doi.org/10.1109/TAC.2016.2545098

Suttle W, Yang ZR, Zhang KQ, et al., 2019. A multi-agent off-policy actor-critic algorithm for distributed reinforcement learning. https://arxiv.org/abs/1903.06372

Sutton RS, McAllester DA, Singh SP, et al., 2000. Policy gradient methods for reinforcement learning with function approximation. Advances in Neural Information Processing Systems, p.1057-1063.

Sutton RS, Szepesvári C, Maei HR, 2008. A convergent $O(n)$ algorithm for off-policy temporal-difference learning with linear function approximation. Proc 21st Int Conf on Neural Information Processing Systems, p.1609-1616.

Sutton RS, Maei HR, Precup D, et al., 2009. Fast gradient-descent methods for temporal-difference learning with linear function approximation. Proc 26th Annual Int Conf on Machine Learning, p.993-1000. https://doi.org/10.1145/1553374.1553501

Sutton RS, Mahmood AR, White M, 2016. An emphatic approach to the problem of off-policy temporal-difference learning. *J Mach Learn Res*, 17(1):2603-2631.

Tesauro G, 1995. Temporal difference learning and TD-Gammon. *Commun ACM*, 38(3):58-68. https://doi.org/10.1145/203330.203343

Tsitsiklis JN, van Roy B, 1997. Analysis of temporal-diffference learning with function approximation. Advances in Neural Information Processing Systems, p.1075-1081.

Tu SY, Sayed AH, 2012. Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks. *IEEE Trans Signal Process*, 60(12):6217-6234. https://doi.org/10.1109/TSP.2012.2217338

Varshavskaya P, Kaelbling LP, Rus D, 2009. Efficient distributed reinforcement learning through agreement. In: Asama H, Kurokawa H, Ota J, et al. (Eds.), Distributed Autonomous Robotic Systems. Springer, Berlin, p.367-378. https://doi.org/10.1007/978-3-642-00644-9_33

Wai HT, Yang Z, Wang ZR, et al., 2018. Multi-agent reinforcement learning via double averaging primal-dual optimization. Advances in Neural Information Processing Systems, p.9649-9660.

Wang XF, Sandholm T, 2003. Reinforcement learning to play an optimal Nash equilibrium in team Markov games. Proc 15th Int Conf on Neural Information Processing Systems, p.1603-1610.

Watkins CJCH, Dayan P, 1992. Q-learning. *Mach Learn*, 8(3-4):279-292. https://doi.org/10.1007/BF00992698

Williams RJ, 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach Learn*, 8(3-4):229-256. https://doi.org/10.1007/BF00992696

Xiao L, Boyd S, Kim SJ, 2007. Distributed average consensus with least-mean-square deviation. *J Parall Distrib Comput*, 67(1):33-46. https://doi.org/10.1016/j.jpdc.2006.08.010

Ying BC, Yuan K, Sayed AH, 2018. Convergence of variance-reduced learning under random reshuffling. IEEE Int Conf on Acoustics, Speech and Signal Processing, p.2286-2290. https://doi.org/10.1109/ICASSP.2018.8461739

Yu HZ, 2015. On convergence of emphatic temporal-difference learning. Proc 28th Conf on Learning Theory, p.1724-1751.

Zazo S, Macua SV, Sánchez-Fernández M, et al., 2016. Dynamic potential games with constraints: fundamentals and applications in communications. *IEEE Trans Signal Process*, 64(14):3806-3821. https://doi.org/10.1109/TSP.2016.2551693

Zhang HG, Jiang H, Luo YH, et al., 2017. Data-driven optimal consensus control for discrete-time multi-agent systems with unknown dynamics using reinforcement learning method. *IEEE Trans Ind Electron*, 64(5):4091-4100. https://doi.org/10.1109/TIE.2016.2542134

Zhang KQ, Lu LQ, Lei C, et al., 2018a. Dynamic operations and pricing of electric unmanned aerial vehicle systems and power networks. *Transp Res Part C Emerg Technol*, 92:472-485. https://doi.org/10.1016/j.trc.2018.05.011

Zhang KQ, Yang ZR, Liu H, et al., 2018b. Finite-sample analyses for fully decentralized multi-agent reinforcement learning. https://arxiv.org/abs/1812.02783v5

Zhang KQ, Yang ZR, Liu H, et al., 2018c. Fully decentralized multi-agent reinforcement learning with networked agents. Proc 35th Int Conf on Machine Learning, p.5867-5876.

Zhang KQ, Yang ZR, Başar T, 2018d. Networked multi-agent reinforcement learning in continuous spaces. IEEE Conf on Decision and Control, p.2771-2776. https://doi.org/10.1109/CDC.2018.8619581

Zhang KQ, Yang ZR, Başar T, 2019. Multi-agent reinforcement learning: a selective overview of theories and algorithms. https://arxiv.org/abs/1911.10635

Zhang QC, Zhao DB, Lewis FL, 2018. Model-free reinforcement learning for fully cooperative multi-agent graphical games. Int Joint Conf on Neural Networks, p.1-6. https://doi.org/10.1109/IJCNN.2018.8489477

Zhang Y, Zavlanos MM, 2019. Distributed off-policy actor-critic reinforcement learning with policy consensus. https://arxiv.org/abs/1903.09255